```
**********************
*    "RECORD COPY"    *
* Schedule No. 7-6-2  *
* Retain Until 12-99  *
**********************
```


POWERTRAIN ELECTRONIC CONTROL STRATEGY BOOK

| STRATEGY LEVEL "LHBH1"

FOR USE WITH 2ND GENERATION EEC-IV MODULE: EFI-SD20

THE PROCEDURE FOR OBTAINING COPIES OF THIS BOOK OR ANY OTHER AVAILABLE "LH" DOCUMENTATION IS EXPLAINED ON THE NEXT PAGE.

LARRY SIERSMA
PHONE: (313)390-2159
STRATEGY PLANNING & PROCESS DESIGN SECTION
CONTROL SYSTEMS DEPT
POWERTRAIN ELECTRONICS
DEVELOPMENT - PTE

| SOFTWARE RELEASE DATE: MARCH 24, 1993
| STRATEGY BOOK RELEASE DATE: MARCH 25, 1993

"LH" STRATEGY DOCUMENTATION

All current strategy documentation is stored on the VAX computer cluster. Documentation can be obtained by logging into a VAX computer (I.E. SYS2, SYS3, SYS4, ETC.) and issuing one or more of the following VAX/DCL commands.

Generally, two types of documentation are available:

1. UPDATE PACKAGES - Change bars at the left margin are used to indicate where changes in text have occurred since the previous level. Some of these changes may simply be enhancements or corrections to the text of the previous level and may be unrelated to the strategy level change. This file can be used as a quick reference to show the changes which have been made for this release. The file name format is LH***UP.MEM, where *** is the desired new strategy level.

2. COMPLETE BOOKS - The file name format is LH***.MEM, where *** is the desired strategy level. Changes in text which have occurred since the previous level book will be indicated with change bars. The INDEX contains an entry, "CHANGED PAGES," which lists all pages containing changes.

The following VAX/DCL commands may be helpful in working with Strategy Book documentation:

TO DETERMINE IF A SPECIFIC STRATEGY BOOK IS AVAILABLE, TYPE:

```
DIR STRATEGY:LH***.MEM
DIR STRATEGY:LH***UP.MEM
```

TO OBTAIN A LINE PRINTER COPY OF A GIVEN DOCUMENT, TYPE:

```
PRINT/NOFEED STRATEGY:LH***.MEM     where *** = the desired strategy level
```

TO OBTAIN A XEROX COPY OF A GIVEN DOCUMENT, TYPE:

```
XEROX STRATEGY:LH***.MEM/DEST=EEE/USERNAME=name/COPIES=no/PMODE=P

    where: *** = the desired strategy level
           name = your user name
           no = desired number of copies (i.e. 1)
```

TO DETERMINE TARGETING OF EMR'S FOR FUTURE RELEASES, TYPE:

```
EDTS
```

AT THE EDTS MAIN MENU, SELECT "Standard Reports Menu" (number 6)
THEN, SELECT "EMR's Within EMR Group" (number 3)

TO DETERMINE THE STATUS OF STRATEGY BOOK DOCUMENTATION, TYPE:

```
@STRATEGY:BOOKSTATUS
```

TABLE OF CONTENTS LHBH1
PED-PTE, FoMoCo, PROPRIETARY & CONFIDENTIAL

CHAPTER 1

STRATEGY EVOLUTION

STRATEGY EVOLUTION

RELEASE FILE	BASE FILE	ERD/ URD NO.	EMR NO.	DESCRIPTION OF CHANGE
LHBH1 (03/24/93	LHBH0			
		10169	94-1237	COMMUNICATION - DCL - Running change needed to correct lose of ISCDTY.
		10394	94-1247	COMMUNICATION - DCL - Update code to properly set bits in DCL BITMAP_1 register.
LHBH0 (09/28/92)	LHBG0			
		8143	93-769	IDLE SPEED - IPSIBR CALC - IPSIBR not updating during RUNNING SELF-TEST.
		8384	93-77	EGR - OTHER - Bypass EGR enabled check during wiggle for 327/337.
		8629	93-775A	EGR - EGR FMEM - EMR # 93-775, was improperly implemented in LHBGA and LHBH0.
		7682	94-745	SPARK - OTHER - Add anti plug fouling strategy.
		8458	94-820C	OBD-I CONT TEST EGO SWITCHING TEST - update LH ego test.
		8458	94-820D	OBD-I CONT TEST EGO SWITCHING TEST - lazy/buzz logic not in code.
		8458	94-820E	OBD-I CONT TEST EGO SWITCHING TEST - associate parameter with LH.
		8458	94-820F	OBD-I CONT TEST EGO SWITCHING TEST - change V_EGO_BYPS := 0 to ... := 1
		8392	94-858	GENERAL - OTHER - ALT_CAL_FLG is set during PWR UP aftr battery discon-nect.
		8447	94-873	SPARK - OTHER - Documentation clean up.
		8559	94-92	SPARK - OTHER - Corrections to anti plug fouling strategy.
		7564	95-122	FUEL - CRANK - Allow greater flexi-bility with de-choke and APT.
		7564	95-122A	FUEL - CRANK - Strat. err: TP_REL>TP_DECHOKE should be TP_REL<=TP_DECHOKE.
LHAG0 (07/15/92)	LHAF0			
		2185	91-303	SELF-TEST - KOER - Greater flexibility for RUNNING EGO SENSOR TEST.
		7781	93-761	FUEL - ADAPTIVE - Function enhancement.
		7005	94-588B	TRANSMISSION - SELF TEST/FMEM - 4X4 low input error detection.
		7005	94-588C	TRANSMISSION - SELF TEST/FMEM - 4X4 low input error detection.
		7082	94-603	TRANSMISSION - SHIFT CONTROL - Improve engagm't strategy by using TOT and NEBART.
		7231	94-641	IDLE SPEED - OTHER - Induction of the A/C pressure switch in 94-E&F CALIB.

		7231	94-641A	IDLE SPEED - OTHER - Induction of the A/C pressure switch in 94-E&F CALIB.
		7741	94-718	TRANSMISSION - EPC - EPC spikes during engagements.
		7184	95-102	GENERAL - OTHER - RPM calc. does not take advantage of full 0.25 accuracy.
		7783	95-141	THERMACTOR - UP/DOWN STREAM - Allow capability to dump thermactor during closed loop fuel.

==

LHBF0 (06/04/92)	LHBE0			
		7202	93-740	TRANSMISSION - SELF TEST/FMEM - Replace N with nebart in vehicle speed sensor test.
		7519	93-746	EGR - EGR CONTROL - Correct software error in EGR RATE calculation.
		7745	93-757	EGR - EGR CONTROL - Software: omitted FN240(WOTTMR) from EGRATE calculation.

==

LHBE0 (05/29/92)	LHBD0			
		7078	93-710A	EGR - EGR CONTROL - Change EGR logic to enable EGR for periods at WOT.
		7078	93-710B	EGR - EGR CONTROL - Change EGR at WOT logic to provide MAP hysteresis.
		7405	93-734	THERMACTOR - UP/DOWN STREAM - Modify Thermactor Dump as a f'n of RPM, avoid backfires.
		7405	93-734A	THERMACTOR - UP/DOWN STREAM - Correct error in flop flop. Required change.
		7705	93-734B	THERMACTOR - UP/DOWN STREAM - Revise thermactor dump at low TP_REL to allow air at idle.
		7408	93-737	FUEL - ADAPTIVE - Function enhancement that will allow more adaptive learning.
		6049	94-280	SELF-TEST - KOER - Correct spark control test in koer - software only.

		7523	94-477B	FUEL - ADAPTIVE - Do not use reference cell values for update rates.
		7025	94-551C	OBD-I CONT TEST EGR - Lock out 327, 337 til EGR enabled. Convert to ladder diagram.
		7025	94-551H	EGR - EGR ENABLE - Correct strategy error in EGR enable logic.
		7025	94-551I	OBD-I CONT TEST EGR - Match strategy to existing software. Sonic EGR test.
		7005	94-588	TRANSMISSION - SELF TEST/FMEM - 4X4 low input error detection.
		7385	94-647	OBD-I ENG OFF - OUTPUT TEST MODE - Software error in output test mode self-test.
		7637	94-690	FMEM - FAILURE RECOGNITION - Many failures of code 628 have been reported.
		7667	94-702	FUEL - DECEL FUEL SHUT-OFF - Software error found in fuel pulsewidth calculation.

==

LHBD0 (05/01/92)	LHBC0			
		6727	92-562A	TRANSMISSION - SELF TEST/FMEM - Modify FMEM action for Transmission Overtemperature - LH,AC.
		5089	93-150G	SELF-TEST - CONTINUOUS - Create variant strategy for 5.8l/o to alter ego test.
		7035	93-708	GENERAL - OTHER - Avoid future software/strategy errors.
		6623	94-431	COMMUNICATION - DOL - Addition of distance-to-empty feature (DOL).
		6834	94-471	GENERAL - OTHER - Update SYMBOLOGY chapter to current symbology usage.
		6862	94-477	FUEL - ADAPTIVE - Four adaptive improvements.
		7302	94-477A	FUEL - ADAPTIVE - Random number algorithm will not function properly.
		7241	94-589	FUEL - OTHER - FUEL_SUM should be in ticks for LL and LH.

==

LHBC0 (01/23/92)	LHBB1			
		6100	93-657	FUEL - ADAPTIVE - Include logic to limit the time to adapt in WOT.
		6100	93-657A	FUEL - ADAPTIVE - Correct parameter XMAPPA base value in Parameter Dictionary.

==

LHBB1 (11/11/91)	LHBB0			
		6076	93-616	TRANSMISSION - MLPS - IPDL getting intermittant values outside correct range.
		6098	93-621	TRANSMISSION - EPC - Fix TV from going to 127.5 every background loop.

==

LHBBO LHBAO
(10/02/91)

4724	93-333	TRANSMISSION - Revise coverter clutch validation error code setting logic.
5409	93-365B	FUEL - DFSO - Thermactor and DFSO need different d_tp_dt terms.
5056	93-396	TRANSMISSION - FMEM - Rename OCIL, ASML, and ASMIL to TCIL.
5056	93-396H	TRANSMISSION - FMEM - Correct Parameter Transactions for LH.
5264	93-432	TRANSMISSION - NORM TV CALC - SCSLPXXX, and SCSLPX cal constants overflow a byte.
5340	93-454	TRANSMISSION - DET SHFT SOL - Main Control Rev'ns require calibratible sol. states in Neu.
5359	93-461	TRANSMISSION - FMEM - Request to Improve FMEM Action for E4OD.
5359	93-461A	TRANSMISSION - FMEM - Add C657_KAM_BIT to AC, LH, and AG.
5359	93-461D	TRANSMISSION - FMEM - Define TOT_OTEMP for LH,AC and AG strategies.
5437	93-500	SELF-TEST - KOEO - MLPS test improvement.
5437	93-500D	SELF-TEST - KOEO - New MLPS test improvement.
5563	93-534	FUEL - CLOSED LOOP - allow switching into/out-of closed loop for f450 superduty.
5713	93-546	TRANSMISSION - NORM TV CALC - Calculate converter clutch torque capacity.
5713	93-546A	TRANSMISSION - NORM TV CALC - Calculate converter clutch torque capacity.
5707	93-564	FMEM - UNKNOWN - Driveability in FMEM.
5714	95-037	TRANSMISSION - OTHER - Coast Clutch Control Software Error.

```
                5714    95-037A   TRANSMISSION - OTHER - Coast Clutch
                                  Control Software Error.
======================================================================

LHBA0      LHAZ1
(06/10/91)
                4516    91-589    OTHER - N/A - Software documentation
                                  needs 1991 copyright date.
                4650    93-327    TRANSMISSION - Revise shift validation
                                  logic error codes, improve FMEM.
                5271    93-327A   TRANSMISSION - Delete OCIL_HP from LH
                                  strategy.
                4834    93-334    OTHER - Revise purge strategy for 7.5L.
                4579    93-365    FUEL - DFSO - Add d/dt(TP) to DFSO and
                                  THERMACTOR to control backfire.
                4579    93-365A   FUEL - DFSO - Match timer resolution to
                                  strategy.
                5355    93-453    MISCELLANEOUS - INFERRED BP - Incorrect
                                  read done for BPPTWTLO for rolav.
======================================================================

LHAZ1      LHAZ0
(03/20/91)
                4922    92-507    EGR - EVR CONTROL - EGRACT and EGRERR were
                                  incorrectly calculated.
========================================================================

LHAZ0      LHAY0
(03/04/91)
                4204    92-324L   MISCELLANEOUS - Provide a separate LAMMAX
                                  for PURGE logic; documentation error.
                3541    92-325    SELF TEST - CONTINUOUS - TRLOAD is byte
                                  value, but loaded as word.
                3123    92-368    SYSTEM COMM - DCL - Verify BITMAP_x
                                  definition and add new PID codes.
                3123    92-368A   SYSTEM COMM - DCL - Replace PID code 2E
                                  (adder of LTMTB1) with IOCC.
                3123    92-368D   SYSTEM COMM - DCL - Revisions to PID
                                  tables and BITMAPS.
                4494    92-469    SELF TEST - CONTINUOUS - False SELF TEST
                                  code 173 (EGO1 sensor fault-rich).
                4513    92-477    TRANSMISSION - Lack of power complaint
                                  after light throttle 1-2 upshifts.
                4307    93-055A   FUEL - Provide a reset calibration switch
                                  for CRKPIP_CTR.
                3494    93-074    MISCELLANEOUS - Ensure validity of queue
                                  subroutine pointers.
                3774    93-150    SELF TEST - Byte thrift.
                4527    93-150C   SELF TEST - CONTINUOUS - PURGING and
                                  PRGFLG not representative of state of
                                  purge.
                3719    93-191    SELF TEST - Allow KOEO self test in
                                  neutral instead of PARK, E4OD.
                3876    93-196    FUEL - OPEN LOOP - 7.0L thermactor switch
                                  open loop error.
                2830    93-221    TRANSMISSION - Some of the governor
                                  parameters are incorrectly defined.
                4533    93-273    SELF TEST - KOEO - Software in KOEO
                                  canister purge OCC test.
```

4554	93-283	TRANSMISSION - Filter MAP_WORD for transmission TQ_NET calculation.
4554	93-283A	TRANSMISSION - Clarify MAP_WORD filtering technique for transmission use.
4612	93-304	MISCELLANEOUS - PURGE - Improve base calibration for PURG_ADP_SF.
4581	93-320	FUEL - PURGE - Remove the 2nd canister purge output.

===

LHAY0 LHAXA
(10/29/90)

1758	91-158B	TRANSMISSION - ENGAG/STALL TV; START_UP TV; TV PRES GUIDE - Correct errors in EMR 91-158.
4133	92-139C	SELF TEST - KOEO - Implementation of EMR's 92-139 and 92-139B was not correct.
4107	92-324K	SELF TEST - CONTINUOUS - Revise Canister Purge Filter.
3364	92-356	MISCELLANEOUS - INPUT CNVRT - Calculate VSBART_FM -based on NIBART, NOBART or NEBART.
3938	92-391	TRANSMISSION - EPC - Improve VFS temperature compensation.
3936	92-431	ISC - IPSIBR CALC - Remove IPSIBR reset to 0 on closed to part throttle change.
4212	92-442	AIR MEASUREMENT - MAP SAMPLE - Fuel drops out during high speed wot to pt tip outs.
3321	93-055	FUEL - Fix CRKPIPCTR logic so that a partial reset is possible.
3343	93-116B	TRANSMISSION - FMEM - Limit VS with VSS failure.
4080	94-010H	TRANSMISSION - GRCMKIV appears in EDTS for LH, but not code/strategy book.

===

LHAXA LHAX0
(09/13/90)

3995	91-409Y	SELF TEST - CONTINUOUS - Protect EGO test from vacuum controlled purge overload.

===

LHAX0 LHAW0
(09/07/90)

3752	91-409W	SELF TEST - CONTINUOUS - Prevent rich EGO failure due to purging.
0140	92-139	MISCELLANEOUS - Protect background from infinite loop with watchdog.
3346	92-139B	MISCELLANEOUS - Change parameter dictionary for RAM/ROM byte thrift.
3146 3177	92-200	FUEL - INJECTOR OUTPUT - Collision with asynchronous AE may cause improper fuel pulse.
3146 3177	92-200A	FUEL - INJECTOR OUTPUT - Collision with asynchronous AE may cause improper fuel pulse.
3597	92-313	TRANSMISSION - Allow application of coast clutch for PDL = 4, GR_CM < 4/

		3820	92-324B	SELF TEST - MIL - Lack of HEGO switching codes 173 and 177 are being set.
		3863	92-324D	OTHER - Purge strategy resolution improvement.
		3863	92-324F	OTHER - Add EDTS parameter transaction for PRG_INIT_FLG.
		4004	92-324G	MISCELLANEOUS - Purge decrement clipping normal purge.
		4004	92-324H	MISCELLANEOUS - PURGE - Limit PRG_DEC 0.99 to make code shorter.
		4028	92-324I	FUEL - MIL light turns on/off/on during rich purge condition.
		3728	92-366	TRANSMISSION - CSDYN cold TV modifier problem.
		3887	92-408	ISC - ISCKAM UPDATE - Documentation commonality needed for ISCKAM UPDATE logic.
		2858	93-060	MISCELLANEOUS - Match engine parameters to MY 91-1/2 7.0L LL strategy.
		2858	93-060A	MISCELLANEOUS - Match engine parameters to MY 91-1/2 7.0L LL strategy.
		3334	93-060B	MISCELLANEOUS - Want to transfer engine calibration from LL calibration.
		3769	93-115	ISC - DASHPOT - Minimum DASPOT clip logic was not governed with respect to N/D.

==

LHAW0 (05/24/90)	LHAV0			
		2096	91-229	SELF TEST - Unused IGNCNT should be removed from dictionary and code.
		3446	91-568	MISCELLANEOUS - Incorrect description of parameter V_EGO_ENA in DOC file.
		3318	92-241	TRANSMISSION - OFMFMG - Force TV_COUNTS to zero when ETV_TEST = 1 (FN622 > 15).
		3300	92-242	SELF TEST - KOER - Logic gates for Engine Running Self Test initialization incorrect.
		3429	92-253	ISC - Unable to calibrate FN825A and FN825B using cal-console.
		3503	92-287	TRANSMISSION - DYN TV CALC - Harsh shifts.
		NONE	93-064A	OTHER - Documentation change from DDVS to VSLIM.

==

LHAV0 (04/20/90)	LHAU2			
		1651	91-178	SELF TEST - KOER - Provide new logic and add throttle adjust modes to strategies.
		1775	91-230	SELF TEST - KOER - Surge during VIP.
		2732	91-230A	SELF TEST - KOER - Documentation for EMR 91-230.
		2964	91-483	SELF TEST - MIL - Change strategy book to use DISABLE_NOSTART.
		NONE	91-531A	TRANSMISSION - DYN TV CALC - Inappropriate upshift TV ramp during verification.
		2764	91-540	TRANSMISSION - SHIFT VAL - TP Failure and Shift Validation.
		3236	91-554	TRANSMISSION - FMEM - Software error in

```
                                LHAT2.
                 NONE    92-155K   TRANSMISSION - DSRD GEAR DET - PPH -
                                   Correct parameter specifications.
                 NONE    92-155L   TRANSMISSION - DSRD GEAR DET - PPH -
                                   Correct parameter specifications.
======================================================================

LHAU2      LHAU1
(03/26/90)
                 3299    91-565    TRANSMISSION - EPC - FN622(TOT) does
                                   not return signed results, but should.
                 3280    92-223    OTHER - RCON macros corrupt SMP.
======================================================================
LHAU1      LHAU0
(03/06/90)
                 3217    91-555    TRANSMISSION - NORM TV CALC - Software
                                   errors in LHAU0.
                 2131    91-556    TRANSMISSION - SHIFT CONTROL - Make
                                   FN12_DC, FN23_DC, and FN34_DC signed
                                   functions.
======================================================================

LHAU0      LHAT1
(02/16/90)
                 1542    91-199    SELF TEST - KOER - Engine running
                                   initialization revisions and new
                                   documentation.
                 NONE    91-199A   SELF TEST - KOER - Engine running
                                   initialization revisions and new
                                   documentation.
                 NONE    91-199C   SELF TEST - KOER - Documentation to
                                   support EMR 91-199/B and 3-digit codes.
                 2582    91-412    SPARK - Thrift; reduce resolution of
                                   SPK_LAMBSE.
                 3171    91-412A   TRANSMISSION - NORM TV CALC - EMR
                                   91-412 cannot be implemented as written.
                 NONE    91-428A   ISC - Improvement to LOWVOL strategy.
                 2623    91-435    SELF TEST - KOER - Delete the old E.R.
                                   spout test.
                 2629    91-437    TRANSMISSION - EPC - Shift quality varies
                                   with temperature.
                 2310    91-440    ISC - DUTY CYCLE - Improve ISC's DEBYMA
                                   output.
                 2863    91-454    TRANSMISSION - EPC - Inconsistent shift
                                   qualify with altitude.
                 2832    91-464    SELF TEST - CONTINUOUS - Remove VIP
                                   EGO switching repetitive one-shot test.
                 NONE    91-464A   FMEM - EGO SENSOR - EMR 91-464 did not
                                   cover the FMEM chapter.
                 3005    91-527    TRANSMISSION - EPC - TVCHARGE and cold
                                   engagement strategies may not work.
                 2856    91-528    MISCELLANEOUS - THERMACTOR - OBD-1
                                   implementation with thermactor.
                 2976    91-537    OTHER - Delay in AM2 turn-off.
                 NONE    91-537A   OTHER - Timer resolution, AM2 turn-off.
                 3167    91-542    AIR MEASUREMENT - MAP SAMPLE - Incorrect
                                   long MAP averaging at wide open throttle.
                 3186    91-543    SELF TEST - KOER - Extend clips on band
                                   limits for running idle adjust test.
```

		3200	91-546	TRANSMISSION - FMEM - Remove VSFMFLG being set by C452_KAM_BIT on power-up.
		1891	92-086	MISCELLANEOUS - Cannot load two regions into calibration console.
		NONE	92-155D	MISCELLANEOUS - INPUT CNVRT - Specify VS_RATEPH calculation.
		NONE	92-155G	TRANSMISSION - DSRD GEAR DET - To prevent powertrain hunting with E-transmissions.
		NONE	92-155H	TRANSMISSION - DSRD GEAR DET - To prevent powertrain hunting with E-transmissions.
		2540	92-161	SELF TEST - CONTINUOUS - Upgrade fuel pump test and documentation.
		2937	92-161A	SELF TEST - CONTINUOUS - Error in EMR 92-161, calibration parameter name used for RAM flag.
		3134	92-161B	SELF TEST - Correct typo's in fuel pump test.
		2913	92-170	MISCELLANEOUS - Correct potential problem.
		3041	92-178	TRANSMISSION - FMEM - OFMFLG is always set using base calibration.
		3100	92-201	MISCELLANEOUS - Install shadow instruction for IPPC(R-CON) function.

==

LHAT1 (01/22/90)	LHAT0			
		3018	91-126F	SELF TEST - CONTINUOUS - MIL light remains on after correction of EGR flow fault.
		2947	91-170G	SELF TEST - CONTINUOUS - Bulletproofing of continuous BOO test.
		3037	91-394C	FMEM - Problem with FMEM fuel when MAP vacuum failure recognized.
		NONE	91-394D	SELF TEST - CONTINUOUS - Correction in self test strategy.
		2987	91-409N	SELF TEST - CONTINUOUS - EGO test/ mechanical purge.
		3016	91-409O	SELF TEST - CONTINUOUS - V_LEGOTMR1 is held to zero after a jumpback with failure present.
		2773	91-411	SELF TEST - KOER - Revise running self test code 998 abort with corresponding code 128.
		2945	91-466	SELF TEST - Software coding error.
		2988	91-485	SELF TEST - CONTINUOUS - Change 3-digit slow code pulsewidth to 0.4 seconds.
		2968	91-487	SELF TEST - CONTINUOUS - Adaptive test prevents ECT open test from failing.
		2762	91-500	TRANSMISSION - CLUTCH VAL - Erroneous VIP code 62's.
		2849	91-503	FMEM - VSS failure detection.
		NONE	91-503A	SELF TEST - KOEO - MLPS service code in KOEO.
		NONE	91-503B	SELF TEST - CONTINUOUS - Amend EMR 91-503 for accuracy.
		NONE	91-503C	FMEM - Modify setting and clearing of flag VSFMFLG.

```
                3000    91-509    SELF TEST - CONTINUOUS - Software error
                                  in continuous ego software test.
======================================================================

LSAT0      LHAS0
(12/12/89)
                2829    91-170F   SELF TEST - CONTINUOUS - Documentation.
                2790    91-191C   SELF TEST - KOEO - DCL documentation
                                  and VIP executive.
                NONE    91-409M   SELF TEST - CONTINUOUS - Add constraint
                                  on adaptive clip test.
                2209    91-414    OTHER - Prevent thermactor in WOT.
                1656    91-439    ISC - BYPASS - RPM flairs when
                                  restarting a warm engine.
                2860    91-442    SELF TEST - V_LESTMR will not count
                                  up due to software error.
                2877    91-445    FUEL - Improve the LAMBDA average
                                  calculation for VIP test.
                2930    91-462    SELF TEST - CONTINUOUS - STO turns
                                  on when starting engine with STI
                                  grounded.
======================================================================

LHAS0      LHAR0
(11/16/89)
                NONE    89-554E   TRANSMISSION - Rename FNXXDC's to
                                  FNXX_DC's for AC, LD and LH.
                NONE    89-554I   TRANSMISSION - Document TP_REL high
                                  byte is the X-input to fox function.
                904     90-190A   SELF TEST - CONTINUOUS - Cleanup of EGO
                                  switching test.
                904     90-190C   SELF TEST - CONTINUOUS - Provide a level
                                  of fuel system testing.
                1995    90-190E   SELF TEST - CONTINUOUS -  More specific
                                  failure criteria for continuous EGO/fuel.
                NONE    90-190G   SELF TEST - CONTINUOUS - More specific
                                  failure criteria for continuous EGO/fuel.
                NONE    90-190H   SELF TEST - CONTINUOUS - EGO test
                                  modifications.
                NONE    90-190J   SELF TEST - CONTINUOUS - Development.
                NONE    90-190K   SELF TEST - CONTINUOUS - Development.
                NONE    90-241M   SELF TEST - CONTINUOUS - Development.
                2376    90-362A   MISCELLANEOUS - THERMACTOR - Thermactor
                                  air does not bypass with failed ego.
                NONE    90-362B   MISCELLANEOUS - THERMACTOR - Incorrect
                                  VIP codes; change from 41 to 144.
                2693    90-369B   TRANSMISSION - TV PRES GUIDE - TP failure
                                  at altitude.
                NONE    91-126A   SELF TEST - Changes made in base EMR
                                  91-126 are also required in LL.
                NONE    91-126B   SELF TEST - MIL - Additional MIL
                                  changes are required for 1991
                                  production.
                2425    91-126C   SELF TEST - MIL - Allow codes 194 and
                                  195 to control EGO FMEM flags.
                1795    91-161    SELF TEST - CONTINUOUS - Thermostat
                                  warranty returns with no trouble found.
                NONE    91-161A   SELF TEST - CONTINUOUS - Thermostat
                                  warranty returns with no trouble found.
```

NONE	91-161B	SELF TEST - CONTINUOUS - Revisions required to implement EMR 91-161A.
NONE	91-170C	SELF TEST - CONTINUOUS - To add continuous BOO test.
NONE	91-170D	SELF TEST - CONTINUOUS - To add continuous BOO test.
NONE	91-190A	MISCELLANEOUS - INPUT CONVERTER - VS is calculated regardless of VSTYPE - Electronic Transmission.
2034	91-191	SELF TEST - Prevent entry into Self Test if vehicle is in Drive or moving.
2130	91-191A	SELF TEST - Conflicting pages (16 and 17) for LH strategy.
2751	91-191B	SELF TEST - KOEO - Documentation for EMR 91-191.
NONE	91-247B	SELF TEST - Erroneous code 29; VSSTMR runs with key on-engine off.
2523	91-272B	MISCELLANEOUS - Revise open loop fuel flag logic; add TP_REL criteria.
2329	91-289	EGR - ENABLE LOGIC - Incorrect DOC file definitions.
NONE	91-289A	EGR - ENABLE LOGIC - Incorrect DOC file definitions.
2538	91-289B	EGR - EMRS 91-289,A not applicable to LL, LH, LU strategies.
2327	91-299	SELF TEST - KOEO - ACC/NDS test revisions.
NONE	91-299A	SELF TEST - KOEO - Documentation correction for A/C switch test EMR 91-299.
2341	91-301	MISCELLANEOUS - SMP vector will be incorrect in production strategies.
2419	91-313	SELF TEST - MIL - MIL light will not turn on if fault 32 is present.
2507	91-336	SELF TEST - CONTINUOUS - Revisions to the Cooling System Test.
NONE	91-336A	SELF TEST - CONTINUOUS - Revisions to the Cooling System Test.
2152	91-339	SELF TEST - Lansdale request for CPU/ RAM/ROM test output pulse spec.
NONE	91-339A	SELF TEST - Lansdale request for CPU/ RAM/ROM test output pulse spec.
2508	91-350	FUEL - Eliminate fuel pump drop-out at power-up.
2534	91-358	SELF TEST - LH strategy must have same EGR updates as all other strategies.
2543	91-367	SELF TEST - KOER - Hot Injector Compensation not correct during Self Test.
2571	91-369	SELF TEST - Software error; incorrect continuous VIP codes can be transmitted.
2286	91-380	SELF TEST - EOL - Unexpected 35 msec pulse on STO during Lansdale Test.
2678	91-382	SPARK - Base value for DWL_XS_MIN inadequate for generic modules.
2679	91-383	SELF TEST - CONTINUOUS - Change required to Adaptive Test.
NONE	91-383A	SELF TEST - CONTINUOUS - Change required to EGO Test.
NONE	91-383C	SELF TEST - Corrections.
NONE	91-383G	SELF TEST - Corrections.

		2171	91-394	SELF TEST - Enhancement to continuous MAP sensor test.
		NONE	91-394A	SELF TEST - Change logic of continuous MAP VAC test for robustness.
		2708	91-396	SPARK - Error found on DV tester in ICCD mode output.
		2753	91-409	SELF TEST - CONTINUOUS - Software reviewers' convenience.
		NONE	91-409D	SELF TEST - Consolidation of EMRS.
		NONE	91-409H	SELF TEST - Eliminate EGO test failure void.
		2819	91-423	SELF TEST - Commonality of VIP software used by AA, AC, GT, LL and LH.
		2729	91-427	SELF TEST - Change from stereo to mono ego.
		2553	92-137	FMEM - Remove TOTFMFLG logic in all strategies.
		2395	93-017	MISCELLANEOUS - Background loop times for generic strategies are high.
		2467	93-027	MISCELLANEOUS - 7.0L governor strategy required in LH strategy.
		NONE	93-027A	MISCELLANEOUS - 7.0L governor strategy required in LH strategy.

==

LHAR0 (09/14/89)	LHAQ0			
		2486	90-345C	TRANSMISSION - Comp torque calculation in EMR 90-345 needs additional variable.
		2398	91-310	ISC - Add ISC-BPA KAM adaptive clips and zero KAM if outside clip.
		2541	91-353	TRANSMISSION - Torque calculation/ engine calibration issues.
		NONE	91-353A	TRANSMISSION - Define AMT and AMPEMT which were used in the base EMR.
		NONE	91-353D	TRANSMISSION - Torque calculation/ engine calibration issues.

==

LHAQ0 (08/29/89)	LHAP0			
		NONE	89-130B	SELF TEST - Spout circuit test documention clarification.
		NONE	89-554D	TRANSMISSION - Revise base calibration for FNXXT and X-input for FNXXDC.
		945	90-089	SELF TEST - Strategy documentation improvements.
		NONE	90-089A	SELF TEST - Documentation clarification for EMR 90-089.
		1977	90-348	SELF TEST - Convert from two digit to three digit self test codes.
		NONE	90-348E	SELF TEST - Create three digit self test parameters for LH strategy.
		NONE	90-348P	SELF TEST - Miscellaneous corrections for three digit codes.
		NONE	90-348T	SELF TEST - Miscellaneous corrections for three digit codes.
		NONE	90-348V	SELF TEST - Miscellaneous corrections for three digit codes.

2468	90-373	FUEL - Correct error in bias table lookup.
2021	91-190	MISCELLANEOUS - VS is calculated regardless of VSTYPE.
895	91-235	SYSTEM COMMUNICATION - Revise DCL diagnostic mode protocol for self test codes.
1596	91-254	SELF TEST - Eliminate the possibility of setting a false code 33.
2170	91-256	MISCELLANEOUS - Document the use of NEW_PIP.
2121	91-257	FMEM - Documentation error in EGR FMEM logic.
2308	91-285	SYSTEM COMMUNICATION - Correct interface issue with 9.1 cal con update.
NONE	91-287A	ISC - Assist in calibrating ISC.
2205	91-288	ISC - DASPOT airmass is clipped under utilizing feature.

==

EVOLUTION FILE	SOURCE FILE	EMR NO.	DESCRIPTION/REMARKS
LHAP0 (06/01/89)	LHAN0	90-326	FUEL - CRANK - Software error in FCA file for FN023. ERD 1768.
		90-345	TRANSMISSION - OTHER - Compensate torque calculation for SAF. ERD 1856.
		90-345A	TRANSMISSION - OTHER - Compensate torque calculation for SAF. No ERD.
		90-347	TRANSMISSION - SHIFT CONTROL - Improve downshift quality. ERD 1911.
		90-351	ISC - DESIRED RPM - Unable to get hicam kick down. ERD 2068.
		91-149	OTHER - Correct documentation to Keep Alive Memory chapter. ERD 1806.
		91-154A	TRANSMISSION - EPC - Possible damage to electronic transmission. No ERD.
		91-165	MISCELLANEOUS - Lack of MPG mode. ERD 1807.
		91-172	ISC - BYPASS - Multiple versions of ISC MODE_SELECT documentation. ERD 1873.
		91-172A	ISC - BYPASS - Correct documentation error shown in 91-172. ERD 1873.
		91-174	FUEL - Correct timer reset(s) logic upon a LAMBSE reset. ERD 1934.
		91-174A	FUEL - Correct timer reset(s) logic upon a LAMBSE reset. No ERD.
		91-177	ISC - BYPASS - Multiple versions of ISC dashpot documentation. ERD 1886.
		91-179	FUEL - CLOSED LOOP - Errors in closed loop LAMBSE documentation. ERD 1932.
		91-197	FUEL - Change vector clips on function FN023. ERD 2037.
		91-205	ISC - DASHPOT - Improve ability to handle tip-in/tip-out clunk. ERD 1390.
		91-222	FUEL - CLSD LOOP ENABL - Allow closed loop fuel control when in MPG mode. ERD 1951.
		92-065	MISCELLANEOUS - OTHER - The codes transmitted to the SBDS on the DCL could be incomplete. ERD 1661.

LHAN0 (02/24/89)	LHMA	89-552	FMEM - TRAN OIL TEMP - Correct FMEM INIT_TOT Calculation. ERD 1778.
		90-271B	VIP - Correct implementation of 90-271: the vector range maximum value of V_VSS_NMIN should be 16,383.75 RPM, not 10,000 RPM. URD 1708.
		90-308A	S/W - The software documentation for all current & future S/W version prereleases needs to be upgraded to a 1989 copyright date beginning 01/01/89. URD 1639.
		90-317	FMEM - Revise ISC FMEM to exclude ACT and ECT effects. URD 1709.
		90-325	TRANS - Some engagements may be preformed with inappropriate EPC pressure. URD 1734.
		90-325A	TRANS - ENGAG/STALL TV - Correct error in original EMR. ERD 1734.
		90-332	TRANSMISSION - TQ_IALFA CALC - Correct clip check for TQ_IALFA CALC. ERD 1814.
		91-063D	MISC - DCL - Revise BITMAP_0 and BITMAP_1 documentation. URD 861,985,1205.
		91-130A	SELF TEST - Continuous code 49 changed to 79 to make common. ERD 1821.
		91-130B	SELF TEST - Change of service code not desired from year to year. ERD 1851.
		91-130C	SELF TEST - Add change pages that should have been on 91-130B. No ERD.
		91-135A	VIP - S.M.A.C. recommendation to improve diagnostics. URD 1606,1694.
LHMA (02/09/89)	LHM1	89-550	TRANS - Insufficient EPC pressure if vehicle speed sensor is questionable. URD 1754.
		89-551	TRANS - Shift error flags can be set erroneously if PDL_ERROR is set. URD 1753.
		90-323	TRANS - Incorrect clipping of EPC pressure. URD 1745.
		90-324	VIP - DASPOT function revisions required before signoff. URD 1715.
		90-324A	VIP - Documentation changes required to implement EMR 90-324. URD 1715.
LHM1 (01/12/89)	LHM0	0V-271A	VIP - Correct implementation of 0V-271 to refresh VSFMFLG in continuous VIP as well as in the VIP executive. URD 1352.
		0V-314	FMEM- EPT sensor (open/short) will not turn on MIL light, doees not set failure flag (codes 31/35 are set). URD 1674.
LHM0 (12/21/88)	LHL0	9-541	FUEL - Reset TSLEGO and ACCUM upon a lambse reset to 1.0. URD 1561.
		9-542	TRANS - Neutral to manual 1 lever movement results in 1st gear at all speeds. URD 1583.
		0V-301	MISC - DCL to VIP interface. URD 1500.
		0V-302	FUEL - VIP requires idle fuel modulation to stabilize engine during the throttle adjust mode. URD 931.
		0V-302A	FUEL - Documentation pages included

			in EMR 0V-302 were incorrect. URD 931.
		0V-302B	MISC - Documentation pages included in EMR 0V-302 were incorrect. URD 931.
		0-311	MISC - Set communication flags every background loop. URD 1655.
		1-106A	MISC - DCL - Processor resets occasionally when DCL is enabled. URD 1600.
LHL0 (22/11/88)	LHK1	9-454	TRANS - Incorrect shift logic documentation. URD 799.
		9-511E	MISC - CANP - Modify canister purge strategy to handle high vapor production. commonize canister purge logic with the final version developed for the 1989 running changes. URD 1272.
		9-519A	FUEL - LAMBSE will be clipped to the minimum value (0.01) instead of the maximum value (1.99) if LMBJMP calculation on tip-in results in a LAMBSE greater than 1.99. URD 1282.
		9-520	SPARK - Allow more flexible octane adjustment which accounts for all combinations of ECT and ACT. URD 1296.
		9-520A	SPARK - The table specification was shown as a 7 x 7 instead of a 6 x 6. URD 1296.
		9-520B	SPARK - EMR 9-520 was written primarily for running change LU and LH. In order to incorporate the ECT, ACT spark table in the mainline LL and LH, it is necessary to make changes for BRDRLN_SPK which is used in Generic ISC. URD 1296.
		0V-117F	VIP - Evaluation of TP adjust mode requires modifications of test conditions. URD 931.
		0V-117G	VIP - EMR 0V-117F requires modification. URD 931.
		0V-117I	VIP - EMR 0V-117F requires modification. Cancel EMR 0V-117G. URD 931.
		0V-117J	VIP - Base strategy KAM qualification test must be revised in order that fuel cells are not reset on exit from the VIP throttle adjust mode. URD 931.
		0V-117K	VIP - Clean up EMR is required to implement all previous associated EMRs. URD 931.
		0V-117L	VIP - Byte thrift and clarification of EMR 0V-117K. Cancel KAM changes from EMRs 0V-117E and 0V-117F. Cancel EMR 0V-117J. URD 931.
		0V-118	VIP - Improvements to Engine Running Goose Test for robustness. URD 328,996.
		0V-118B	VIP - Strategy documentation updates required to implement 0V-118. URD 996.
		0V-118D	VIP - Final updates required to

	implement EMRs 0V-118 and 0V-118B into Speed Density strategies. URD 996.
0V-118F	VIP - Revise Goose Test documentation for added clarity. URD 1581.
0-144B	TRANS - Attachment 9 of EMR 0-144A contains an error. URD 1092.
0V-190D	VIP - Allow EGO Full Time to switch before indicating a failure. URD 904.
0V-195	VIP - Review of EMR 9V-399 for release of 'TT' strategy required changes to the documentation of the KAM Code Erase logic. URD 945.
0-197A	MISC - Cleanup EMR. URD 1444.
0-210	TRANS - RPM flare when converter unlocks in first gear due to slight reduction in throttle position. URD 1175.
0-219	TRANS - Compensation of EPC is required for transmission oil temperature and 4x4 Low operation. Engagement TV pressure needs to be a fraction of engine RPM and time since engagement began. URD 1176,1194,1204,1249,1281.
0-219A	TRANS - Change dynamic EPC from engine RPM to throttle position. Correct specifications of EPC parameters introduced by EMR 0-219. URD 1326,1381.
0-219B	TRANS - Correct errors in original EMR 0-219. URD not required.
0-219C	TRANS - Correct errors in original EMR 0-219. URD not required.
0-221	TRANS - The optimum speed ratio delta to unlock/relock the torque converter during upshifts varies with throttle position, but current logic uses scalar value independent of TP. URD 1178.
0-226	SPARK - KNOCK - The 4.9L engine is knock limited above a certain RPM. To allow maximum use of the knock sensor, spark is advanced beyond SAF which causes detonation when the engine enters then knock limited speed range. URD 1294.
0-226A	SPARK - KNOCK - FN146B is listed in strategy book and FN146A is in the code. FN146A is a word function and is correct for the code. URD 1294.
0-232	FUEL - INJECTOR TIMING - If the INJCNT register becomes corrupted, the recovery code will not set IBETA and INJCNT to zero as desired since the foreground temporary register that is stored as IBETA and INJCNT will not contain a zero value. This could prevent correct injecor timing. URD 1148.
0-235	TRANS - It is difficult to detect shift errors on 1 -> 2 shift because of a single minimum vehicle speed criteria to validate a shift. Dividing the vehicle speed paraemter by RT_NOVS prior to comparing to VSBART is extensively used

	in transmission control. Time/byte thrift is possible. URD 1123,1250.
0V-252	VIP - Code used only for GS strategy is done for all strategies. URD 1344.
0-255	TRANS - VEHICLE SPEED - Electronic transmissions require a rapidly responding rationality test of vehicle speed to avoid unwanted shifts if the vehicle sensor input is providing incorrect data. Vehicle speed calculation needs bullet-proofing modifications. URD 1177.
0-255B	TRANS - Correction/cleanup of EMR 0-255. Original EMR incorrectly specified "VSCNT" instead of "VSCTR" in "GR_DR_AUTO" module for selecting shift logic. URD N/A.
0-256	SYSEQUAT - VSBART - Calibrating the time constant for VSBART to a large value results in a significant deadband. The Rolling Average remainder is not currently saved, and VSBART will not continue to update when within the deadband. URD 1347.
0V-271	VIP - Changes in E-TRANS FMEM require that VSFMFLG be able to set when in 4x4 mode. URD 1352.
0-272	FMEM - All DIS logic should have been removed including FMEM logic. DIS_FMEM is still included in the MILTMR logic and should have been removed with the DIS logic. URD 1405.
1-070	MISC - Add clock in engineering unit, CLOCK_SEC. Add capability to gather real time in engineering units when data logging. URD 1318.
1-084	MISC - MPGTMR - To enable fuel economy mode during dynamometer certification of "over 8500" truck applications, where the vehicle speed sensor input is non-functional. URD 1125.
1-085	FUEL - INJECTOR OUTPUT IN CRANK - No fuel output on 1st pip down edge. When down edge fueling for crank is selected, and the first edge processing is a PIP down edge, no fuel is output because the foreground fuel routine is called from the PIP up edge and FUELPW is still zero. URD 1362.
1-085A	FUEL - INJECTOR OUTPUT IN CRANK - No fuel output on 1st pip down edge. When down edge fueling for crank is selected, and the first edge processing is a PIP down edge, no fuel is output because the foreground fuel routine is called from the PIP up edge and FUELPW is still zero. Cancel EMR 1-085. URD 1362.
1-091	MISC - DCL - The 184 byte block of RAM, which was previously reserved for SBDS use, is now needed for engine control strategy. This reduces the RAM available

			to the SBDS for downloaded programs from 600 to 416 bytes. URD 376.
LHK1 (10/13/88)	LHK0	1-043D	MISC - Permit use of V8.2 cal console. Floating SMP data prohibits the use of V8.2 cal consoles. LHK0 does not exceed 8K of SMP data, so V8.2 cal consoles would be able to access all parameters. URD 1340.
LHK0 (09/28/88)	LHJ0	9-511	MISC - PURGE/THERMACTOR - Thermactor and/or purge inducted Open Loop Fuel on high rpm extended idles is undesirable. URD 454,568,1151.
		9-511A	MISC - Clean up existing EMR (9-511). Byte savings can be realized and revisions to start book documentation to reflect actual paragraph names. URD 454,568,1151.
		9-511B	FUEL - Fuel goes lean when PRGTMR is reset to zero because LAMBSE is close to the upper clip. URD 1272.
		9-511C	FUEL - EMR 9-511B did not include proper "new" attachment 2 for LUVF strategy. Some applications don't want to clip LAMBSE when PRGTMR is reset. URD 1272.
		9-519	CLOSED LOOP - If the LAMBSE jumpback calculates a requested value which is greater than 2.0, LAMBSE will be set to 0.01 instead of 1.99 prior to the LAMMIN/LAMMMAX clips. Therefore, the clipped LAMBSE value will be LAMMIN when it should have been LAMMAX. URD 1282.
		0V-035	VIP - Engine running: evaluate new approach to testing the spout circuit. URD 310.
		0V-035A	VIP - Strategy change and documentation update required for clarification. URD 310.
		0V-117	VIP - Evaluate proposed throttle setting mode in Engine Running test. This EMR cancels EMR 0V-097, which was created for EAO strategy (CE). URD 777,778,931.
		0V-117A	VIP - Changes of original EMR required for implementation of EMR 0V-117. URD 931.
		0V-117C	VIP - Changes of original EMR required for implementation of EMR 0V-117,0V-117A. URD 931.
		0-141	TRANS - To correct errors in the strategy book documentation for E4OD. URD 802,836,837.
		0-146	SPARK - TRANSIENT SPARK - TLOFLG can change state each pass once started. this can cause spark errors in delivery with resultant driveability complaints. URD 1067.
		0-161	FUEL - Incorrect FUEL_A calculation. URD 1132.
		0-163	ISC - Commonize RPMERR documentation

	for Generic ISC. Multiple documentation versions exist for the Generic ISC RPMERR strategy modules, even though all describe identical logic. URD 1137.
0-163D	ISC - Correct parameter definitions in 0-163. Certain parameter definitions were left out of list in EMR 0-163. URD 1137.
0-172	ISC - FHEM strategy for Generic Idle Speed. URD 223.
0-172A	ISC - FHEM for Generic ISC. Revisions to 0-172. HCAMFG was accidentally shown as HCAMFLG in 0-172. Page 4 of 0-172 shows a logical AND which should be an OR. URD 223.
0-174	MISC - Landsdale tester is not compatible with 56K EPROM EEC-IV. URD 1005.
0-180	MISC - Add RAM initialization pages to strategy books. URD 324.
0V-190	VIP - More robust continuous EGO switching test. URD 904.
0V-190A	VIP - Cleanup of EGO switching test. URD 904.
0V-190B	VIP - Provide more explicit continuous EGO test initialization logic. Modify logic to reflect actual implemementation. URD 904.
0-197	MISC - Supply flags for continuous EGO VIP test. URD 1227.
0V-204	VIP - Prevent high TV pressure when in Engine Running Self Test. Commonize all electronic trans. control during Engine Running. URD 1122.
0V-205	VIP - Software error when coding TOT testing. Multiple failure (ECT and TOT concurrently) cannot be detected in KOEO VIP. URD 1197.
0-212A	ISC - Commonize overview documentation for Generic ISC. Multiple documentation versions exist for the Generic ISC OVERVIEW strategy. URD 1137.
0-212B	ISC - Generic ISC OVERVIEW software is incorrect. While doing commonization of the Generic ISC OVERVIEW documentation, it was discovered that certain software implementations were incorrect. URD 1137.
0-246	SPARK - OUTPUT SCHEDULING - Missing SPOUT signal with ICCD and ECHO PIP. Transitions from ECHO PIP mode to normal spark mode, and from falling edge dwell to ECHO PIP mode, fail to put out SPOUT signals URD 1333.
0-246A	SPARK - OUTPUT SCHEDULING - Wrong parameter set to DATA_TIME. URD 1333.
0V-264	VIP - More efficient code and increased robustness. Provide a single exit point for common housekeeping whenever continuous VIP is exited early due to

			being in crank mode or being in the first four seconds since power up. URD 1376.
		1-043	MISC - Change S/W calling sequence to support new 48K cal console. URD 1102.
		1-043A	MISC - Correct 48K cal console calling routine. An EEC-IV system reset is required before the cal console will enter the CS mode. URD 1325.
		1-043C	OTHER - SMP data is still being truncated with version 9 cal console. URD 1340.
LHJ0 (06/02/88)	LHI0	0-144	TRANS - improve execution time and create common strategy modules. URD 1092.
		0-144A	TRANS - Clarify documentation.
		0-145	SPARK - avoid premature TFI failure. URD 715.
		0-145A	SPARK - revise 0-145. URD 715.
		0-162	FUEL - Increase calibration resolution. URD 970.
		0-164	TRANS - avoid engine RPM flare on subsequent tip-in before transmission upshifts correctly when VSS fails. URD 1113.
LHI0 (05/13/88)	LHH0	9-092	OTHER/SW - New macro. URD 127.
		9-294	MISC - Common CRKTMR. URD 556.
		9-497	ISC/SW - Add generic idle.
		9-500	TRANS - Revise converter unlock. URD 1022.
		9-501	FUEL - Premature VS limiting in 4x4 low mode. URD 1032.
		0-071	FUEL - Revise CRANK fuel to be a function of PIPS (not time) in CRANK. URD 854.
		0V-072	VIP - Define KAM bits associated with continuous error codes. URD 135.
		0-099	DCL - Add DCL.
		0-099A	DCL - Revise PID table/Bit Map. URD 376.
		0-102	DCL - modifications. URD 701.
		0-102A	DCL - modifications. URD 701.
		0-106	ISC - Revert to former IPSIBR calculation. URD 923.
		0-131	AE - Revise AEFUEL enable/disable logic. URD 978.
		0-137	MISC - Revise A0 Base value; too large. URD 1045.
		0-166	DCL - Reset RAM/CART when changing modes. URD 1121.
		0-168	E40D/SW - Correct Gear Selection logic. URD 1140.
LHH0 (03/14/88)	LHG1	9V-489	VIP-OCIL operates normally during ENGINE-RUNNING S/T. URD 967.
		9-491	OTHER - Add proprietary messages. URD 974
		0V-046C	VIP - Rename FMEM_MONITR1 to 2.
		0-092	ISC - Add Generic Idle. URD 017

		0-092A	ISC - Revisions to 0-092. URD 017
		0V-098	VIP - Delete unused calibration constants, VTOT3, VTOT5. URD 966.
		0V-101	VIP/ISC - Update to coincide with Generic Idle Speed. URD 017.
LHG1 (02/25/88)	LHG0	9-486	SW/FMEM - Correct erroneous OFMFLG definition. URD 955.
		9-487	FUEL - Do Open loop fuel control during speed limiting. URD 956.
LHG0 (02/19/88)	LHF0	9-456A	TRANS - Move ETVOCM conversion logic from System Equations to ATOD CNVRT. URD 914.
		9-470	TRANS - Create Calibration switch to inc/exclude "shift errors" from transmission fault(s) causing OCIL flash. Refer to URD 887.
		9V-476	VIP/SW - Revise EPC Solenoid Test in KOEO Self Test. URD 903.
		9V-476A	Revisions to 9V-476. URD 903.
		9-477	TRANS - Avoid harsh WOT upshifts. Refer to URD 929.
		9-480	MISC - Reduce memory. (No URD #)
		9-481	TRANS - Insure correct entry to Torque Truncation at high throttle angles when EPC circuit open. URD 939.
		9-482	TRANS - Insure EPC on high engine RPM engagements and on first auto shift. Refer to URD 938.
		0-047	MISC - Redefine registers containing flag bits. Refer to URD 748.
LHF0 (1/27/88)	LHE0	9V-032E	VIP/KOEO - Fix sw error in ETV solenoid test performance. URD 816
		9-154B	MISC - Correct documentation of original EMR. URD 729.
		9-155B	TRANS - Correct documentation of original EMR. URD 729.
		9-267B	TRANS/SW - Correct Shift validation software error. URD 877
		9-268A	TRANS - Byte thrift.
		9-268B	TRANS - Cancel 9-268A.
		9-387	MISC - Add copyrights.
		9-389C	FMEM/TRANS - Correct software error in ETV current monitor test. URD 828.
		9-393A	TRANS/SW - Update FLG_PWR determination flip-flops every BG Loop. Refer to URD 692.
		9-394	TRANS - Revise Coast Boost logic to use FN3CB if GEAR_CUR >OR= 3. Refer to URD 757.
		9-406	TRANS - Modify Shift logic to enable downshift while upshift in progress. Refer to URD 758.
		9-411	TRANS - Avoid long delays for manual downshifts. URD 759.
		9-415	MISC/THERMACTOR - Replace AWOTMR with WOTTMR in the Thermactor Air Control

			Bypass logic. URD 771.
		9-416	MISC - Thrift, remove DPLGHP. URD 446.
		9-417	SPARK - Respecify FN1128 to signed table, resolution: 0.0156, range -2.00 to 1.98. URD 792.
		9-417A	SPARK - Correct calculation of FN1127*FN1128. Refer to URD 849.
		9-434	FUEL - TORQUE Trunction, improve function of torque limiting strategy. Refer to URDs 630, 631, 825.
		9-435	TRANS - delay transmission shifts during torque limiting to improve drive. Refer to URD 831.
		9-437	SYSEQUA/FMEM - ECT & TOT start rolling averages routines revised. URD 606.
		9V-440	VIP - Avoid shift val. & converter clutch monitoring errors during running Wiggle mode. URD 848.
		9-442	MISC - Revise POWSFG definition. Refer to URD 668.
		9-443	SW/TRANS - Revise Base Calibration. URDs 679,718.
		9-445	MISC - Incorrect FLG_NO_TV_UP. Refer to URD 725.
		9-446	MISC - Correct VBAT computation documentation. URD 739.
		9-450	MISC - Incorrect BIHP documentation.
		9-452	TRANS - Revise shift logic documentation. Refer to URD 782.
		9V-455	VIP - Fix fault filter subroutine so error-detect flag is always = 0 before subroutine returns. URD 851.
		9-456	TRANS - compensate ETVOCM for VREF variability. Refer to URD 731.
		9-457	TRANS - Avoid harsh transmission engagements after a very cold start. Refer to URD 844.
		9-461	TRANS - Modify NOV_ACT calculation pacing. Refer to URD 866.
		0-048	SYMB - Add hexidecimal explanation.
LHE0 (11/28/87)	LHD0	9-001H	TRANS - Documentation corrections. URDs 520, 582, 588, 591.
		9V-032A	VIP - Addition of ETV solenoid test to KOEO. URD 285.
		9V-032B	VIP - Revise KOEO ETV solenoid eror code.
		9V-032C	VIP - Scaling revisions for ETV parameters.
		9V-032D	VIP - Revise ETV test.
		9-038C	TRANS - Documentation corrections. URD 638.
		9V-102D	VIP - Corrections to 9V-102C.
		9V-170	VIP - Insure correct RPM if engine in FMEM mode, upon entry to Self Test. URD 266.
		9V-170A	VIP - Modify 9V-170 implementation.
		9V-244B	VIP/SW - Avoid false indications of EGR failure. URD 698.
		9V-246	VIP -Prevent execution Engine Running Test if vehicle moving. URD 396.
		9V-261	VIP - Correct erratic TP sensor operation. URD 471.

		9V-261A	VIP - Cancel 9V-261.
		9V-265	VIP - Software thrift. URD 477
		9V-292	VIP - KOEO MAF test revision; ECT, ACT Test documentation correction. URD 462.
		9-341	TRANS - Add TP requirement to unlock the converter clutch when brake applied. Refer to URD 602.
		9-341A	TRANS - Update 9-341, replace TP_REL and FIPL_REL with DD_UN_UNL. URD 602.
		9-344	VIP compatibility. Cancel 9-145, 9-145A 9-300. Refer to URDs 612, 614, 623, 628.
		9-345	TRANS - Cleanup: delete TM_LK_RATE. Refer to URD 527.
		9V-359	VIP - Reduce potential storage of erroneous VSS error code. URD 664.
		9-361	TRANS - Avoid unwanted downshifts. URD 558.
		9-364	FMEM - S/W - force full h/w reset if BG_POINT range check fails. URD 661.
		9-375	FUEL - Modify DFSO/FMEM interaction. URD 425.
		9V-375B	VIP - Clarify VSS Test documentation.
		9V-375C	VIP - Correct logic error in Attachment #2 of 9V-375B. URD 425
		9-381	ISC - Clear FFMTMR each time BGCNT is clearedn in FAM. Refer to URD 708.
		9V-382A	VIP - General cleanup: documentation.
		9-385	FMEM - modify with RPM/RP sensor failures.
		9-387	S/W - Copyright paragraphs. NO URD.
		9-389	TRANS - Modify TV Guide. Add test for open/short if failure indicated. URD 731.
		9-389A	TRANS - Deletes 9-389; Modify TV Guide and OFMFLG logic. URD 731.
		9-389B	TRANS - Modify register use.
		9-391	TRANS - Improve shift error detection. URDs 716, 738.
		9-399	S/W - Modify VIP KAM code. URD 742.
		9V-400	VIP - Correct warmup counter logic. URD 479.
		9-401	TRANS - thrift/delete unused scheduled 1st gear converter clutch lock-up.
		0-019	FMEM - Delete immobile TP test in FMEM. Refer to URD 625
		0V-019A	VIP - Modify 0-019.
LHD0 (10/2/87)	LHC0	8V-165E	VIP - Clutch switch failure.
		9-003B	ISC - Correct Bypass Air Idle speed documentation.
		9-029D	S/W - Correct Closed loop fuel ramp rates.
		9-105	FUEL - Update of fuel computation.
		9-105A	Cancel 9-105.
		9-145	BACKGROUND - BG_MANAGER, load register Watchdog_BG. Refer URD 140
		9-145A	BACKGROUND - Refine 9-145. Refer to URD 140.
		9-146	FUEL - provide more consistent CRANK fuel.
		9V-158	VIP - Correct code in VIP_KAMOUT_O.AST.
		9V-166	VIP - Minimize Vectorfile compare effort.
		9V-200	VIP - enhance continuous EGO.
		9V-200A	VIP - revisions to strategy.
		9V-200B	VIP - Correct documentation error.
		9-226	TRANS - Commonize Converter Clutch Routines.

		9-235	FUEL - INJ timing. CIBETA - TOTAL DELAY_<0 set TOTAL DELAY = 0.
		9-239	TRANS - Flash OCIL light if failure occurs. Refer to URD 419
		9-239A	TRANS - FLASH OCIL if ETV and shift failure occurs. URD 419.
		9-243	FUEL - FN071(MAP) current load input FN1343, 1354, 1355.
		9-251A	S/W - TRANS - byte thrift.
		9V-259A	VIP - S/W thrift. Apply 9-259 to LH. Refer to URD 285.
		9-267	TRANS - Correct shift validation logic. URD 496.
		9V-267A	VIP - Continuous test for Conv. Clutch. Refer to URD 442,478, 496.
		9-268	TRANS - Correct convertor clutch validation. Refer to URD 442.
		9-269	TRANS - Delete VBAT Check in PRNDL convert. Refer to URD 478.
		9-274	TRANS - S/W bullet-proofing. Refer to URD 529.
		9-281	MISC/THRIFT - remove unused code. Refer to URD 540.
		9-282	S/W - MISC-Byte and Time thrift.
		9-283	S/W - MISC-Notification of Memory Overrun.
		9-290	EGR EGRDC calculated & stored before corresponding values of EGRCNT & EGRPRF.
		9V-290A	VIP&S/W - eliminates EGRDC = 0 check.
		9-297	DOCUMENTATION - $SW_RELK --> $SW_RLK.
		9-314	S/W - SYSEQUAT - Temp at start routine uses byte instructions.
		9V-318	VIP & S/W - Replace allign stmts with conditional assembly.
		9V-319	VIP-S/W set VSFMFLG according to C29 KAM BIT in procedure.
		9V-320	VIP & S/W - thrift bytes.
		9-321	TRANS - ETV overcurrent failure flag not setting properly. Refer to URD 531.
		9-324	TRANS - Move setting of GR_DS_TV from Delay Shift Logic to PRNDL Based Desired Gear Determination. Refer to URD 559.
		9V-327	VIP & S/W - fuel pump circuit monitor check function. URD 597.
LHC0 (8/12/87)	LHB1	8V-165	VIP - Provide Continuous clutch switch test.
		8V-165A	VIP - Modify clutch switch test.
		8V-165B	VIP - Remove MTXSW logic.
		8V-165C	VIP - Correct 8V-165B.
		8V-165D	VIP - Add new Continuous clutch test.
		9V-010D	VIP - Correct the SW implementation of 9V-010 - 9-010C.
		9-029C	S/W - Improve implementation of fox functions. Refer to URD #113.
		9V-032	VIP - Add VIP test logic to the E4OD transmission strategy.
		9-049C	Modify Air Charge calculation.
		9V-052A	VIP - Split LU self test strategy into LL, LH, and LD. Refer to URD #96.

		9V-052B	VIP - Change from Dual EGR to Sonic.
		9V-058	VIP - Free up service code 75 for other use. Refer to URD #76.
		9V-058A	VIP - Delete service code 75 from IVSC test.
		9V-058B	VIP - Revise flag names and sense of flags used in KOER brake test.
		9V-129	VIP - SW.
		9V-130	VIP - Restructure Engine-running Self Test Refer to URD #219.
		9V-130A	VIP - Clarify variable names.
		9-131C	S/W - Correct PPCTR initialization.
		9V-132	VIP - Byte and time thrift in Continuous VIP. Refer to URD #139.
		9V-133	VIP - Revise fault filter call.
		9V-147	VIP - Byte thrift.
		9V-165	VIP - Clarify source code for VIP KOER. Refer to URD #299.
		9V-183	VIP - Provide for two types of IDM. Refer to URD #518.
		9-184	Revise CRKTMR logic to supply correct cranking fuel. Refer to URD #323.
		9-184A	Clearly define the CRKTMR strategy.
		9-187	S/W - Protect future assemblies. Refer to URD #317.
		9-197	SYSEQUA - Improve MAP average during large pulsations in manifold. Refer to URD #209.
		9-197A	Cleanup EMR for original.
		9V-199A	VIP - Make VIP compatible with strategy EMR 9-199. Refer to URD #11.
		9-205	SPARK - Revise OSCMOD to apply to decel as well as off idle.
		9-220	Add Transmission Oil Strategy. Refer to URD #309.
		9-220B	Use ECT for TOT if ITOT is out of limits.
		9V-223	VIP - Add TOT sensor tests. Refer to URD #309.
		9V-223A	VIP - Revise parameter and register names.
		9V-224	VIP - Byte thrift. Refer to URD #411.
		9-227	Correct TSLAMU update. Refer to URD #421.
		9V-240	VIP - Boo Test. Refer to URD #436.
		9-251	Add a calibration parameter (TVPMIN) and clip TV_PRES to TVPMIN as a minimum. URD #456.
		9-258	Set FLG_ENG_TV to 0 on F/R and R/F manual shifts. Refer to URD #461.
		9-271	Correct documentation of DSTPBR.
		9-273	Clip daspot to 1.99 or FN882 * FN891, whichever is less. Refer to URD #499.
		9-276	Documentation - Delete reference to VSMFLG and VSFAIL. Refer to URD #506.
		9-278	S/W - Correct temperature _@ start routine. Refer to URD #532.
LHB1 (6/30/87)	LHB0	9-038B	TRANS - Correct misspelling of FLG_SF_AUTO.
		9-144A	FUEL S/W - Correct KAMREF lookups. Refer to URD #174.

LHB0 (6/12/87)	LHA0	9-049	FUEL - Add torque limiting strategy for ETV overcurrent failure. Refer to URD #98.
		9-049B	FUEL - Clarify 9-049.
		9-087B	TRANS - Cleanup ETVOCM logic.
		9-152A	TRANS - Update SPD_RATIO once per BG pass. Refer to URD #358.
		9-155A	TRANS - Do not delay shift logic for manual shifts. Refer to URD #278.
		9-189	TRANS - Revise TV Pressure logic for Reverse engagements. Refer to URD #326.
		9-198	TRANS S/W - Correctly scale parameters. Refer to URD #355.
LHA0 (5/28/87)	LUX0	9-014	TRANS - Delete SIL.
		9V-032	VIP - Add VIP for E4OD.
		9-052	MISC - Split LH from LU. Refer to URD #96.
		9V-052A	VIP - Split LU into LL, LH, and LD.
		9V-052B	VIP - Drop PFEEGR.
		9-052C	EGR - Remove PFE EGR from LH.
		9-087	TRANS - Add ETV failure flags. Split from LD. Refer to URD #98.
		9-087A	TRANS - Add OFMFLG logic.
		9-112B	TRANS - Add failsafe to Dynamic TV logic. Refer to URD #186.
		9-112C	TRANS - Revise failsafe to Dynamic TV logic.
		9-131	FUEL - Add Generic Open Loop fuel. Refer to URD #165.
		9-131A	FUEL - Clean-up for 9-131.
		9-131B	FUEL - Correct CL LAMBSE reset.
		9-144	FUEL - Expend VOLEFF & Adaptive fuel tables. Refer to URDs #174 & #253.
		9-149	TRANS - Combine the tip-out and TV shift delays. Refer to URD #279.
		9-149A	MISC - Correct documentation in 9-149.
		9-150	TRANS - Reverse the sign of the speed ratio check during manual downshifts. Refer to URD #272
		9-151	TRANS - Only delay the manual 1-2 upshifts during power-off mode. Refer to URD #273.
		9-152	TRANS - Modify Converter Clutch routines. Refer to URD #274.
		9-153	TRANS - Revise the back-out upshift logic. Refer to URD #276.
		9-154	TRANS - Revise TV strategy to provide coast boost TV when coast clutch is on. Refer to URD #271.
		9-154A	TRANS - Clear FLG_DEL_MDN when shift is complete.
		9-155	TRANS - Recognize a 4-2/2-4 shift. Refer to URD #278.
		9-156	TRANS - Revise the "Infer Coast Clutch Engaged Logic." Refer to URD #291.
		9-164	TRANS - S/W clean-up. Implement 9-119. Refer to URD #307.

9-164A	TRANS S/W - Clarify 9-164.
9-164B	TRANS S/W - Correct flag code.
9-172	MISC - Clean-up TP-RATCH vs TP_REL substitution. Refer to URD #314.
9-172A	MISC - Clean-up TP-RATCH vs TP_REL substitution. Refer to URD #314.

CHAPTER 2

SYMBOLOGY

SYMBOLOGY

DEFINED PARAMETERS

A defined parameter is a variable or a constant that is defined in EEC-IV source code according to its definition in the strategy parameter dictionary. Defined parameters are represented in the strategy description by identifiers whose alphabetic characters are in the upper case.

STRATEGY SPECIFICATION PARAMETERS

A strategy specification parameter is a variable or a constant that is not defined in the strategy parameter dictionary. Strategy specification parameters are represented in the strategy description by identifiers whose alphabetic characters are in the lower case. These parameters are used only to facilitate the description of strategy function. A strategy specification parameter need not be defined in the EEC-IV source code if the implementation structure does not require it.

The scope of the identifier representing a strategy specification parameter may include more than one strategy module, but it is strictly local to one strategy chapter. A strategy specification variable cannot be used to pass information between strategy modules that execute asynchronously. The value of a strategy specification variable does not persist between repeated executions of any particular strategy module in which it is referenced.

INPUTS

The inputs to most logical operations will be conditional statements of the form:

$$X > Y$$

where, X is a variable (RAM), and Y is a calibration constant, fox function or table look-up, or a mathematical expression. In some cases, Y may also represent a variable.

Typically, six types of conditional statements will appear in the strategy logic diagrams. They are; X > Y, X < Y, X >= Y, X <= Y, X = Y, and X <> Y.

SYMBOL	MEANING
=	EQUAL TO
<>	NOT EQUAL TO
>	GREATER THAN
>=	GREATER THAN OR EQUAL TO
<	LESS THAN
<=	LESS THAN OR EQUAL TO

It should be noted that when the expression X > Y or X < Y is encountered, the conditional statement can be calibrated such that it will never be true, and the appropriate strategy action will never take place. For example, if the variable X has a range of 0 to 255, and the calibration constant in the logical statement, X > Y, is selected to be 255, the statement will always be false. This provides a means for calibrating out certain strategy functions.

When any conditional statement is true, the INPUT STATE to the logical operation is said to be 'TRUE', and is assigned a value of '1'. When the statement is false, the INPUT STATE is 'FALSE', and is assigned a value of '0'.

LOGICAL OPERATIONS

Two logical operations are used, the 'AND' gate and the 'OR' gate. An 'AND' gate is represented by the following symbol:

```
A ---------------------|
                       |AND -| C
B ---------------------|
```

where A and B are INPUT STATES and C is defined as the OUTPUT STATE of the logical 'AND' operation. The value of the OUTPUT STATE is a function of the INPUT STATES as shown in the following truth table:

AND GATE

INPUT STATE A	INPUT STATE B	OUTPUT STATE C
0	0	0
0	1	0
1	0	0
1	1	1

Likewise, the 'OR' gate is represented by:

```
A ---------------------|
                       |OR --| C
B ---------------------|
```

And the OUTPUT STATE for various INPUT STATES is given by the 'OR' truth table:

OR GATE

INPUT STATE A	INPUT STATE B	OUTPUT STATE C
0	0	0
0	1	1
1	0	1
1	1	1

OUTPUTS

The output of all logical operations results in one of two possible paths:

1) The output is an input to another logical operation.

```
A ----------|
            |AND -- C --|
B ----------|           |OR -| E
                        |
D ----------------------|
```

In this case, OUTPUT STATE C is an input to an 'OR' gate. It should be treated like any other conditional statement when determining the value of the final OUTPUT STATE E.

2) ACTION is taken based upon the OUTPUT STATE.

```
A ---------------|
                 |AND -- C --| Energize OUTPUT
B ---------------|           |
                             | --- ELSE ---
                             |
                             | De-energize OUTPUT
```

The ACTION described in the action box is taken when OUTPUT STATE C is 'true'. If an ALTERNATE ACTION is required when OUTPUT STATE C is 'false', the alternate action is described below an ELSE statement in the action box. If no alternate action is required, no ELSE statement will appear.

Multiple "ELSE/ACTION" blocks can appear in a logic diagram in which three or more alternate actions are possible. Consider the following example:

```
A ---------------|
                 |AND -- C --| ACTION #1
B ---------------|           |
                             | --- ELSE ---
D ---------------|           |
                 |AND -- F --| ACTION #2
E ---------------|           |
                             | --- ELSE ---
                             |
                             | ACTION #3
```

The procedure is:

1. Test for ACTION #1. If "C" is true, perform ACTION #1.

2. Otherwise, test for ACTION #2. If "F" is true, perform ACTION #2.

3. Otherwise, perform ACTION #3.

Notes about multiple "ELSE/ACTION" logic:

1. When logic has multiple "ELSE/ACTION" blocks, only one action block can be performed during a program pass. Priority is always top down. In the example, if "C" and "F" are simultaneously true, "C" takes precedence and ACTION #1 is performed.

2. If the final "ELSE/ACTION" block does not have logic as input pointing to it (as in the example), the final action block is performed when no preceeding action block is true. Action is always performed during each program pass with this type of logic.

3. If the final "ELSE/ACTION" block has logic as input pointing to it, the final action block is performed only when no preceeding action block is true and when its input logic is true. Action is not always performed during each program pass with this type of logic.

HYSTERESIS

Hysteresis in a strategy is a situation in which the logic used to make an output true is different from the logic used to make the output false. An example use would be to prevent on/off cycling of an output because of jitter in an input parameter.

Hysteresis is represented in strategy logic diagrams using the following "flip-flop" notation:

```
A ---------------------|S  Q-| C
                       |
B ---------------------|C
```

The actual conditional statements and direction of hysteresis will depend upon the specific application in each strategy module The action of this hysteresis notation is given by the following truth table:

HYSTERESIS FLIP-FLOP

S(SET) A	C(CLEAR) B	Q-OUTPUT C
0	0	no change
0	1	0
1	0	1
1	1	1

The action of the flip-flop can be described as follows:

When the "A" (set) input is true, regardless of the "B" (clear) input level, the flip-flop sets and the "C" output is true. When the "B" input is true and the "A" input is false, the flip-flop clears and the "C" output is false. When "A" and "B" are both false, the "C" output remains unchanged.

ADVICE:

1. Since the intent of a flip-flop is to provide hysteresis, the state of a flip-flop must be remembered from pass to pass. The output is usually defined as a flag.

2. All flip-flops must be serviced every pass through the program, even though some flip-flops are shown in portions of logic that may not execute each pass. The normal practice is to lump and service the flip-flops together at the beginning of a routine. This guarantees that all flip-flop outputs will reflect the correct state based on current input conditions. Then, when logic refers to a flip-flop, the logic only needs to look at the flag which represents the state of the flip-flop.

3. Clear (0) is the default initial value of the output flag for a flip-flop. The strategy specification must explicitly state if the initial value should be set (1).

4. The set input always takes precedence over the clear input. When both are true, the flip-flop output should set. In some instances, the software practice has been to perform the clear logic first, followed by the set logic. The procedure may initially clear the flag and then reverse the decision later. This practice could cause problems if the flip-flop output flag is tested during an EOS interrupt because the EOS can catch the flag in the wrong state.

   The flip-flop procedure should always be:

```
IF set condition met
     THEN set flip-flop output flag
ELSE IF clear condition met
     THEN clear flip-flop output flag
ELSE
     No change to flip-flop output flag
ENDIF
```

Some flip-flops are specified without a hysteresis term to save bytes; this is the preferred method of presenting flip-flops. In this case there are two calibration parameters, one for the set condition, and one for the clear condition. When flip-flops are specified this way, the calibration parameter used to set the flip-flop will end in _SH or _SL. The parameter used to clear will end in _CH or _CL. The H or L determines the larger (H) or smaller (L) parameter.

Example:

```
DEL_LOAD < DLOD_SL ----|S  Q-| C
                       |
DEL_LOAD >= DLOD_CH ---|C
```

```
               1.0 |
                   |
DEL_LOAD           |          /|\                |    o = state change
                   |           | clear region    |
        DLOD_CH    |    -------o-----------------|--------------
                   |           |                 |
                   |           |                 |
                   |           | no change  (hysteresis)
        DLOD_SL    |    -------|-----------------o--------------
                   |           |                 |
                   |           |  set region    \|/
                   |
               0.0 ---------------------------------------------------------
```

HOW TO INTERPRET THE HEXADECIMAL REPRESENTATION OF A REGISTER CONTAINING BIT FLAGS

1) Display the register on the calibration console.

2) Press the 'HEX' button on the calibration console to display the register in hexadecimal format.

3) Two hexadecimal digits will be displayed. In order to determine which bit flags are set, use the following hexadecimal-to-binary conversion chart:

HEXADECIMAL DIGIT DISPLAYED	BINARY EQUIVALENT STRING
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
A	1010
B	1011
C	1100
D	1101
E	1110
F	1111

The LEFTMOST hexadecimal digit represents the state of bit flags in bit positions 7 through 4. The RIGHTMOST hexadecimal digit represents the state of bit flags in bit positions 3 through 0.

Consider the following: The 8-bit binary string representing the leftmost and rightmost hexadecimal digits together, read from left to right, represents the state of bit flags in bit position 7 through bit position 0.

EXAMPLE:

You want to examine the state of NDSFLG, a bit flag in bit position 5 of a register which is at address B0.

1) Display the contents of B0 on the calibration console.

2) Display the contents in hexadecimal format.

3) The hexadecimal value '2F' is displayed.

HEXADECIMAL	2				F			
BINARY	0	0	1	0	1	1	1	1
BIT POSITION	7	6	5	4	3	2	1	0

This means that bit flags in position 5 (NDSFLG), 3, 2, 1 and 0 are set to 1.

THE "DO:" STATEMENT

The DO: statement is used to call a logic subroutine in strategy. It is not used, as it often is by software, to avoid repeating a frequently used piece of code. It is primarily used to allow a process to be broken up into smaller pieces so that the strategy can be easily represented without resorting to large, cumbersome, confusing pieces of logic. The DO: statement can also provide an "executive routine" or calling structure for a series of sequential pieces of logic.

The DO: statement can call either an entire strategy module or strategy process. A strategy module is a segment of strategy that has an overview, definitions and a process and usually corresponds to a software module. A strategy process is a subset of a strategy module, usually used to conveniently break up a logic diagram in a strategy book. Do not use the version type extension for strategy modules, i.e. COM1, COM5. This makes the calling module independent of various module varieties. The syntax is:

```
DO: DSDRD_GR MODULE  or  DO: RPM LIMITER PROCESS
```

The DO: statement is used on the "ACTION" or right hand side of logic diagram.

The format for the called subroutine (Module or Process) is that it must have an explicitly stated start and end point. This shows that the piece of strategy contained within the start and end points is executed as a subroutine, not necessarily every background loop through the strategy. The syntax is:

```
 BEGIN: DSDRD_GR MODULE              BEGIN: RPM LIMITER PROCESS
           .                                   .
           .                                   .
           .                                   .
CONTINUE: DSDRD_GR MODULE           CONTINUE: RPM LIMITER PROCESS
      (if required)                         (if required)
           .                                   .
           .                                   .
           .                                   .
  END: DSDRD_GR MODULE                 END: RPM LIMITER PROCESS
```

The CONTINUE: syntax is used to identify the continuation of a logic diagram on to a page that does not have either a START: or an END: statement. The intent is to have any page of a strategy book be able to be identified as being part of a subroutine or not. For example, a logic diagram that takes up 3 pages will begin with a START: statement on the first page, begin with a CONTINUE: statement on the second page and end with an END: statement on the third page.

At the end of the module or process, there is an implicit "RETURN" statement, that is, the next execution step of the strategy must return to the place where the DO: statement was called. For example, if a DO: statement is executed in the middle of an ELSE/ACTION block, the strategy continues execution after returning to the originating point.

```
A ---------------------|
                       |AND -| ACTION 1
B ---------------------|     |
                             | --- ELSE ---
C ---------------------|     |
                       |AND -| DO: XYZ PROCESS ------> BEGIN: XYZ PROCESS
D ---------------------|     |                 <--|              .
                             | --- ELSE ---       |              .
                             |                    |              .
                             | ACTION 2           |      END: XYZ PROCESS
                                                  |
                                                  |              |
                                                  |--------------|


E > F -----------------------| ACTION 1
                             |
                             | --- ELSE ---
                             |
                             | ACTION 2
```

If the first AND gate containing "A" and "B" is false and the second AND gate containing "C" and "D" is true, then the XYZ PROCESS is called and executed. After returning to the DO: statement, strategy execution continues with the evaluation of the "E > F" condition in the second ELSE/ACTION block.

DO: statements can be nested, that is, one DO: statement can call a subroutine which contains another DO: statement within itself. There are no restrictions on this other than the basic rule continues to apply: Each process or module that is executed must return to the originating point upon its execution.

Flip flops should not be included within a DO: statement since they must be evaluated every background loop. The strategy designer must have the flip flops shown as being executed every background loop and use the appropriate flop flop output flags within the DO: statement.

The "ROLAV" Statement

The ROLAV statement is used to invoke the rolling average routine in the strategy. The EEC-IV filters inputs using this rolling average routine. This is a difference equation implementation of a first order low pass filter. The filter behaviour is defined by the discrete time solution to the first order differential equation and takes the form:

$$f = 1 - e^{-(FK\_TMR/TC)}$$

Where "f" is called the filter constant in the difference equation:

$$new\_average = f * new\_value + (1 - f) * old\_average$$

Using the first two terms of the series expansion:

$$e^{x} = 1 + x + \frac{x^2}{2!} + \cdots$$

Simplifying, the exponential "f" becomes:

$$f = 1 - e^{-(FK\_TMR/TC)} = 1 - \frac{1}{e^{(FK\_TMR/TC)}} = \ldots$$

$$f = 1 - \frac{1}{1 + (FK\_TMR / TC)} = \frac{1}{1 + (TC / FK\_TMR)} .$$

The time constant (TC) is a function of the input being filtered. It is calibratable. Generally, a longer time constant filters more heavily, but also introduces more time lag into the signal.

For most filters, the sampling period (FK_TMR) will equal the background loop time (BG_TMR). In those case where the sample period is not equal to the background loop time, the true sample period is to be passed to the rolling average routine. See the calling convention below.

The ROLAV statement is used on the "ACTION" or right hand side of a logic diagram. The strategy will specify calls to the rolling average routine using the following convention:

```
condition -------------| new_average = ROLAV(new_value,TC[,FK_TMR])
```

Where:
- old_average = Current value of new_average prior to filtering. This parameter is implicit in the call to the rolling average routine.
- new_average = Output value of rolling average filter. This parameter becomes the old_average on the next filtering event.
- new_value = Input value to be filtered.
- TC = Time constant.
- FK_TMR = Elapsed time between successive calculations. This is an optional argument only to be specified if it is different from background loop time.

The "ABS" Statement

The ABS function returns a value which is the absolute value of parameter x. ABS has the following form:

```
ABS(x)
```

The parameter x cannot be of flag type.

The "CLIP" Statement

The CLIP function returns the value x clipped between the range low and high. low is the lower limit and high is the upper limit. low and high MUST be specified in the given order. CLIP has the following form:

```
CLIP( x, low, high)
```

EXAMPLE:

```
y = CLIP( x, low, high)
```

This is equivalent to the following logic chart.

```
x > high ---------------------| y = high
                              |
                              | --- ELSE ---
                              |
x < low ----------------------| y = low
                              |
                              | --- ELSE ---
                              |
                              | y = x
```

NOTE:

If the limits high and low are calibrated such that low >= high, then the output will be high. The high limit ALWAYS take priority.

The "MAX" Statement

The MAX function returns the maximum value of a list of parameters x1,...xn. MAX has the following form:

MAX(x1,...,xn)

The parameters x1 through xn cannot be of flag type.

Example:

y = MAX(x, 10)

This is equivalent to the following logic chart.

```
x < 10 -----------------------| y = 10
                              |
                              | --- ELSE ---
                              |
                              | y = x
```

The "MIN" Statement

The MIN function returns the minimum value of a list of parameters x1,...xn. MIN has the following form:

MIN(x1,...,xn)

The parameters x1 through xn cannot be of flag type.

Example:

y = MIN(x, 15)

This is equivalent to the following logic chart.

```
x > 15 -----------------------| y = 15
                              |
                              | --- ELSE ---
                              |
                              | y = x
```

The "SQR" Statement

The SQR function returns a value which represents the square of the parameter x. SQR has the following form:

```
SQR(x)
```

The parameter x cannot be a flag type.

The "SQRT" Statement

The SQRT function returns a value which represents the square root of the parameter x. SQRT has the following form:

```
SQRT(x)
```

The parameter x cannot be a flag type, and MUST NOT be NEGATIVE.

The "pid_def()" Construct

A PID is a way to access parameter values by name via a communication protocol between the Powertrain Control Module and an outside requestor.

Parameter values from within the PCM can be accessed from the outside environment via the communication network. This is done by the generic scan tool as required by CARB-OBDII regulations to obtain information in reference to OBDII system monitors and their status. This can also be done by SBDS or other systems capable of establish and sustaining communications using the SCP.

CARB regulations require the availability of a number of PIDs regarding the state of the on board diagnostic system. These, as well as the Ford specific PIDs, will be defined via the strategy documents in a similar manner. A construct will be used in the strategy documents to define the data conveyed in a PID request. That construct is the:

```
pid_def(pid_number, description)
```

Where:

"pid_number" is the symbol used to refer to the PID. For example, all PIDs defined by SAE standard J1979 will by of the form j1979_xx[_xxx]. The portion inside the brackets [] is optional. Other conventions will be used for j2190 or SBDS specific PIDs.

"description" is the actual data to be used in the transmission of the PID. This could reference a register (RAM, KAM or ROM) that is maintained in the strategy. For example ECT or TEST_SW. This could also be an equation used to compute the PID. This equation may or may not reference other internal registers in the strategy. An example of this would be (((ECT - 32) * 5/9) + 40). The PID may also be a bit map of a number of flag bits. In this case the following convention is used for the "description" parameter:

```
pid_def(xxxx_xx_xxx,(b0: FLG_1,
                     b1: FLG_2,
                     b2: 1,
                     b3: 0,
                     b4: FLG_3,
                     b5: FLG_4,
                     b6: 0,
                     b7: 0)
```

Whenever a parameter is referenced in the "description" and is in lower case in the pid_definitions context, logic will be present in the context that describes how that parameter is to be determined. The pid_definitions context is delineated by a "BEGIN: pid_defintions" statement and a "END: pid_defintions" statement. The pid_defintions context is located in the module in the strategy that is most relavent to the PIDs being defined. The pid_definitions contexts will therefore be distrubuted throughout the strategy. See the following for an example of a module with PID definitions:

The "pid_def()" constructs do not imply anything with respect to implementation. It is left up to the best judgement of the software designer as to how to implement the PID scheme.

MODULE NAME

OVERVIEW

Put overview text here. . .

DEFINITIONS

Put definitions text here. . .

PROCESS

STRATEGY MODULE: STRATEGY_MODULE_COM1n

```
BEGIN: pid_definitions
 ;comment describing that this section is defining PIDs.

REGISTER1 = Y --------------|
                            |AND -| flag1 := 1
REGISTER2 = X --------------|     |
                                  | --- ELSE ---
                                  |
                                  | flag1 := 0

unconditionally ------------------| parameter1 := (REGISTER3 * M) + B

pid_def(j1979_xx_xxx, REGISTER4)

pid_def(j2190_xx, parameter1)

pid_def(sbds_xx_xxx, FNnnn(REGISTER5))

pid_def(xxxx_xx_xxx, (REGISTER6 * M) + B))

pid_def(xxxx_xx, bo: flag1,
                 b1: FLG_1,
                 b2: 0,
                 b3: 1,
                 b4: FLG_2,
                 b5: CAL_SW_1,
                 b6: 0,
                 b7: 0)

END: pid_definitions
```

Continue with the control strategy logic here. . .

The "send()" Construct

The SCP interface strategy allows for numerous messages to be supported by the Powertrain Control Module and the vehicle network databus.

These messages can be sent from various locations in the strategy, depending on their functional intent, to be received by one or more recieving nodes on the bus.

In order to provide a uniform method of sending an SCP message, a construct will be used in the strategy documents to define the data to be transmitted in an SCP message. That construct is the

```
send(msg_name: scp_data_[])
```

Where:

"msg_name" is the name of the particular message to be sent. For example, a response to an SAE standard J1979 request might be named REPORT_OBDII_PID.

"scp_data_[]" is the actual data to be sent in the message transmission. Since there are up to seven (7) data bytes available to be sent in each message, the values of scp_data_[] could all be defined. These values could be either register names, or constant hex values, or even a PID which was defined in the "pid_def()" construct. While not all seven of the scp_data_[] values are required to be defined in a "send()" command, the construct does not allow for data to be missing in between defined values.

An example of the use of the "send()" construct follows:

```
send(REPORT_OBDII_PID: 41h, 01h,
                       j1979_01_011,
                       j1979_01_012,
                       j1979_01_013,
                       j1979_01_014)
```

..which tells the software designer to send a message called REPORT_OBDII_PID over the network with the following defined values:

```
scp_data_1 = 41h
scp_data_2 = 01h
scp_data_3 = the value defined in pid_def(j1979_01_011,...)
scp_data_4 = the value defined in pid_def(j1979_01_012,...)
scp_data_5 = the value defined in pid_def(j1979_01_013,...)
scp_data_6 = the value defined in pid_def(j1979_01_014,...)
```

NOTES: SCP messages which contain no variable data bytes are to be sent from the strategy without the "scp_data_[]" portion defined.

The priority/type and target specifier bytes are unique values for each message, and thus are NOT defined in the "send()" construct. As a result, all messages referenced by "send()" must also be defined in the SCP interface strategy for the software designers' reference.

The "send()" constructs do not attempt to imply software implementation, but rather a means of communicating information to the software designer to transmit over the SCP network.

CHAPTER 3

EEC OVERVIEW

ELECTRONIC ENGINE CONTROL SYSTEM OVERVIEW

The Electronic Engine Control system is intended to provide a more optimum engine control strategy than is possible through a strictly mechanical system. This is accomplished by using a microprocessor which interprets input data from a number of engine parameter sensors, and based on a control strategy in the microprocessor's program chips, generates output control signals to a number of actuators.

The control strategy is divided into two segments, an engine control strategy, and self test diagnostics. The diagnostics will be discussed in another section. The engine control strategy is segmented into three principal modes:

-CRANK
-UNDERSPEED
-RUN

The strategy description and the entrance and exit conditions for CRANK/UNDERSPEED/RUN are shown on the following pages. RUN is of particular interest because it contains the control logic for most engine operating regions. The RUN strategy is further broken down into three modes to facilitate optimum control. Based primarily on throttle position, they are:

-CLOSED THROTTLE
-PART THROTTLE
-WIDE OPEN THROTTLE

The specific entrance and exit conditions for these modes are described in the throttle mode selection section.

The remainder of this document describes the normal engine control strategy (RUN) for the various outputs, including fuel, spark, EGR, thermactor air, and idle speed control (ISC). It also contains the utility functions, filters, ratchets, and timers, and a parameter dictionary of calibration constants, fox functions and tables.

HARDWARE CALIBRATION SWITCHES

The LH strategy is an EFI, speed density strategy designed to handle any vehicle speed or non-vehicle speed engine application. Hardware complexity is taken into account via a set of user accessible software calibration switches. These switches are detailed below:

- BIHP - If a brake on/off switch is present, set BIHP = 1, otherwise set BIHP = 0.

- CANPHP - If EEC controlled canister purge hardware is present, set CANPHP = 1, otherwise set CANPHP = 0 to bypass the canister purge logic.

- DOLHP - If Data Output Link is being utilized, set DOLHP = 1, otherwise set DOLHP = 0 to bypass the DOL fuel calculation.

- GOVHP - If output to stand-alone governor is present, set GOVHP = 1; otherwise set = 0.

- PFEHP - If a Sonic EGR system is being used, set PFEHP = 0. Sonic EGR systems utilize an EVR (Electronic Vacuum Regulator) output. In order to conserve memory, registers and fox functions have been scaled such that both the PFE and Sonic EGR logic can share calibration locations. For example; both EGR strategies use FN219, FN221, and FN239. Also, the EVR output routine, EGR enable/disable logic, and the desired EGR rate are common software code segments. If PFEHP is set to 1 or 2, then the EGR Strategy is always disabled.

- PSPSHP - If a power steering pressure switch is present, set PSPSHP = 1, otherwise set PSPSHP = 0. PSPSHP is used in the Idle Speed Control logic.

- THRMHP - If Thermactor air pump hardware is present, set THRMHP = 1, otherwise set THRMHP = 0. Also, the following logic sets CHKAIR = 1 for proper function of the Closed Loop/Open Loop fuel logic when Thermactor Air is not used.

```
  THRMHP = 0 ----------------------------| CHKAIR = 1
                                         |
                                         | --- ELSE ---
                                         |
                                         | Do NOT modify CHKAIR
                                           (Set within Thermactor logic)
```

- TSTRAT = Transmission Strategy switch - 0 -> No Transmission Control; 1 -> Shift Indicator Light Control; 2 -> A4LD with Vehicle Speed Sensor; 4 -> C6E4 Electronic Transmission Control.

- TRLOAD = 0 Manual trans, no clutch or gear switch, forced neutral (NDSFLG = 0); TRLOAD = 1 Manual trans, no clutch or gear switch; TRLOAD = 2 Manual trans, one clutch or gear switch; TRLOAD = 3 Manual trans; TRLOAD = 4 Auto trans, non-electronic, Neutral Drive Switch; TRLOAD = 5 Auto trans, non-electronic, Neutral Pressure Switch (AXOD); TRLOAD = 6 Auto trans, electronic, PRNDL sensor Park, Reverse, Neutral, Overdrive, Manual2, Manual1 configuration.

- IMS - If no IMS (Inferred Mileage Sensor) hardware is present, IMS defaults to a value of 1 (1 -> no IMS hardware or high mileage). IMS is referenced in the EGR and Thermactor Air logic.

- VSTYPE - If a Vehicle Speed Sensor is present, set VSTYPE = 1, otherwise set VSTYPE = 0.

DCL HARDWARE PRESENT SWITCHES

Hardware present switches define the types of devices connected to the EEC module through the wiring harness. (See also DCL Chapter)

```
LINK_SW 00 -> UART MODE disabled / no DOL, no DCL
        01 -> UART MODE disabled / DOL present, no DCL
        02 -> UART MODE disabled / no DOL , DCL present
        03 -> UART MODE disabled / DOL present, DCL present
        04 -> UART MODE enabled / no DOL, no DCL
        05 -> UART MODE enabled / DOL present, no DCL
        06 -> UART MODE enabled / no DOL, DCL present
        07 -> UART MODE enabled / DOL present, DCL present
```

NOTE: All applications, except FN-9, should have UART mode enabled.

INITIALIZATION ROUTINE

After power is first applied, the software initializes all of the Read/Write (RAM) registers before executing the strategy. All RAM registers are set equal to zero unless initialized to another value (either set value or a calculated value) as shown below. NOTE: The parameters are not necessarily initialized in the order shown.

RAM/FLAG	INITIAL VALUE
A3CTMR	FFFF (63.99 SEC)
ACT	60.00 DEG F
AEMAP	MAP
APT	-1.0 (CLOSED THROTTLE)
BP_INTR	1.0
CFIEPT	650.0 COUNTS
CONPR	370.0 COUNTS
CRKFLG	1.0 (CRANK MODE)
ECT	60.0 DEG F
EGRBAR	307.0 COUNTS
EOFF	307.0 COUNTS
EPTBAR	650.0 COUNTS
FIEPT	650.0 COUNTS
GEAR_CUR	1.0
GEAR_OLD	1.0
GR_CM	1.0
GR_CM_LST	1.0
GR_DS	1.0
GR_DS_LST	1.0
GR_OLD	1.0
IEGR	307.0 COUNTS
INJ_PIP_CNT	1.0
IPDL	7.0
ISCMOD	1.0
LAM_OLD	1.0
LAMAVE	1.0
LAMBSE	1.0
LAMMUL	0.996
LOACT	245.0 DEG F
MAP	27.0 IN. HG
MAPBAR	27.0 IN. HG
MAP_FREQ	150.0 HZ
MAP_WORD	27.0 IN. HG
MKAY	1.0
MULTMR	255.0 SEC
NOV_ACT	NVBASE
NOV_ACT_LST	NVBASE
OLFLG	1.0 (OPEN LOOP)
PDL	7.0
PDL_LST	4.0
PPCTR	PIPNUM
PUMP	1.0
RANNUM	8193
RATCH	RACHIV (250 COUNTS)
RLKCTR	1.0
RT_GR_CUR	GRRAT1
RT_GR_OLD	GRRAT1
RT_NOVS	1.0

INITIALIZATION ROUTINE (CONTINUED)

RAM/FLAG	INITIAL VALUE
SAF	10.0 DEG. BTDC
SYNCTR	1.0
TBART	RACHIV (250 COUNTS)
TLSCTR	FFFF (ALL BITS EQUAL ONE)
TPBAR	RACHIV (250 COUNTS)
TPBART	RATIV
TPBARTC	RATIV
TPBARTV	RATIV
TSLMPH	0.249 SEC
UNDSP	1.0 (UNDERSPEED)
VBAT	12.5 VOLTS
WINDOW_BETA	0.95

CHAPTER 4

CRANK/UNDERSPEED/RUN MODE SELECTION

CRANK/UNDERSPEED/RUN MODE SELECTION

The EEC-IV strategy operation is divided into three distinct strategy segments. These are:

1) CRANK
2) UNDERSPEED
3) RUN

The CRANK mode is entered after a power-up initialization or after an engine stall. CRANK employs a special strategy to aid engine starting. When the CRANK logic first becomes false, the UNDERSPEED mode is entered. The UNDERSPEED mode employs a special spark and fuel strategy in place of the normal engine control strategy (RUN). After start, the RUN mode is entered and the normal engine control strategy is executed. If the engine stumbles during RUN mode, the UNDERSPEED mode can again be entered to help recover from the stumble and prevent a stall.

The specific strategies are:

CRANK STRATEGY

Fuel	Energize all injector ports simultaneously every CRKPIP PIPS on PIP Falling Edges. Injector Synchronization Logic is disabled. See the BACKGROUND FUEL PULSEWIDTH CALCULATION section of the FUEL Chapter for a description of the FUELPW calculation.
Spark Advance	10 degrees BTDC (on PIP signal)
Thermactor Air	bypass
EGR	disabled
Purge	disabled
ISC	disabled (0% duty cycle if N = 0, 100% duty cycle if N <> 0)

UNDERSPEED STRATEGY

Fuel	Energize all injector ports in the same manner as in the RUN mode, referenced to PIP Rising Edges. See the FUEL Chapter for the FUELPW calculation.
Spark Advance	10 degrees BTDC (on PIP signal)

Other outputs are the same as the RUN mode.

RUN STRATEGY

Injector Synchronization Logic is enabled if SIGNATURE PIP distributor is present. (See the FUEL Section)

The normal engine control strategy is described in the remainder of this book.

CRANK/UNDERSPEED/RUN MODE SELECT

DEFINITIONS

INPUTS:

Registers:

- ECTCNT = Number of times ECT sensor input was read.
- N = Engine RPM.
- PIPCNT = Number of PIPs which have occurred.
- TSLPIP = Time since last PIP occurred, msec.

Bit Flags:

- CRKFLG = Flag indicating engine mode. 1 -> cranking; 0 -> run or underspeed mode.
- FIRST_PIP = Bit Flag set to 1 if First PIP has been received.

Calibration Constants:

- CRKPIP = Number of PIPs between injector firing.
- IGN_TYPE = Inidcator of ignition type (0 = TFI, 1 = TFI_ICCD, 2 = LDR-DIS).
- NCNT = Minimum number of PIPs necessary to exit CRANK Mode.
- NRUN = Minimum Engine Speed to exit CRANK Mode.
- NSTALL = Engine Stall speed to re-enter CRANK Mode.
- STALLN = Stall RPM: If the first RPM calculated is greater than this value assume that there was a reinit.
- UNRPM = Underspeed Engine Speed.
- UNRPMH = Hysteresis term for UNDERSPEED Mode.

OUTPUTS

Registers:

- ECTCNT = See above.
- INIT_TOT = Temperature of transmission oil at start-up, deg. F.
- N = See above.
- PIPCNT = See above.
- RUNUPTMR = Time since RUNUP_FLG was set, sec.

- TCSTRT = Temperature of Engine Coolant at Startup, deg F.

Bit Flags:

- CRKFLG = See above.
- FLG_STALL = Flag indicating a stall has occurred; transition from underspeed/run to crank.
- REFLG = Reinit flag: 1 -> reinit occurred; 0 -> no reinit.
- RUNUP_FLG = Flag indicating that Runup is complete; 1 -> Runup complete.
- UNDSP = Flag indicating engine mode: 1 -> cranking or underspeed, 0 -> run mode.

CRANK/UNDERSPEED/RUN MODE SELECTION LOGIC

```
CRKFLG = 1 ---------------|
 (Crank mode)             |     | CRANK MODE
                          |AND -| CRKFLG = 1
N <= NRUN ----------|     |     | UNDSP = 1
                    |     |     | FLG_STALL = 0
ECTCNT < 8 ---------|OR --|     |
                    |           |  --- ELSE ---
PIPCNT < NCNT ------|           |
                                | A Stall has occurred
CRKFLG = 0 ---------------|     |
 (Run or Underspeed mode) |AND -| CRKFLG = 1
                          |     | UNDSP = 1
N < NSTALL ---------------|     | ECTCNT = 0
                                | TCSTRT = 0
                                | INIT_TOT = 0
                                | FLG_STALL = 1
                                | RUNUP_FLG = 0
                                | RUNUPTMR = 0
                                |
                                | --- ELSE ---
                                |
CRKFLG = 1 ---------|           | UNDERSPEED Mode
                    |OR --|S Q--| CRKFLG = 0
N < UNRPM ----------|     |     | UNDSP = 1
                          |     | FLG_STALL = 0
N > UNRPM + UNRPMH -------|C    |
                                |  --- ELSE ---
                                |
                                | RUN Mode
                                | CRKFLG = 0
                                | UNDSP = 0
                                | FLG_STALL = 0
```

PIP COUNTER AND ECT COUNTER CONTROL LOGIC

```
CRKFLG = 1 -------------|
 (CRANK mode)           |AND -| Count PIP signals as they occur
                        |     |   (PIPCNT is the counter)
N > NRUN ---------------|     |
                              | --- ELSE ---
                              |
                              | Stop counting PIP signals
                              |   PIPCNT = 0
```

NOTE: If the PIP period (time elapsed since the last PIP signal) becomes >= 800 msec, the engine speed RPM is set to zero. This insures that if the PIP signal goes away because of a stall, RPM will become zero to trigger CRANK mode.

ENGINE RUNNING REINIT STRATEGY

The reinit strategy attempts to differentiate an engine running reinit from a normal start engine run-up. After a reinit, a "first RPM" is calculated from the first two PIP rising edges. If the calculated RPM is greater than idle RPM, then a reinit is assumed.

```
"first RPM calculation" >= STALLN -------| REFLG = 1
                                           (Set reinit flag)


TSLPIP >= 800 msec ----------------------| N = 0
                                         | FIRST_PIP = 0
                                         | REFLG = 0
```

When the engine is not moving and a LDR-DIS ignition system is used, the SPOUT output from the EEC computer to the ignition module should be held in the HIGH state to prevent the coil from charging.

```
IGN_TYPE = 2 ---------------------|
 (LDR-DIS system)                 |
                                  |AND -| SPOUT to high
TSLPIP > 800 msec ----------|     |
 (no PIPs in long time)     |OR --|
                            |
FIRST_PIP = 0 --------------|
 (no PIPs yet)
```

CHAPTER 5

THROTTLE MODE SELECTION

THROTTLE MODE SELECTION

OVERVIEW

The throttle mode scheduler is used to determine what engine operating region is currently extant. The variable APT (At Part Throttle flag) is used to indicate throttle mode and is assigned the following values:

THROTTLE MODE	APT
Closed Throttle	-1
Part Throttle	0
Wide Open Throttle	1

The value of APT is determined by the logic shown on the following page. Briefly, throttle angle breakpoints, in terms of counts, are used to define the CLOSED/PART THROTTLE and PART/WIDE OPEN THROTTLE transitions. Hysteresis is incorporated in both breakpoints to prevent jitter between modes.

TP_REL is a parameter which indicates the amount of throttle movement beyond the closed throttle/idle setting. TP_REL is computed by subtracting RATCH from TP. Larger values of TP_REL indicate wide open throttle, smaller values of TP indicate part throttle, and near zero TP_REL indicates closed throttle. The variable RATCH is the output of a ratchet algorithm which continuously seeks the minimum throttle angle corresponding to a CLOSED THROTTLE position. This alleviates the necessity to set the throttle position sensor at an absolute position and compensates for system changes and differences between vehicles. The ratchet algorithm uses filtered throttle position for the determination of RATCH. RATCH is initialized to the non-calibratable value RATIV, currently set to 250 counts.

A more detailed explanation of the throttle position ratchets and throttle position filter is contained in the SYSTEM EQUATIONS section.

DEFINITIONS

INPUTS

Registers:

- APT = Throttle Mode Flag; -1 -> closed throttle, 0 -> part throttle, 1 -> wide open throttle.

- TP_REL = Relative Throttle Position.

Bit Flags:

- CRKFLG = Flag indicating engine mode; 1 -> cranking, 0 -> run or underspeed mode.

Calibration Constants:

- DELTA = CT/PT Breakpoint Value above RATCH.

- HYST2 = Hysteresis term to enter WOT Mode.
- HYSTS = Hysteresis term to exit Closed Throttle Mode.
- THBP2 = PT/WOT Breakpoint Value above RATCH.

OUTPUTS

Registers:

- APT = Throttle Mode Flag; -1 -> closed throttle, 0 -> part throttle, 1 -> wide open throttle.
- CTPTFG = Closed throttle to PT/WOT transition flag.
- PTSCR = Part throttle mode since exiting CRANK flag.

PROCESS

The logic described below considers the current position of the throttle and compares that value to the RATCH, Closed Throttle, plus the change in throttle position from the last setting. If both flip-flops in the logic clear, then Part Throttle is set.

```
TP_REL <= DELTA ----------------|S Q--| CLOSED THROTTLE MODE
                                |     | APT = -1
TP_REL >= DELTA + HYSTS --------|C    |
                                      | --- ELSE ---
                                      |
TP_REL > THBP2 + HYST2 ---------|S Q--| WIDE OPEN THROTTLE MODE
                                |     | APT = 1
TP_REL <= THBP2 ----------------|C    |
                                      | --- ELSE ---
                                      |
                                      | PART THROTTLE MODE
                                      | APT = 0
```

```
Previous APT = -1 -------------|
                               |
Current APT <> -1 -------------|AND -| CTPTFG = 1
                               |     |  (Closed Throttle to Part/WOT
CRKFLG = 0 --------------------|     |   transition)
                                     | PTSCR = 1
                                     |  (Part Throttle since Crank)
                                     |
                                     | --- ELSE ---
                                     |
                                     | CTPTFG = 0
```

NOTE: PTSCR and CTPTFLG are initialized to zero.

CHAPTER 6

FUEL STRATEGY

OVERVIEW

The purpose of the Fuel Control Strategy is to provide fuel to the engine in appropriate quantities to achieve the desired Air/Fuel ratio in the combustion chambers. The desired A/F ratio is determined by the fuel control strategy and calibration for all operating conditions. It can either be a predetermined value that is calibration dependent and can vary with engine operating conditions (Open Loop Control); or the EEC may ramp the value up and down in a limit cycle to maintain an average stoichiometric mixture, as determined by the EGO sensor (Closed Loop Control).

The fuel control actuators, or fuel injectors, consist of a solenoid and metering needle or pintle which is moved off a seat by energizing the solenoid, thus releasing fuel through a nozzle. Each cylinder has an injector installed in the intake manifold to direct fuel toward the intake valve. The length of time the solenoid is energized (pulsewidth) determines the amount of fuel delivered.

Fuel is supplied to the injectors by a high pressure electric fuel pump, controlled by the EEC. Fuel supply pressure is modulated by a regulator sensing MAP to maintain the pressure differential across the injectors constant.

A group of 2,3 or 4 injectors are energized simultaneously by a single output port of the processor. If only one output port is used, all injectors are energized every PIPOUT pips. If two output ports are used, they can be energized individually in an alternating manner, with each port being energized PIPOUT/NUMOUT pips after the other port, or simultaneously every PIPOUT pips. The calibration parameter OUTINJ selects the injection scheme.

Except while in CRANK mode, until the #1 cylinder is identified from a Signature Pip System, or if a Signature Pip System is not used, the injectors are energized on the rising edge of PIP. If it is desired to optimize the fuel delivery timing relative to intake valve opening, the injectors must be grouped into banks with each bank containing cylinders in consecutive spark firing order. Bank "A" must contain cylinder #1. This provides a time (or window) during the engine cycle when all intake valves for a given injector bank are closed. After synchronization, the injections can be delayed to occur after the rising edge of the reference (signature) pip signal. CIBETA (in pip periods) is the delay. CIBETA should be calibrated to cause the injection during the closed valve window.

Except while in CRANK mode, the amount of air entering the engine is divided by the desired A/F ratio to obtain the desired fuel flow. The desired A/F ratio is expressed in terms of lambda, where lambda is the desired A/F ratio divided by 14.64, the chemically correct ratio for complete combustion (stoichiometry). This desired fuel flow is then converted to a pulsewidth for the injectors, based on the engine RPM, number of cylinders, injector flow characteristics, number of injectors and the desired injection frequency.

In CRANK mode, since the airflow measurement is unreliable, the injector pulsewidth is based on ECT, ACT, MAP and elapsed time in CRANK. Also, to maximize the voltage available to activate the injectors, they are energized on PIP Falling Edges.

The strategy has the ability to modify the fuel pulsewidth to account for intake manifold wall wetting (TRANSIENT FUEL), injector flow reduction due to high temperatures (HICOMP/A0COR), or while operating in UNDERSPEED mode. The pulsewidth is also modulated during idle to stabilize idle RPM on Speed Density systems (ISCMOD). The fuel can also be turned off during decelerations (DFSO). Asynchronous with the main pulses, additional pulses can be issued to account for the manifold filling effect during throttle openings (AEFUEL).

The strategy also has the ability to adjust the fuel pulsewidth just prior to energizing the injectors. The base pulsewidth is calculated in the background sequence, using the then available MAP value. It may be a significant time period before that pulsewidth is actually used, and MAP (and airflow) may have changed. MAP is updated in the foreground on PIP up edges prior to performing the fuel logic. For small changes in MAP, airflow can be approximated as a linear function of MAP. Therefore, a more accurate fuel pulsewidth can be obtained by adjusting the pulsewidth by the ratio of the most current MAP value to the value used to calculate airflow. This technique is called "Foreground Fuel".

FUEL MODE DESCRIPTION

The purpose of the FUEL_MODE module is to determine the fuel control mode (Open Loop/Closed Loop) and the value of LAMBSE. The fuel control strategy consists of 3 mutually exclusive modes:

OPEN LOOP (OLFLG = 1)
CLOSED LOOP (OLFLG = 0)
SELF TEST OPEN LOOP (OLFLG = 1)

OPEN LOOP MODE

During open loop operation, the computer calculates the injector fuel pulsewidths required to provide a pre-determined A/F ratio or lambda value. The desired lambda value (LAMBSE) can vary with engine operating conditions and is calibration dependent.

CLOSED LOOP MODE

During closed loop operation, the computer ramps the desired lambda value (LAMBSE) in a limit cycle manner about stoichiometry. Using the EGO (Exhaust Gas Oxygen) sensor, the computer increases or decreases LAMBSE at a calculated rate of change. The rate at which LAMBSE changes is calibration dependent.

SELF TEST OPEN LOOP MODE

During Self Test, the computer calculates lambda values (LAMBSEL, LAMBSER) that will exercise the fuel, EGO, and thermactor systems. These calculations are done in Self Test, outside of "Base Fuel Strategy" (See the SELF TEST SECTION).

ACCELERATION ENRICHMENT

(Called from SOFTWARE TAR routine which is done in the CONVRT module)

OVERVIEW

Whenever the rate of change of throttle angle exceeds a certain value additional injection pulses are delivered during throttle opening transients until the manifold filling effect is completed (MAP stops increasing). These pulses are added to the normal pulse train to provide Acceleration Enrichment (AE).

AEFUEL = The acceleration enrichment desired fuel flow rate, lb/hr

The AE pulse period (AEPP) is controlled by FN332 which adds pulses at a rate determined by the desired AE fuel flow (AEFUEL). The duration of each AE pulse is given by AEPW.

AEPW is then used to update the DOL summer register FUEL_SUM_TKS.

NOTE: The "MINPW" minimum pulsewidth clip is only applied to FUELPW, the main fuel pulsewidth. The clip does not apply to AEPW. Since AEPW is determined by the AEFUEL and AE pulse period calibration, it is expected that the developer will not request AE pulses in the non-linear injector range.

ADDITIONAL REQUIREMENTS

All injectors are energized together when delivering AE pulses.

AE pulses are asynchronous to normal fuel pulsewidths. If a fuel pulse is not in progress when an AE pulse is required, the AE pulse is sent immediately. In this case, the injector offset (FN367) is added to the AE pulsewidth calculation (AEPW).

If a normal fuel pulsewidth is in progress when an AE pulse is required, the AE pulse is added to the base pulse for that injector. In this case, the injector offset (FN367) is not added to the AE pulsewidth calculation (AEPW) (it is done once in the normal pulsewidth calculation).

For throttle angle rates of change below that of the first column of the TAE table (AETAR), TAE is set to zero.

AE pulses are enabled during run and underspeed modes only.

The AE fuel is turned off by setting TAR equal to zero in the S/W TAR logic. See SOFTWARE TAR CALCULATION in the System Equations Chapter.

DEFINITIONS

INPUTS

Registers:

- ACT = Air charge temperature, degrees F.
- A0COR = Corrected fuel flow rate of injectors, lb/sec (see Fuel chapter).
- BP = Barometric pressure, inches Hg.
- ECT = Engine coolant temperature, degrees F.
- FUEL_SUM_TKS = Register for DOL summer, ticks.
- MAP = Manifold absolute pressure, inches Hg.
- NBAR = Rolling average RPM.
- RATCH = Kicker off lowest filtered TP, counts.
- TAR = Throttle Angle Rate, deg/sec.
- TLSPAT = Torque limiting strategy injection pattern.
- TP = Throttle position, counts.
- VBAT = Battery voltage, volts.

Bit Flags:

- CRKFLG = Flag indicating status of engine mode.

Calibration Constants:

- AEM = ECT/ACT weighting factor, unitless.
- AETAR = TAR above which AE may be enabled.
- FN019A(TAR) = x-axis input to FN1303.
- FN020(FRCTAE * ACT + (1-FRCTAE) * ECT) = y-axis input to FN1303.
- FN1303(TAR,TEMP) = Desired AE fuel flow, lbm/hr.
- FN324(MAP) = Accel enrichment fuel flow multiplier, unitless.
- FN331A(TP-RATCH) = Accel enrichment fuel flow multiplier, unitless.
- FN332(AEFUEL) = Accel enrichment pulse period, seconds.
- FN378(BP) = Accel enrichment fuel flow multiplier, unitless.
- FN379(NBAR) = Accel enrichment fuel flow multiplier, unitless.
- FN367(VBAT) = Injector offset, millisec.

- FRCTAE = ACT to ECT proportioning factor, unitless.
- INJOUT = Number oof injectors fired by each output port.
- NUMOUT = Number of injector output ports, unitless.
- stcf = Seconds to clock ticks conversion factor, ticks/second.

OUTPUTS

Registers:

- AEFUEL = The acceleration enrichment desired fuel flow rate, lb/hr.
- AEPP = The AE pulse period as defined by FN332, sec.
- AEPW = Acceleration Enrichment Pulsewidth, ticks.
- FUEL_SUM_TKS = Register for DOL summer, ticks.

PROCESS

STRATEGY MODULE: FUEL_AE_COM5

AE ENABLE LOGIC

```
CRKFLG = 0 -------------|
   (RUN/UNDERSPEED)     |
                        |AND -| Enable AE
TAR > AETAR ------------|     | AEFUEL = AEM * FN1303 * FN331A * FN378
                        |     |            * FN379 * FN324
TLSPAT = 65535 ---------|     | AEPP = FN332(AEFUEL)
                              | AEPW = [ AEFUEL * FN332 / ( NUMOUT
                              |          * INJOUT * 3600 * A0COR )
                              |           + FN367 / 1000 ] * stcf
                              | FUEL_SUM_TKS = FUEL_SUM_TKS + AEPW
                              |
                              | --- ELSE ---
                              |
                              | Disable AE
                              | AEFUEL = 0
                              | AEPW = 0
                              | AEPP = 0
                              | FUEL_SUM_TKS = 0
```

WARM EGO LOGIC

OVERVIEW

The Warm EGO Logic determines if the EGO sensor is warm enough to enter Closed Loop control. Time since start-up and coolant temperature at start-up are used to determine if the sensor is warm. The output from the logic is the flag, 'WRMEGO'.

DEFINITIONS

INPUTS

Registers:

- ATMR1 = Time since start (time since exiting crank mode), sec.
- ATMR2 = Time since ECT became greater than TEMPFB, sec.
- ECT = Engine Coolant Temperature, degrees F.
- EGOSSS = Number of EGO switches since start-up.
- MFAMUL = MFA table ramp-in Multiplier, unitless.

Bit Flags:

- CRKFLG = Flag indicating engine mode; 1 -> cranking, 0 -> run or underspeed mode.
- FLG_ECTSTABLQ = ECT stabilized flag; 1 -> ECT stabilized, use FN1360.
- MPGFLG = Flag that indicates whether in Fuel Economy mode; 1 -> In Fuel Economy mode, 0 -> Not in Fuel Economy mode.
- SWTFL = EGO switch flag; 0 -> no EGO switch, 1 -> EGO switch this background loop.

Calibration Constants:

- CTHIGH = Hot start engine coolant temperature, deg F.
- CTLOW = Cold start engine coolant temperature, deg F.
- ECTSTABL = Minimum ECT to use stabilized engine open loop fuel table FN1360.
- ECTSTHYS = Hysteresis for ECTSTABL. This value should be larger than the drop in ECT when the thermostat opens on a 0 degree cold warm-up.
- MFARMP = MFAMUL ramp increment when ramping into MPG table. MFARMP is added every background loop.
- OPCLT1 = Cold start closed loop delay, seconds.

- OPCLT2 = Mid-ambient start closed loop delay, seconds.

- OPCLT3 = Hot start closed loop delay time, seconds.

- TCSTRT = ECT at start-up.

OUTPUTS

Registers:

- MFAMUL = MFA table ramp-in Multiplier, unitless.

Bit Flags:

- FFULFG = Foreground fuel flag; 1 -> Compute fuel pulsewidth in foreground using latest computed manifold absolute pressure, 0 -> otherwise use background fuel pulsewidth.

- WRMEGO = Flag that is set equal to 1 if the EGO sensor is warm and reset to zero if the sensor has cooled off.

PROCESS

STRATEGY MODULE: FUEL_WRMEGO_COM3

```
CRKFLG = 1 -----------------------------| WRMEGO = 0
                                        | FFULFG = 0
                                        | Exit FUEL_MODE
                                        |
                                        | --- ELSE ---
                                        |
                                        | Continue with FUEL_MODE
                                        |  and LAMBSE DETERMINATION
```

EGOSSS LOGIC

```
SWTFL = 1 ------------------------------| Increment EGOSSS
 (EGO switch)                           | Clip at 255
                                        |
                                        | --- ELSE ---
                                        |
                                        | Freeze EGOSSS
```

WRMEGO LOGIC

```
TCSTRT >= CTHIGH -----------|
                            |AND -|
ATMR1 >= OPCLT3 ------------|     |
                                  |
CTLOW < TCSTRT < CTHIGH ----|     |
                            |AND -|OR --| WRMEGO = 1
ATMR1 >= OPCLT2 ------------|     |     |
                                  |     | --- ELSE ---
TCSTRT <= CTLOW ------------|     |     |
                            |AND -|     | WRMEGO = 0
ATMR2 >= OPCLT1 ------------|
```

STABILIZED ECT FLIP/FLOP LOGIC
(FLG_ECTSTABLQ)

```
ECT >= ECTSTABL ------------------|S Q--| FLG_ECTSTABLQ = 1
                                  |     |  (ECT stabilized)
ECT < ECTSTABL - ECTSTHYS --------|C    |
                                        | --- ELSE ---
                                        |
                                        | FLG_ECTSTABLQ = 0
                                        |  (cold engine)
```

MFAMUL LOGIC
(MFAMUL ramps LAMBSE to the MPG mode table FN1328)

```
MPGFLG = 0 -----------------------------| MFAMUL = 0
                                        |
                                        | --- ELSE ---
                                        |
                                        | MFAMUL = MFAMUL + MFARMP
                                        | Clip MFAMUL to 1.0 as a maximum
```

LAMMUL RESET LOGIC

OVERVIEW

LAMMUL provides a means to adjust LAMBSE (rich or lean) momentarily on transmission engagements while in Open Loop. LAMMUL and FN371 have a range of 0 through 1.99. The LAMMUL RESET and RAMP BACK logic are done both in Open Loop and Closed Loop fuel. LAMMUL, however is only applies in Open Loop fuel.

DEFINITIONS

INPUTS

Registers:

- LAMMUL = Fuel multiplier for Neutral-to-Drive transitions used to prevent cold engine stalls following transmission engagement.

- MULTMR = Time since incrementing LAMMUL, sec.

Bit Flags:

- ALT_CAL_FLG = Flag to indicate use of alternate calibration.

- DNDSUP = Delayed Neutral/Drive flag; 0 -> neutral, 1 -> drive.

- NEUFLG = Neutral/Drive transition occurred.

Calibration Constants:

- FN371 = Initial LAMMUL as a function of ECT.

- FN371_ALT = Alternative FN371.

- MULTM = Minimum time interval incrementing LAMMUL.

- TRLOAD = Transmission load switch.

OUTPUTS

Registers:

- LAMMUL = Fuel multiplier for Neutral-to-Drive transitions used to prevent cold engine stalls following transmission engagement.

PROCESS

STRATEGY MODULE: FUEL_LAMMUL_COM3

```
ALT_CAL_FLG = 1 ------------------------| fn371 = FN371_ALT(ECT)
                                        |
                                        | --- ELSE ---
                                        |
                                        | fn371 = FN371(ECT)


TRLOAD <= 3 ----------------------------| No LAMMUL RESET
 (manual transmission)                  | No change to LAMMUL
                                        | No change to NEUFLG
                                        |
                                        | --- ELSE ---
DNDSUP = 1 -----------------------|     |
 (transmission in gear)           |     |
                                  |AND -| LAMMUL = fn371(ECT)
NEUFLG = 1 -----------------------|     |  (LAMMUL reset)
 (transition from neutral)              | NEUFLG = 0
                                        |  (N/D fuel enrichment)
                                        |
                                        | --- ELSE ---
                                        |
DNDSUP = 0 -----------------------------| NEUFLG = 1
 (transmission in neutral)
```

LAMMUL RAMP BACK LOGIC

```
MULTMR >= MULTM ------------------------| LAMMUL =  LAMMUL + .0039
 (free running timer)                   | Clip at .996 maximum
                                        | MULTMR = 0
```

OPEN LOOP FLAG DETERMINATION

OVERVIEW

This module determines the state of the Open Loop Flag, FLG_OPEN_LOOP.

DEFINITIONS

INPUTS

Registers:

- APT = At Part Throttle; -1 -> Closed throttle, 0 -> Part throttle, 1 -> Wide Open throttle.
- APTMR = Timer to limit time in close loop fuel control when at Wide Open Throttle.
- BIAS = A/F biasing term: FN1355(N,MAP). Units are lambdas.
- EGO_CNT_IDLE = Number of EGO switches which have occurred since entering Idle Fuel Modulation.
- IDLTMR = Time since entering Idle mode, seconds. IDLTMR is defined in the TIMER Chapter.
- LAMAVE = Average LAMBSE between EGO switches.
- MAP = Manifold Absolute Pressure, inches Hg.
- N = RPM.
- TP_REL = Relative TP (TP - RATCH).
- XAPT = Time elapse afterwhich you leave Closed Loop at Wide Open Throttle, to return to Open Loop.

Bit Flags:

- CHKAIR = Thermactor forced open loop flag; 0 -> thermactor logic forcing open loop, 1 -> closed loop permitted.
- DFSFLG = Indicates decel fuel shut off.
- MPGFLG = Flag that indicates whether in Fuel Economy mode; 1 -> In Fuel Economy mode, 0 -> Not in Fuel Economy mode.
- OFMFLG = ETV over-current monitor failure flag: 0 -> ETV O.K.; 1 -> ETV failure mode.
- TLS_NV_FLG = Engine speed/Vehicle speed limiting flag; 0 -> not limiting speed, 1 -> limiting speed.

Calibration Constants:

- EGO_IDLE = Number of EGO switches required to enter Open Loop Fuel at extended idles.

- FN311 = Minimum TP_REL required to force open loop.

- FN1360 = Stabilized open loop fuel table, an 8 x 10 table of lambda values as a function of engine speed N, and load, MAPPA. FN070B is the normalizing function for N. FN072C is the normalizing function for MAPPA.

- HLDTIM = Time delay before high load forced open loop, sec.

- LAMDLT = LAMAVE delta to allow open loop fuel.

- LAMRHYS = Hysteresis for LAMRICH.

- LAMRICH = FN1360 Lambda value below which enrichment is requested.

- LMAP = Minimum MAP for open loop decels, inches Hg.

- LOMAPH = Hysteresis for LMAP, inches Hg.

- MPG_CL_SW = MPG mode closed loop switch; 1 -> operate in closed loop, 0 -> operate in open loop.

- OLITD1 = Time delay to go open loop at idle, seconds.

- T70LSW = 7.0L thermactor application switch.

- TP_HYS_OL = TP_REL hysteresis for operation in open loop.

OUTPUTS

Bit Flags:

- FLG_OPEN_LOOP = Open Loop Fuel flag; 1 -> Open Loop fuel, 0 -> Closed Loop fuel may be permitted.

- MPGTFG = MPG transition mode flag; 1 -> MPG mode exit into Closed Loop fuel.

PROCESS

STRATEGY MODULE: FUEL_OLFLG_COM7

```
FN1360 <= LAMRICH -------------|S Q -|
 (enrichment required)         |     |
                               |     |
FN1360 > LAMRICH + LAMRHYS ----|C    |AND -|
 (return to closed loop)             |     |
                                     |     |
HLTMR >= HLDTIM ---------------------|     |
 (high load delay time)              |     |
                                     |     |
OPEN_TMR <= TIME_OL -----------------|     |
 (O.K. to remain in open loop; not time    |
  to adaptive or force purge control)      |
                                           |
MAP < LMAP - LOMAPH -----------------|S Q -|
 (decel load)                        |     |
                                     |     |
MAP >= LMAP -------------------------|C    |
                                           |
APT = 1 -----------------------------|     |
                                     |AND -|OR --| FLG_OPEN_LOOP = 1
APTMR >= XAPT -----------------------|     |     |  (open loop conditions
                                           |     |   met)
MPG_CL_SW = 0 -----------------------|     |     | MPGTFG = 0
                                     |AND -|     | Stop MPG mode fuel ramp
MPGFLG = 1 --------------------------|     |     |
 (MPG mode)                                |     | --- ELSE ---
                                           |     |
DFSFLG = 1 --------------------------------|     | Check to see if EGO sensor
 (decel fuel shutoff)                      |     |  ready or fuel ramp in
                                           |     |  progress
IDLTMR > OLITD1 ---------------------|     |     | FLG_OPEN_LOOP = 0
 (idle)                              |     |     |  (closed loop operation
                                     |     |     |   possible)
EGO_CNT_IDLE >= EGO_IDLE ------------|AND -|
                                     |     |
|1.0 + BIAS - LAMAVE| <= LAMDLT -----|     |
                                           |
CHKAIR = 0 --------------------------|     |
 (thermactor forced open loop)       |     |
                                     |AND -|
T70LSW = 0 --------------------------|     |
 (1 = 7.0L application; no thermactor      |
  forced open loop)                        |
                                           |
OFMFLG = 1 --------------------------------|
 (ETV sol. shorted to ground)              |
                                           |
TLS_NV_FLG = 1 ----------------------------|
 (vehicle/engine speed limiting)           |
                                           |
TP_REL > FN311(N) -------------------|S Q -|
                                     |
TP_REL <= FN311(N) - TP_HYS_OL ------|C
```

Where: BIAS = FN1355(N,MAP).

CALCULATION OF LAMAVE

OVERVIEW

This module calculates the LAMBSE average to be used in Self Test to determine if the adaptive tables have reached their clip.

The LAMBSE average, LAMAVE, is only calculated in Closed Loop. Also, it requires at least two EGO switches after entering Closed Loop to calculate an average.

DEFINITIONS

Registers:

- LAMBSE = Desired Air/Fuel ratio. LAMBSE(N) = New LAMBSE. LAMBSE(O) = previously calculated LAMBSE.

- LAM_OLD = Value of LAMBSE at previous EGO switch.

Bit Flags:

- OLFLG = Open Loop Fuel flag; 1 -> open loop, 0 -> closed loop possible.

OUTPUTS

Registers:

- LAMAVE = Average LAMBSE between EGO switches.

- LAM_OLD = See above.

PROCESS

STRATEGY MODULE: FUEL_LAMAVE_COM2

```
OLFLG = 1 -----------------------------| LAM_OLD = 0
                                       | LAMAVE = 1.0
                                       |
                                       | Exit FUEL_LAMAVE_COM1 module


EGO switch ----------------------|
                                 |
OLFLG = 0 -----------------------|AND -| LAM_OLD = LAMBSE
                                 |     |  (not enough EGO switches
LAM_OLD = 0 ---------------------|     |   since going Closed Loop
                                       |   to calculate LAMAVE)
                                       |
                                       | --- ELSE ---
                                       |
EGO switch ----------------------|     |
                                 |AND -| LAMAVE = (LAM_OLD + LAMBSE) / 2
OLFLG = 0 -----------------------|     | LAM_OLD = LAMBSE
                                       |  (update LAM_OLD for next time)
```

NOTE: LAMAVE and LAM_OLD are initialized to 1.0.

OPEN LOOP LAMBSE CALCULATION

OVERVIEW

Due to the current design of the EGO sensor, Closed Loop operation about stoichiometry only can be utilized. In the future, Universal EGOs will be able to provide feedback at points either rich or lean of stoich. Closed loop operation is also required for adaptive fuel to "learn" the variabilities associated with production tolerances of airflow measuring and fuel metering devices. Given, however, that closed loop fuel is not always the appropriate mode of operation, the FUEL_MODE logic performs two functions:

- determine whether open loop fuel is appropriate, and

- schedule the optimum equivalence ratio (LAMBSE).

Optimum Air/Fuel (A/F) ratio at any given engine speed-load is that which will develop the required torque with the lowest fuel consumption consistent with smooth reliable operation. This optimum A/F ratio is not constant but depends on many factors. The proper A/F ratio for each particular set of operating conditions is most conveniently viewed under the two headings, Stabilized and Cold Engine/Startup. Stabilized is taken to mean continuous operation with normal engine temperatures. Cold engine/startup includes starting and warming up or when the EGO sensor is not ready to switch.

COLD/START-UP OPEN LOOP FUEL

ENTRY CONDITIONS

- WRMEGO = 0, EGO sensor may not be ready to switch or,

- ECT < ECTSTABL, engine coolant temperature too cold and FLG_OPEN_LOOP = 1, standard open loop conditions met. (see Stabilized Open Loop entry condition below.)

FUEL SCHEDULING

- Fuel may be scheduled as a function of load, temperature and time since start.

STABILIZED ENGINE OPEN LOOP FUEL

Mixture requirements for lowest fuel consumption can generally be adequately described as a function of engine load.

Idle, extremely light loads or deceleration:

- No useful work is being done at idle other than driving accessory loads, i.e. brake torque is zero. The lowest fuel consumption is that A/F which provides steady, reliable cylinder firing. Extremely light loads or decels at high RPMs may require A/F scheduling. This is because the mass of residual gas tends to be constant, therefore, at closed throttle/slight part throttle decels, the percentage of residual diluting the fresh charge increases, hence the need to enrichen the incoming charge.

- During Closed Throttle, High RPM Decels, the fuel can be completely turned off to improve fuel economy and/or limit catalyst temperatures. See Decel Fuel Shut-off logic in this chapter.

Medium loads, typical part throttle operation at road load:

- Best economy is around 18:1 A/F ratio, however, this may lead to excessive highway NOx levels. Also, current catalyst technology makes use of the closed loop fuel limit cycle to maximize conversion efficiency. This, closed loop fuel operation is desirable from both an adaptive fuel and catalyst efficiency standpoint. Provisions are in place to schedule lean open loop operation during steady state cruise modes (MPG mode).

Full load, typical W.O.T operation:

- A rich A/F ratio is required to maximize the torque output of the engine during periods of high driver demand. Maximum torque output is about 13:1 A/F ratio (LBT). Best fuel economy is sacrificed in return for higher torque output.

- Best economy at a given load is independent of RPM at least down to idle type loads, under conditions at which fuel distribution is good. Departures from best fuel economy may be required due to the following:

  - Poor distribution (usually a function of TP and RPM).

  - To reduce temperature of hot spots such as exhaust valves, spark plug points or piston crowns - that is, to assist cooling.

STABILIZED OPEN LOOP FUEL

ENTRY CONDITIONS

- WRMEGO = 1 - engine coolant hot and engine is stabilized, EGO sensor is ready to switch.

- High Load enrichment required - scheduled as a function of speed/load. An open loop delay time is provided to prevent unnecessary enrichment during high load spikes such as during shifts. A second enrichment timer provides the ability to schedule fuel as a function of time in the enrichment mode to enhance durability in heavy truck applications.

- Decel loads - either enrichment or enleanment can be scheduled as a function of speed/load.

- Extended idle - the ability to go open loop for idle stability is provided as a function of time at idle.

- Lean cruise mode uses a unique speed/load table to schedule open loop lambdas.

- W.O.T. mode - open loop is automatically scheduled as a function of wide open throttle mode.

- CHKAIR = 0 - thermactor air control can put upstream air into the exhaust manifold ahead of the EGO sensor causing the EGO to always read lean. Open loop is therefore required.

- Lack of EGO switching requires the scheduling of open loop operation.

- Decel Fuel Shutoff is an open loop state.

FUEL SCHEDULING

- Fuel may be scheduled as a function of load, RPM, temperature and time at high load.

SPECIAL EXIT CONDITIONS

- If MPG mode or Decel Fuel Shut-Off logic had previously dictated Open Loop control, and that mode is no longer desired, special logic maintains Open Loop control and decrements LAMBSE to 1.0 before allowing closed loop control.

DEFINITIONS

INPUTS

Registers:

- APT = At Part Throttle; -1 -> Closed throttle, 0 -> Part throttle, 1 -> Wide Open throttle.
- APTMR = Timer to limit time in close loop fuel control when at Wide Open Throttle.
- ATMR1 = Time since start (time since exiting crank mode), sec.
- BIAS = A/F biasing term: FN1355(N,MAP). Units are LAMBDAS.
- ECT = Engine coolant temperature.
- EGOSSS = Number of EGO switches since start-up.
- EGOSW_OL_CTR = Number of EGO switches in closed loop; used to determine when o.k. to return to open loop.
- LAMAVE = Average of last two LAMBSE values.
- LAMBSE = Desired ratiometric air/fuel ratio.
- LAMMUL = Fuel multiplier for Neutral-to-Drive transitions used to prevent cold engine stalls following transmission engagement.
- MAPPA = MAP/BP, inches of Hg.
- MFAMUL = MFA table ramp-in Multiplier, unitless.
- N = Engine speed, rpm.
- PPCTR = PIP counter; updated at PIP rising edge before injector pulsewidth is calculated and output.

Bit Flags:

- ALT_CAL_FLG = Flag to indicate use of alternate calibration.
- DNDSUP = Delayed Neutral/Drive flag; 0 -> neutral, 1 -> drive.
- CLFLG = Closed Loop Flag; 0 -> open loop, 1 -> closed loop.
- FLG_ECTSTABLQ = ECT stabilized flag; 1 -> ECT stabilized, use FN1360.
- FLG_NOT_ADP = Adaptive enabled within an act and ect range.
- FLG_OPEN_LOOP = Open Loop Fuel flag; 1 -> Open Loop fuel, 0 -> Closed Loop fuel may be permitted.
- ISCFLG = ISC mode flag; 1 -> RPM control mode.
- LESFLG = Lack of EGO switching flag; 0 -> EGO switching, 1 -> EGO did not switch for LESTIM seconds.

- MPGFLG = Flag that indicates whether in Fuel Economy mode; 1 -> In Fuel Economy mode, 0 -> Not in Fuel Economy mode.

- MPGTFG = MPG transition mode flag; 1 -> MPG mode exit into Closed Loop fuel.

- OLFLG = Open Loop fuel flag; 1 -> open loop fuel, 0 -> closed loop fuel.

- SWTFL = EGO switched flag.

- WOTTMR = Time of WOT.

- WRMEGO = Flag that is set equal to 1 if the EGO sensor is warm and reset to zero if the sensor has cooled off.

Calibration Constants:

- EGOCL1 = Number of EGO switches required to enter Closed Loop.

- DFSLAM = Relative LAMBSE value to exit decel fuel shut-off.

- EGO_CNT_OL = Number of EGO switches required in closed loop to return to open loop control.

- FN301(N) = Multiplier for closed throttle as a function of engine speed N.

- FN303(N) = Multiplier for WOT as a function of engine speed N.

- FN310(WOTTMR) = WOT fuel multiplier as a function of time in WOT mode. WOTTMR is a count up/count down timer to prevent resets during normal shifts. FN310 is used only within a WOTRPL and WOTRPH RPM range.

- FN325(ECT) = Multiplier as a function of ECT for FN1360.

- FN1325L = LTMTBL learning/use control table.

- FN1328 = MPG fuel table, lambdas. It is a 10 x 8 table of fuel economy open loop lambda values as a function of N and MAPPA.

- FN1360 = Stabilized open loop fuel table, an 8 x 10 table of lambda values as a function of engine speed N, and load, MAPPA. FN070B is the normalizing function for N. FN072C is the normalizing function for MAPPA.

- FN1361 = Start-up fuel table = an 8 x 10 table of lambda values as a function of [FRCSFT * ACT + (1 - FRCSFT) * ECT] and MAPPA. FN022A is the temperature normalizing function. FN018A is the time normalizing function.

- FN1362 = Base fuel table = an 8 x 10 table of lambda values as a function of [FRCBFT 8 ACT + (1 - FRCBFT) * ECT] and MAPPA. FN022A is the temperature normalizing function.

- FN1362 = Alternatice FN1362.

- LAMDLT_OL = LAMAVE DELTA to allow return to open loop fuel control.

- LAMRICH = FN1360 Lambda value below which enrichment is requested.

- MPG_CL_SW = MPG mode closed loop switch; 1 -> operate in closed loop, 0 -> operate in open loop.

- MPGDEC = Lambse decrement when exiting MPG mode.

- OLIDRV = Drive Open Loop Idle Fuel Multiplier.

- OLINEUT = Neutral Open Loop Idle Fuel Multiplier.

- OLMCL = Open loop fuel multiplier for development only.

- PIPNUM = Number of PIPs for DFSO exit fuel ramp.

- TRLOAD = Transmission load switch.
- WOTRPH = Maximum RPM to use FN310 for WOT enleanment.
- WOTRPL = Minimum RPM to use FN310 for WOT enleanment. Set to WOTRPL to 10,000 RPM to disable use of FN310.
- XAPT = Time elapse afterwhich you leave Closed Loop at Wide Open Throttle, to return to Open Loop.
- XMAPPA = Calibratable MAPPA value above which APTMR is enabled.

OUTPUTS

Registers:

- APTMR = See above.
- EGOSW_OL_CTR = See above.
- LAMBSE = Desired air/fuel ratio.
- OPEN_TMR = TIME IN OPEN LOOP, SECONDS.

Bit Flags:

- ACCUM = Accumulator for closed loop LAMBSE ramp increments. Used for jumpback when TSLEGO <= transport lag.
- CLFLG = See above.
- OLFLG = See above.
- MPGTFG = MPG transition mode flag; 1 -> MPG mode exit into Closed Loop fuel.

PROCESS

STRATEGY MODULE: FUEL_OL_LAMBSE_COM3

FINAL FUEL MODE DETERMINATION (OLFLG/CLFLG)
AND OPEN LOOP LAMBSE CALCULATION

```
ALT_CAL_FLG = 1 ---------------------| fn1362 = FN1362_ALT(TEMP,MAPPA)
                                     |
                                     | --- ELSE ---
                                     |
                                     | fn1362 = FN1362(TEMP,MAPPA)


MPGFLG = 1 --------------------|
                               |AND -| LAMBSE = 1 + [FN1328(N,MAPPA) *
MPG_CL_SW = 0 -----------------|     |          (MFAMUL)]
                                     | OLFLG = 1
                                     | CLFLG = 0
                                     | ACCUM = 0
                                     |  (lean cruise mode)
                                     |
                                     | --- ELSE ---
MPGTFG = 1 --------------------|     |
 (exiting MPG mode)            |     |
                               |AND -| LAMBSE = LAMBSE - MPGDEC
MPG_CL_SW = 0 -----------------|     | Do: MPGTFG RESET LOGIC
                                     | OLFLG = 1
                                     | CLFLG = 0
                                     | ACCUM = 0
                                     |  (lean cruise mode exit into
                                     |   closed loop fuel (ramp fuel))
                                     |
                                     | --- ELSE ---
PPCTR < PIPNUM ----------------|     |
 (decel fuel shut-off)         |     |
                               |AND -| LAMBSE = 1 + DFSLAM -
FLG_OPEN_LOOP = 0 -------------|     |          [(PPCTR / PIPNUM)
 (closed loop conditions met)        |          * DFSLAM]
                                     | OLFLG = 1
                                     | CLFLG = 0
                                     | ACCUM = 0
                                     |  (decel fuel shut-off exit into
                                     |   closed loop fuel (ramp fuel))
                                     |
                                     | --- ELSE ---
```

(continued on next page)

(continued from previous page)

```
WRMEGO = 0 --------------------|      |
 (cold EGO)                    |      |
                               |      |
FLG_OPEN_LOOP = 1 -|           |      |
 (open loop req'd) |           |OR --| Do: 'C' MULTIPLIER LOGIC
                   |           |      | LAMBSE = [fn1362(TEMP,MAPPA) -
EGOSSS < EGOCL1 ---|OR --|     |      |          FN1361(TEMP,ATMR1)] * C *
 (not enough EGO   |     |     |      |          LAMMUL * OLMCL
  switches)        |     |     |      | OLFLG = 1
                   |     |AND -|      | CLFLG = 0
LESFLG = 1 --------|     |            | ACCUM = 0
 (lack of EGO            |            |  (cold engine/start-up fuel tables)
  switching)             |            |
                         |            |
FLG_ECTSTABLO = 0 -------|            |
 (ECT cold)                           |
                                      | --- ELSE ---
WRMEGO = 1 --------------------|      |
 (warm EGO)                    |      |
                               |AND -| Do: 'C' MULTIPLIER LOGIC
FLG_OPEN_LOOP = 1 -------|     |      | LAMBSE = FN1360(N,MAPPA) *
 (open Loop required)    |     |      |          FN325(ECT) * C *
                         |     |      |          LAMMUL * OLMCL
EGOSSS < EGOCL1 ---------|OR --|      | OLFLG = 1
 (not enough EGO         |            | CLFLG = 0
  switches)              |            | ACCUM = 0
                         |            |  (stabilized engine fuel table)
LESFLG = 1 --------------|            |
 (lack of EGO switching)              |
                                      | --- ELSE ---
                                      |
                                      | Do: CLOSED LOOP LAMBSE CALCULATION
                                      | OLFLG = 0
                                      | CLFLG = 1
                                      |  (closed loop fuel)
```

Compute OPEN_TMR to determine if conditions exist to return to open loop:

```
FN1360 <= LAMRICH -------------------|
                                     |
CLFLG = 1 ---------------------------|
                                     |
EGOSW_OL_CTR >= EGO_CNT_OL ----------|AND -|
                                     |     |
|1.0 + BIAS - LAMAVE| <= LAMDLT_OL --|     |
                                           |
FN1325L < 0 -------------------------------|
 (not in an adaptable area; forced         |OR --| OPEN_TMR = 0
  adaptive not required)                   |     |
                                           |     |
ISCFLG = 1 --------------------------------|     |
 (at idle; forced adaptive not required)   |     |
                                           |     |
APTMR >= XAPT -----------------------------|     |
                                           |     |
FLG_NOT_ADP = 0 ---------------------------|     |
                                                 | --- ELSE ---
                                                 |
CLFLG = 0 ---------------------------------------| Increment OPEN_TMR.
                                                 |
                                                 | --- ELSE ---
                                                 |
                                                 | Freeze OPEN_TMR.


ACT > AFACT1 ------------------------------|
                                           |
ACT < AFACT2 ------------------------------|
                                           |AND -| FLG_NOT_ADP = 1
ECT <= AFECT2 -----------------------------|     |
                                           |     |
ECT >= AFECT1 -----------------------------|     |
                                                 | --- ELSE ---
                                                 |
                                                 | FLG_NOT_ADP = 0
```

Increment EGOSW_OL_CTR when EGO switches during closed loop:

```
CLFLG = 1 ---------------------------------|
                                           |AND -| Increment EGOSW_OL_CTR.
SWTFL = 1 ---------------------------------|     |
 (EGO switched)                                  |
                                                 | --- ELSE ---
                                                 |
CLFLG = 0 ---------------------------------------| EGOSW_OL_CTR = 0
```

Increment APTMR to limit time in close loop when at Wide Open Throttle.

```
MAPPA >= XMAPPA ---------------------------|
                                           |AND -| increment APTMR
OLFLG = 0 ---------------------------------|     |
  (closed loop)                                  | --- ELSE ---
                                                 |
                                                 | APTMR = 0
```

GLOBAL LAMBSE CLIP

```
always ------------------------------------------| Clip LAMBSE to 1.999969 as
                                                 |  a maximum and 0.0000305
                                                 |  as a minimum
```

"C" MULTIPLIER LOGIC

```
DNDSUP = 1 --------------|
 (in drive)              |
                         |OR --|
TRLOAD <= 3 -------------|     |
 (manual trans)                |
                               |
APT = -1 ----------------------|AND -| "C" = FN301(N)
                               |     |  (closed throttle decel/idle, FN1360
WRMEGO = 0 --------------|     |     |   not available)
 (cold EGO)              |     |     |
                         |OR --|     |
FLG_ECTSTABLQ = 0 -------|           |
 (ECT cold)                          |
                                     | --- ELSE ---
DNDSUP = 0 --------------------|     |
 (in neutral)                  |     |
                               |AND -| "C" = OLINEUT
APT = -1 ----------------------|     |  (neutral idle)
 (closed throttle)                   |
                                     | --- ELSE ---
DNDSUP = 1 --------------------|     |
 (in drive)                    |     |
                               |AND -| "C" = OLIDRV
APT = -1 ----------------------|     |  (drive idle)
                                     |
                                     | --- ELSE ---
APT = 1 -----------------------|     |
                               |AND -| "C" = FN303(N) * FN310(WOTTMR)
WOTRPL <= N <= WOTRPH ---------|     |  (wide open throttle)
                                     |
                                     | --- ELSE ---
                                     |
APT = 1 -----------------------------| "C" = FN303(N)
                                     |  (wide open throttle)
                                     |
                                     | --- ELSE ---
                                     |
                                     | "C" = 1
```

MPGTFG RESET LOGIC

```
MPGTFG = 1 --------------------|
                               |AND -| MPGTFG = 0
LAMBSE <= 1.0 -----------------|     |  (fuel ramp to stoic. is done)
```

CLOSED LOOP LAMBSE CALCULATION

(Background software module CLFUEL, called from FUEL_MODE)

OVERVIEW

The goals of the closed loop strategy are:

- add the capability of introducing A/F ratio biasing,
- maximize the feedback limit cycle frequency for all bias values, and
- maintain a simple calibration procedure to describe the closed loop limit cycle.

The fuel flow is driven in a limit cycle manner about stoichiometry. Using the EGO (Exhaust Gas Oxygen) sensor, the computer increases or decreases the injector pulsewidths in a controlled manner. If the EGO reads rich, the pulsewidths will be decreased (made leaner) at a calculated rate. If the EGO reads lean, the pulsewidths will be increased (made richer) at a calculated rate.

When an EGO switch occurs, an instantaneous change (or "jumpback") is made in the A/F ratio back towards stoichiometry. The jump is made relative to the A/F ratio (LAMBSE) value at the EGO switch.

The limit cycle can be biased to operate on the average richer or leaner of stoichiometry.

An example of the closed loop limit cycle is shown on the next page.

LIMIT CYCLE DESCRIPTION

```
   ---|-------------------*                   *-|-
      |                  **                  ** |
      |                 * *                 * * |
      |                *  *                *  * |
      |               *   *               *   * |
      |              *    *              *    * |
 FN1354(N,MAP)      *     *          RF *     * (RF*T) OR (ACCUM)
   (PTPAMP)        *      *            *      * |
- - - | - | - - - * - - - * - - - - - - * - - - - * | - - - - - - - mean A/F
      |  BIAS    *        *          *        * |
------|---|-----*---------*---------*---------*-|-------*-|-- stoc
      |         |         | *       *           *       * |
      |         |         |   *     *             *     * |
      |         |         |     *   *           RS  *   * (RS*T) OR (ACCUM)
      |         |         |       * *                 * * |
   ---|---------|---------|---------*                   *-|-
                |<TSLEGO->|<-TSLEGO>|
```

***** NOTE *****

The direction of the bias is controlled by the sign of the bias value. If the bias term is negative, a rich bias is indicated. If the sign of the bias term is positive, a lean bias is indicated.

****

****
***** WARNING *****
****

It is imperative that an accurate value for the system transport delay be entered. An incorrect value will result in greatly reduced catalyst efficiencies due to excessively fast or slow ramp rates, incorrect jumpback amounts, etc.

****
***** WARNING *****
****

Prior to calculating a new Closed Loop LAMBSE value, LAMBSE may be clipped to 1.0 maximum and/or multiplied by JMPMUL under the conditions listed below. LAMBSE is not reset in Open Loop fuel control because the value of LAMBSE is calculated using the Open Loop fuel logic. LAMBSE is reset in Closed Loop as follows:

- when entering or exiting the filtered idle air mass region (FAM),
- when changing load states within the filtered idle air mass region (FAM), and
- any time a transition is made from Open Loop to Closed Loop fuel control.

LAMBSE is always clipped to 1.0 as a maximum. The intent is to allow rich errors and to prevent lean errors, given that running rich does not cause any driveability concerns.

CALIBRATION PHILOSOPHY

Although appearing somewhat complicated, this closed loop algorithm has been designed to be easy to use.

There are 3 tables which must be calibrated. They are:

1) limit cycle peak to peak amplitude; FN1354(N,MAP); a typical value is 0.034 lambdas (0.034 * 14.64 = 0.49776 A/F ratio) = PTPAMP

2) fuel system transport delay; FN1343(N,MAP); typical values have been 5 - 10 engine revolutions, but note earlier WARNING.

3) BIAS; FN1355(N,MAP) for closed throttle mode:

    a) a positive bias value is lean; a negative bias value is rich

    b) any calculated absolute value of BIAS/PTPAMP exceeding .2 is clipped to .2. This is done to avoid extremely long limit cycle periods.

DEFINITIONS

INPUTS

Registers:

- ACCUM = Accumulator for closed loop LAMBSE ramp increments. Used for jumpback when TSLEGO <= transport lag.

- APT = Throttle mode flag; -1 -> Closed throttle, 0 -> Part throttle, 1 -> Wide-Open throttle.

- BIAS = A/F biasing term: FN1355(N,MAP). Units are lambdas.

- ISFLAG = Indication of engine load state at idle (See IDLE SPEED CONTROL Chapter); 0 -> Drive, 1 -> Drive and A/C clutch engaged, 2 -> Neutral, 3 -> Neutral and A/C clutch engaged.

- ISLAST = Register which tracks the state of engine load from the previous background pass. Used in determining when it is necessary to increment the filtered air mass (FAM) and clip the C/L idle speed integrator to a minimum value.

- JUMP = Amount of jumpback after an EGO switch, either ACCUM or based on table FN1343.

- LAMBSE = Desired Air/Fuel ratio. LAMBSE(N) = New LAMBSE. LAMBSE(O) = previously calculated LAMBSE.

- PTPAMP = Limit cycle peak to peak amplitude.

- R = Ramp rate; can be set to either RF or RS.

- RF = Fast ramp rate used on the side of stoichiometry that bias is desired: RF = FN1354(N,MAP) * FN372(|BIAS/PTPAMP|) * N/FN1343(N,MAP). Units are lambdas/second.

- RS = Slow ramp rate used on the side of stoichiometry that bias is not desired: RS = FN1354(N,MAP) * (0.01660 - FN372(|BIAS/PTPAMP|)) * N/FN1343(N,MAP). Units are lambdas/second.

- TDSEC = 60 * FN1343/N (temporary register).

- TSLAMU = Time since the last LAMBSE update (1 background loop).

- TSLEGO = Time since last EGO switch occurred, seconds.

Bit Flags:

- JMPFLG = LAMBSE reset flag.

- OLFLG = Open Loop fuel flag; 1 -> Open Loop fuel, 0 -> Closed Loop fuel.

- REFFLG = Indication of Idle Air Flow; 1 -> Idle Air Flow.

Calibration Constants:

- FN025 = MAP normalizing function for Bias, Peak-to-Peak and Transport Delay Tables. Input = MAP, Output = Table Entry Point.

- FN372(|BIAS/PTPAMP|) = Ramp rate multiplier (not a calibration item) - provides correct multiplier to produce desired waveform; function of |BIAS/PTPAMP|. Units are minutes/sec.

- FN1343(N,MAP) = System transport lag time; time delay from when a fuel change is made until the EGO sensor indicates this change. Units are REVS. X-input = FN039 normalized engine speed, RPM; Y-input = FN025 normalized manifold absolute pressure, MAP; output = transport delay, REVS.

- FN1354(N,MAP) = Closed loop Peak-to-Peak amplitude: units are lambdas. X-input = FN039 normalized engine speed, RPM; Y-input = FN025 normalized manifold absolute pressure, MAP; output = Peak-to-Peak amplitude, PTPAMP.

- FN1355(N,MAP) = Amount of BIAS from stoichiometry: X-input = FN039 normalized engine speed, RPM; Y-input = FN025 normalized manifold absolute pressure, MAP; output = BIAS from stoichiometry.

- JMPMUL = FAM exit LAMBSE reset multipler.

- LAMMAX = Maximum closed loop LAMBSE clip.

- LAMMIN = Minimum closed loop LAMBSE clip.

OUTPUTS

Registers:

- ACCUM = See above.

- BIAS = See above.

- JUMP = See above.

- LAMBSE = See above.

- R = See above.

- TDSEC = See above.

- TSLEGO = See above.

Bit Flags:

- JMPFLG = See above.

- V_LAMJMP = Flag when 1 indcates base strategy caused a LAMBSE jump since last EGO switch.

PROCESS

STRATEGY MODULE: FUEL_CL_LAMBSE_COM3

LAMBSE RESET - LOAD STATE CHANGE

```
APT = -1 --------------------------|
 (closed throttle)                 |
                                   |
ISFLAG <> ISLAST ------------------|AND -| Clip LAMBSE to 1.0 as a
 (load state change)               |     |  maximum
                                   |     | TSLEGO = 0
LAMBSE > 1.0 ----------------------|     | ACCUM = 0
                                         | V_LAMJMP = 1
```

OPEN LOOP/CLOSED LOOP LAMBSE RESET

```
Previous OLFLG = 1 ----------------|
 (last pass was open loop)         |
                                   |
Current OLFLG = 0 -----------------|AND -| Clip LAMBSE to 1.0 as a
 (current pass is closed loop)     |     |  maximum
                                   |     | TSLEGO = 0
LAMBSE > 1.0 ----------------------|     | ACCUM = 0
                                         | V_LAMJMP = 1
```

FAM ENTRY/EXIT LAMBSE RESET

```
REFFLG = 1 ------------------------|
                                   |
JMPFLG = 0 ------------------------|AND -| JMPFLG = 1
                                   |     | Clip LAMBSE to 1.0 as a
LAMBSE > 1.0 ----------------------|     |  maximum
                                         | TSLEGO = 0
                                         | ACCUM = 0
                                         |  (entering idle region)
                                         | V_LAMJMP = 1
                                         |
                                         | --- ELSE ---
REFFLG = 1 ------------------------|     |
                                   |AND -| JMPFLG = 1
JMPFLG = 0 ------------------------|     |
                                         | --- ELSE ---
REFFLG = 0 ------------------------|     |
                                   |AND -| JMPFLG = 0
JMPFLG = 1 ------------------------|     | LAMBSE = LAMBSE * JMPMUL
                                         |  (exiting idle region)
                                         | V_LAMJMP = 1
                                         |
                                         | --- ELSE ---
                                         |
                                         | No change
```

CALCULATE PTPAMP, RF AND RS

```
always ------------------------------| PTPAMP = FN1354(N,MAP)
                                     | RF = PTPAMP * FN372(|BIAS/PTPAMP|) *
                                     |       N / FN1343(N,MAP)
                                     | RS = PTPAMP * (0.01660 -
                                     |        FN372(|BIAS/PTPAMP|)) *
                                     |        N / FN1343(N,MAP)
```

RAMP RATE CALCULATIONS BASED ON BIAS AND EGO STATE

```
BIAS = 0 ----------------------|
 (no bias)                     |AND -| LAMBSE(N) = LAMBSE(O) -
                               |     |             RF * TSLAMU
EGO IS LEAN -------------------|     | ACCUM = ACCUM + RF * TSLAMU
                                     |  (ramp in rich direction)
                                     |
                                     | --- ELSE ---
BIAS = 0 ----------------------|     |
 (no bias)                     |AND -| LAMBSE(N) = LAMBSE(O) +
                               |     |             RF * TSLAMU
EGO IS RICH -------------------|     | ACCUM = ACCUM + RF * TSLAMU
                                     |  (ramp in lean direction)
                                     |
                                     | --- ELSE ---
BIAS > 0 ----------------------|     |
 (lean bias)                   |AND -| LAMBSE(N) = LAMBSE(O) -
                               |     |             RS * TSLAMU
EGO IS LEAN -------------------|     | ACCUM = ACCUM + RS * TSLAMU
                                     |  (ramp in rich direction)
                                     |
                                     | --- ELSE ---
BIAS > 0 ----------------------|     |
 (lean bias)                   |AND -| LAMBSE(N) = LAMBSE(O) +
                               |     |             RF * TSLAMU
EGO IS RICH -------------------|     | ACCUM = ACCUM + RF * TSLAMU
                                     |  (ramp in lean direction)
                                     |
                                     | --- ELSE ---
BIAS < 0 ----------------------|     |
 (rich bias)                   |AND -| LAMBSE(N) = LAMBSE(O) -
                               |     |             RF * TSLAMU
EGO IS LEAN -------------------|     | ACCUM = ACCUM + RF * TSLAMU
                                     |  (ramp in rich direction)
                                     |
                                     | --- ELSE ---
BIAS < 0 ----------------------|     |
 (rich bias)                   |AND -| LAMBSE(N) = LAMBSE(O) +
                               |     |             RS * TSLAMU
EGO IS RICH -------------------|     | ACCUM = ACCUM + RS * TSLAMU
                                     |  (ramp in lean direction)
```

NOTE: TSLAMU is time since the last LAMBSE update (1 background loop).

JUMPBACK CALCULATIONS BASED ON BIAS AND EGO STATE

```
BIAS = 0 ------------------------|
 (no bias)                       |AND -| LAMBSE(N) = LAMBSE(O) + JUMP
                                 |     | R = RF
EGO SWITCHED FROM LEAN TO RICH --|     | ACCUM = 0
                                       |  (jumpback in lean direction)
                                       | TSLEGO = 0
                                       |
                                       | --- ELSE ---
BIAS = 0 ------------------------|     |
 (no bias)                       |AND -| LAMBSE(N) = LAMBSE(O) - JUMP
                                 |     | R = RF
EGO SWITCHED FROM RICH TO LEAN --|     | ACCUM = 0
                                       |  (jumpback in rich direction)
                                       | TSLEGO = 0
                                       |
                                       | --- ELSE ---
BIAS > 0 ------------------------|     |
 (lean bias)                     |AND -| LAMBSE(N) = LAMBSE(O) + JUMP
                                 |     | R = RS
EGO SWITCHED FROM LEAN TO RICH --|     | ACCUM = 0
                                       |  (jumpback in lean direction)
                                       | TSLEGO = 0
                                       |
                                       | --- ELSE ---
BIAS > 0 ------------------------|     |
 (lean bias)                     |AND -| LAMBSE(N) = LAMBSE(O) - JUMP
                                 |     | R = RF
EGO SWITCHED FROM RICH TO LEAN --|     | ACCUM = 0
                                       |  (jumpback in rich direction)
                                       | TSLEGO = 0
                                       |
                                       | --- ELSE ---
BIAS < 0 ------------------------|     |
 (rich bias)                     |AND -| LAMBSE(N) = LAMBSE(O) + JUMP
                                 |     | R = RF
EGO SWITCHED FROM LEAN TO RICH --|     | ACCUM = 0
                                       |  (jumpback in lean direction)
                                       | TSLEGO = 0
                                       |
                                       | --- ELSE ---
BIAS < 0 ------------------------|     |
 (rich bias)                     |AND -| LAMBSE(N) = LAMBSE(O) - JUMP
                                 |     | R = RS
EGO SWITCHED FROM RICH TO LEAN --|     | ACCUM = 0
                                       |  (jumpback in rich direction)
                                       | TSLEGO = 0


TSLEGO <= TDSEC -----------------------| JUMP = ACCUM
 (time since last EGO <=               |
  transport lag)                       | --- ELSE ---
                                       |
                                       | JUMP = R * TDSEC
```

```
APT = -1 ------------------------|
                                 |AND -| Clip LAMBSE to 1.0 as a maximum
ISFLAG <> ISLAST ----------------|     |
                                       | --- ELSE ---
                                       |
                                       | Clip LAMBSE to LAMMAX as a maximum


always --------------------------------| Clip LAMBSE to LAMMIN as a
                                       |  minimum
```

DAC REGISTER CALCULATION
(software module DACEQN, called from FUEL_MODE)

OVERVIEW

A special register, DSLMBS, has been added to assist in calibration development by increasing the resolution of LAMBSE for display purposes. DSLMBS is updated every time LAMBSE is updated, both Open & Closed Loop.

DEFINITIONS

INPUTS

Registers:

- LAMBSE = Desired ratiometric air/fuel ratio.

OUTPUTS

Registers:

- DSLMBS = Special display version of LAMBSE.

PROCESS

STRATEGY MODULE: FUEL_DAC_REG_COM2

The value of DSLMBS is calculated as shown:

DSLMBS = LAMBSE - 1.0

Because DSLMBS is a signed word quantity, a value of zero will be output as 5 volts.

ADAPTIVE FUEL LOGIC
(Background software module ADAPT)

OVERVIEW

Fuel injected systems may exhibit vehicle to vehicle steady state A/F ratio errors due to normal variability in fuel system components.

The adaptive fuel strategy attacks this problem by memorizing the characteristics of the individual fuel system being used. This memorized information is used to predict what the system will do based on past experience.

The ability to predict fuel system behavior improves both open loop and closed loop fuel control. As an example, the memorized information can be used on cold starts to achieve better open loop fuel control before the EGO sensor reaches operating temperature.

The chief benefit of the adaptive fuel strategy will be to reduce the effects of product variability in the field.

The memorized or adaptive information is stored in table form in the Keep Alive Memory (KAM). KAM is continuously powered by the vehicle battery even when the vehicle is shut off. As a result, the table is not lost on vehicle shutdown.

ADAPTIVE FUEL TABLE

The adaptive fuel table, LTMTBL, is a 2-dimensional array of learned fuel system corrections. Ideally, if LAMBSE = 1.0 and data from a mature adaptive fuel table is used, a stoichiometric A/F ratio would result at whatever speed-load point adaptive learning had taken place.

Present table size is 10 (rows) x 10 (columns) plus 6 special idle adaptive cells, for a total of 106 cells.

The total learned fuel system correction is called KAMREF where KAMREF = 0.5 + LTMTBLrc.

During adaptive learning, only the LTMTBL cells are modified. Therefore, the range of the KAMREF multiplier is (0.5 + 0.0) to (0.5 + 1.0) or 0.5 to 1.5.

The range of the LTMTBL cells can be further restricted by use of the calibration parameter clips, MINADP and MAXADP.

The precise location where KAMREF is used is shown in the FUELPW equation.

If KAM fails the KAM validation test (described later), all LTMTBL cells are initialized to 0.5 or 80 (HEX) resulting in a value of KAMREF = 0.5 + 0.5 = 1.0.

When allowed, updates to LTMTBL are statistically distributed in the vicinity of the speed-load operating point except in the case of the idle cells. Only the current idle cell is updated, no statistical distribution is done.

Data extracted from the table undergoes a 4 point linear interpolation. This is explained further under the FN1325L description. Note that idle cells do not undergo four point interpolation. Only the current idle cell is used.

LTMTBL and FN1325L share the same normalizing functions. FN031 is the MAPOPE normalizing function. FN070L is the engine speed N normalizing function.

The adaptive fuel table, LTMTBL, is shown on the next page.

ADAPTIVE FUEL TABLE (LTMTBL)

LTMTBLrc CELLS

```
special idle
adaptive cells ----->    |100 101 102 103 104 105
                         |
                    9    |90  91  92  93  94  95  96  97  98  99
                         |
                    8    |80  81  82  83  84  85  86  87  88  89
                         |
                    7    |70  71  72  73  74  75  76  77  78  79
                         |
                    6    |60  61  62  63  64  65  66  67  68  69
                         |
                    5    |50  51  52  53  54  55  56  57  58  59
                         |
       FN031        4    |40  41  42  43  44  45  46  47  48  49
     NORMALIZED          |
       ENGINE       3    |30  31  32  33  34  35  36  37  38  39
       LOAD              |
      (MAPOPE)      2    |20  21  22  23  24  25  26  27  28  29
                         |
                    1    |10  11  12  13  14  15  16  17  18  19
                         |
                    0    |00  01  02  03  04  05  06  07  08  09
                         ---------------------------------------

                           0   1   2   3   4   5   6   7   8   9

                               FN070L  NORMALIZED
                                 ENGINE SPEED
                                     (N)
```

FN1325L DESCRIPTION

FN1325L is an 11 (row) x 10 (column) table containing 1 cell corresponding to each cell in the adaptive fuel table LTMTBL. (The 11th row is used to reference the idle cells and is not accessible from FN031 which only goes from 0 to 9)

The normalizing functions for FN1325L and LTMTBL are shared. FN031 is the MAPOPE normalizing function. FN070L is the engine speed N normalizing function.

FN1325L is designed to do the following:

- Identify LTMTBL cells where learning is allowed to occur.
    - Learning is allowed in any LTMTBLrc cell whose corresponding FN1325Lrc cell contains a value >= 0. Negative FN1325L cell values disallow learning in the corresponding LTMTBL cell.

- Define a high confidence speed-load region that can be referenced from any other speed-load point.
    - This occurs whenever a negative value is entered into a FN1325L cell. The negative number serves as an offset to LTMTB00. If 1 of the 4 cells used by the 4 point linear interpolation LTMTBL table lookup routine contained -42, the cell value used by the interpolation routine for the cell that contained the -42 would be the value found in the LTMTBL cell located at the intersection of row 4 and column 2. In the extreme, if -42 was entered into every cell of FN1325Lrc except for the cell corresponding to LTMTB42, learning would be allowed only in cell LTMTB42 and the learned correction in LTMTB42 would be applied to all speed-load points (every cell referenced by the 4 point linear interpolation routine during the LTMTBL table lookup would point to cell LTMTB42). This calibration for FN1325L is shown on the next page.

- Specify the values of LOPCT1 and LOPCT2 required to update an individual LTMTBL cell.
    - This is done by entering a value into FN1325L that is >= 0. The value entered represents 1/2 the required update value. A value of 20 entered would require LOPCT1 and/or LOPCT2 to be greater than 40 for an update to occur.

LTMTBL LEARNING/USE CONTROL TABLE (FN1325L)

FN1325L CELLS

FN031 NORMALIZED ENGINE LOAD (MAPOPE)	0	1	2	3	4	5	6	7	8	9
special idle adaptive cells -->	-42	-42	-42	-42	-42	-42				
9	-42	-42	-42	-42	-42	-42	-42	-42	-42	-42
8	-42	-42	-42	-42	-42	-42	-42	-42	-42	-42
7	-42	-42	-42	-42	-42	-42	-42	-42	-42	-42
6	-42	-42	-42	-42	-42	-42	-42	-42	-42	-42
5	-42	-42	-42	-42	-42	-42	-42	-42	-42	-42
4	-42	-42	20	-42	-42	-42	-42	-42	-42	-42
3	-42	-42	-42	-42	-42	-42	-42	-42	-42	-42
2	-42	-42	-42	-42	-42	-42	-42	-42	-42	-42
1	-42	-42	-42	-42	-42	-42	-42	-42	-42	-42
0	-42	-42	-42	-42	-42	-42	-42	-42	-42	-42

FN070L NORMALIZED
ENGINE SPEED
(N)

DEFINITIONS

INPUTS

Registers:

- ACT = Air Charge Temperature, deg F.
- ADPTMR = Adaptive Learning Enable Timer (see TIMER section).
- AEFUEL = Acceleration enrichment fuel flow, lb/hr.
- BIAS = A/F biasing term: FN1355(N,MAP). Units are lambdas.
- CHKSUM = KAM word containing the sum of the LTMTBL contents.
- ECT = Engine Coolant Temperature.
- EFTR = EQUIL fuel transfer rate BIN 16 LBM/SEC.
- EGOCNT = Number of EGO switches required before allowing updates to the Adaptive Fuel cell.
- FUELPW = Fuel Pulsewidth.
- KWUCTR = KAM Warm Up counter. Stores number of warm ups in KAM. Reset to zero if KAM is corrupted (battery disconnect, etc.).
- LAMBSE = Desired air/fuel ratio.
- LAMWIN = LAMBSE window outside which adaptive is enabled.
- LSTROW = Last pass normalized row.
- LTMTBL = Adaptive Fuel Table.
- MAPOPE = MAP/ABS exhaust pressure.
- N = Engine speed RPM.
- PTPAMP = Limit cycle peak_to_peak amplitude
- PURGDC = Canister Purge Duty Cycle.
- RANNUM = Random numbers used to statistically distribute the corrections to the Adaptive Fuel Table among four adjacent cells.
- TCSTRT = ECT at start-up.
- UPRATE = KAM update rate.

Bit Flags:

- AFMFLG = ACT Failure flag set to 1 if the ACT fails range check.

- CFMFLG = ECT Failure flag set to 1 if the ECT fails range check.

- DFSFLG = Indicates decel fuel shut off.

- DISABLE_ADAPT = Adaptive fuel disable flag; 1 -> disable adaptive fuel.

- HCAMFG = Flag indicating the completion of Hi-Cam; 0 -> no desired engine speed adder exists, 1 -> an rpm adder above base idle is present. Flag is used in the ISC adaptive update routine to disable updates when HCAMFG = 1.

- ISCFLG = ISC MODE Flag; 1 -> rpm control mode.

- ISFLAG = Indication of engine load state at Idle (See ISC Chapter).

- LIMIT_PURGE = Flag which indicates Purge Duty Cycle is being limited due to LAMBSE being clipped; 1 -> limited Purge.

- MFMFLG = MAP Failure flag set to 1 if MAP sensor fails.

- REFFLG = Indication of Idle Air Flow; 1 -> Idle Air Flow.

- SWTFL = EGO switch flag; 0 -> no EGO switch, 1 -> EGO switch this background loop.

- TFMFLG = TP Failure flag set to 1 if TP sensor fails range check.

- WARM_UP = Engine Warm-up flag; 1 -> engine warmed-up.

Calibration Constants:

- ADAPTM = Adaptive learning enable time delay (seconds).

- ADEFTR = Transient fuel threshold to update adaptive fuel.

- ADEGCT = Number of EGO switches required to permit Adaptive Learning within the cell boundaries.

- AELIM = Maximum acceleration enrichment fuel flow to allow adaptive learning, lb/hr.

- AFACT1 = Minimum ACT to Update Adaptive Fuel Table, deg F.

- AFACT2 = Maximum ACT to Update Adaptive Fuel Table, deg F.

- AMPMUL = Multiplier to determine LAMWIN from PTPAMP.

- DELAMB = Deadband (around LAMBSE = 1.0) within which loop counter values are not altered.

- DELCOL = Calibration constant (normalized engine speed N) which provides the ability to lock out table updates under transient conditions; establishes an operating range (engine speed) within which the appropriate loop counter may be incremented.

- DELROW = Same function as DELCOL but for normalized load MAPOPE.

- FAEGCT = Fast Adaptive EGO count. Number of EGO switches required to permit adaptive learning when KWUCTR < KWUCNT. Should be set to 0 to permit fast adaptive learning for the first few warm-up cycles.

- FN031 = MAPOPE normalizing function used with both FN1325L and LTMTBL.

- FN025 = MAP normalizing function for Bias, Peak-to-Peak and Transport Delay Tables. Input = MAP, Output = Table Entry Point.

- FN039 = Engine Speed (N) normalizing function for FN1355.

- FN070L = Normalizing function for N, used with FN1325L as well as LTMTBL.

- FN1325Lrc = LTMTBL learning/use control table.

- FN1355 (N,MAP) = Amount of BIAS from stoichiometry, units are lambdas. X-input = FN039 (Normalized engine speed, rpm). Y-input = FN025 (Normalized Manifold Absolute pressure, Inches Hg), Output = BIAS from stoichiometry, lambdas.

- HCAMSW = Calibration switch which allows the developer to select how the adaptive fuel idle cells are to be used.

  If HCAMSW is set to 0, the adaptive fuel idle cells are used as soon as the filtered air mass region is entered (REFFLG = 1).

  If HCAMSW is set to 1, the adaptive fuel idle cells are used only when in the filtered air mass region and no rpm adder above base idle is present (HCAMFG = 0). This includes FN825A, FN825B, FN826, and BZZRPM.

- KWUCNT = Maximum number of warm-up cycles to use fast adaptive EGO count. Should be set to approximately 3 to 5 warm-ups.

- MAXADP = Maximum adaptive correction.

- MINADP = Minimum adaptive correction.

- MINPW = Minimum pulsewidth clip value.

- RANMUL = Multiplier for random number generation.

- VECT3 = Minimum coolant temperature, engine on.

- VECT5 = Starting coolant temperature for warm-up counter.

OUTPUTS

Registers:

- COLTBU = Column address of Adaptive cell to be updated, integer = FN070L + 0.5 + upper byte of RANNUM (where the upper byte of RANNUM is a random number ranging from -0.5 to 0.496).

- EGOCNT = See above.
- FUELPW = See above.
- KWUCTR = See above.
- LAMWIN = See above.
- LOPCT1 = See above.
- LOPCT2 = See above.

  NOTE: A background loop occurs when the computer reaches the same point in the program after executing all other necessary instructions.

- LSTCOL = Last pass normalized column.
- LSTROW = See above.
- LTMTBL = Adaptive Fuel Table.
- RANNUM = See above.
- ROWTBU = Row address of adaptive cell to be updated, integer = FN031 + 0.5 + lower byte of RANNUM (where the lower byte of RANNUM is a random number ranging from -0.5 to 0.496).
- UPRATE = See above.

Bit Flags:

- DISABLE_ADAPT = Adaptive fuel disable flag; 1 -> disable adaptive fuel.
- WARM_UP = Engine Warm-up flag; 1 -> engine warmed-up.

PROCESS

STRATEGY MODULE: FUEL_ADAPT_COM4

Calculate the random number to be used this background pass:

```
always -----------------------------------| RANNUM = RANNUM * RANMUL +
                                          |            RANMUL/4
                                          | rancol = rannum_hi
                                          |        = the high byte of the
                                          |          low word of RANNUM
                                          | ranrow = rannum_lo
                                          |        = the low byte of the low
                                          |          word of RANNUM
                                          | RANNUM = the low word of RANNUM
                                          |          (save for next pass)
```

Calculate the cell to be updated this background pass:

```
REFFLG = 1 -------------------------|
                                    |
ISCFLG = 1 -------------------------|AND -| kamcol = ISFLAG
                                    |     | COLTBU = ISFLAG
HCAMFG = 0 --------------------|    |     | kamrow = 10
                               |OR --|    | ROWTBU = 10
HCAMSW = 0 --------------------|          |  (special idle cells)
                                          |
                                          | --- ELSE ---
                                          |
                                          | kamcol = FN070L(N)
                                          | COLTBU = kamcol + 0.5 + rancol
                                          | kamrow = FN031(MAPOPE)
                                          | ROWTBU = kamrow + 0.5 + ranrow
```

Calculate the DISABLE_ADAPT value:

```
DFSFLG = 1 -------------------------------| DISABLE_ADAPT = 1
                                          |
                                          | --- ELSE ---
                                          |
FUELPW < MINPW ---------------------------| DISABLE_ADAPT = 1
                                          | FUELPW = MINPW
                                          |
                                          | --- ELSE ---
                                          |
                                          | DISABLE_ADAPT = 0
```

Calculate EGO switch logic:

```
SWTFL = 1 --------------------------------| Increment EGOCNT
 (EGO switch)
```

Determine if adaptive is enabled/disabled of if adaptive area has changed:

```
AFMFLG = 1 --------------------------|
                                     |
CFMFLG = 1 --------------------------|
                                     |
TFMFLG = 1 --------------------------|
                                     |
MFMFLG = 1 --------------------------|
                                     |
ACT <= AFACT1 -----------------------|
                                     |
ACT >= AFACT2 -----------------------|
                                     |OR --| LOPCT1 = 0
DISABLE_ADAPT = 1 -------------------|     | LOPCT2 = 0
                                     |     | EGOCNT = 0
OPEN LOOP FUEL CONTROL --------------|     | LSTCOL = kamcol
                                     |     | LSTROW = kamrow
ADPTMR < ADAPTM ---------------------|     |
                                     |     | Do: NEW_UPRATE_CALC
|kamrow - LSTROW| > DELROW ----------|     |
                                     |     | Exit FUEL_ADAPT_COM4
|kamcol - LSTCOL| > DELCOL ----------|     |
                                     |     |
AEFUEL > AELIM ----------------------|     |
                                     |     |
|EFTR| >= ADEFTR --------------------|     |
                                     |     |
REFFLG = 1 --------------------|     |     |
                               |AND -|     |
ISCFLG <> 1 -------------------|           |
                                           | --- ELSE ---
                                           |
                                           | Continue with FUEL_ADAPT_COM4
```

Calculate the LAMBSE window, LAMWIN:

```
always ------------------------------------| LAMWIN = DELAMB + (PTPAMP *
                                           |           AMPMUL)
```

Calculate the EGO based learning rate:
(This logic gives the system the capability to learn faster during 'green engine conditions').

```
WARM_UP = 0 -------------------------|
                                     |
TCSTRT < VECT5 ----------------------|
                                     |AND -| WARM_UP = 1
ECT > VECT3 -------------------------|     | KWUCTR = KWUCTR + 1
                                     |     | Clip KWUCTR to 255
RUN mode ----------------------------|


KWUCTR < KWUCNT ---------------------------| egolearn_rat = FAEGCT
 (first few warm-up cycles)                |  (use fast learning rate)
                                           |
                                           | --- ELSE ---
                                           |
KWUCTR >= KWUCNT --------------------------| egolearn_rat = ADEGCT
                                           |  (use normal learning rate)
```

Calculate loop counters:

(LOPCT1 and LOPCT2)

```
EGO IS RICH -------------------------|
                                     |
LAMBSE >= (1 + BIAS + LAMWIN) -------|
                                     |AND -| Increment LOPCT1 (max 255)
EGOCNT >= egolearn_rat --------------|     |
                                     |     |
LIMIT_PURGE = 0 ----------------|    |     |
                                |OR --|    |
PURGDC = 0 ---------------------|          |
                                           | --- ELSE ---
EGO IS LEAN -------------------------|     |
                                     |     |
LAMBSE <= (1 + BIAS - LAMWIN) -------|AND -| Increment LOPCT2 (max 255)
                                     |     |
EGOCNT >= egolearn_rat --------------|     |
                                           | --- ELSE ---
                                           |
                                           | No change to LOPCT1 or LOPCT2
```

Determine if adaptive cells should be incremented/decremented:
LTMTBL cells are updated when the following conditions are satisfied:
Note that r = ROWTBU and c = COLTBU in the following charts.

```
FN1325Lrc >= 0 ---------------------|
                                    |
LOPCT1 > 2 * UPRATE ----------------|AND -| LTMTBLrc = LTMBLrc - 0.0039
                                    |     | Decrement CHKSUM
LTMTBLrc > MINADP ------------------|     | LOPCT1 = 0
                                          | EGOCNT = 0
                                          |
                                          | --- ELSE ---
                                          |
LOPCT1 > 2 * UPRATE ----------------------| LOPCT1 = 0
                                          | EGOCNT = 0
                                          |
                                          | --- ELSE ---
LOPCT2 > 2 * UPRATE ----------------|     |
                                    |     |
LTMTBLrc < MAXADP ------------------|AND -| LTMTBLrc = LTMTBLrc + 0.0039
                                    |     | Increment CHKSUM
FN1325Lrc >= 0 ---------------------|     | LOPCT2 = 0
                                          | EGOCNT = 0
                                          |
                                          | --- ELSE ---
                                          |
LOPCT2 > 2 * UPRATE ----------------------| LOPCT2 = 0
                                          | EGOCNT = 0
```

BEGIN: NEW_UPRATE_CALC

This algorithm calculates the average loop counter value to be used during the adaptive algorithm. The standard four point interpolation routine is used. If, however, one or more of the cells in the four points is a reference cell, then the average of the values in the positive cells is used in place of the negative values.

If the system is in the special idle cells, the UPRATE value is equal to the value in FN1325L specified by kamcol.

```
Note:  kamrow_max = maximum number of table rows (excluding the Special Idle
                    Cells row).
       kamrow_spec = row number for the Special Idle Adaptive Cells.
       kamcol_max = maximum number of table columns.


kamrow = kamrow_spec ------------| Read the value in FN1325L specified
                                 |  by kamcol.
                                 | Load this value in UPRATE.
                                 |
                                 | Exit the subroutine NEW_UPRATE_CALC
                                 |
                                 | --- ELSE ---
                                 |
                                 | Continue with subroutine NEW_UPRATE_CALC
```

Separate kamcol and kamrow into an integer and remainder.

```
always --------------------------| kamcol_rem = kamcol - int(kamcol)
                                 | kamcol = int(kamcol)
                                 | kamrow_rem = kamrow - int(kamrow)
                                 | kamrow = int(kamrow)
```

Check for row and column boundary conditions:

```
kamcol < kamcol_max -------|     | adapt1_uprat = FN1325L(kamcol,kamrow)
                           |AND -| adapt2_uprat = FN1325L(kamcol + 1,kamrow)
kamrow < kamrow_max -------|     | adapt3_uprat = FN1325L(kamcol,kamrow + 1)
                                 | adapt4_uprat =
                                 |           FN1325L(kamcol + 1,kamrow + 1)
                                 |
                                 | --- ELSE ---
kamcol = kamcol_max -------|     |
                           |AND -| adapt1_uprat = FN1325L(kamcol,kamrow)
kamrow = kamrow_max -------|     | adapt2_uprat = adapt1_uprat
                                 | adapt3_uprat = adapt1_uprat
                                 | adapt4_uprat = adapt1_uprat
                                 | (use FN1325L 4 times so not to wrap
                                 |   around table or use special idle cells)
                                 |
                                 | --- ELSE ---
```

(continued on next page)

(continued from previous page)

```
                                 |
kamcol = kamcol_max -------------| adapt1_uprat = FN1325L(kamcol,kamrow)
                                 | adapt2_uprat = adapt1_uprat
                                 | adapt3_uprat = FN1325L(kamcol,kamrow + 1)
                                 | adapt4_uprat = adapt3_uprat
                                 | (use kamcol_max values twice so not to
                                 |   wrap around table)
                                 |
                                 | --- ELSE ---
                                 |
kamrow = kamrow_max -------------| adapt1_uprat = FN1325L(kamcol,kamrow)
                                 | adapt2_uprat = FN1325L(kamcol + 1,kamrow)
                                 | adapt3_uprat = adapt1_uprat
                                 | adapt4_uprat = adapt2_uprat
                                 | (use kamrow_max values twice so not to
                                 |   wrap around table or use special
                                 |   idle cells)


always --------------------------| uprat_cnt = 0


Check for reference cells:

adapt1_uprat >= 0 ---------------| uprat_cnt = uprat_cnt + 1
                                 | adapt1_uprat_ref = 0
                                 |
                                 | --- ELSE ---
                                 |
                                 | adapt1_uprat = 0
                                 | adapt1_uprat_ref = 1


adapt2_uprat >= 0 ---------------| uprat_cnt = uprat_cnt + 1
                                 | adapt1_uprat_ref = 0
                                 |
                                 | --- ELSE ---
                                 |
                                 | adapt2_uprat = 0
                                 | adapt2_uprat_ref = 1


adapt3_uprat >= 0 ---------------| uprat_cnt = uprat_cnt + 1
                                 | adapt3_uprat_ref = 0
                                 |
                                 | --- ELSE ---
                                 |
                                 | adapt3_uprat = 0
                                 | adapt3_uprat_ref = 1
```

```
adapt4_uprat >= 0 ---------------| uprat_cnt = uprat_cnt + 1
                                 | adapt4_uprat_ref = 0
                                 |
                                 | --- ELSE ---
                                 |
                                 | adapt4_uprat = 0
                                 | adapt4_uprat_ref = 1
```

If all cells are referenced cells, set UPRATE to a maximum value. Else, find UPRATE average.

```
uprat_cnt = 0 -------------------| UPRATE = 127 (maximum value)
                                 |
                                 | --- ELSE ---
                                 |
                                 | uprat_avg = (adapt1_uprat +
                                 |      adapt2_uprat + adapt3_uprat +
                                 |      adapt4_uprat) / uprat_cnt
                                 |
                                 | Truncate UPRATE to an integer


adapt1_uprat_ref = 1 ------------| adapt1_uprat = uprate_avg


adapt2_uprat_ref = 1 ------------| adapt2_uprat = uprate_avg


adapt3_uprat_ref = 1 ------------| adapt3_uprat = uprate_avg


adapt4_uprat_ref = 1 ------------| adapt4_uprat = uprate_avg


always --------------------------| Make kamcol and kamrow real numbers
                                 |   kamcol = kamcol + kamcol_rem
                                 |   kamrow = kamrow + kamrow_rem
                                 |
                                 | Do four point interpolation using
                                 | kamrow and kamcol, and the four
                                 | adaptn_uprat values  calculated
                                 | above.  Store this value in UPRATE.
```

END: NEW_UPRATE_CALC

KAM ADAPTIVE FUEL LOGIC
(Background software module KAMREF, called from BG_FUELPW)

OVERVIEW

The adaptive fuel table stored in KAM is used as a reference for both open and closed loop fuel control. The use of KAMREF is shown in the pulsewidth equation section of this chapter.

NOTE: The following Adaptive Fuel Logic is not executed during ENGINE RUNNING Self Test MODE.

DEFINITIONS

INPUTS

Registers:

- ISFLAG = Indication of engine load state at idle (See IDLE SPEED CONTROL Chapter); 0 -> Drive, 1 -> Drive and A/C clutch engaged, 2 -> Neutral, 3 -> Neutral and A/C clutch engaged.

- LTMTBL = Adaptive fuel table.

Bit Flags:

- CRKFLG = Flag indicating engine mode; 1 -> Cranking, 0 -> Run or Underspeed mode.

- HCAMFG = Flag indicating the completion of Hi-Cam; 0 -> no desired engine speed adder exists, 1 -> an rpm adder above base idle is present. Flag is used in the ISC adaptive update routine to disable updates when HCAMFG = 1.

- REFFLG = Indication of Idle Air Flow; 1 -> Idle Air Flow.

Calibration Constants:

- HCAMSW = Calibration switch which allows the developer to select how the adaptive fuel idle cells are to be used.

  If HCAMSW is set to 0, the adaptive fuel idle cells are used as soon as the filtered air mass region is entered (REFFLG = 1).

  If HCAMSW is set to 1, the adaptive fuel idle cells are used only when in the filtered air mass region and no rpm adder above base idle is present (HCAMFG = 0). This includes FN825A, FN825B, FN826, and BZZRPM.

OUTPUTS

Registers:

- KAMREF = Adaptive fuel strategy correction factor.

PROCESS

STRATEGY MODULE: FUEL_KAM_ADAPT_COM3

```
CRKFLG = 1 ----------------------------------| KAMREF = 1.0
 (crank mode)                                |  (use no interpolation)
                                             |
                                             | --- ELSE ---
REFFLG = 1 ----------------------------|     |
                                       |AND -| KAMREF = 0.5 + LTMTBLrc
HCAMFG = 0 ----------------------|     |     |  (where r = 10 and
 (no rpm adder)                  |OR --|     |   c = ISFLAG)
                                 |           |  (use no interpolation)
HCAMSW = 0 ----------------------|           |
 (ignore HCAMFG)                             | --- ELSE ---
                                             |
                                             | KAMREF = 0.5 + LTMTBLrc
                                             |  (use 4-point
                                             |   interpolation)
```

NOTE: For purposes of interpolation, the LTMTBL100 to LMTBL105 cells are not included. These cells should correspond to the special idle cells.

TRANSIENT FUEL COMPENSATION
(Background software module TFCOMP, called from BG_FUELPW)

OVERVIEW

Transient Fuel is variously referred to as manifold wall wetting, puddling, filling, and fuel film condensation/evaporation.

A liquid fuel film resides on the walls of the intake manifold. The film mass varies primarily with manifold absolute pressure and manifold wall temperature. During steady state conditions, the film mass is constant. The rates of condensation and evaporation on the manifold walls are equal. During transients, the film mass changes creating air/fuel ratio errors.

- During accelerations, the film mass increases. Fuel will condense faster on the manifold walls until equilibrium is reached. In an uncompensated system at stoichiometry, fuel is diverted from the cylinders, resulting in a momentary lean condition.

- During decelerations, the film mass decreases. Fuel will evaporate faster from the manifold walls until equilibrium is reached. In an uncompensated system at stoichiometry, fuel is added to the cylinders, resulting in a momentary rich condition.

The problem is magnified in closed loop fuel systems because the fuel control will incorrectly chase the transient air/fuel excursions.

INTENT

The Transient Fuel Compensation Strategy (TFC) augments the closed/open loop fuel control to keep cylinder events at the desired air/fuel ratio during all engine transients. The goals are:

- eliminate lean air/fuel excursions during accelerations, and

- eliminate rich air/fuel excursions during decelerations.

NOTE: Transient Fuel Compensation is not run in Self Test.

APPROACH

The computer adjusts fuel flow to match the transient fuel flow to or from the manifold fuel film.

```
COMPENSATED   CLOSED/OPEN LOOP   FILM MASS RATE OF CHANGE
 FUEL FLOW  =    FUEL FLOW     +   OR FUEL FLOW
 (ACTUAL)      (BASE STRATEGY)    (TFC STRATEGY)
```

The film mass rate of change is proportional to the amount of fuel that must be added to or subtracted from the manifold film.

```
                                     /                      \
FILM MASS            1              |  STEADY       ACTUAL  |
RATE OF      = --------------- *    |  STATE    -           |
CHANGE          TIME CONSTANT       | FILM MASS   FILM MASS |
                                     \                      /
```

The time constant and steady state film mass are calculated from MAP and temperature variables and must be calibrated for different applications.

The actual film mass is a time integration of the film mass rate of change.

```
                             /                            \
ACTUAL      ACTUAL          | FILM MASS        TIME SINCE |
FILM    =   FILM     +      | RATE OF     *      LAST     |
MASS        MASS            | CHANGE             UPDATE   |
                             \                            /
```

DEFINITIONS

INPUTS

Registers:

- AISF = Actual Intake Surface Fuel Calculation, lb.
- ATMR1 = Time since Start-up, sec.
- DELTIM = Time since last AISF update, sec.
- EFTR = Equilibrium fuel transfer rate, fuel flow to and from the manifold puddle.

Bit Flags:

- DFSFLG = Indicates decel fuel shut off.
- EFFLG1 = Equilibrium Fuel Transfer flag.
- ISCFLG = ISC Mode Flag. 1 -> RPM control; 2 -> RPM lockout
- MPGFLG = Manage fuel air state flag.
- REFFLG = Indicates idle fuel modulation enabled.

Calibration Constants:

- AISFM = Multiplier on AISF when in DFSO (0 - 2). Determines Fuel Puddle size upon re-entering Normal Fuel.
- ALPHA = Multiplier proportioning the dependency of ACT to ECT.
- EFTC = TEFTC(FN1322) = an 8 X 10 table of equilibrium fuel transfer time constants as a function of ALPHA * ACT + (1 - ALPHA) * ECT and MAP.
- EISF = TEISF(FN1321) = an 8 X 10 table of fuel mass values as a function of ALPHA * ACT + (1 - ALPHA) * ECT and MAP.
- FN307(N_BYTE) = MTEFTC Multiplier as a function of N_BYTE.
- FN1321(TEISF) = Equlibrium intake surface fuel function.
- KFT = Multiplier (if set = 0, disables TFC) when not in MPG mode.
- KFTMPG = Transient Fuel multiplier in MPG mode.
- MEFTRA = Transient fuel PW multiplier during accels.
- MEFTRD = Transient fuel PW multiplier during decels.
- MTEISF = Multiplier for FN1321.
- TFCBITS = Minimum difference in Equilibrium Intake Surface Fuel to trigger transient fuel. 1 bit = 0.000015259, therefore to have a deadband of 5 bits, set TFCBITS = 0.000076295.

  NOTE: To see how many bits are contained in the desired fuel puddle, calculate FN1321 * MTEISF/0.000015259. For example, if FN1321 = 0.0014648, MTEISF = 0.0625, then the number of bits in the puddle is 6. Thus even a 1 bit change would require an 18% TFCDED to eliminate it. TFCBITS resolves this issue when puddle values are extremely small.

- TFCDED = Percentage deadband around Equilibrium Intake Surface Fuel to turn off Transient Fuel.

- TFCISW = Switch for Transient Fuel Control.

- TFCTM = Time since entering Transient Fuel, sec.

- TFRFSW = Trans fuel REFFLG enable switch; 0 = disable REFFLG logic.

OUTPUTS

Registers:

- AISF = Actual Intake Surface Fuel Calculation, lb.

- EFTR = Equilibrium fuel transfer rate, fuel flow to and from the manifold puddle.

- EFTRFF = Equilibrium fuel flow.

Bit Flags:

- EFFLG1 = Equilibrium Fuel Transfer flag.

PROCESS

STRATEGY MODULE: FUEL_TRANS_COM5

```
CRANK mode --------------------|
 (CRKFLG = 1)                  |
                               |
UNDERSPEED mode ---------------|OR --| EFFLG1 = 0
 (UNDSP = 1)                   |     | EFTR = 0
                               |     | EFTRFF = 0
ATMR1 < TFCTM -----------------|     |  (do not run transient
                                     |   fuel compensation)
                                     |
                                     | --- ELSE ---
EFFLG1 = 0 --------------------|     |
                               |     |
RUN mode ----------------------|AND -| AISF = MTEISF * FN1321(TEISF)
 (CRKFLG = 0, UNDSP = 0)       |     | EFFLG1 = 1
                               |     |  (do not run transient
TFCISW = 1 --------------------|     |   fuel compensation)
 (assume wet manifold at start-up)   |
                                     | --- ELSE ---
EFFLG1 = 0 --------------------|     |
                               |     |
RUN mode ----------------------|AND -| AISF = 0
                               |     | EFFLG1 = 1
TFCISW = 0 --------------------|     |  (do not run transient)
 (assume dry manifold at start-up)   |
                                     | --- ELSE ---
RUN mode ----------------------|     |
                               |     |
DFSFLG = 1 --------------------|     |
 (in DFSO)                     |AND -| AISF = MTEISF * FN1321(TEISF) * AISFM
                               |     | EFFLG1 = 1
EFFLG1 = 1 --------------------|     | EFTR = 0
                                     |  (do not run transient)
                                     |
                                     | --- ELSE ---
EFFLG1 = 1 --------------------|     |
                               |AND -| Do ACTUAL INTAKE SURFACE
RUN mode ----------------------|     |  FUEL CALCULATIONS
 (CRKFLG = 0, UNDSP = 0)             | Do EQUILIBRIUM FUEL
                                     |  TRANSFER CALCULATIONS
                                     |  (run transient fuel
                                     |   compensation)
```

NOTE: During ENGINE RUNNING Self Test MODE, the above logic is not executed; the Self Test software prevents Transient Fuel Compensation.

EQUILIBRIUM FUEL TRANSFER CALCULATIONS

These calculations are performed during each program pass (background loop) while Transient Fuel Compensation is enabled. The general form of the rate calculation is:

EFTR = A * [(EISF * MTEISF - AISF) / (EFTC * FN307(N_BYTE))]

```
MPGFLG = 0 ------------------------------------| A = KFT
 (not in MPG mode)                             |
                                               | --- ELSE ---
                                               |
MPGFLG = 1 ------------------------------------| A = KFTMPG
 (in MPG mode)
```

ACTUAL INTAKE SURFACE FUEL CALCULATION (AISF)

This calculation is performed during each program pass (background loop) while TFC is enabled. AISF is a time integration of the fuel flow to and from the manifold puddle. Clip AISF to 0 as a minimum.

AISF = AISF + (EFTR * DELTIM)

TRANSIENT FUEL FLOW MULTIPLIER

```
REFFLG = 1 ------------------------|
                                   |AND -|
TFRFSW = 1 ------------------------|     |
 (enable REFFLG logic)                   |
                                         |
ISCFLG > 0 ------------------------|     |
 (in RPM control)                  |     |
                                   |AND -|OR --| EFTRFF = 0
TFISCW = 1 ------------------------|     |     |  (stop adding
 (enable ISCFLG logic)                   |     |   transient fuel to
                                         |     |   pulsewidth continue
|MTEISF * FN1321 - AISF| <= TFCBITS -----|     |   AISF update)
 (difference is less than a few bits)    |     |
                                         |     |
|(MTEISF * FN1321 - AISF) /              |     |
    (MTEISF * FN1321)| <= TFCDED --------|     |
 (percentage difference is small)              |
                                               | --- ELSE ---
                                               |
MTEISF * FN1321 < AISF ------------------------| EFTRFF = MEFTRD * 60
                                               |    EFTR
                                               |  (use decel multiplier)
                                               |
                                               | --- ELSE ---
                                               |
                                               | EFTRFF = MEFTRA * 60
                                               |    * EFTR
                                               |  (use accel multiplier)
```

HOT INJECTOR COMPENSATION
(Done in background software module CNVRT)

OVERVIEW

Under conditions of high injector tip temperatures, injector fuel delivery decreases as a function of increasing injector tip temperature. The amount of vaporized fuel delivered by the injector increases as hot soak time increases, or as conducted heat (from cylinder head) and/or radiated heat (from intake/exhaust manifold) increases. Higher fuel pressure or lower fuel volatility helps the situation, but fuel volatility is beyond the developers control. Hot Injector compensation has therefore been applied to the fuel delivery slope A0. Tip temperature has been characterized to be a function of ECT/ACT, ACSTRT, ATMR3 and AM as follows:

- ECT/ACT for conducted or radiated heat.
- ACSTRT for soak time.
- ATMR3 for fuel mass to cool the injector.
- AM for fuel mass to cool the injector.

Hot injector compensation, as a percent above base A0 is, HICOMP = FN1349(TEMP,ACSTRT) * FN1348(ATMR3,AM).

A0 corrected = A0/(1 + HICOMP) = A0COR

Thus when HICOMP goes to 0, no enrichment is desired. FN1350 has been added for crank fuel enrichment as a function of ACT and CRKPIP_CTR. ACT is the input - in case of a stall and no key off, you will get a better enrichment factor as opposed to ACSTRT.

DEFINITIONS

INPUTS

Registers:

- ACSTRT = ACT at start-up; arithmetic average of the first 8 ACT readings.
- ACT = Air Charge Temperature, degrees F.
- AM = Air mass flow, lb/min.
- ATMR3 = Time since entering RUN mode, secs.
- CRKPIP_CTR = Foreground PIP counter for crank fuel.
- ECT = Coolant temp, degree F.

Calibration Constants:

- FN1349 = Fuel enrichment factor. X = FN005(TEMP) Y = FN005(ACSTRT)
- FN1348 = Time multiplier. X = FN008(ATMR3) Y = FN007(AM)
- FN1350 = CRANK fuel enrichment multiplier as a function of "number of PIPs" in CRANK and ACT. X = FN023(CRKPIP_CTR) Y = FN024(ACT)
- FRCHIC = Fraction of ECT or ACT to use in FN1349. If FRCHIC = 1.0, all ACT is used, if FRCHIC = 0, all ECT is used.

OUTPUTS

Registers:

- HICOMP = Hot injector compensation enrichment factor.

PROCESS

STRATEGY MODULE: FUEL_HOT_INJ_COMP_COM2

HICOMP = FN1349(TEMP,ACSTRT) * FN1348(ATMR3,AM)

INJECTOR DELAY LOGIC

OVERVIEW

UNSYNCHRONIZED (SYNFLG = 0 or FUEL_SYNC = 0) The injector timing routine is disabled (IBETA = 0). All injector output ports are fired on the rising edge of their respective reference PIPs.

SYNCHRONIZED (SYNFLG = 1 and FUEL_SYNC = 1) All injector output port firings are delayed CIBETA PIP periods from their respective reference PIPs.

NOTE: A Signature PIP distributor is required in order to delay injector firing by CIBETA PIP periods. If a Signature PIP distributor is not used, SYNFLG will always be 0.

CALIBRATION PHILOSOPHY

CIBETA should be calibrated so that the injector firings occur within the optimum window determined by camshaft geometry.

$$\text{CIBETA} = \frac{\text{DESIRED INJECTOR DELAY IN CRANK DEGREES - 10 DEG}}{\text{CRANK DEGREES BETWEEN PIPS}}$$

The range of CIBETA is 0 to 6.5 PIP periods.

DEFINITIONS

INPUTS

Registers:

- TOTAL_DELAY = Register which is equivalent to CIBETA. TOTAL_DELAY changes during synchronization until the requested CIBETA is obtained.

Bit Flags:

- FUEL_SYNC = Flag which indicates that PIP and fuel are in synch.

- NEW_DELAY = Flag to indicate that new TOTAL_DELAY is being requested by the FUEL SERVICE module.

- SYNFLG = Signature PIP correctly identified flag: 1 -> Signature PIP in correct place; 0 -> not Signature PIP or in wrong place.

Calibration Constants:

- CIBETA = Number of PIPs to delay Bank A from rising edge of Signature PIP.

- PIPOUT = Number of PIP periods between injector outputs on each injector port.

OUTPUTS

Registers:

- NEW_DELAY = See above.
- TOTAL_DELAY = See above.

PROCESS

STRATEGY MODULE: FUEL_INJDLY_COM4

```
SYNFLG = 0 ----------------------|
                                 |
FUEL_SYNC = 0 -------------------|
 (fuel not sync'd)               |
                                 |
SYNFLG = 1 ----------------|     |
                           |     |
FUEL_SYNC = 1 -------------|     |
                           |     |
NEW_DELAY = 1 -------------|     |
 (requesting new delay)    |AND -|OR --| TOTAL_DELAY = 0
                           |     |     | NEW_DELAY = 0
CIBETA > PIPOUT -----------|     |     | Exit FUELPW ROUTINE
 (requested delay too long)      |     |
                                 |     |
SYNFLG = 1 ----------------|     |     |
                           |     |     |
FUEL_SYNC = 1 -------------|     |     |
                           |AND -|     |
NEW_DELAY = 1 -------------|           |
                           |           |
CIBETA < TOTAL_DELAY ------|           |
 (requested delay less than            |
  present delay)                       |
                                       | --- ELSE ---
SYNFLG = 1 ----------------------|     |
                                 |     |
FUEL_SYNC = 1 -------------------|AND -| Exit FUELPW ROUTINE
                                 |     |
NEW_DELAY = 0 -------------------|     |
                                       | --- ELSE ---
                                       |
CIBETA - TOTAL_DELAY > 0.5 ------------| TOTAL_DELAY =
 (requested delay more than 1/2 PIP    |     TOTAL_DELAY + 0.5
  away)                                | NEW_DELAY = 0
                                       | Exit FUELPW ROUTINE
                                       |
                                       | --- ELSE ---
                                       |
                                       | TOTAL_DELAY = CIBETA
                                       | NEW_DELAY = 0
                                       | Exit FUELPW ROUTINE
```

NOTE: TOTAL_DELAY is used by the FUEL SERVICE routine to compute IBETA.

IDLE FUEL MODULATION

OVERVIEW

Idle fuel modulation is used to enhance the idle stability of speed density systems. The strategy achieves A/F modulation by multiplying the FUELPW equation by the parameter ISCMOD, which has a nominal value of 1.0. ISCMOD is inversely proportional to rate of change of rpm. If rpm is increasing, the value of ISCMOD becomes less than 1.0, which leans the A/F and tends to stop the increase. The opposite happens if rpm is decreasing. To prevent the A/F modulation from being excessive, ISCMOD is clipped between maximum and minimum values.

Idle speed modulation is enabled when TP is near RATCH, and rpm is near DSDRPM. The flag REFFLG indicates that idle speed modulation is enabled.

DEFINITIONS

INPUTS

Registers:

- DNDT_ISC = Rate of change of rpm. From the rpm calculation.

- DSDRPM = Desired rpm. Calculated in Idle Speed Control.

- N_BYTE = Byte value of rpm. From the rpm calculation.

- RATCH = Lowest filtered throttle position (see System Equations Chapter).

- TP_REL = Throttle Position relative to RATCH; TP_REL = TP - RATCH.

Bit Flags:

- DISABLE_ISC = Flag set in Running VIP to disable ISCMOD; 1-> disable ISCMOD.

- REFFLG = Idle fuel modulation flag; 1 -> idle fuel modulation enabled.

- RUNUP_FLG = Flag indicating that a stall has occurred; 1 -> Runup rpm exceeded.

- VRUN_ISCFLG = Flag which indicates that idle speed is being controlled by Engine Running VIP; 1 -> in Engine Running VIP, 0 -> not in Engine Running VIP.

Calibration Constants:

- DELRAT = Throttle position adder to RATCH. Used to describe a throttle position below which idle fuel modulation is enabled.

- DLHYST = Hysteresis for DELRAT.

- ISCMOD_MAX = Maximum clip on ISCMOD. Limits the maximum rich excursion that will result from the idle fuel modulation strategy.

- ISCMOD_MIN = Minimum clip on ISCMOD. Limits the maximum lean excursion that will result from the idle fuel modulation strategy.

- ISCMOD_RPM = Incremental adder to DSDRPM; total defines an engine speed below which idle fuel modulation can occur. Should be kept to a minimum to avoid unnecessary activation of the fuel modulation routine.

- KDNDT = Gain term for idle fuel modulation. Larger values result in more fuel being added when engine speed falls and more fuel being taken away when engine speed rises. Too small a value results in unstable idle and too large a value results in unnecessary A/F excursions at idle.

- V_ISCMOD_MAX = VIP maximum clip on ISCMOD when in VIP throttle adjust mode.

- V_ISCMOD_MIN = VIP minimum clip on ISCMOD when in VIP throttle adjust mode.

- V_KDNDT = gain term for idle fuel modulation when in VIP Throttle Adjust mode.

OUTPUTS

Registers:

- EGO_CNT_IDLE = Number of EGO switches which have occurred since entering Idle Fuel Modulation.

- ISCMOD = FUELPW equation multiplier for idle fuel modulation.

- LAM_OLD = Value of LAMBSE at previous EGO switch.

- LAMAVE = Average LAMBSE between EGO switches.

Bit Flags:

- REFFLG = See above.

CALIBRATION INFORMATION

Typical Values:
- DELRAT = 15 counts.
- DLHYST = 10 counts.
- ISCMOD_MAX = 1.2 (allows up to 20_% rich A/F to correct for decreasing rpm).
- ISCMOD_MIN = 0.9 (allows up to 10_% lean A/F to correct for increasing rpm).
- ISCMOD_RPM = 75 to 150 rpm.
- KDNDT = 0.0003 sec/rpm.

PROCESS

STRATEGY MODULE: FUEL_IDLE_MOD_COM3

```
DISABLE_ISC = 1 ---------------------------------| REFFLG = 1
 (freeze ISCDTY for VIP)                         | ISCMOD = 1.0
                                                 |  (disable idle fuel
                                                 |   modulation for
                                                 |   VIP)
                                                 |
                                                 | --- ELSE ---
                                                 |
VRUN_ISCFLG = 1 ---------------------------------| REFFLG = 1
 (throttle adjust mode)                          | ISCMOD = (1 -
                                                 |          V_KDNDT *
                                                 |          DNDT_ISC)
                                                 | Clip ISCMOD between
                                                 |  V_ISCMOD_MIN and
                                                 |  V_ISCMOD_MAX
                                                 |  (enable VIP Idle Fuel
                                                 |   Modulation)
                                                 |
                                                 | --- ELSE ---
TP_REL <= DELRAT + DLHYST -----------------|     |
                                           |     |
RUNUP_FLG = 1 -----------------------------|     |
 (initial runup complete)                  |AND -| REFFLG = 1
                                           |     | ISCMOD = (1 -
TP_REL <= DELRAT --------------|           |     |          KDNDT *
                               |AND -|     |     |          DNDT_ISC)
N_BYTE <= DSDRPM + ISCMOD_RPM -|     |OR --|     | Clip ISCMOD between
                                     |           |  ISCMOD_MIN and
previous value of REFFLG = 1 --------|           |  ISCMOD_MAX
                                                 |  (enable idle fuel
                                                 |   modulation)
                                                 |
                                                 | --- ELSE ---
                                                 |
                                                 | REFFLG = 0
                                                 | ISCMOD = 1.0
                                                 |  (disable idle fuel
                                                 |   modulation)


Previous value of REFFLG = 0 --------------|
                                           |AND -| EGO_CNT_IDLE = 0
Current value of REFFLG = 1 ---------------|


EGO switch --------------------------------------| Increment EGO_CNT_IDLE
```

NOTE:

- REFFLG = 1 indicates idle fuel modulation is enabled.
- ISCMOD is a multiplier on the FUELPW equation.
- DNDT_ISC is a filtered dn/dt (see rpm calculation)
- ISCMOD_RPM = Delta rpm above DSDRPM to enable idle fuel modulation.
- Most MAF systems should not use Idle Fuel Modulation (set KDNDT = 0).

DECEL FUEL SHUT-OFF LOGIC
(Software module DECEL_FUEL_SHUTOFF, called by BG_FUELPW)

OVERVIEW

This logic turns off the fuel on a deceleration condition.

DEFINITIONS

INPUTS

Registers:

- BG_TMR = Background loop time, sec.
- CTTMR = Time at closed throttle, sec.
- DSDRPM = Desired engine speed. See the IDLE SPEED CONTROL Chapter Overview section for definition of the various uses of this register.
- D_TP_DT = Time derivitive of TP (ticks/sec).
- DFSO_A_TMR = Free running, down counting, TP based thermactor air shut off timer.
- DFSO_F_TMR = Free running, down counting, TP based DFSO timer.
- ECT = Engine Coolant Temperature.
- MAP = Manifold Absolute Pressure, " Hg (byte value).
- N = Engine speed, rpm.
- NACTMR = Time since leaving Closed Throttle mode. NACTMR is defined in the TIMERS Chapter.
- NOVS = The ratio of engine speed (NBAR) over vehicle speed (VSBAR).
- OLD_TP_DFSO = Previous value of TP used in DFSO logic.
- TP = Throttle position, counts.
- VSBAR = Time dependent rolling average of instantaneous vehicle speed, VS.

Bit Flags:

- DFSVS_HYS_FG = Decel fuel vehicle speed flip-flop.
- FLG_DFSO_NOVS = Flip flop state flag using transmission gear to engage DFSO.
- FPWQ2 = Fuel pulsewidth flip-flop 2.

- FPWQ3 = Fuel pulsewidth flip-flop 3.

- MFMFLG = MAP failure flag; 1 -> MAP sensor has failed.

- NDSFLG = Flag that indicates Neutral/Drive switch position; 0 -> neutral, 1 -> drive.

- RUNNING = Flag which indicates that idle speed is being controlled by Engine Running VIP; 1 -> in Engine Running VIP, 0 -> not in Engine Running VIP.

- TFMFLG = Flag indicating that the TP sensor is in/out of range; 1 -> failed range check.

- VSFMFLG = Flag indicating the Vehicle Speed Sensor has failed; 1 -> failed sensor.

Calibration Constants:

- CTDSFO = Time at closed throttle for DFSO, sec.

- CTEDSO = Time at extended decel for DFSO, sec.

- DFNOVH = Hysteresis for DENOVS, rpm/mph.

- DFNOVS = NOVS value below which DFSO is permitted. Used to disable DFSO in lower transmission gears to prevent large torque changes.

- DFSECT = Minimum ECT to do DFSO, degrees F.

- DFSMAP = Minimum MAP for DFSO. Should be calibrated to not allow DFSO when the engine is making power so as to minimize torque change at fuel shut off point, Hg.

- DFSO_ECT = Ect above which TP based DFSO is allowed.

- DFSO_OUT = Switch to disable DFSO under a throttle position faliure; 1 -> DFSO disabled.

- DFSMPH = Hysteresis for DFSMAP, Hg.

- DFSRPH = Hysteresis value for DFSRPM.

- DFSRPM = Minimum value of (N - DSDRPM) for DFSO, rpm.

- DFSTM = Time delay before DFSO, sec.

- DFSVS = Minimum vehicle speed to allow Decel Fuel Shut-Off. Used to limit DFSO when in a high gear, low speed decel. Also prevents use of DFSO in parking lots, etc, mph.

- DFSVSH = Hysteresis value for DFSVS, mph.

- D_TP_DT_F = TP tip out rate needed to trigger TP based DFSO (should never be greater that zero). TP tip out can be turned off by setting this scalar to its maximum negative value.

- FND_TP_DT_A = TP rate at which to activate air bypass.
- TRLOAD = Transmission load switch.
- FN222A(n) = The amount of time which thermactor air is shut off, sec.
- FN222F(n) = The amount of time which DFSO is active, sec.

OUTPUTS

Registers:

- D = Decel fuel shut-off multiplier.
- D_TP_DT = See above.
- DFSO_A_TMR = See above.
- DFSO_F_TMR = See above.
- OLD_TP_DFSO = See above.

Bit Flags:

- DFSFLG = Decel Fuel Shut-Off flag; 1 -> the decel fuel multiplier is not one.
- FLG_DFSO_NOVS = See above.
- DFSVS_HYS_FG = See above.
- FPWQ2 = See above.
- FPWQ3 = See above.

PROCESS

STRATEGY MODULE: FUEL_DFSO_COM2

```
always ------------------------------------------| D_TP_DT = (TP -
                                                 |     OLD_TP_DFSO) / BG_TMR
                                                 | OLD_TP_DFSO = TP


ECT > DFSO_ECT ----------------------------|
  (ECT high enough)                        |
                                           |AND -| [dfso_condition] = 1
                                           |     |
TFMFLG = 0 --------------------------------|     |
 (TP sensor OK)                                  |
                                                 | --- ELSE ---
                                                 |
                                                 | [dfso_condition] = 0


[dfso_condition] = 1 ----------------------|
 (condition right)                         |
                                           |
DFSO_F_TMR = 0 ----------------------------|AND -| DFSO_F_TMR = FN222F(N)
 (initialize once)                         |     |
                                           |     |
D_TP_DT < D_TP_DT_F -----------------------|     |
 (tipping out)                                   |
                                                 | --- ELSE ---
                                                 |
                                                 | decrement DFSO_F_TMR
                                                 | (clip at zero)


[dfso_condition] = 1 ----------------------|
 (condition right)                         |
                                           |
DFSO_A_TMR = 0 ----------------------------|AND -| DFSO_A_TMR = FN222A(N)
 (initialize once)                         |     |
                                           |     |
D_TP_DT < FND_TP_DT_A(N) ------------------|     |
 (tipping out)                                   |
                                                 | --- ELSE ---
                                                 |
                                                 | decrement DFSO_A_TMR
                                                 | (clip at zero)
```

Evaluate each flip flop each background pass:

```
NACTMR > DFSTM ----------------------------|S Q -| FPWQ3
 (long tip-in)                             |
                                           |
NACTMR > 0 --------------------------------|C


VSBAR >= DFSVS ----------------------------|S Q -| DFSVS_HYS_FG
                                           |
VSBAR < DFSVS - DFSVSH --------------------|C


NOVS <= DFNOVS ----------------------------|S Q -| FLG_DFSO_NOVS
 (in higher gear)                          |
                                           |
NOVS > DFNOVS + DFNOVH --------------------|C


N - DSDRPM >= DFSRPM ----------------|
 (rpm high enough)                   |
                                     |AND -|S Q -| FPWQ2
MFMFLG = 1 --------------------|     |     |
                               |OR --|     |
MAP <= DFSMAP -----------------|           |
 (not making power)                        |
                                           |
N - DSDRPM < DFSRPM - DFSRPH --------|     |
                                     |OR --|C
MAP > DFSMAP + DFSMPH ---------|     |
                               |AND -|
MFMFLG = 0 --------------------|
```

```
FPWQ2 = 1 ---------------------------------|
                                           |
RUNNING = 0 -------------------------------|
 (not in KOER VIP)                         |
                                           |
TRLOAD < 2 --------------------------|     |
 (ignore N/D input)                  |OR --|
                                     |     |
NDSFLG = 1 --------------------------|     |
 (in gear)                                 |
                                           |
ECT > DFSECT ------------------------------|
 (engine warm enough)                      |
                                           |
TFMFLG = 0 --------------------|           |
                               |AND -|     |
DFSO_OUT = 0 ------------------|     |     |
                                     |     |
MFMFLG = 0 --------------------|     |     |
                               |AND -|OR --|AND -| "A" = 1
DFSO_OUT = 0 ------------------|     |     |     |
                                     |     |     |
TFMFLG = 0 --------------------|     |     |     |
                               |AND -|     |     |
MFMFLG = 0 --------------------|           |     |
                                           |     |
TFMFLG = 1 --------------------------|     |     |
                                     |OR --|     |
CTTMR > CTDSFO ----------------|     |     |     |
 (closed throttle long enough) |     |     |     |
                               |AND -|     |     |
CTTMR > CTEDSO ----------|     |           |     |
 (extended closed throt.)|     |           |     |
                         |OR --|           |     |
FPWQ3 = 1 ---------|     |                 |     |
                   |AND -|                 |     |
FLG_DFSO_NOVS = 1 -|                       |     |
                                           |     |
VSFMFLG = 1 -------------------------|     |     |
 (VS sensor bad)                     |OR --|     |
                                     |           |
DFSVS_HYS_FG = 1 --------------------|           |
                                                 | --- ELSE ---
                                                 |
                                                 | "A" = 0


DFSO_F_TMR <> 0 ---------------------------|
                                           |OR --| FUEL OFF
"A" = 1 -----------------------------------|     | D = 0
                                                 | DFSFLG = 1
                                                 |
                                                 | --- ELSE ---
                                                 |
                                                 | FUEL ON
                                                 | D = 1
                                                 | DFSFLG = 0
```

FUEL PULSEWIDTH CALCULATION (BGFUEL)
(Background Module BG_FUELPW)

OVERVIEW

Except for crank mode and Asynchronous Acceleration Enrichment, fuel pulsewidths are calculated from the following equation:

$$BGFUEL = FUEL\_A * \left[\frac{AM * KAMREF}{14.64 * LAMBSE} + EFTRFF \right] * \frac{1}{N * ENGCYL * A0COR} * \frac{PIPOUT}{INJOUT * NUMOUT} + FN367$$

DEFINITIONS

INPUTS

Registers:

- A0COR = Corrected fuel injector slope.
- ACT = Air Charge Temperature.
- AM = Air Mass flow as defined in the SYSTEM EQUATIONS Chapter, lb/min.
- BASE_EM = Fuel requirement based on EGR flow (non-displayable).
- BASEFF = Fuel amount to provide stoichiometric operation based on inducted air mass (AM) = (KAMREF * AM)/(14.64 * LAMBSE).
- CRKPIP_CTR = Foreground PIP counter for crank fuel.
- CRKPIP_CTR_BG = Background equivalent of CRKPIP_CTR.
- D = Decel fuel shutoff multiplier.
- ECT = Engine Coolant Temperature.
- EFTRFF = Equilibrium fuel transfer rate for transient fuel compensation (lbf/min).
- EM = Actual EGR mass flow.
- FUEL_A = Fuel pulsewidth multiplier for Idle Fuel Modulation / Decel / Underspeed operation, unitless (non-displayable).
- FUELPW = Foreground/background calculated fuel pulsewidth.
- ISCMOD = Multiplier for idle speed fuel.

- KAMREF = Adaptive fuel strategy correction factor.
- LAMBSE = Desired ratiometric air/fuel ratio. Normalized to stoichiometry (14.7 a/f).
- MAP = Manifold Absolute Pressure, "Hg (byte value).
- MAPWBG = MAP_WORD updated once per background pass at calculation of AMPEM. Used in fuel pulsewidth calculation.
- N = Engine speed in revolutions per minute.
- OFMFLG = ETV overcurrent monitor failure flag; 0 -> ETV O.K., 1 -> ETV failure mode.
- PWCF = Pulsewidth Conversion Factor - converts total computed for engine into amount per injector = [1/(N * ENGCYL)]/[PIPOUT/(NUMOUT * INJOUT)].
- RT_NOVS_KAM = Ratio of actual N-OVER-V to base N-OVER-V.
- TLS_NV_FLG = Torque limiting strategy - no fuel flag; 0 -> normal fuel, 1 -> no fuel.
- TLS_24_FLG = Torque limiting strategy - 1/2 fuel flag; 0 -> normal fuel, 1 -> 1/2 fuel.
- TLS_34_FLG = Torque limiting strategy - 3/4 fuel flag; 0 -> normal fuel, 1 -> 3/4 fuel.
- TP_REL = TP - RATCH.
- VBAT = Battery voltage.
- VSBAR = Time dependent rolling average of instantaneous vehicle speed, VS.
- VSBART_FM = VS calculated based on NIBART, NOBART, or NOBART.
- VSFMFLG = Vehicle speed sensor FMEM flag.

Bit Flags:

- ALT_CAL_FLG = Flag to indicate use of alternate calibration.
- CRKFLG = Flag indicating engine mode; 1 -> cranking, 0 -> run or underspeed mode.
- DSFFLG = Decel Fuel Shutoff flag; 1 -> the decel fuel multiplier is not one.
- MFMFLG = MAP sensor failure flag.
- REFFLG = Idle air flow region flag; 1 -> in region.
- TFMFLG = TP sensor failure flag; 1 -> TP failure.
- UNDSP = Run/Underspeed Engine Mode flag; 1 -> Underspeed/Crank, 0 -> Run mode.

Calibration Constants:

- BFULSW = Calibration switch to force use of background calculation of fuel pulsewidth; 1 -> do all fuel calculations in background, 0 -> use foreground fuel flag logic.

- ENGCYL = Number of cylinders per engine revolution.

- FN348(ECT) = Crank fuel pulsewidth as a function of ECT.

- FN367(VBAT) = Injector offset as a function of VBAT.

- FN387(ECT) = Fuel pulsewidth multiplier as a function of ECT.

- FN387_ALT = Alternative FN387.

- FN1350(CRKPIP_CTR_BG,ACT) = Cranking fuel pulsewidth multiplier as a function of number of PIPs in crank and air charge temperature.

- FREQ18 = Seconds to clock ticks conversion factor. 1 clock tick = 3 * 10E-6 seconds.

- INJOUT = Number of injectors fired by each output port.

- MINPW = Minimum pulsewidth for repeatable fuel delivery.

- NLMT = Overspeed RPM.

- NLMTH = Hysteresis for overspeed RPM.

- NUMOUT = Number of injector output ports.

- OUTINJ = Injector scheme selection switch; 1 -> alternate injections, 2 -> simultaneous injections.

- PIPOUT = Number of PIP periods between injector outputs on each injector port.

- TLSNV = Torque limiting pattern for engine RPM/vehicle speed, unitless.

- TLS24D = Torque limiting pattern for 1/2 fuel, double fire, unitless.

- TLS34D = Torque limiting pattern for 3/4 fuel, double fire, unitless.

- TLS24S = Torque limiting pattern for 1/2 fuel - single fire.

- TLS34S = Torque limiting pattern for 3/4 fuel - single fire.

- TP_DECHOKE = TP value above which to de-choke.

- TQMAX1 = Maximum torque before 3/4 fuel.

- TQMAX2 = Maximum torque before 1/2 fuel.

- TQMAXH = Hysteresis for TQMAX1.

- VSLIM = Maximum vehicle speed, MPH.

- VSLIMH = Hysteresis for maximum vehicle speed, MPH.

OUTPUTS

Registers:

- BASE_EM = See above.
- BASEFF = See above.
- BGFUEL = Background fuel pulsewidth, clock ticks.
- FFULC = The constant that is added in the foreground fuel pulsewidth equation, lb/cyl.
- FFULM = The value that is multiplied by MAP_WORD in the foreground fuel pulsewidth equation (lb/cyl - "Hg).
- FUEL_A = See above.
- FUEL_PIPS = Number of PIPs between injections.
- LAMBSE = See above.
- TLS_NV_FLG = See above.
- PWCF = See above.
- TLS_24_FLG = See above.
- TLS_34_FLG = See above.
- TLSPAT = Torque limiting strategy injection pattern.

Bit Flags:

- DISABLE_ADAPT = Adaptive fuel disable flag; 1 -> disable adaptive fuel.
- FFULFG = Foreground fuel flag; 1 -> Compute fuel pulsewidth in foreground using latest computed manifold absolute pressure, 0 -> otherwise use background fuel pulsewidth.

PROCESS

STRATEGY MODULE: FUEL_BG_PW_DET_COM5

CRANK MODE FUEL PULSEWIDTH

```
ALT_CAL_FLG = 1 ---------------------| fn387 = FN387_ALT(ECT)
                                     |
                                     | --- ELSE ---
                                     |
                                     | fn387 = FN387(ECT)


CRKFLG = 1 --------------------|
                               |AND -| BGFUEL = FN348(ECT) *
TP_REL <= TP_DECHOKE ----------|     |          FN1350(CRKPIP_CTR_BG,ACT) *
                               |     |          (MAP/29.875)
TFMFLG = 0 --------------------|     |  (clip to 0.249 sec as a max)
 (tp sensor o.k.)                    | Goto "CONVERT PULSEWIDTH TO CLOCK
                                     |       TICKS"
                                     | Exit FUELPW Routine
                                     |
                                     | --- ELSE ---
                                     |
CRKFLG = 1 --------------------------| BGFUEL = 0
                                     |  (WOT de-choke mode)
                                     | Goto "CONVERT PULSEWIDTH TO CLOCK
                                     |       TICKS"
                                     | Exit FUELPW Routine
```

CALCULATE KAMREF

```
always ------------------------------| Do "KAMREF" Module
```

CALCULATE TRANSIENT FUEL

```
always ------------------------------| Do "TFCOMP" Module
```

DETERMINE BASE FUEL FLOW (BASEFF)

BASEFF is used in the foreground fuel calculation and contains base fuel flow, unadjusted for transient fuel, AE fuel, or injector hardware.

$$BASEFF = \frac{AM * KAMREF}{14.64 * LAMBSE}$$

DETERMINE BASE EM (BASE_EM)

BASE_EM is used in the foreground fuel calculation to adjust the base fuel for EGR mass that may be present at the time.

$$BASE\_EM = \frac{EM * KAMREF}{14.64 * LAMBSE}$$

CALCULATE PULSEWIDTH CONVERSION FACTOR (PWCF)

PWCF converts total fuel required for the engine into an amount required per injection.

PWCF = [1/(N * ENGCYL * A0COR)] * [PIPOUT/(INJOUT * NUMOUT)]

CALCULATE IDLE FUEL MODULATION/DECEL/UNDERSPEED MULTIPLIER (FUEL_A)

```
always ------------------------------| Do "DECEL FUEL SHUTOFF" Logic
                                     |  (determine the value of D)
                                     | Do "IDLE FUEL MODULATION" Logic
                                     |  (determine the value of "ISCMOD")


UNDSP = 1 ---------------------------| FUEL_A = D * ISCMOD * fn387(ECT)
                                     |
                                     | --- ELSE ---
                                     |
                                     | FUEL_A = D * ISCMOD
```

APPLY MINIMUM FUELPW AND LAMBSE CLIPS

```
DFSFLG = 1 --------------------------| DISABLE_ADAPT = 1
 (decel fuel shutoff)                |  (disable adaptive fuel)
                                     |
                                     | --- ELSE ---
                                     |
FUELPW < MINPW ----------------------| DISABLE_ADAPT = 1
                                     |  (do not learn adaptive fuel
                                     |   at minimum clip)
                                     | BGFUEL = MINPW
                                     | LAMBSE = 1.0
                                     |
                                     | --- ELSE ---
                                     |
                                     | DISABLE_ADAPT = 0
```

ENGINE AND VEHICLE SPEED LIMITER LOGIC

```
VSFMFLG = 1 -------------------------| vsbar = VSBART_FM
                                     |
                                     | --- ELSE ---
                                     |
                                     | vsbar = VSBAR


N > NLMT ----------------|S  Q-|
                         |     |
N <= NLMT - NLMTH -------|C    |OR --| TLS_NV_FLG = 1
                               |     |
vsbar > VSLIM /                |     |
       RT_NOVS_KAM ------|S  Q-|     |
                         |           |
vsbar <= (VSLIM /        |           |
    RT_NOVS_KAM) -       |           |
           VSLIMH -------|C          |
                                     | --- ELSE ---
                                     |
                                     | TLS_NV_FLG = 0
```

NOTE: This strategy does not limit based on vehicle speed. It limits based on propshaft speed as a result of dividing VSLIM by RT_NOVS_KAM, and assumes that with a change in axle ratio, the equivalent change is made to the VSS drive gear.

ENGINE TORQUE LIMITING STRATEGY

```
OFMFLG = 0 --------------------|
 (TV solenoid OK)              |       (Full Fuel)
                               |OR --| TLS_24_FLG = 0
TQ_OFM < TQMAX1 -              |     | TLS_34_FLG = 0
          TQMAXH --------|S  Q-|     |
                         |           |
TQ_OFM >= TQMAX1 --------|C          |
                                     | --- ELSE ---
                                     |
                                     |   (1/2 Fuel)
TQ_OFM >= TQMAX2 --------------|S  Q-| TLS_24_FLG = 1
                               |     | TLS_34_FLG = 0
TQ_OFM < TQMAX2 - TQMAXH ------|C    |
                                     | --- ELSE ---
                                     |
                                     |   (3/4 Fuel)
                                     | TLS_24_FLG = 0
                                     | TLS_34_FLG = 1
```

```
TLS_NV_FLG = 1 ----------------------| TLSPAT = TLSNV
                                     |
                                     | --- ELSE ---
PIPOUT/ENGCYL = 2 -------------|     |
 (single fire)                 |     |
                               |AND -| TLSPAT = TLS24S
TLS_24_FLG = 1 ----------------|     |
                                     | --- ELSE ---
PIPOUT/ENGCYL = 2 -------------|     |
                               |AND -| TLSPAT = TLS34S
TLS_34_FLG = 1 ----------------|     |
                                     | --- ELSE ---
PIPOUT/ENGCYL = 1 -------------|     |
 (double fire)                 |     |
                               |AND -| TLSPAT = TLS24D
TLS_24_FLG = 1 ----------------|     |
                                     | --- ELSE ---
PIPOUT/ENGCYL = 1 -------------|     |
                               |AND -| TLSPAT = TLS34D
TLS_34_FLG = 1 ----------------|     |
                                     | --- ELSE ---
                                     |
                                     | TLSPAT = 65535
```

CONVERT PULSEWIDTH TO CLOCK TICKS
(can be entered from CRANK MODE FUEL PULSEWIDTH)

```
always ------------------------------| BGFUEL = [FUEL_A * (BASEFF +
                                     |          EFTRFF) * PWCF] * FREQ18
                                     |  (FREQ18 = 1/3 * 10E-6 ticks per sec)
```

CALCULATE NUMBER OF PIPS BETWEEN INJECTIONS (FUEL_PIPS)
(used in FUEL SERVICE FOREGROUND routine)

```
                                                 PIPOUT * OUTINJ
always ------------------------------| FUEL_PIPS = ---------------
                                                     NUMOUT
```

DETERMINE STATE OF FOREGROUND FUEL FLAG (FFULFG)

```
BFULSW = 1 --------------------|
                               |
MFMFLG = 1 --------------------|
                               |OR --| FFULFG = 0
CRKFLG = 1 --------------------|     |  (do background fuel)
                               |     |
REFFLG = 1 --------------------|     |
                                     | --- ELSE ---
                                     |
                                     | FFULFG = 1
                                     |  (do foreground fuel)
```

CALCULATE FOREGROUND PARTS OF THE FUEL EQUATION (FFULM, FFULC)

```
always ------------------------------| FFULM = FUEL_A * [(BASEFF +
                                     |          BASE_EM) / MAPWBG] *
                                     |          PWCF
                                     | FFULC = FUEL_A * (EFTRFF -
                                     |          BASE_EM) * PWCF
```

CALCULATE INJECTOR DELAY (CIBETA)

```
always ------------------------------| Do "INJECTOR DELAY" Logic
```

VBAT INJECTOR OFFSET

FN367(VBAT) is compensation for low battery voltage. It is added to the fuel pulsewidth in foreground as the pulse is sent out. The displayed pulsewidth does not include FN367.

FUEL PUMP CONTROL LOGIC

OVERVIEW

EFI vehicles are equipped with an electric fuel pump controlled by the computer via a relay. The fuel pump relay is energized according to the logic below.

DEFINITIONS

INPUTS

Registers:

- TSLPIP = Timer indicating time since last PIP low-to-high transition, sec.

OUTPUTS

Registers:

- PUMP = Bit flag indicating fuel pump mode; 0 -> fuel pump disabled, 1 -> fuel pump enabled.

PROCESS

STRATEGY MODULE: FUEL_PUMP_COM1

```
TSLPIP < 1 SECOND -----------------| PUMP = 1
                                   |
                                   | --- ELSE ---
                                   |
                                   | PUMP = 0
```

PPCTR CONTROL
(Updated during PIP_DATA foreground routine)

OVERVIEW

This counter counts PIPs when not in Decel Fuel Shut-Off. PPCTR is updated at PIP rising edge before injector pulsewidth is calculated and output.

DEFINITIONS

INPUTS

Bit Flags:

- DFSFLG = Decel Fuel Shut-Off flag; 1 -> the decel fuel multiplier is not one.

Calibration Constants:

- PIPNUM = Number of PIPs to remain in Open Loop fuel after DFSO. Prevents LAMBSE from ramping off rich due to normal transport delay time. Set to 1 to calibrate out.

OUTPUTS

Registers:

- PPCTR = PIP counter; updated at PIP rising edge before injector pulsewidth is calculated and output.

PROCESS

STRATEGY MODULE: FUEL_PPCTR_COM2

```
DFSFLG = 0 -----------------------------------| Increment PPCTR every PIP
                                              | Clip at PIPNUM
                                              |
                                              | --- ELSE ---
                                              |
                                              | PPCTR = 1
```

FUEL SERVICE ROUTINE
(foreground module FUEL_SERVICE executed on PIP falling edge in CRANK and PIP rising edge in UNDSP & RUN)

OVERVIEW

The purpose of the Fuel Service module is to issue the on-off signals to the injectors to attain the desired amount of fuel (FUELPW) at the desired time in the engine cycle (TOTAL_DELAY). This module also updates the background calculated fuel pulsewidth to the most current MAP if so desired (Foreground Fuel).

A software switch, OUTINJ, can be calibrated to yield the desired injection scheme:

```
OUTINJ = 1 -----------------------------------| ALTERNATE INJECTIONS
                                              |
                                              | --- ELSE ---
                                              |
OUTINJ = 2 -----------------------------------| SIMULTANEOUS INJECTIONS
```

DEFINITIONS

INPUTS

Registers:

- BGFUEL = Background fuel pulsewidth, clock ticks.
- CRKPIP = Number of PIPs between injector firing.
- ENGCYL = Number of PIPs (or injections) per revolution.
- FFULM = The valve which is multiplied by MAP_WORD in the foreground fuel pulsewidth equation, lb/cyl -- " Hg.
- FUEL_PIPS = Number of PIPs which have occurred between injections.
- FUEL_SUM_TKS = Register for DOL summer, ticks.
- FUELPW = Fuel pulsewidth, displayed in clock ticks.
- IBETA = Fractional part of total injector delay, PIPs.
- INJ_PIP_CNT = Counter which counts the number of PIPs between injections.
- INJCNT = Injector portion of total delay.
- MAP_WORD = Manifold absolute pressure, " Hg.
- N = Engine rpm.
- NUMOUT = Number of injector output ports.
- OUTINJ = Injector scheme selection switch; 1 -> alternate injections, 2 -> simultaneous injections.
- SYNCTR = Counter which counts PIP signals until its value equal NUMCYL; always initialized to zero.
- TLSCTR = Torque limiting strategy injection counter.
- TLSPAT = Torque limiting strategy injection pattern.

Bit Flags:

- CRKFLG = Flag indicating engine mode; 1 -> crank mode, 0 -> run or underspeed.
- DSFFLG = Decel fuel shutoff flag; 1 -> the decel fuel multiplier is not one.
- FFULFG = Foreground fuel flag; 1 -> compute fuel pulsewidth in foreground using latest computed manifold absolute pressure, 0 -> otherwise use background fuel pulsewidth.
- FUEL_FINISHED = Flag indicating status of fuel calculations; 1 -> fuel calculations are complete.
- INJ_BANK = Flag indicating which injector bank is being energized; 0 -> Bank A, 1 -> Bank B.
- NO_SYNC = 1 -> Fuel injectors are not sychronized with the Signature PIP.
- PIPOUT = Number of PIP periods betwen injector outputs on each injector port.
- RUNNING = Engine running VIP enable flag.
- STALL = Flag which indicates Run to Crank transition.
- SYNFLG = Signature PIP correctly identified flag; 1 -> Signature PIP in correct place, 0 -> not Signature PIP or in wrong place.
- TLS_24_FLG = Torque limiting strategy - 1/2 fuel flag; 0 -> normal fuel, 1 -> 1/2 fuel.
- TLS_34_FLG = Torque limiting strategy - 3/4 fuel flag; 0 -> normal fuel, 1 -> 3/4 fuel.
- TLS_NV_FLG = Engine RPM/Vehicle speed limiting flag.
- UNDSP = Engine mode flag; 1 -> cranking or underspeed, 0 -> run.

Calibration Constants:

- DT12S = The value, in clock ticks, of the current PIP period.
- FFULC = The constant which is added in the foreground fuel pulsewidth equation, lb/cyl.
- FREQ18 = Seconds to clock ticks conversion factor. 1 clock tick = 3 * 10E-6 seconds.
- MINPW = Minimum pulsewidth for repeatable fuel delivery.
- STALLN = Stall rpm: If the first rpm calculated is greater than this value, then assume a reinit occurred, rpm.

OUTPUTS

Registers:

- FUELPW = See above.
- FUEL_SUM_TKS = See above.
- IBETA = See above.
- INJ_PIP_CNT = See above.
- INJCNT = See above.
- SYNCTR = See above.
- TLS_SHFTR = Foreground scratch register for injection pattern.
- TLSCTR = See above.

Bit Flags:

- FUEL_FINISHED = See above.
- FUEL_SYNC = See above.
- INJ1_PIP = Injector number one PIP occurred, if set to one.
- INJ2_PIP = Injector number two PIP occurred, if set to one.
- NO_SYNC = See above.
- STALL = See above.
- SYNFLG = See above.

PROCESS

STRATEGY MODULE: FUEL_INJ_OUT_COM6

FUEL SERVICE START

```
STALL = 1 ----------------------------| STALL = 0
 (N < STALLN)                         | SYNFLG = 0
                                      | SYNCTR = 0
                                      | FUEL_SYNC = 0
                                      | NO_SYNC = 0
                                      | INJ_PIP_CNT = 1
```

Divide TOTAL_DELAY into an integer portion, INJCNT and a fraction portion, IBETA.

```
TOTAL_DELAY = 0 ----------------------| IBETA = 0
                                      | INJCNT = 0
                                      | SYNCH_VALUE = 0
                                      | Do: "FUEL FIRING PIP"
                                      |
                                      | --- ELSE ---
                                      |
                                      | IBETA = TOTAL_DELAY - INJCNT


IBETA < 0 ----------------------------| IBETA = 0
                                      | INJCNT = 0
                                      | SYNCH_VALUE = 0
                                      | Do: "Fuel Firing PIP"
                                      |
                                      | --- ELSE ---
                                      |
IBETA >= 1.0 -------------------------| IBETA = IBETA - 1
                                      | INJCNT = INJCNT + 1
                                      | INJ_PIP_CNT = INJ_PIP_CNT + 1
                                      |  (number of PIPs between injections)


always -------------------------------| SYNCH_VALUE = (2 * ENGCYL) - INJCNT


SYNCH_VALUE <= 0 ---------------------| IBETA = 0
                                      | INJCNT = 0
                                      | SYNCH_VALUE = 0
                                      | Do: "FUEL FIRING PIP"


SYNCH_VALUE = 2 * ENGCYL ------------| SYNCH_VALUE = 0
```

FUEL FIRING PIP

Check to see if this is the correct PIP for firing fuel. SYNCH_VALUE is the integer portion of TOTAL_DELAY.

```
SYNFLG = 1 --------------------|
 (PIP in synch)                |
                               |
SYNCTR = SYNCH_VALUE ----------|
 (proper PIP for fuel firing)  |
                               |AND -| FUEL_SYNC = 1
INJ_BANK = 0 ------------------|     |  (PIP is synched with fuel)
 (signature PIP must contain   |     |
  bank A)                      |     |
                               |     |
INJ_PIP_CNT = 1 ---------------|     |
                                     | --- ELSE ---
SYNFLG = 1 --------------------|     |
                               |AND -| NO_SYNC = 1
SYNCTR = SYNCH_VALUE ----------|     |  (PIP is synched but fuel is on
                                     |   wrong bank)


always ------------------------------| Decrement INJ_PIP_CNT


INJ_PIP_CNT > 0 ---------------------| EXIT "FUEL SERVICE" routine
                                     |  (no fuel to be output this PIP)
```

FOREGROUND FUEL DETERMINATION

```
FFULFG = 1 --------------------------| FUELPW = (FFULM * MAP_WORD) + FFULC
                                     |  (use foreground fuel calculation)
                                     |
                                     | --- ELSE ---
                                     |
FFULFG = 0 --------------------------| FUELPW = BGFUEL
                                     |  (use background fuel calculation)
```

MINIMUM PULSEWIDTH CLIP

```
DSFFLG = 0 --------------------|
 (not in DFSO)                 |
                               |
RUNNING = 0 -------------------|
 (not in VIP)                  |AND -| FUELPW = MINPW * FREQ18
                               |     |  (convert to clock ticks)
FUELPW < MINPW ----------------|     |
 (less than minimum pulse)     |     |
                               |     |
FFULFG = 1 --------------------|     |
                                     | --- ELSE ---
FUELPW >= 0.250 ---------------|     |
                               |AND -| FUELPW = .25 * FREQ18
FFULFG = 1 --------------------|     |
                                     | --- ELSE ---
                                     |
FFULFG = 1 --------------------------| FUELPW = FUELPW * FREQ18
                                     |  (FREQ18 = 1/3 * 10E-6 ticks per sec)


always ------------------------------| INJ_PIP_CNT = FUEL_PIPS
                                     |  (FUEL_PIPS = [PIPOUT * OUTINJ]
                                     |   /NUMOUT)
                                     | FUEL_SUM_TKS = FUEL_SUM_TKS + FUELPW


CRKFLG = 1 --------------------------| INJ_PIP_CNT = CRKPIP
 (crank mode)                        |  (fire every CRKPIP PIPs in
                                     |   crank mode)


UNDSP = 0 ---------------------|
 (run mode)                    |AND -| Decrement INJ_PIP_CNT
                               |     |  (walk injector bank B
NO_SYNC = 1 -------------------|     |   back 1 PIP at a time)
 (fuel not synched)
```

```
INJ_PIP_CNT <= 0 --------------------| INJ_PIP_CNT = 1
                                     | FUEL_FINISHED = 0
                                     |
                                     | --- ELSE ---
                                     |
                                     | FUEL_FINISHED = 0


OUTINJ = 2 --------------------------| INJ_BANK = 0
 (simultaneous injections)           |  (always start with bank A)


always ------------------------------| NEW_DELAY = 1
                                     |  (request new IBETA)
```

FUEL_REPEAT

This section sets up the fuel output edge, if required.

```
TLS_NV_FLG = 0 ----------------|
                               |
TLS_24_FLG = 0 ----------------|AND -| TLSCTR = 1
                               |     | Do: "SET UP INJECTOR OUTPUT EDGE"
TLS_34_FLG = 0 ----------------|     |
                                     | --- ELSE ---
                                     |
                                     | Decrement TLSCTR
                                     |  (check next bit in pattern)


TLSCTR = 0 --------------------------| TLSCTR = 16
 (pattern completed)                 | TLS_SHFTR = TLSPAT


Always ------------------------------| Shift TLS_SHFTR 1 bit to the left


carry bit = 1 -----------------------| Do: "SET UP INJECTOR OUTPUT EDGE"
 (OK to output fuel this PIP)        |
                                     | --- ELSE ---
                                     |
                                     | SKIP "SET UP INJECTOR OUTPUT EDGE"
                                     |  but continue through FUEL_REPEAT
```

SET UP INJECTOR OUTPUT EDGE

```
UNDSP = 1 ---------------------------| injector edge = (Time of last PIP
                                     |                  rising edge)
                                     | Set immediate output request
                                     |
                                     | --- ELSE ---
                                     |
                                     | injector edge = (Time of last PIP
                                     |                  rising edge) +
                                     |                  (DT12S * IBETA)


FUELPW <> 0 -------------------|
                               |AND -| INJ1_PIP = 0
INJ_BANK = 0 ------------------|     |  (injection A occurred)
                                     |
                                     | --- ELSE ---
                                     |
                                     | INJ2_PIP = 0
                                     |  (injection B occurred)


NUMOUT = 2 --------------------------| Toggle INJ_BANK
 (2 banks)                           |  (do next bank)
```

```
NUMOUT = 2 --------------------|
 (2 outputs)                   |
                               |
FUEL_FINISHED = 0 -------------|AND -| FUEL_FINISHED = 1
                               |     | NO_SYNC = 0
CRKFLG = 1 --------------|     |     | Do: "FUEL_REPEAT"
                         |     |     |
OUTINJ = 2 --------------|OR --|     |
                         |           |
PIPOUT = 2 --------|     |           |
                   |AND -|           |
NO_SYNC = 1 -------|                 |
                                     | --- ELSE ---
                                     |
                                     | NO_SYNC = 0
                                     | EXIT "FUEL SERVICE" routine
```

INJECTOR TIMING ROUTINE
(Done during PIP_DATA foreground routine)

OVERVIEW

CRANK or UNDERSPEED mode (UNDSP = 1) OR MHPFD = 0.99

The injector synchronization routine is disabled and SYNFLG is cleared.

RUN mode (UNDSP = 0) AND MHPFD < 0.99

The injector synchronization routine is enabled. The objective of the routine is to identify the cylinder #1 PIP and to alter the injector timing schedule so that Injector Output Port "A", which fires Injector #1, is synchronized with the cylinder #1 PIP.

NOTE: THE USER MUST SET UP THE TWO INJECTOR OUTPUT PORTS SUCH THAT THE CYLINDER #1 INJECTOR IS FIRED BY INJECTOR OUTPUT PORT "A".

A Signature PIP distributor must be used in order to achieve the identification of the cylinder #1 PIP. Refer to the Spark Section for a description of the Signature PIP Distributor.

When a new PIP down-edge-interrupt is received, HFDLTA (the elapsed time since the last up-edge-interrupt was seen) is calculated.

Next, the fractional difference between HFDLTA and the previous up-edge to down-edge elapsed time, PHFDLT, is calculated and compared to a critical value, MHPFD, as shown below:

(PHFDLT - HFDLTA) / PHFDLT > MHPFD ?

MHPFD is a calibration constant which is dependent only upon number of cylinders and the value of the Signature PIP duty cycle. The user must calibrate MHPFD to the appropriate value as shown below:

- If 8-cyl & Signature PIP duty cycle <= 35%, then set MHPFD = .20

- If 6-cyl & Signature PIP duty cycle <= 30%, then set MHPFD = .24

- If 4-cyl & Signature PIP duty cycle <= 30%, then set MHPFD = .29

NOTE: IF A SIGNATURE PIP DISTRIBUTOR IS NOT PRESENT, THEN SET MHPFD = .99

When the above comparison is true, then the current PIP is the Signature PIP. If Injector Output Port "A", which fires Injector #1, is timed from the up-edge of the Signature PIP, then the system is synchronized.

If the system is determined to be unsynchronized, then Injector Output Port "B" is fired one PIP earlier than normal. This causes the injector firing schedule to be shifted one PIP per revolution until synchronization is achieved.

A decrementing counter, SYNCTR, that starts at the number of cylinders and counts down to 0, 1 count per PIP, is used to predict when the Signature PIP should be seen again after it is first identified.

Each time that SYNCTR = 0 and the above comparison is true, a synchronization flag, SYNFLG, is set to 1 and SYNCTR is reset to the number of cylinders.

If the above comparison is ever false when SYNCTR = 0, then SYNFLG is set to 0, and the entire synchronization routine of first finding the Signature PIP and then "Stepping" the injection firing schedule to it is repeated.

DEFINITIONS

INPUTS

Registers:

- ENGCYL = Number of PIPs (or injections) per revolution.
- HFDLTA = Most recent PIP first half period.
- MHPFD = Signature PIP difference check value.
- MKAY = Half period multiplier to correct for average error caused by Hall effect sensor in distributor and armature.
- PHFDLT = Previous PIP first half period.
- SIGKAL = Signature PIP half period multiplier - initial value
  = 1.66666 for 30% duty cycle signature PIP
  = 1.42857 for 35% duty cycle signature PIP.
- SYNCTR = Counter which counts PIP signals until its value is equal to NUMCYL (number of cylinders). SYNCTR is initialized to 0.

Bit Flags:

- SIGPIP = A flag that indicates that signature PIP half period has been identified; 1 -> signature PIP, 0 -> not signature PIP.
- UNDSP = Flag indicating engine mode; 1 -> Cranking or Underspeed, 0 -> Run mode.

OUTPUTS

Registers:

- HFDLTA = See above.
- SYNCTR = See above.

Bit Flags:

- SIGPIP = See above.
- SYNFLG = Signature PIP correctly identified flag; 1 -> Signature PIP in correct place, 0 -> not Signature PIP or in wrong place.

PROCESS

STRATEGY MODULE: FUEL_INJ_TIM_COM2

```
(PHFDLT - HFDLTA) / PHFDLT > MHPFD -| SIGPIP = 1
                                    |  (this is signature PIP)
                                    |
                                    | --- ELSE ---
                                    |
                                    | SIGPIP = 0
                                    |  (not signature PIP)


SYNCTR <> 0 ------------------------| EXIT
 (not cylinder #1)                  |
                                    | --- ELSE ---
                                    |
UNDSP = 1 --------------------------| SYNCTR = 2 * ENGCYL
 (not RUN mode)                     | EXIT
                                    |  (do not attempt to
                                    |   synchronize fuel in
                                    |   CRANK or UNDERSPEED)
                                    |
                                    | --- ELSE ---
SIGPIP = 1 -------------------|     |
 (signature PIP)              |AND -| SYNFLG = 1
                              |     |  (in synch OK to
SYNCTR = 0 -------------------|     |   synchronize fuel)
 (cylinder #1)                      | HFDLTA = (HFDLTA *
                                    |           SIGKAL) / MKAY
                                    |  (correct signature PIP to
                                    |   50% duty cycle)
                                    | SYNCTR = 2 * ENGCYL
                                    | EXIT
                                    |
                                    | --- ELSE ---
                                    |
                                    | SYNFLG = 0
                                    |  (not in synch)
                                    | SYNCTR = (2 * ENGCYL) - 1
                                    | EXIT
```

CHAPTER 7

IGNITION TIMING STRATEGY

BASE SPARK ANGLE CALCULATION
(background calculation)

OVERVIEW

The spark advance provided by the Ignition Timing Strategy depends on the engine operating mode. The three modes are:

1. SELF TEST - See SELF TEST Section

2. CRANK/UNDERSPEED MODE

   The Spark Advance, SAF, is set at 10 deg BTDC. The spark is fired when the PIP rising edge signal is received.

3. RUN MODE (includes all throttle modes)

   During RUN MODE, the spark strategy can operate in any one of the following four distinct states:

   - state 0 = Normal spark state. Spark advance is calculated from the spark tables and modifying functions. Normal spark is used as the starting point for the transition ramp into idle spark, and as the ending point for the transition out of idle spark.

   - state 1 = Entry spark state. The purpose of this state is to provide a smooth transition into feedback spark state. Spark is ramped from its last value in normal spark state to the mean operating point for feedback spark state. At the same time, the proportional component is increased to its maximum contribution.

   - state 2 = Feedback spark state. In this state, there is a mean value for idle spark. A proportional gain term increases spark above the mean value if RPM is too low, and decreases spark below the mean value if RPM is too high. The error term, RPMERR_S, is filtered using time constant TCFBS.

   - state 3 = Exit spark state. The purpose of this state is to provide a smooth transition from feedback spark state to normal spark state. Spark is filtered toward the value of normal spark at a rate which is proportional to TAR using filter constant FKEXIT. At high throttle rates, spark moves immediately to the normal value.

Depending on the value of SPK_STATE, the base spark advance in RUN MODE equals:

```
SAF = SPK_NORM  (SPK_STATE = 0)
         or
      SPK_ENTRY (SPK_STATE = 1)
         or
      SPK_FBS   (SPK_STATE = 2)
         or
      SPK_EXIT  (SPK_STATE = 3)
```

SPARK STATE TRANSITION DIAGRAM
(For transitions between normal and feedback spark)

```
                                 DEFAULT
                                ---------   __ POWER UP
                               |         | /
                             2 |         V V
                           -------------------
                          |   normal spark    |   ENTRY CONDITIONS
          EXIT COMPLETE   |                   |         TRUE
                --------->|   SPK_STATE = 0   |----------
               |          |  SAF = SPK_RUNUP  | 1        |
               |          |      or SPK_NORM  |          |
               |           -------------------           |
               |                                         |
               | 2          ENTRY CONDITIONS             V
D   3  -----------------         TRUE          -----------------       D
E  ---|   exit spark    |--------------------->|   entry spark   |<---  E
F |   |                 | 1                   |                 |    | F
A |   |  SPK_STATE = 3  |                     |  SPK_STATE = 1  |    | A
U |   |  SAF = SPK_EXIT |                   1 | SAF = SPK_ENTRY |    | U
L  -->|                 |<--------------------|                 |----  L
T      -----------------    EXIT CONDITIONS    -----------------  3    T
                                 TRUE                 2 |
               |                                         |
               |                                         |
               |           -------------------           |
               |          |  feedback spark   |          |
               |        1 |                   |          |
                ----------|   SPK_STATE = 2   |<---------
        EXIT CONDITIONS   |   SAF = SPK_FBS   |   ENTRY COMPLETE
             TRUE         |                   |
                           -------------------
                              2 |
                                |         |
                                 ---------
                                  DEFAULT
```

NOTES:
- Boxes represent the 4 states in which the RUN MODE spark can exist.
- The contents of the boxes show the actions which take place during that state.
- Arrows represent transitions from one state to another (from one box to another).
- Numbers on the arrows indicate priority of that transition, compared to the priorities of other transitions out of the same state (out of the same box).
- Labels on the arrows represent logic which determines whether of not that particular transition is to take place.

DEFINITIONS

INPUTS

Registers:

- ACT = Air Charge Temperature, deg F.

- APT = Throttle Mode; -1 -> Closed Throttle, 0 -> Part Throttle, 1 -> Wide Open Throttle.

- ATMR1 = Time since leaving CRANK Mode, sec. See TIMER chapter.

- BRDRLN_SPK = Maximum SAF clip in spark states 1 and 2. Calculated using MBT and OCTANE subtractor terms in SPK_NORM equation, deg BTDC.

- DASPOT = Dashpot contribution to idle air flow, ppm. See Idle Speed Control chapter for calculation. Used here as input to Dashpot Spark multiplier, FN839.

- DNDT_SPK = Filtered rate of change of engine RPM for OSCMOD, RPM/sec. Filtered using time constant TCNDT_SPK.

- DSDRPM = Desired engine idle speed, RPM.

- ECT = Engine Coolant Temperature, deg F.

- EGRACT = Actual EGR Percentage, _%.

- FKEXIT = TAR/TARMAX, unitless. Filter constant for spark ramping function in SPK_STATE 3. FKEXIT is clipped between FKEXIT_MAX as a maximum, and FKEXIT_MIN as a minimum.

- IDLTMR = Time since entering Idle mode, sec. See TIMER Chapter.

- ISCFLG = ISC mode indicator; -1 -> Dashpot Mode; 0 -> Dashpot Preposition Mode; 1 -> Closed Loop RPM Control Mode; 2 -> Closed Loop RPM Control (Lock-out entry to RPM control).

- KSPARK = Gain term for feedback spark, (deg/RPM). There are 4 values: KSPKDO, KSPKDU, KSPKNO and KSPKNU for drive/speed high, drive/speed low, neutral/speed high, and neutral/speed low, respectively.

- MAP = Manifold Absolute Pressure, "Hg.

- N = Engine Speed, RPM.

- N_BYTE = Engine Speed, rpm; resolution is 16 RPM.

- OSCMOD = Oscillation mode spark multiplier, unitless.

- OSCTMR = OSCMOD spark delay timer, sec. See TIMERS chapter.

- RPMERR = DSDRPM - N. Idle speed error term, RPM.

- RPMERR_S = RPM error term for feedback spark. Time dependent rolling average filter of the instantaneous RPM error, RPMERR, using time constant TCFBS.

- SAF = Final Spark Advance, deg BTDC. Method of calculation depends on SPK_STATE.

- SPKAD(n) = Spark advance adjustment term for knock for cylinder "n". See KNOCK Strategy.

- SPK_ENTRY = Spark advance used when in SPK_STATE "1" to ramp from normal spark to feedback spark, deg BTDC.

- SPK_EXIT = Spark advance used when in SPK_STATE "3" to ramp from feedback spark to normal spark, deg BTDC.

- SPK_FBS = Spark advance used in SPK_STATE "2" to do feedback spark control, deg BTDC.

- SPK_IDLE = Mean operating point for spark in feedback spark state, deg BTDC. Equals SPKIDR in drive, or SPKINU - FN180 in neutral.

- SPK_NORM = Normal Mode Spark value, deg BTDC.

- SPK_RAMP = Time dependent rolling average filter of spark advance, deg BTDC. Uses SPK_IDLE as the "new" value and time constant TCRAMP. Used in calculating SPK_ENTRY in SPK_STATE "1" to ramp spark from the last value in the previous state to the spark for feedback spark mode.

- SPK_STATE = Spark State indicator; 0 -> Normal spark, 1 -> entry spark, 2 -> feedback spark, 3 -> exit spark.

- SPKTMR = Spark feedback entry transition timer, sec. Used to pace the transition into feedback spark control. Set to 0 on entry into SPK_STATE "1". Otherwise, counts up.

- TAPBAR = A time and MAP dependent rolling average of TP, counts.

- TAR = Throttle angle rate of change, deg/sec.

- TCSTRT = ECT at start-up, deg F.

- TIPRET = Tip-in Spark retard term, deg BTDC. See KNOCK section.

- TP = Throttle Position, counts.

- TP_REL = Relative Throttle Position, counts. TP - RATCH.

- TPDLBR = Filtered change of throttle position, counts. Time constant is TCTPDL.

- TRANS_T = SPKTMR/STTIM, unitless. Transition pacer used to ramp in the effect of feedback spark during SPK_STATE "1". Set to 0 on transition into state 1, and clipped to 1.0 maximum. Feedback spark cannot be entered until TRANS_T reaches ENTRY_T.

- VSBAR = Filtered vehicle speed, MPH.

Bit Flags:

- ALT_CAL_FLG = Flag to indicate use of alternate calibration.

- CSSFLG = Cold start spark flag: 0 -> no cold start spark required; 1 -> cold start spark required.

- DNDSUP = Delayed Neutral/Drive state flag. Set equal to NDSFLG when delay is complete; 1 -> drive, 0 -> neutral. See SYSTEM EQUATIONS chapter.

- MPGFLG = Flag that indicates state of Fuel Economy mode: 1 -> in Fuel Economy mode; 0 -> Not in Fuel Economy mode.

- NEWSA = Flag which indicates that a new spark advance calculation is required; 1 = new PIP received since last spark calculation.

- PTSCR = Part throttle since crank mode flag: 0 -> driver has not tipped in since start; 1 -> driver tipped in, kick down desired RPM.

- RUNUP_FLG = Flag indicating if initial runup is complete; 0 -> runup not complete, 1 -> runup complete.

- SA10FG = Flag indicating if spark advance should echo PIP in RUN mode; 0 -> do RUN mode spark advance logic, 1 -> Set SAF to 10 deg. BTDC and fire spark on rising edge of PIP.

- UNDSP = Flag indicating Engine mode; 0 -> RUN mode, 1 -> UNDERSPEED or CRANK mode.

- V_MODE_SETUP = VIP throttle adjust mode enabled flag; 1 -> enabled.

Calibration Constants:

- CSHIGH = Maximum TCSTRT for cold start spark, deg F.

- CSLOW = Minimum TCSTRT for cold start spark, deg F.

- CSSPRK = Cold start spark multiplier, unitless.

- CSSTIM = Maximum time to use cold start spark, sec.

- DELTA_SPK = Deadband to determine when spark transitions are complete, deg BTDC.

- DFTRPM = Maximum engine rpm to issue spark on the rising edge of PIP when at WOT, RPM.

- DFTRPH = Hystersis term for DFTRPM, RPM.

- DRBASE = Base desired engine idle speed in drive, RPM.

- DRBASE_ALT = Alternative Cal DRBASE.

- ENGCYL = Number of PIP up edges per revolution; (number of cylinders/2).

- ENTRY_T = Time threshold for entry into feedback spark, sec.

- FBS_MIN = Minimum clip on SAF in states 1 and 2, deg BTDC.

- FBS_MIN_ALT = Alternative Cal FBS_MIN.

- FKEXIT_MAX = Maximum clip for FKEXIT, SPK_EXIT filter constant, unitless.

- FKEXIT_MIN = Minimum clip for FKEXIT, SPK_EXIT filter constant, unitless.

- FN070(N) = RPM normalizing function for FN1120, FN1121, FN1122, FN1126, FN1127 and FN1129.

- FN071(MAP) = MAP normalizing function for FN1120, FN1121, FN1122, FN1126, FN1127 and FN1129.

- FN091(TEMP_FRAC) = TEMP_FRAC normalizing function for FN1128, where TEMP_FRAC = FRCCTM * ACT + (1 - FRCCTM) * ECT.

- FN092(TP - TAPBAR) = TP - TAPBAR normalizing function for FN1128.

- FN151(ECT) = Octane Table (FN1122) multiplier vs. ECT.

- FN152(ACT) = Octane Table (FN1122) multiplier vs. ACT.

- FN153(N) = WOT Spark Adder for Fuel Enrichment (APT = 1) vs engine speed.

- FN180(IDLTMR) = Spark reduction vs. time at idle (IDLTMR). Used as part of the inspection/maintenance strategy.

- FN182(DNDT_SPK) = Oscillation mode spark adder vs rate of change of engine speed.

- FN183(VSBAR) = Multiplier on oscillation mode spark adder (FN182) vs VSBAR.

- FN839(DASPOT) = Decel spark multiplier as a function of dashpot air flow.

- FN1120(N,MAP) = Base MBT Spark Table, deg BTDC.

- FN1121(N,MAP) = Spark Advance Adder Table for EGR, deg BTDC per 1% EGR.

- FN1122(N,MAP) = Spark Advance Reduction Table for Octane, deg BTDC.

- FN1126(N,MAP) = Base spark table for MPG mode, deg BTDC.

- FN1127(N,MAP) = Spark Advance Adder Table for Cold Temperatures and tip-ins.

- FN1128 = Multiplier for FN1127 vs TEMP_FRAC and TP - TAPBAR

- FN1129(N,MAP) = Spark Advance Adder Table for EGR in MPG mode, deg BTDC per 1_% EGR.

- FN1150 = Spark octane multiplier of FN1122(N,MAP); inputs are FN051(ECT) and FN052(ACT).

- FN1150_ALT = Alternate FN1150.

- FRCCTM = ACT/ECT proportioning factor for temperature input to FN1128.

- KS1 = Spark Adder, deg BTDC.

- KSPKDO = Feedback spark gain - Drive/speed high, deg BTDC/RPM.

- KSPKDU = Feedback spark gain - Drive/speed low, deg BTDC/RPM.

- KSPKNO = Feedback spark gain - Neutral/speed high, deg BTDC/RPM.

- KSPKNU = Feedback spark gain - Neutral/speed low, deg BTDC/RPM.

- MINMPH = Minimum speed to enter Closed Loop RPM control and do feedback spark control, MPH. ..Typical value - 3 MPH.

- NUBASE = Base desired engine idle speed in neutral, RPM.

- NUBASE_ALT = Alternative Cal NUBASE.

- OSCDLY = OSCMOD disable time after large negative change in TP, secs.

- SPKCTL = Maximum difference between DSDRPM and DRBASE/NUBASE to enable spark feedback, RPM.

- SPKCTL_ALT = Alternative Cal SPKCTL.

- SPKIDR = Nominal feedback spark operating point in drive, deg BTDC.

- SPKIDR_ALT = Alternative Cal SPKIDR.

- SPKINU = Nominal feedback spark operating point in neutral, deg BTDC.

- SPKINU_ALT = Alternative Cal SPKINU.

- SPK_RUNUP = Value used for SAF after entering RUN mode and before initial runup is complete, deg BTDC.

- SPLCLP = Lower spark clip for total spark advance (SAFTOT, including knock and TIPRET terms), deg BTDC.

- SPKLIM = Percent of "crank degrees between PIPs" used to determine the maximum spark advance increase allowed between consecutive spark events. Do NOT calibrate higher than 0.06 without Ignition Department approval. Maximum spark advance increase between events varies with the number of cylinders; Maximum increase = SPKLIM * 360/ENGCYL.

```
For 4 cyl; 0.06 * 360/2 = 10.8 degrees
    6 cyl; 0.06 * 360/3 =  7.2 degrees
    8 cyl; 0.06 * 360/4 =  5.4 degrees
```

- SPUCLP = Upper spark clip for total spark advance (SAFTOT, including knock and TIPRET terms), deg BTDC.

- STTIM = Time after transition into SPK_STATE "1" when TRANS_T will equal 1.0, sec. Controls rate at which feedback spark is included in SPK_ENTRY.

- TARMAX = Maximum TAR to ramp into normal spark. deg/sec. Higher TARs will cause SAF to jump to SPK_NORM.

- TCFBS = Time constant for RPMERR_S rolling average filter, sec.
- TCRAMP = Time constant for SPK_RAMP rolling average filter, sec.
- TPDLMX = Maximum filtered TP change for oscillation mode spark, counts.
- TPOBP2 = Maximum TP_REL for oscillation mode spark, counts.
- TPOH2 = Hysteresis for TPOBP2, counts.
- Y = Calibration development spark multiplier, unitless.
- VSOMAX = Maximum VSBAR for oscillation mode spark, MPH.
- VSOMXH = Hysteresis for VSOMAX, MPH.

OUTPUTS

Registers:

- BRDRLN_SPK = See above.
- FKEXIT = See above.
- KSPARK = See above.
- OSCMOD = See above.
- SAF = See above.
- SAFTOT = Total spark advance, including knock and tip-in retard, deg BTDC. SAFTOT = SAF + SPKAD(n) + TIPRET
- SPK_ENTRY = See above.
- SPK_EXIT = See above.
- SPK_FBS = See above.
- SPK_IDLE = See above.
- SPK_NORM = See above.
- SPK_RAMP = See above.
- SPK_STATE = See above.
- SPKTMR = See above.
- TRANS_T = See above.

Bit Flags:

- CSSFLG = See above.
- NEWSA = See above.
- SA10FG = See above.

PROCESS

STRATEGY MODULE: SPARK_BASE_COM2

NORMAL SPARK CALCULATION (SPK_NORM)
(All spark states)

SPK_NORM refers to the value of spark advance determined from the spark tables and appropriate modifying functions as shown below. SPK_NORM is the value used for SAF when SPK_STATE = 0. However, SPK_NORM is always calculated, even if SPK_STATE is not 0, to provide the correct value to the SPK_EXIT calculation when leaving feedback spark.

```
APT = 1 -------------------------|
                                 |AND -|S Q--| SA10FG = 1
N_BYTE < DFTRPM -----------------|     |     |
                                       |     | --- ELSE ---
APT < 1 -------------------------|     |     |
                                 |OR --|C    | SA10FG = 0
N_BYTE > DFTRPM + DFTRPH --------|


UNDSP = 1 -----------------------------|
                                       |OR --| SAF = 10 deg BTDC
SA10FG = 1 ----------------------------|     | EXIT Base Spark Angle Logic
                                             |
                                             | --- ELSE ---
                                             |
NEWSA = 0 -----------------------------------| Do NOT update SAF
 (previous value not used yet)               | EXIT Base Spark Angle Logic
                                             |
                                             | --- ELSE ---
                                             |
NEWSA = 1 -----------------------------------| Continue with Base Spark
                                             |  Angle Logic
                                             | Calculate new SAF based on
                                             |  SPK_STATE
```

```
ALT_CAL_FLG = 1 -----------------------------| fn1150 = FN1150_ALT(ECT,ACT)
                                              |
                                              | --- ELSE ---
                                              |
                                              | fn1150 = FN1150(ECT,ACT)
```

```
SPK_NORM = {(FN1120 or FN1126                      MBT SPARK
            + FN1127 * FN1128                      COLD TEMP and TIP IN ADDER
            + (FN1121 or FN1129) * EGRACT          EGR ADDER
            - FN1122 * fn1150(ECT,ACT)             OCTANE SUBTR.
           [+ FN153]                               WOT ADDER
           [+ OSCMOD])                             OSCILLATION SPARK ADDER
           [* FN839]                               DASHPOT MULT.
           [* CSSPRK]                              COLD START MULT.
            * Y}                                   DEVELOPMENT MULT.
            + KS1                                  DEVELOPMENT ADDER.
```

```
MPGFLG = 0 -----------------------------| Use FN1120(N,MAP) and FN1121(N,MAP)
 (not MPG mode)                         |
                                        | --- ELSE ---
                                        |
                                        | Use FN1126(N,MAP) and FN1129(N,MAP)
```

NOTE: Terms enclosed by "[ ]" are optional. See following logics to determine their usage.

WIDE OPEN THROTTLE SPARK ADDER

```
APT = 1 -----------------------------| Include FN153(N) in SPK_NORM
 (WOT mode)                          |
                                     | --- ELSE ---
                                     |
                                     | Do NOT include FN153
```

LIGHT LOAD RPM OSCILLATION ADDER

Spark advance can be modulated by to reduce engine RPM oscillations under light load conditions as follows:

```
TP_REL <= TPOBP2 --------------|S Q--|
 (Near closed throttle)        |     |
                               |     |
TP_REL > TPOBP2 + TPOH2 -------|C    |
                                     |
VSBAR <= VSOMAX ---------------|S Q--|AND -| Include OSCMOD =
 (Not at high speed)           |     |     | FN182(DNDT_SPK) * FN183(VSBAR)
                               |     |     | in SPK_NORM
VSBAR > VSOMAX + VSOMXH -------|C    |     |
                                     |     | --- ELSE ---
TPDLBR <= TPDLMX --------------------|     |
 (Not a quick tip in)                |     | Do NOT include OSCMOD in
                                     |     | SPK_NORM
OSCTMR >= OSCDLY --------------------|
 (Gear change delay)                 |
                                     |
ISCFLG <= 0 -------------------------|
 (Dashpot or preposition)
```

DASHPOT SPARK MULTIPLIER

```
ISCFLG = -1 ---------------------------| Include FN839(DASPOT) in SPK_NORM
 (Dashpot mode)                        |
                                       | --- ELSE ---
                                       |
                                       | Do NOT include FN839
```

COLD START SPARK FLAG AND MULTIPLIER

```
CSLOW < TCSTRT < CSHIGH ---------|
 (Warm start)                    |
                                 |
ATMR1 < CSSTIM ------------------|AND -| CSSFLG = 1
 (Cold start spark time)         |     |  (Set cold start spark flag)
                                 |     | Include CSSPRK in SPK_NORM
PTSCR = 0 -----------------------|     |
                                       | --- ELSE ---
                                       |
                                       | CSSFLG = 0
                                       | Do NOT include CSSPRK in SPK_NORM
```

MISCELLANEOUS SPARK CALCULATIONS

TRANS_T CALCULATION

```
Always ----------------------------------| TRANS_T = SPKTMR / STTIM
                                         |  Clip TRANS_T to 1.0 as a maximum
```

SPK_IDLE SELECT LOGIC

```
DNDSUP = 1 ---------------------|
                                |AND -| SPK_IDLE = SPKIDR_ALT
ALT_CAL_FLG = 1 ----------------|     |
                                      | --- ELSE ---
                                      |
DNDSUP = 1 ---------------------------| SPK_IDLE = SPKIDR
                                      |
                                      | --- ELSE ---
                                      |
V_MODE_SETUP = 1 ---------------------| SPK_IDLE = SPKINU
 (VIP Throttle Adjust mode)           |
                                      | --- ELSE ---
                                      |
ALT_CAL_FLG = 1 ----------------------| SPK_IDLE = SPKINU_ALT -
                                      |   FN180(IDLTMR)
                                      |
                                      | --- ELSE ---
                                      |
                                      | SPK_IDLE = SPKINU -
                                      |   FN180(IDLTMR)
```

KSPARK SELECT LOGIC

```
V_MODE_SETUP = 1 ------------------------|  KSPARK = 0
 (VIP Throttle Adjust Mode)0             |
                                         | --- ELSE ---
DNDSUP = 1 ------------------------|     |
                                   |AND -| KSPARK = KSPKDO
RPMERR_S < 0 ----------------------|     |  (Drive, speed high)
                                         |
                                         | --- ELSE ---
DNDSUP = 1 ------------------------|     |
                                   |AND -| KSPARK = KSPKDU
RPMERR_S >= 0 ---------------------|     |  (Drive, speed low)
                                         |
                                         | --- ELSE ---
DNDSUP = 0 ------------------------|     |
                                   |AND -| KSPARK = KSPKNO
RPMERR_S < 0 ----------------------|     |  (Neutral, speed high)
                                         |
                                         | --- ELSE ---
DNDSUP = 0 ------------------------|     |
                                   |AND -| KSPARK = KSPKNU
RPMERR_S >= 0 ---------------------|     |  (Neutral, speed low)
```

ENTRY LOGIC -- ENTRY INTO IDLE SPARK

```
ISCFLG >= 1 ------------------------------|
 (RPM control or lockout)                 |
                                          |
ALT_CAL_FLG = 1 --------------|           |
                              |           |
DNDSUP = 1 -------------------|AND -|     |
                              |     |     |
DSDRPM - DRBASE_ALT           |     |     |
    <= SPKCTL_ALT ------------|     |     |
                                    |     |
DNDSUP = 1 -------------------|     |     |
                              |AND -|     |
DSDRPM - DRBASE <= SPKCTL ----|     |     |
 (Near Idle RPM)                    |OR --|AND -| ENTRY CONDITIONS TRUE
                                    |     |     |
DNDSUP = 0 -------------------|     |     |     | --- ELSE ---
                              |AND -|     |     |
DSDRPM - NUBASE <= SPKCTL ----|     |     |     | ENTRY CONDITIONS NOT TRUE
                                    |     |
                                    |     |
ALT_CAL_FLG =1 ---------------|     |     |
                              |     |     |
DNDSUP = 0 -------------------|AND -|     |
                              |           |
DSDRPM - NUBASE_ALT           |           |
    <= SPKCTL_ALT ------------|           |
                                          |
CSSFLG = 0 -------------------------------|
 (Cold Start spark not in use)
```

EXIT LOGIC -- EXIT FROM IDLE SPARK

```
APT >= 0 -----------------------------|
 (Tip in)                             |OR --| EXIT CONDITIONS TRUE
                                      |     |
VSBAR > MINMPH -----------------------|     | --- ELSE ---
 (Vehicle moving)                           |
                                            | EXIT CONDITIONS NOT TRUE
```

ENTRY COMPLETE LOGIC -- CONTINUE WITH FEEDBACK SPARK

```
|SPK_RAMP - SPK_IDLE| <= DELTA_SPK ---|
 (Spark ramp to idle done)            |AND -| ENTRY COMPLETE
                                      |     |
TRANS_T >= ENTRY_T -------------------|     | --- ELSE ---
 (Transition time complete)                 |
                                            | ENTRY NOT COMPLETE
```

EXIT COMPLETE LOGIC -- CONTINUE WITH NORMAL SPARK

```
|SPK_NORM - SAF| <= DELTA_SPK --------------| EXIT COMPLETE
 (Spark has filtered to normal value)       |
                                            | --- ELSE ---
                                            |
                                            | EXIT NOT COMPLETE
```

SPARK STATE DETERMINATION LOGIC

```
SPK_STATE = 0 ------------------------|      |  (Do fixed runup spark)
 (Current state is 0)                 |AND -| SPK_STATE = 0
                                      |      | SAF = SPK_RUNUP
RUNUP_FLG = 0 ------------------------|      | NEWSA = 0
 (Initial runup not complete)                |
                                             | --- ELSE ---
SPK_STATE = 0 ------------------|            |
 (Current state is 0)           |OR --|      |
                                |     |      |  (Transition to entry spark)
SPK_STATE = 3 ------------------|     |AND -| SPK_STATE = 1
 (Current state is 3)                 |      | SPKTMR = 0
                                      |      | TRANS_T = 0
ENTRY CONDITIONS TRUE ----------------|      | SPK_RAMP = SAF
 (See entry logic)                           | Do SPK_ENTRY calculations
                                             | SAF = SPK_ENTRY
                                             | NEWSA = 0
                                             |
                                             | --- ELSE ---
SPK_STATE = 0 ------------------------|      |
 (Current state = 0)                  |      |  (Do normal spark)
                                      |OR --| SPK_STATE = 0
SPK_STATE = 3 ------------------|     |      | SAF = SPK_NORM
 (Current state = 3)            |AND -|      | NEWSA = 0
                                |            |
EXIT COMPLETE ------------------|            |
 (See exit complete logic)                   |
                                             | --- ELSE ---
SPK_STATE = 1 ------------|                  |
 (Current state is 1)     |OR --|            |
                          |     |            |
SPK_STATE = 2 ------------|     |AND -|      |
 (Current state is 2)           |     |      |
                                |     |      |  (Do exit spark)
EXIT CONDITIONS TRUE -----------|     |OR --| SPK_STATE = 3
 (See exit logic)                     |      | Do SPK_EXIT calculations
                                      |      | SAF = SPK_EXIT
SPK_STATE = 3 ------------------------|      | NEWSA = 0
 (Current state is 3)                        |
                                             | --- ELSE ---
```

(continued on next page)

(continued from previous page)

```
SPK_STATE = 1 ------------------|           |
 (Current state is 1)           |AND -|     |
                                |     |     |  (Do feedback spark)
ENTRY COMPLETE -----------------|     |OR --| SPK_STATE = 2
 (See entry complete logic)           |     | Do PK_FBS calculations
                                      |     | SAF = SPK_FBS
SPK_STATE = 2 ------------------------|     | NEWSA = 0
 (Current state is 2)                       |
                                            | --- ELSE ---
                                            |
                                            |  (Continue entry spark)
SPK_STATE = 1 ------------------------------| SPK_STATE = 1
 (Current state is 1)                       | Do SPK_ENTRY calculations
                                            | SAF = SPK_ENTRY
                                            | NEWSA = 0
```

ENTRY SPARK CALCULATION (SPK_ENTRY)
(state 1)

The entry spark calculation is performed in entry spark state (SPK_STATE = 1). The purpose of entry spark state is to perform a smooth transition into feedback spark. This is accomplished by ramping spark from the last value in the previous state to the mean value in feedback spark state. This 'ramp' is referred to as SPK_RAMP. Superimposed on SPK_RAMP is the feedback spark term, KSPARK * RPMERR_S. This term is phased in by a multiplier, TRANS_T, which starts at 0 when the entry state is first entered, and increases to 1.0 at a calibratable rate (the parameter, STTIM defines when TRANS_T reaches 1.0). Therefore, at the same time SPK_RAMP is approaching SPK_IDLE, the feedback component increases from 0 to its maximum contribution (TRANS_T = 1.0). SPK_ENTRY is clipped between FBS_MIN as a minimum and BRDRLN_SPK as a maximum. The minimum clip is done last, so that if BRDRLN_SPK is less than FBS_MIN, SPK_ENTRY will equal FBS_MIN.

SPK_ENTRY = SPK_RAMP + TRANS_T * KSPARK * RPMERR_S

clips: maximum clip (done first): SPK_ENTRY <= BRDRLN_SPK
minimum clip (done last): SPK_ENTRY >= FBS_MIN
(or FBS_MIN_ALT if ALT_CAL_FLG = 1)

where,

- BRDRLN_SPK = FN1120(N,MAP) - FN1122(N,MAP) * FN1150(ECT,ACT)
- KSPARK = KSPKDO, KSPKDU, KSPKNO or KSPKNU
- RPMERR_S = ROLAV(RPMERR, TCFBS)
- SPK_IDLE = SPKIDR or (SPKINU - FN180)
- SPK_RAMP = ROLAV(SPK_IDLE, TCRAMP)
  - SPK_RAMP is initialized to the previous value of SAF on the transition into state 1.
  - The ramp is considered complete when SPK_RAMP is within DELTA_SPK of SPK_IDLE, and SPK_RAMP is set equal to SPK_IDLE.
  - |SPK_RAMP - SPK_IDLE| <= DELTA_SPK ---------| SPK_RAMP = SPK_IDLE
- TRANS_T = SPKTMR/STTIM
  SPKTMR and TRANS_T are set to zero on the transition into state 1. Otherwise, SPKTMR always counts up, and TRANS_T is clipped to 1.0 maximum.

FEEDBACK SPARK CALCULATION (SPK_FBS)
(state 2)

The feedback spark calculation is performed in feedback spark state (SPK_STATE = 2). Spark is increased or decreased about a mean value based on a filtered RPM error term (RPMERR_S). Note the equation for SPK_FBS is identical to SPK_ENTRY with TRANS_T = 1.0, and SPK_RAMP = SPK_IDLE. The feedback spark gain has four values: KSPKDO, KSPKDU, KSPKNO and KSPKNU for drive/neutral and overspeed/underspeed, based on DNDSUP and RPMERR_S. SPK_FBS is clipped between FBS_MIN as a minimum and BRDRLN_SPK as a maximum. The minimum clip is done last, so that if BRDRLN_SPK is less than FBS_MIN, SPK_FBS will equal FBS_MIN. SPK_IDLE has values for both neutral and drive.

SPK_FBS = SPK_IDLE + KSPARK * RPMERR_S

clips: maximum clip (done first): SPK_ENTRY <= BRDRLN_SPK
minimum clip (done last): SPK_ENTRY >= FBS_MIN
(or FBS_MIN_ALT if ALT_CAL_FLG = 1)

where,

- BRDRLN_SPK = FN1120(N,MAP) - FN1122(N,MAP) * FN1150(ECT,ACT)
- KSPARK = KSPKDO, KSPKDU, KSPKNO or KSPKNU
- RPMERR_S = ROLAV(RPMERR, TCFBS)
- SPK_IDLE = SPKIDR or (SPKINU - FN180)

NOTE: KSPARK is set to zero when in VIP throttle adjust mode;
V_MODE_SETUP = 1

EXIT SPARK CALCULATION (SPK_EXIT)
(state 3)

The exit spark calculation is performed in exit spark state (SPK_STATE = 3). The purpose of exit spark is to perform a smooth transition to normal spark state. This is accomplished by filtering spark from the last value of SAF in the previous state to the present value of normal spark. Normal spark is calculated in all spark states so it will be available as an input to the exit spark equation.

The exit rate is controlled by the filter constant, FKEXIT, which is a function of TAR. Higher TARs result in faster filter constants. This allows the exit rate to vary with the type of tip-in which occurs - fast tip-ins have a filter constant of 1.0, so SPK_EXIT goes immediately to SPK_NORM. The relationship between the filter constant and TAR is calibratable.

SPK_EXIT = (1 - FKEXIT) * SAF + FKEXIT * SPK_NORM

where,

- FKEXIT = TAR/TARMAX

  clips: FKEXIT <= FKEXIT_MAX
  FKEXIT >= FKEXIT_MIN

- SAF = Spark advance from the previous calculation. On the transition into state 3, it will be the last value from the previous state. Otherwise, it will be the previous value of SPK_EXIT.

NOTES:

- The register SAF does not include the output of the individual cylinder knock strategy. The knock registers, SPKAD(n) and TIPRET, may be displayed separately and are added to SAF by the EOS when the waiting time is calculated. SAFTOT does include SPKAD(n) and TIPRET. However, since SAFTOT is updated on PIP interupts, it may NOT display every update if the background loop time is longer than a PIP period. Refer to the knock strategy documentation within this chapter for additional information on SPKAD and TIPRET.

- The final value of spark advance, SAFTOT, is limited to the range:

  SPLCLP <= SAFTOT <= SPUCLP

  where,

  SAFTOT = SAF + SPKAD(n) + TIPRET

  SPLCLP is the lower spark clip. SPUCLP is the upper spark clip. SPLCLP and SPUCLP are calibrated to match the rotor registry of the distributor. Intermediate spark calculations and results are maintained in an unlimited fashion.

- The software allows the lower spark clip, SPLCLP, to be calibrated to values down to -10 deg (10 deg ATC). This feature has been initially provided for the sole use of the Ignition Department in performing rotor registry tests. Unless prior approval has been received from the Ignition Department, Engine Systems engineers are hereby requested to refrain from calibrating SPLCLP to a value which is less than the minimum value of the "Spark Range" which is shown on the Rotor Registry page of this Chapter. Otherwise, such a calibration may result in mis/crossfire.

- Due to physical time constraints for arming the coil and firing the next spark, the largest spark advance increase allowed between consecutive spark events is limited to SPKLIM*360/ENGCYL degrees. There is no limit on the amount of spark advance decrease allowed on consecutive spark events. NOTE: SPKLIM is set to 0.06 and should not be increased without the prior approval of the Ignition Department. This clip is performed just prior to issuing the the spark, and is not reflected in SAF or SAFTOT. Therefore, the actual delivered spark may not be as advanced as indicated by SAF or SAFTOT.

  new actual spark <= previous actual spark + SPKLIM * 360/ENGCYL

DWLBSE/DWLCOR CALCULATION

OVERVIEW

This strategy is designed to work with both Thick Film Ignition - Improved Computer Control Dwell (TFI-ICCD)(IGN_TYPE = 1) and Low Data Rate - Distributorless Ignition System (LDR-DIS)(LGN_TYPE = 2) ignition systems. When in operation, this strategy will provide control of the ignition coil charge time as well as the correct positioning of the ignition spark with both events being commanded by the EEC-IV module. Because the requirement remains that spark timing is paramount, the positioning of the SPOUT up-edge could not be changed. The difference between the two ignition systems, as it effects the strategy, is the type of signal available from the ignition module on the Ignition Diagnostic Monitor (IDM) input to the EEC computer.

For the TFI-ICCD module, the signal provides information indicating when the coil starts to charge and when it reaches current limit. The LDR-DIS module IDM signal is a digital signal that indicates coil start to charge and discharge. The strategy can use the current limit information in the IDM signal from the TFI-ICCD module to reduce the dwell so the ignition system operates at near zero excess dwell at low engine speeds. Because the current limit information is not available from the LDR-DIS module, the strategy will not be able to reduce the dwell from the base dwell value and will have to operate with some excess dwell at all times.

The Computer Controlled Dwell strategy is designed to provide a function that determines when both the dwell edge of SPOUT and the spark edge of SPOUT can be positioned within the time limits after the PIP down edge and the spark rotor registry. In those instances when the dwell edge can not be positioned after the PIP down edge, the dwell strategy provides PIP acceleration and spark change factors to the dwell calculation. Under steady state conditions, this will produce some excess dwell, but will protect for cases of acceleration rates of up to 8000 RPM/second and/or spark changes of up to 6 percent of a PIP period.

During CRANK or UNDERSPEED engine modes, the strategy schedules the dwell signal at the down edge of PIP due to the highly variable acceleration rates of the engine and the low data rate of the incoming PIP signal. For LDR-DIS systems, to protect the ignition module, the SPOUT signal is held high to prevent the coil from charging, when the engine stalls or during power-up before the first PIP edge is detected.

Falling edge dwell mode: (DWLELD = 1)

Calculations:

```
                                       |<-- desired spark position
                                -->|  |<-- DWLIDM
                                   -->||<-- excess dwell
                                -->|   |<-- DWELL
     max. spark advance permitted -->| |<-- (2 * SPKLIM *HFDLTA)
                    -->|             |<-- NEXT_SPOUT_ADVANCED
                    -->|           |<-- SPOUT_ICCD_DELTA
```

Schedule SPOUT edge from vertical bar using information computed on down edge of PIP:

```
                       |---------->|-->|                  |---------->|-->|


      _________________                  _________________                  __
PIP   |                |                 |                |                 |
______|                |_________________|                |_________________|

    _______________________________    _______________________________    ____
    |                              |   |                              |   |
|___|   SPOUT                      |___|                              |___|

   _                                  _                                  _
  / |  COIL CURRENT                  / |                                / |
_/  |_______________________________/  |_______________________________/  |___

  _                                 __                                  _
 | |   IDM as seen at CPU          |  |                                | |
_| |_______________________________|  |________________________________| |____

_   _______________________________    ________________________________   ____
 | |                               |  |                                | |
 |_| IDM at Corporate Connector    |__|                                |_|
```

Rising edge dwell mode: (DWLELD = 0)

Dwell is calculated on the up edge of PIP but is not used until after the up-edge of SPOUT has been scheduled. Up-edge of SPOUT is calculated and queued from the down of edge of PIP if there is sufficient time.

Calculations:

```
                                     |<-- desired spark position
                         |           |<-- excess dwell
                     -->|            |<-- DWELL
   max. spark advance permitted -->| |<-- (2 * SPKLIM *HFDLTA)
                    -->|           |<-- NEXT_SPOUT_ADVANCED
                    -->||<-- SPOUT_ICCD_DELTA
->|                    |<-- SPOUT_LOW_DELTA
                    -->|             |<-- dwell_extra
```

Schedule SPOUT edge from vertical bar using spark information computed on down edge of PIP and dwell information computed on previous up edge of PIP:

```
                       |------------>|------------------>|
```

```
      _________________                  _________________                  __
PIP   |                 |                 |                 |                 |
______|                 |_________________|                 |_________________|

  ____________________            ____________________            ______
  |                    |              |                    |              |
__|     SPOUT          |______________|                    |______________|

__                        ____________                        ____________
  |    COIL CURRENT      /             |                     /             |
  |____________________/              |____________________/              |_____

                         _                                   _
  IDM as seen at CPU    | |                                 | |
_______________________| |_________________________________| |_________________

_______________________   _________________________________   _________________
 IDM at Corporate       | |                                 | |
          Connector     |_|                                 |_|
```

DWLBSE/DWLCOR_CALCULATION
(background calculation)

DEFINITIONS

INPUTS

Registers:

- DWELL = Value for system required dwell.
- DWLBSE = Dwell required for particular battery voltage.
- DWLCOR = Interactive correction to DWLBSE.
- DWLIDM = Measured Coil Rise Time.
- VBAT = Battery voltage.

Bit Flags:

- DWLELD = Dwell edge leads spark edge 1 <- falling edge dwell.
- NEW_DWLIDM = Flag indicating when new excess dwell can be computed.
- UNDSP = Flag indicating Engine mode; 0 -> RUN mode, 1 -> UNDERSPEED or CRANK mode.

Non-Calibratable:

- FREQ18 = Seconds to clock ticks conversion factor. 1 clock tick = 3 * 10E-6 seconds.

Calibration Constants:

- DWELLA = Base Dwell additive element.
- DWELLM = Base Dwell multiplicative element.
- DWLMIN = Minimum dwell allowed.
- DWL_XS_MIN = Minimum excess dwell in falling edge dwell.
- ENGCYL = Number of PIP up edges per revolution; (number of cylinders/2).
- IGN_TYPE = Indicator of ignition type (0 = TFI, 1 = TFI_ICCD, 2 = LDR-DIS).
- PACOFF = Offset in RPM-PIP accel bata function.
- PACPER = PIP period time switchpoint for change between PIPACL equations.
- VBAT_DWELL = Minimum battery voltage to use IDM for dwell correction.
- VBAT_DWL_HYS = Battery voltage to enable the use of IDM for dwell correction.

OUTPUTS

Registers:

- DWELL = Value for system required dwell.
- DWLBSE = Dwell required for particular battery voltage.
- DWLCOR = Interactive correction to DWLBSE.
- POFFENG = Foreground value of PACOFF divided by ENGCYL.
- PPERENG = Foreground value of PACPER divided by ENGCYL.

Bit Flags:

- CCD_HP = Flag indicating presence of computer controlled dwell hardware.
- NEW_DWLIDM = Flag indicating when new excess dwell can be computed.

PROCESS

1. Determine if there is hardware present that will support computer controlled dwell.

```
IGN_TYPE = 1 ----------------------|
(TFI-ICCD)                         |
                                   |OR -----| CCD_HP = 1
ING_TYPE = 2 ----------------------|        |
(LDR-DIS)                                   | --- ELSE ---
                                            |
                                            | CCD_HP = 0
```

2. When CCD hardware is present, the amount of dwell required to charge a coil is related to the battery voltage and the values used for DWELLM and DWELLA are calibrated by the Ignition Department to include the worst case coil. The calculation of base dwell is:

```
CCD_HP = 1 ----| DWLBSE(ticks) = [1 / (DWELLM * (VBAT - DWELLA))]
                                                    * FREQ18 (ticks/sec)
```

3. Calculate values for use in the maximum PIP acceleration calculation that is performed in the foreground:

PPERENG = PACPER / ENGCYL
POFFENG = PACOFF / ENGCYL

DWLBSE/DWLCOR_CALCULATION (continued)

4. The base dwell will be larger than the coil requires in most cases and in the TFI-ICCD system, can be reduced by a correction that uses the information supplied by the Ignition Diagnostic Monitor (IDM) signal. If the IDM signal is not present to allow the calculation of a new DWLIDM, the base dwell correction is not altered. The calculation of dwell and base dwell correction are determined by the logic below:

```
UNDSP = 1 -------------------------|
 (Not in run mode)                 |
                                   |
IGN_TYPE = 2 ----------------------|OR ------| DWLCOR = 0
 (LDR-DIS hardware)                |         | DWELL = DWLBSE
                                   |         |  (Initial dwell)
VBAT < VBAT_DWELL -------|S  Q ----|         | NEW_DWLIDM = 0
 (Low battery voltage)   |                   |
                         |                   |
VBAT > VBAT_DWL_HYS -----|C                  |
 (Slightly higher voltage)                   |
                                             | --- ELSE ---
IGN_TYPE = 1 ----------------------|         |
 (TFI-ICCD hardware)               |         |
                                   |         |
DWLELD = 1 ------------------------|         |
 (Falling edge dwell mode)         |         |
                                   |         |
UNDSP = 0 -------------------------|AND -----| DWLCOR = DWLCOR +
 (In run mode)                     |         |    [(DWELL - DWLIDM) / 2]
                                   |         | DWELL = DWLBSE - DWLCOR
DWLIDM < (DWELL - DWL_XS_MIN) -----|         | NEW_DWLIDM = 0
 (Too much excess dwell)           |         |
                                   |         |
NEW_DWLIDM = 1 --------------------|         |
 (New dwell error available)                 |
                                             | --- ELSE ---
                                             |
IGN_TYPE = 1 ----------------------|         |
                                   |         |
DWLELD = 1 ------------------------|         |
                                   |         |
UNDSP = 0 -------------------------|AND -----| DWLCOR = DWLCOR -
                                   |         |  DWL_XS_MIN
DWLIDM >= DWELL -------------------|         | (Clip DWLCOR to zero as
 (Insufficient charge time)        |         |  a minimum)
                                   |         | DWELL = DWLBSE - DWLCOR
NEW_DWLIDM = 1 --------------------|         | NEW_DWLIDM = 0
                                             |
                                             | --- ELSE ---
                                             |
IGN_TYPE = 1 --------------------------------| DWELL = DWLBSE - DWLCOR
                                             | NEW_DWLIDM = 0
```

5. Prevent the dwell value from being computed below a calibratable value by:

```
DWELL < DWLMIN ------------------------------| DWELL = DWLMIN
```

DWELL_CALCULATION
(called from various foreground routines)

OVERVIEW

Calculate the percent change in PIP period under maximum acceleration for use in the calculation of the possible dwell requirement. The acceleration factor is comprised of two linear functions which can be related to the current PIP period. The decision on which function to use is based on the number of cylinders in the engine and the possible acceleration rate for that engine. There is a clip on the maximum amount of acceleration factor since as the engine speed goes down the amount of time required to protect for any acceleration increases expontentially, and the amount of time desired for the coil to be in current limit has a finite limit.

DEFINITIONS

INPUTS

Registers:

- DT12S = The value, in clock ticks, of the current pip period.
- DWELL = Value for system required dwell.
- HFDLTA = Most recent PIP first half period.
- PIPACL = Percentage change in PIP under acceleration of 8K RPM/sec.
- POFFENG = Foreground value of PACOFF divided by ENGCYL.
- PPERENG = Foreground value of PACPER divided by ENGCYL.

Bit Flags:

- CCD_HP = Flag indicating presence of computer controlled dwell hardware.
- DWLELD = Dwell edge leads spark edge 1 <- falling edge dwell.

Calibration Constants:

- DWLMAX = Maximum dwell allowed.
- DWLTSW = Time switchpoint for maximum percentage dwell.
- MINDLA = Maximum percent of PIP period not charging coil for PIP periods greater than DWLTSW.
- MINDLB = Maximum percent of PIP period not charging coil for PIP periods less than DWLTSW.
- PACLIM = PIP period acceleration factor for dwell.
- PACSLO = Slope in RPM-PIP accel beta function.

- SPKLIM = Percent of "crank degrees between PIPs" used to determine the maximum spark advance increase allowed between consecutive spark events. Do NOT calibrate higher than 0.06 without Ignition Department approval. Maximum spark advance increase between events varies with the number of cylinders; Maximum increase = SPKLIM * 360/ENGCYL.
  For 4 cyl; 0.06 * 360/2 = 10.8 degrees
  6 cyl; 0.06 * 360/3 = 7.2 degrees
  8 cyl; 0.06 * 360/4 = 5.4 degrees

OUTPUTS

Registers:

- PIPACL = Percentage change in PIP under acceleration of 8K RPM/sec.

- SPOUT_LOW_DELTA = Delta time from spark edge to dwell edge.

PROCESS

```
CCD_HP = 1 --------------|         | PIPACL = PACSLO * DT12S - POFFENG
(ICCD hardware present)  |         | clip PIPACL so that:
                         |         | 0 < PIPACL <= PACLIM
DWLELD = 0 --------------|AND -----| dwell_extra = [(SPKLIM + PIPACL)
(not in ICCD mode)       |         |                * DT12S] + DWELL
                         |         |
DT12S >= PPERENG --------|         |
(RPM below accel bkpt)             |
                                   |
                                   |
                                   | --- ELSE ---
                                   |
CCD_HP = 1 --------------|         | PIPACL = 2 * DT12S
                         |AND -----| dwell_extra = [(SPKLIM + PIPACL)
DWLELD = 0 --------------|         |                * DT12S] + DWELL
```

Under the most adverse conditions, the ignition department has determined that there is a maximum amount of dwell. This amount is the clip, DWLMAX.

```
CCD_HP = 1 --------------|
                         |AND -----| dwell_extra = DWLMAX
dwell_extra > DWLMAX ----|         |
(too much dwell)
```

DWELL_CALCULATION (continued)

In Rising edge computer controlled dwell, it is necessary to schedule the SPOUT down or dwell edge from the SPOUT up or spark edge. The logic below decides if the PIP period minus the dwell period (i.e. time coil is off) is greater than some value, typically 50% at low speeds and 20% at all other speeds, and extends the amount of time the coil is off until the percentage is within these limits.

```
CCD_HP = 1 --------------|
                         |
DWLELD = 0 --------------|
(rising edge mode)       |
                         |AND -----| SPOUT_LOW_DELTA = MINDLA * DT12S
DT12S > DWLTSW ----------|         | (extend coil off time for low speeds)
(RPM below switchpoint)  |         |
                         |         |
[(DT12S - dwell_extra)/  |         |
   DT12S] <= MINDLA -----|         |
(coil off percentage too short)    |
                                   | --- ELSE ---
                                   |
CCD_HP = 1 --------------|         |
                         |         |
DWLELD = 0 --------------|AND -----| SPOUT_LOW_DELTA = MINDLB * DT12S
                         |         | (extend coil off time)
[(DT12S - dwell_extra)/  |         |
   DT12S] <= MINDLB -----|         |
                         |         | --- ELSE ---
                                   |
CCD_HP = 1 --------------|         |
                         |AND -----| SPOUT_LOW_DELTA = DT12S - dwell_extra
DWLELD = 0 --------------|         |
                                   | --- ELSE ---
                                   |
IGN_TYPE = 0 ----------------------| SPOUT_LOW_DELTA = HFDLTA
(TFI hardware present)             |
                                   | --- ELSE ---
                                   |
                                   | Do not calculate value for
                                   |  SPOUT_LOW_DELTA
```

MKAY/SIGKAY CALCULATIONS

DEFINITIONS

INPUTS

Registers:

- DT12S = Last PIP period.
- DT23S = Previous PIP period before DT12S.
- DTPCYC = PIP period ENGCYL * 2 + 1 cylinders previous.
- DTSIG = PIP period of last signature PIP.
- HFDLTA = Last period from PIP up-edge to down-edge.
- HFPCYC = Period from PIP up to down-edge ENGCYL * 2 cylinders previous.
- KAYCTR = A counter to indicate how often to update MKAY.
- MKAY = Half period multiplier to correct for average error caused by Hall effect sensor in distributor and armature.
- PSGDLT = Previous uncorrected signature PIP half period.
- SIGDLT = Uncorrected signature PIP half period.
- SIGKAL = Signature PIP half period multiplier - initial value = 1.66666 for 30% duty cycle signature PIP = 1.42857 for 35% duty cycle signature PIP.

Calibration Constants:

- ENGCYL = The number of cylinders in one engine revolution.
- FKMKAY = Filter constant of update rate to MKAY.
- FKSKAY = Filter constant of update rate to SIGKAL.
- IGN_TYPE = Indicator of ignition type (0 = TFI, 1 = TFI-ICCD, 2 = LDR-DIS-DP, 3 = LDR-DIS, and 4 = HDR-DIS).
- KLLIM = Lowest value for MKAY multiplier - initial value = 0.9.
- KULMT = Highest value for MKAY multiplier - initial value = 1.1.
- SIGKLL = Lowest value for signature PIP multiplier - initial value = 1.42857 for 30% duty cycle signature PIP = 1.25000 for 35% duty cycle signature PIP.
- SIGKLU = Highest value for signature PIP multiplier - initial value = 1.99996 for 30% duty cycle signature PIP = 1.66666 for 35% duty cycle signature PIP.
- SSFCTR = Steady state factor for MKAY and signature KAY calculations.

- SKSSLC = Steady state factor for Signature Kay calculation.

Bit Flags:

- SIGPIP = A flag that indicates that signature PIP half period has been identified.

- SYNFLG = Flag when set indicates Signature PIP has been identified; else Signature PIP not yet seen. It is initialized to 0.

OUTPUTS

Registers:

- DTPCYC = PIP period ENGCYL * 2 + 1 cylinders previous.

- DTSIG = PIP period of last signature PIP.

- HFDLTA = Last period from PIP up-edge to down-edge.

- HFPCYC = Period from PIP up to down-edge ENGCYL * 2 cylinders previous.

- KAYCTR = A counter to indicate how often to update MKAY.

- MKAY = Half period multiplier to correct for average error caused by Hall effect sensor in distributor and armature.

- PSGDLT = Previous uncorrected signature PIP half period.

- SIGDLT = Uncorrected signature PIP half period.

- SIGKAL = Signature PIP half period multiplier - initial value = 1.66666 for 30% duty cycle signature PIP = 1.42857 for 35% duty cycle signature PIP.

PROCESS

PIP_DATA MODULE

```
IGN_TYPE < 4 ------------------------| Call EOS_KAY_CALCULATIONS module
                                     |  (perform MKAY/SIGKAY calculation)
                                     |
                                     | --- ELSE ---
                                     |
                                     | Skip this module and continue
                                     | with PIP_DATA processing
```

MKAY/SIGKAY CALCULATIONS

Foreground module KAY (EOS_KAY_CALCULATIONS)
(called from PIP_DATA during PIP rising edge)

```
SIGPIP = 1 --------------------------|
                                     |AND ---| Do Signature Kay
SYNFLG = 1 --------------------------|       |    Caculation
                                             |
                                             | --- ELSE ---
                                             |
                                             | Skip Signature Kay
                                             | calculation. Go on
                                             | to MKAY calc. entry point
```

SIGNATURE KAY CALCULATION

```
|DT12S - DTSIG| < SKSSLC * DT12S -----|       | DTSIG = DT12S
                                      |       | sigkal' = DT12S/(SIGDLT*2)
|DT12S - DT23S| < SKSSLC * DT12S -----|AND ---| SIGKAL = FKSKAY*sigkal' +
                                      |       |        (1 - FKSKAY)*SIGKAL
|SIGDLT - PSGDLT| < SKSSLC * SIGDLT --|       | SIGKLL < SIGKAL < SIGKLU
                                              |
                                              | --- ELSE ---
                                              |
                                              | DTSIG = DT12S
```

MKAY CALC. ENTRY POINT

```
KAYCTR > ENGCYL * 2 --------------------------| Set KAYCTR = 1
                                              | Set DTPCYC = DT12S
                                              | Set HFPCYC = HFDLTA
                                              |
                                              | Return
                                              |
                                              | --- ELSE ---
                                              |
                                              | Increment KAYCTR
```

```
SIGPIP = 0 --------------------------|
                                     |AND ---| Do MKAY Kay
KAYCTR <= ENGCYL * 2 ----------------|       |    Calculation
                                             |
                                             | --- ELSE ---
                                             |
                                             | Return
```

MKAY CALCULATION

```
|DT12S - DTPCYC| <= SSFCTR * DT12S ------|
                                         |
|HFDLTA - HFPCYC| <= SSFCTR * HFDLTA ----|AND -| mkay' =
                                         |     |    0.5*(DT12S+DTPCYC)/
mkay' < 1.2 -----------------------------|     |    (HFDLTA + HFPCYC)
                                               | MKAY =
                                               |    FKMKAY * MKAY' +
                                               |    (1 - FKMKAY) * MKAY
                                               |
                                               | KLLIM < MKAY < KULMT
                                               |
                                               | --- ELSE ---
                                               |
                                               | Return
```

NOTE: On every PIP down edge transition, in the PIP_DATA routine

```
SIGPIP = 1------------------|
                            |AND ---| PSGDLT = SIGDLT
SYNFLG = 1 -----------------|       | SIGDLT = HFDLTA
                                    | HFDLTA = (HFDLTA*SIGKAL)/MKAY
```

TRANSIENT SPARK COMPENSATION

DEFINITIONS

INPUTS

Registers:

- DIFCTR = Counter for TL0FLG state changes.
- DIFF0 = Steady State Spark TL0 error.
- DIFF1 = Transient Spark TL0 error.
- DT12S = Last PIP period.
- DT23S = Previous PIP period before DT12S.
- HFDLTA = Last period from PIP up-edge to down-edge.
- MKAY = Half period multiplier to correct for average error caused by Hall effect sensor in distributor and armature.
- PHFDLT = Previous time elapsed between up-edge to down-edge of PIP.
- SPOUT = Time to fire spark.
- TPPLW = Actual time at PIP down edge (SPOUT reference).
- TSPKUP = Time to output SPOUT.

Bit Flags:

- TL0FLG = Transient Spark calculation flag.

Calibration Constants:

- DFMIN0 = Minimum number of TS0FLG 1 to 0 state changes.
- DFMIN1 = Minimum number of TS0FLG 0 to 1 state changes.
- TRSRPM = Minimum RPM to enable transient spark routine.
- TRSRPH = Hysteresis for TRSRPM.

OUTPUTS

Registers:

- DIFCTR = see above.

Bit Flags:

- TL0FLG = See above.

TRANSIENT SPARK COMPENSATION LOGIC

```
    :<-----------DT23S----------->:<-----------DT12S----------->:

    :<---PHFDLT--->:             :<---HFDLTA--->:

    .--------------.             .--------------.              .---
    |              |             |              |              |
    |              |             |              |<-----tlo---->|
    |              |             |              |              |
----.              .-------------.              .--------------.

                                                :<--TSPKUP-->:
                                                :            :
                                              TPPLW        SPOUT
```

PROCESS

After a PIP up-edge occurs, the following logic is executed:

```
DIFF0 <= DIFF1 ---------|
                        |AND -|
TLOFLG = 1 -------------|     |
                              |OR --| DIFCTR = DIFCTR + 1
DIFF0 > DIFF1 ----------|     |     |
                        |AND -|     | --- ELSE ---
TLOFLG = 0 -------------|           |
                                    | DIFCTR = 0


DIFF0 <= DIFF1 ---------------|
                              |AND -| TLOFLG = 0
DIFCTR >= DFMIN0 -------------|     | DIFCTR = 0
                                    |
DIFF0 > DIFF1 ----------------|     | --- ELSE ---
                              |AND -|
DIFCTR >= DFMIN1 -------------|     | TLOFLG = 1
                                    | DIFCTR = 0
                                    |
                                    | --- ELSE ---
                                    |
                                    | No change to TLOFLG
```

Where:

    DIFF0 = | DT12S - 2 * HFDLTA * MKAY |

    DIFF1 = | DT12S - HFDLTA - (HFDLTA * (DT23S - PHFDLT))/PHFDLT |

After a PIP down-edge occurs, TLOFLG is checked and the appropriate tlo calculation is included for SPOUT.

```
TLOFLG = 1 -----------------------|
                                  |
N > TRSRPM ---------------|S  Q --|AND ---| tlo = (DT12S - PHFDLT
                          |               |  * MKAY) / PHFDLT
N < TRSRPM - TRSRPH ------|C              |
                                          | --- ELSE ---
                                          |
                                          | tlo = HFDLTA * MKAY
```

PIP_DATA

OVERVIEW

The spark output routine (SPARK_KNOCK_CALCULATION) is called from the PIP edge processing routine. Depending on the conditions and the hardware, the SPARK_KNOCK_CALCULATION routine may be called from either the PIP high transition or the PIP low transition. Additionally, when the engine is in crank mode, underspeed mode, or SA10FG mode, the SPOUT signal will reflect the PIP signal input.

DEFINITIONS

INPUTS

Registers:

Bit Flags:

- CCD_HP = Flag indicating presence of computer controlled dwell hardware.
- DWLELD = Dwell edge leads spark edge 1 <- falling edge dwell.
- ECHO_PIP A flag that indicates when the spark output signal is output coincident with the PIP edges as they are received.
- ECHO_TRANS = A flag that indicates when the spark output signal is in the process of transitioning to or from normal spark output to the ECHO_PIP mode.
- PIP_DOUBLE = A flag indicating which edge is referenced for spark: 1 -> use PIP down edge; 0 -> use PIP up-edge.

OUTPUTS

Registers:

- OLD_BETA = The percentage of PIP period from the reference PIP edge used on the last spark output.

Bit Flags:

- DOUBLE_EDGE = A foreground (DOS) flag used to indicate the current spark output calculation method.
- ECHO_TRANS = A flag that indicates when the spark output signal is in the process of transitioning to or from normal spark output to the ECHOPIP mode.

PROCESS

RISING EDGE CALCULATIONS

Determine if and when to perform the rising edge dwell calculation by the following logic:

```
CCD_HP = 0 --------------------------|
                                     |OR ---| Call DWELL_CALCULATION
DWLELD = 0 --------------------------|
 (Rising edge mode)


                                            | (reflect PIP in SPOUT)
                                            |
ECHO_PIP = 1 ------------|                  | "echo the PIP edge
 (Not in echo mode)      |AND --------------|  transition as a
                         |                  |  SPOUT transition"
ECHO_TRANS = 0 ----------|                  | DOUBLE_EDGE = 0
 (Transition completed)                     | OLD_BETA = 1
                                            | "continue PIP processing"
                                            |
                                            | --- ELSE ---
                                            | (transition from normal
                                            |  spark to PIP echo mode)
                                            |
ECHO_PIP = 1 -------------------------------| DOUBLE_EDGE = 0
                                            | OLD_BETA = 1
                                            | "continue PIP processing"
                                            |
                                            | --- ELSE ---
PIP_DOUBLE = 0 ----------|                  |
 (Double edge spark not  |                  | (transition from PIP echo
  requested)             |                  | to normal spark mode)
                         |                  |
                         |                  | SPOUT_HIGH_EDGE = DATA_TIME
                         |                  |
                         |AND --------------| "echo the PIP edge
ECHO_TRANS = 1 ----------|                  | transition as a
                                            | SPOUT transition"
                                            |
                                            | ECHO_TRANS = 0
                                            | Call SPOUT_CALCULATION
                                            | "continue PIP processing"
                                            |
                                            | --- ELSE ---
                                            |
                                            | (normal spark mode)
                                            |
PIP_DOUBLE = 0 -----------------------------| Call SPOUT_CALCULATION
                                            | "continue PIP processing"
                                            |
                                            | --- ELSE ---
                                            |
                                            | "continue PIP processing"
```

FALLING EDGE CALCULATIONS

```
ECHO_PIP = 1 ------------|          | (transition from falling edge
 (In PIP echo mode)      |          | dwell to PIP echo mode)
                         |          |
ECHO_TRANS = 1 ----------|AND -----| DWLELD = 0
 (In transition)         |          |
                         |          | ECHO_TRANS = 0
DWLELD = 1 --------------|          |
 (In falling edge dwell)            | "Echo the PIP transition as a
                                    |  SPOUT transition"
                                    |
                                    |  "continue with PIP processing"
                                    |
                                    | --- ELSE ---
                                    |
                                    | (transition from normal spark to
ECHO_PIP = 1 ------------|          |  PIP echo mode)
 (In echo pip mode)      |AND -----|
                         |          | ECHO_TRANS = 0
ECHO_TRANS = 1 ----------|          | "continue with PIP processing"
 (In transition)                    |
                                    | --- ELSE ---
ECHO_PIP = 1 ------------|          |
                         |OR ------| (reflect PIP as SPOUT)
PIP_DOUBLE = 0 --|       |          |
 (Double edge    |AND ---|          | "echo the PIP transition as a
  spark nor      |                  |  SPOUT transition"
  spark not      |                  |
  requested)     |                  | "continue with PIP processing"
                 |                  |
ECHO_TRANS = 1 --|                  |
                                    | --- ELSE ---
                                    |
                                    | (normal falling edge spark mode)
                                    |
PIP_DOUBLE = 1 --------------------| Call SPOUT_CALCULATION
                                    | ECHO_TRANS = 0
                                    | "continue with PIP processing"
                                    |
                                    | --- ELSE ---
                                    |
                                    | "continue with PIP processing"
```

SPOUT_KNOCK ROUTINE

OVERVIEW

Falling edge computer controlled dwell can only be used in falling edge spark (PIP_DOUBLE = 1) and when there is sufficient time to permit the dwell edge to be scheduled after the high-to-low transition of PIP. The period of time between the high-to-low transition of PIP and the desired position of spark in time is calculated as a temporary value and also saved as NEXT_SPOUT_BETA.

DEFINITIONS

INPUTS

Registers:

- DWELL = Value for system required dwell.
- HFDLTA = Most recent PIP first half period.
- LAST_HI_PIP = Time of last PIP up-edge.
- NEW_BETA = Percent of PIP period from reference PIP edge to the spark firing signal.
- MKAY = Half period multiplier to correct for average error caused by Hall effect sensor in distributor and armature.
- NEXT_SPOUT_ADVANCED = Delta time from PIP down edge to position of spark with maximum spark advance on next cylinder.
- NEXT_SPOUT_BETA = The percentage of PIP period (betas) from the reference PIP edge to the signal on the spark output (SPOUT) that causes the ignition module to discharge the coil across the spark plug.
- SPOUT_HIGH_EDGE = Time of next scheduled SPOUT transition from low to high.
- SPOUT_ICCD_DELTA = Delta time from PIP down edge to the dwell edge when in falling edge mode.
- SPOUT_LOW_DELTA = Delta time from spark edge to dwell edge.
- SPOUT_LOW_EDGE = Time of next scheduled SPOUT transition from high to low.

Bit Flags:

- CCD_HP = Flag indicating presence of computer controlled dwell hardware.
- DWLELD = Dwell edge leads spark edge; 1 -> falling edge dwell.
- PIP_DOUBLE = A flag indicating which edge is referenced for spark; 1 -> use PIP down-edge, 0 -> use PIP up-edge.

Non-Calibratable:

- TICKS_DOUBLE = The value in clock-ticks when there is sufficient time in Rising-edge mode to put out spark from the falling edge (currently set equivalent to 0.0010 seconds).

- TICKS_SINGLE = The value in clock-ticks when here is insufficient time to put out spark from the falling edge of PIP (currently set equivalent of 0.0010 seconds).

Calibration Constants:

- SPKLIM = The maximum percentage of a PIP period by which the spark may be advanced between two outputs.

OUTPUTS

Registers:

- NEXT_SPOUT_ADVANCED = Delta time from PIP down edge to position of spark with maximum spark advance on next cylinder.

- NEXT_SPOUT_BETA = The percentage of PIP period (betas) from the reference PIP edge to the signal on the aprk output (SPOUT) that causes the ignition module to discharge the coil across the spark plug.

- SPOUT_HIGH_EDGE = Time of next scheduled SPOUT transition from low to high.

- SPOUT_ICCD_DELTA = Delta time from PIP down edge to the dwell edge when in falling edge mode.

- SPOUT_LOW_EDGE = Time of next scheduled SPOUT transition from high to low.

Bit Flags:

- DWLELD = Dwell edge leads spark edge 1 <- falling edge dwell.

PROCESS

1. The temporary value for NEXT_SPOUT_BETA is calculated by the following logic:

```
PIP_DOUBLE = 1 --------------------| beta = (NEW_BETA - 0.5) * 2
 (Falling edge spark)              | temp_value = (tlo * beta) +
                                   |                   (MKAY - 1) * HFDLTA
                                   | NEXT_SPOUT_BETA = temp_value
                                   |
                                   | --- ELSE ---
                                   |
                                   | NEXT_SPOUT_BETA = NEW_BETA * DT12S
```

2. Based on the current requested position of spark and the maximum amount of change allowed in the advance direction, calculate the amount of time from the next PIP down edge to the earliest possible spark position on the next cylinder. This term will be used to decide whether there is a need to switch modes for dwell.

```
CCD_HP = 1 ----| NEXT_SPOUT_ADVANCED = temp_value - (2 * SPKLIM * HFDLTA)
```

3. Compute the time for turning on the coil (dwell edge) for the current cylinder based on the spark time and the amount of dwell required to reach current limit.

```
CCD_HP = 1 ----| SPOUT_ICCD_DELTA = temp_value - DWELL
```

SPOUT_KNOCK ROUTINE

4. Determine the output values and sequence for both edges of SPOUT:

```
CCD_HP = 1 ---------------|
(CCD hardware present)    |
                          |
PIP_DOUBLE = 1 -----------|
(spark from PIP down)     |
                          |
DWLELD = 1 ---------------|AND ----| FALLING EDGE DWELL MODE
(falling edge mode)       |        | (compute time for coil start
                          |        |  to charge and schedule as the
(NEXT_SPOUT_ADVANCED      |        |  SPOUT_LOW_EDGE:)
 - DWELL) >=              |        | SPOUT_LOW_EDGE = LAST_HI_PIP + HFDLTA
     TICKS_SINGLE --------|        |                    + SPOUT_ICCD_DELTA
(sufficient time to output         | (when SPOUT_LOW_EDGE is output
 dwell edge from PIP down)         |  compute time of spark and schedule
                                   |  the SPOUT_HIGH_EDGE:)
                                   | SPOUT_HIGH_EDGE = SPOUT_LOW_EDGE
                                   |                    + DWELL
                                   |
                                   | --- ELSE ---
                                   |
                                   | TRANSITION FROM FALLING EDGE DWELL
                                   | MODE TO RISING EDGE CCD
                                   | DWLELD = 0 (change to CCD mode)
                                   | (compute time for coil start to
                                   |  charge and schedule as the
CCD_HP = 1 ---------------|        |  SPOUT_LOW_EDGE:)
                          |        |
PIP_DOUBLE = 1 -----------|AND ----| SPOUT_LOW_EDGE = LAST_HI_PIP + HFDLTA
                          |        |                  + SPOUT_ICCD_DELTA
DWLELD = 1 ---------------|        | (when SPOUT_LOW_EDGE is output,
                                   |  compute time of spark and
                                   |  schedule the SPOUT_HIGH_EDGE:)
                                   | SPOUT_HIGH_EDGE = SPOUT_LOW_EDGE
                                   |                   + DWELL
                                   | (when SPOUT_HIGH_EDGE is output,
                                   |  call DWELL_CALCULATION to
                                   |  calculate the next coil turn on
                                   |  time, SPOUT_LOW_DELTA and then
                                   |  schedule the SPOUT_LOW_EDGE:
                                   | SPOUT_LOW_EDGE = SPOUT_HIGH_EDGE
                                   |                    + SPOUT_LOW_DELTA
                                   |
                                   | --- ELSE ---
```

(continued on next page)

SPOUT_KNOCK ROUTINE

(continued from previous page)

```
CCD_HP = 1 --------------|          |
                         |AND -----| TRANSITION FROM RISING EDGE DWELL
PIP_DOUBLE = 1 ----------|          | MODE TO FALLING EDGE DWELL MODE
                         |          | DWLELD = 1 (change to ICCD mode)
(NEXT_SPOUT_ADVANCED     |          | (when the SPOUT_LOW_EDGE is no
     - DWELL) >=         |          |  longer pending in the queue,
      TICKS_DOUBLE ------|          |  compute time of spark and schedule)
(sufficient time to output          | SPOUT_HIGH_EDGE:
 dwell edge from PIP down)          | SPOUT_HIGH_EDGE = LAST_HI_PIP
                                    |                   + NEXT_SPOUT_BETA
                                    |                   + HFDLTA
                                    |
                                    | --- ELSE ---
                                    |
                                    | RISING EDGE DWELL MODE OR TFI WITH
                                    | FALLING EDGE SPARK MODE
                                    | (compute time of spark and
                                    |  schedule as the SPOUT_HIGH_EDGE:)
                                    |
PIP_DOUBLE = 1 --------------------| SPOUT_HIGH_EDGE = LAST_HI_PIP
                                    |                   + NEXT_SPOUT_BETA
                                    |                   + HFDLTA
                                    | (when SPOUT_HIGH_EDGE is output,
                                    |  compute time for coil start to
                                    |  charge and schedule as
                                    | SPOUT_LOW_EDGE:)
                                    | SPOUT_LOW_EDGE = SPOUT_HIGH_EDGE
                                    |                   + SPOUT_LOW_DELTA
                                    |
                                    | --- ELSE ---
                                    |
                                    | RISING EDGE DWELL MODE OR TFI WITH
                                    | RISING EDGE SPARK MODE
                                    | SPOUT_HIGH_EDGE = LAST_HI_PIP
                                    |                    + NEXT_SPOUT_BETA
                                    | (when SPOUT_HIGH_EDGE is output,
                                    |  compute time for coil start to
                                    |  charge and schedule as
                                    |  SPOUT_LOW_EDGE:)
                                    | SPOUT_LOW_EDGE = SPOUT_HIGH_EDGE
                                    |                   + SPOUT_LOW_DELTA
```

VIP, EOS_IDM

OVERVIEW

Upon the change in state of the high speed input for Ignition Diagnostic Monitor, the flag IDM_INT is set to one. When new input data is processed, a check is made of the flag IDM_INT, and if set, will cause this module to be called to process the IDM state change. If the transition has been high to low at the CPU (IDM_HIGH = 0), the self test logic is notified by setting the flag NEW_IDM. If the ignition system provides an IDM signal that can be used to determine the amount of excess dwell and the EEC is controlling the dwell (IGN_TYPE = 1), the time of coil charging is determined and a flag, NEW_DWLIDM, is set.

NOTE: IDM_HIGH reflects the state of the High Speed Input (HSI) pin and not the IDM voltage. Because of an inversion, when IDM voltage = 0, IDM_HIGH = 1, and when IDM voltage is greater than 3.5 volts, IDM_HIGH = 0.

DEFINITIONS

INPUTS

Registers:

- DATA_TIME = Time of latest digital input edge.
- SPOUT_LOW_EDGE = Time of next scheduled SPOUT transition from high to low.

Bit Flags:

- DWLELD = Dwell edge leads spark edge 1 <- falling edge dwell.
- IDM_HIGH = Flag that reflects the state of the High speed Input (HSI) pin. Because of an inversion, when IDM voltage = 0, IDMHIGH = 1, and when IDM voltage is greater than 3.5 volts, IDMHIGH = 0.
- IDM_INT = Flag that indicates that a change of state has occurred on the IDM input pin.

Calibration Constants:

- IGN_TYPE = Indicator of ignition type (0 = TFI, 1 = TFI_ICCD, 2 = LDR-DIS).

OUTPUTS

Registers:

- DWLIDM = Measured Coil Rise Time.

Bit Flags:

- IDM_INT = Flag that indicates that a change of state has occurred on the IDM pin.
- NEW_DWLIDM = Flag indicating when new excess dwell can be computed.
- NEW_IDM = A flag that indicates that the EOS has processed new IDM information for use by the self test (VIP) strategy.

PROCESS

```
Always ----------------------------------------| Clear IDM_INT

IDM_HIGH = 0 -----------------|                | Calculate Dwell Time
 (High-to-low transition)     |                |
                              |                | DWLIDM = (DATA_TIME -
DWLELD = 1 -------------------|AND ------------|         SPOUT_LOW_EDGE)
 (In ICCD mode)               |                |
                              |                | Set NEW_DWLIDM
IGN_TYPE = 1 -----------------|                | Set NEW_IDM
                                               |
                                               | --- ELSE ---
                                               |
IDM_HIGH = 0 ----------------------------------| Set NEW_IDM
                                               |
                                               | --- ELSE ---
                                               |
                                               | Return
```

INDIVIDUAL CYLINDER KNOCK

OVERVIEW

KNOCK HARDWARE DESCRIPTION

The knock sensor is a piezo-electric accelerometer which resonates at engine knock frequencies of approximately 5.45, 5.7, 6.0 or 8.05 kHz. The bandwidth of the resonant frequency is quite narrow (<+/- 150 Hz) to avoid resonance due to noise from other sources. The resonation causes the sensor to transmit a positive voltage, KNOCK, to the EEC hardware circuit. This hardware circuit compares the KNOCK voltage to a threshold voltage, NOISE. When KNOCK > NOISE, the hardware circuit sends a KNOCK INPUT signal to the EEC software. This event is represented on the next page as KI = 1. The EEC software stores this information until the next rising edge of PIP is received. At that time, the information is used by the KNOCK LOGIC as described in the remainder of this document.

NOISE, the threshold voltage, is a positive voltage in an RC circuit which is proportional to the Knock Input level at the time that a charging pulse, KTS, is output. This threshold voltage is established to avoid treating rod knock, piston slap, valve train noise and other noise as spark knock.

During normal engine operation, the software opens and closes a window once per PIP period. While the window is open, KTS charges up the capacitor in the RC circuit. While the window is closed, the NOISE level decays (decreases) at a steady rate determined by the time constant of the RC circuit.

$$\text{NOISE} \sim (\text{D.C. Bias} + \text{KNOCK(A)})*(1-\exp(-\text{KTS/RC})) + \text{LAST NOISE}$$

Where, NOISE is the noise threshold level
KNOCK(A) is the Knock input level at the time KTS is being output.
KTS is the pulsewidth (secs) of the charging pulse
RC is the RC time constant.
LAST NOISE is the noise level at the time KTS is output.

WARNING: To avoid raising the NOISE threshold level too high, the KTS pulse should charge the RC circuit only during that portion of the PIP period wherein no Knock is indicated, normally late in the current PIP period, or early in the following PIP period. The calibration of the pulsewidth and timing of the window is described in the Knock Threshold Sense Logic section of this strategy.

Since the noise level is a function of rpm, the NOISE threshold tends to increase with increasing rpm. At high rpm and heavy detonation conditions, knock usually continues well into the following PIP period. To avoid opening the window during this period of knock, the software withholds KTS for WINCLD PIP periods to avoid raising the noise threshold too high.

DEFINITIONS

INPUTS/OUTPUTS:

- APT = Throttle Mode Flag.

- CTFLG = Flag set to 1 to indicate Closed Throttle Tip-in.

- CWCTR = Cancel Window Counter incorporated each PIP period.

- ECT = Engine Coolant Temperature, deg F.

- KI = Knock indicated, knock level is higher than noise level; called KNK_HIGH in code.

- KIHP = Knock hardware present switch; 1 -> Knock sensor present.

- KNOCK_DETECTED = Flag set to 1, if knock occurred in current PIP half-period.

- KNOCK_OCCURRED = Flag set to 1, if knock occurred in current or last PIP period.

- KTS = Pulsewidth (clock ticks) of the charging pulse. (This signal is internal to the EEC.) Start time = LAST HI-PIP + (WINDOW_BETA * MKAY * 2 * HFDLTA).

- "LAST PIP PERIOD" = MKAY*2*HFDLTA. (MKAY and HFDLTA are defined in Base Spark Chapter)

- N_BYTE = Low Resolution rpm.

- RETINC = Calculated as a function of rpm and is subtracted from each SPKAD corresponding to a knocking cylinder. (positive degree)

- SPKADn = Spark adder terms for the nth cylinder. It is added to SAF, may be positive or negative degress.

- TBART = Filtered Throttle position (initialized to RATCH). = UROLAV(TP,TCTPT)

- TCF = Value indicating difference between TP and TBART. TCF = (TP - TBART)

- TIPFLG = Flag set to 1 to indicate a Tip-in.

- TIPRET = Degrees of tip-in retard added to SAF.

- TSLADV = Free-running millisecond timer which counts the time since the spark was last advanced by the KNOCK STRATEGY.

FOX FUNCTIONS:

- FN143 = Retard Increment as function of N (positive degrees).

- FN144 = Width of KTS as a function of N, fraction of pip period.

- FN145 = Position of KTS as a function of N, fraction of PIP period.
- FN146A = Spark Advance rate (secs) as function of N.
- FN190 = Maximum spark advance allowed from SPKAD(n) registers as a function of engine speed (N_BYTE), degrees.

CALIBRATION CONSTANTS:

- ECTIP = Minimum ECT to enable TIP-in Knock logic, deg F.
- ECTNOK = Minimum ECT to enable Knock Strategy, deg F.
- ENGCYL = Number of PIPS per revolution. = (Number of Engine Cylinders/2.)
- KACRAT = Minimum change in TP that indicates a Tip-in, counts.
- KIHP = KNOCK hardware present switch; 1 -> KNOCK sensor present.
- KNKCYL = Calibration constant which can be calibrated equal to number of cylinders, or 1. This number determines whether it is an Individual Cylinder Knock or Multi- cylinder knock strategy.
- LODNOK = Minimum MAPPA at which Knock Strategy is enabled, unitless.
- NTIP = Maximum rpm to enable Tip-in logic, rpm.
- RETLIM = Means of preventing excessive retard; SPKAD is clipped to RETLIM, degrees.
- RPMCNL = Threshold rpm below which the window is always opened, rpm.
- RPMMIN = Minimum rpm to enable Knock Strategy. (Helps prevent Spark Retard at Idle.) rpm.
- TIPINC = Advance per PIP following a Tip-in retard. (Must be a positive number; units are degrees.)
- TIPMAX = Initial amount of retard following a Tip-in. (Must be a negative number; units are degrees.)
- TPFK = Calibratible filter constant.
- WINCLD = Maximum number of PIP periods to withhold KTS KTS (to refresh NOISE threshold level) during periods of sustained knock, PIP periods.
- WINLEN = Minimum KTS pulsewidth, (fraction of PIP period).
- WOPEN = Minimum delay after the rising edge of PIP before the KTS pulse will be output (fraction of PIP period).

PROCESS

STRATEGY MODULE: SPKKNOCK_LL
KNOCK SIGNAL DETECTION

```
        :                                       :
        :............. PIP PERIOD ..............:
        :                                       :

        ---------------------                   ----------
        |                   |                   |
PIP     |                   |                   |
        |                   |                   |
   ------                   ---------------------

        :                                        :
        :<-- (WOPEN + FN145(N)) * PIP PERIOD ->  :
        :                                        :

         ------                                  ------
         |    |                                  |    |
KTS  --->|    |<--- WINLEN+FN144(N)*PIP PERIOD   |    |
         |    |                                  |    |
----------    ------------------------------------    ----

         :    :                                  :    :
         :    :                                  :    :
         :    :                                  :    :
         :    :                **                :    :
         :    :               *  *               :    :
         :    :              *    *              :    :
         :    :             *      *             :    :
         :    :-------     *        *            :    :---
NOISE    :   /:       ----*--        *           :   /:
         :  / :          *   -------  *          :  / :
 ------- : /  :         * :         ---*---      : /  :
   -------/   :        *  :             *  -------/   :
         :    :       *   :            : *       :    :
         :    :      *    :            :  *      :    :
         :    :     *     :            :   *     :    :
KNOCK    :    :    *      :            :    *    :    :
SENSOR   :    :   *       :            :     *   :    :
     *************        :            :      ************
                          :            :
                          :            :
                          :            :
                          :            :

                          --------------
                          |            |
KI                        |            |
                          |            |
    -----------------------            -------------------

    (IF KNOCK > NOISE, KI=1; OTHERWISE, KI=0)
```

STRATEGY DESCRIPTION

The Individual Cylinder Knock Strategy consists of four major sub-strategies:

1. KNOCK STRATEGY ENABLE LOGIC
2. KNOCK THRESHOLD SENSE LOGIC
3. SPARK RETARD LOGIC
4. SPARK ADVANCE LOGIC

KNOCK STRATEGY ENABLE LOGIC

The following logic is checked every background loop:

```
KIHP = 1 -------------------------|
                                  |
MAPPA > LODNOK -------------------|
                                  |AND -| ENABLE KNOCK STRATEGY
ECT > ECTNOK ---------------------|     |
                                  |     | --- ELSE ---
N > RPMMIN -----------------------|     |
                                        | DISABLE KNOCK STRATEGY
                                        | SPKAD(ALL) = 0
                                        | TSLADV = 0
```

LODNOK, ECTNOK, and RPMMIN define the minimum engine operating conditions to enable the Knock Control Strategy. These are calibration parameters accessible through VECTOR and through the calibration console.

SPKAD(ALL) are spark adder terms; SPKAD1, SPKAD2, SPKAD3, ... SPKADn; where n = KNKCYL. If KNKCYL is calibrated to be equal to the number of cylinders, then there is a unique SPKAD term for each cylinder -- INDIVIDUAL CYLINDER KNOCK. If KNKCYL is calibrated to 1, the Knock Strategy functions as a Multi-Cylinder Knock Strategy; i.e., there is only one SPKAD term. It is applied to all cylinders. If one cylinder knocks, then all cylinders get retarded an equal amount. Negative values for SPKAD mean that spark is being retarded.

KNOCK THRESHOLD SENSE (KTS) LOGIC

The software periodically opens a window which allows a Noise threshold charging pulse called KTS to raise the Knock Threshold level of the Hardware circuit. The window always opens once per PIP period unless the rpm exceeds RPMCNL. The engine developer defines the window during which the charging pulse is on by means of two fox functions (FN144, FN145) and two calibration constants (WOPEN, WINLEN). The pulsewidth of KTS defines the period of time that the capacitor in the RC circuit will be charged. Wide KTS pulses cause the threshold to increase. The timing of the KTS pulse must coincide with the optimum non-knocking portion of the PIP period over all engine rpm. Since Knock tends to extend longer through the PIP period with increasing rpm, the KTS pulse should be timed late in the current PIP period, or early in the following PIP period (95 - 110 % PIP period).

Noise threshold elevation will result when the capacitor charging rate greatly exceeds the discharge rate or when the KTS pulse is output during conditions of Knock. When knock occurs at high rpm, the charging pulse window is kept closed for WINCLD PIP periods to prevent elevating the NOISE threshold to the level of KNOCK, thereby preventing the EEC hardware circuit from sensing additional spark knock.

The WINDOW LOGIC and calculations shown below are checked every rising edge of PIP except in Engine Running VIP (RUNNING = 1):

The pulsewidth of KTS is equal to

    WINLEN + FN144(N) * ("LAST PIP PERIOD")

Where, WINLEN is minimum KTS pulsewidth, clock ticks

FN144(N) is fraction of pip period, BETA Units

"LAST PIP PERIOD" is equal to 60/(ENGCYL*N)

ENGCYL is number of PIPS per revolution

At the start of the Goose Test, the pulsewidth of KTS is set to V_KTS.

The timing of KTS is equal to

    (WOPEN + FN145(N)) * ("LAST PIP PERIOD")

Where, WOPEN is the minimum delay after the rising edge of the PIP before the KTS pulse will be output.

FN145(N) is fraction of pip period, BETA Units.

The WINDOW LOGIC show below is checked every falling edge of PIP:

```
KIHP = 0 -------------------------------| DO NOT OUTPUT KTS
 (KNOCK input hardware not present)     |
                                        | --- ELSE ---
N <= RPMCNL ----------------------|     |
                                  |     | CWCTR = 0
CWCTR >= WINCLD ------------------|OR --| OPEN WINDOW AT CALCULATED TIME
                                  |     |
NOT SIGNATURE PIP ----------|     |     | --- ELSE ---
                            |AND -|     |
KNOCK_DETECTED = 0 ---------|           | Increment CWCTR
 (No KNOCK in current PIP               | DO NOT OPEN WINDOW
  half period)
```

NOTE: If KIHP = 1, the KTS pulse is output even if the knock strategy is disabled to refresh the threshold level in the event that the Knock Strategy becomes enabled. The absence of the KTS pulse for more than a few PIP periods would result in full retard upon entering Knock Strategy.

```
        |<--"LAST PIP PERIOD"-->|
        _____________           _____________           ____
        |           |           |           |           |
        |           |           |           |           |
        |           |           |           |           |
PIP  ___|           |___________|           |___________|


                                |<-------A--------->|
                              _____                _______
                              |   |                |     |
                              |   |                |<-B->|
                              |   |                |     |
KTS  _________________________|   |________________|     |_

         Where A = (WOPEN + FN145)*("LAST PIP PERIOD")
               B = WINLEN + FN144*("LAST PIP PERIOD") = KTS

    Note: Range of A is typically 90 - 110 % of PIP period.


            _________             _____
            |       |             |   |
            |       |             |   |
            |       |             |   |
KI   _______|       |_____________|   |__________________________

                ________________    __________________________
                |              |    |                        |
                |              |    |                        |
                |              |    |                        |
KNOCK_DETECTED _|              |____|                        |___

                               ___________________________________
                               |
KNOCK_OCCURRED  _______________|
```

EXAMPLES

EXAMPLE 1: N < RPMCNL

```
          _______________          _______________
          |             |          |             |
          |             |          |             |
          |             |          |             |
PIP    ___|             |__________|             |___

          |             |
          |             |
          |  _______________               _____
          |  |          | |                |   |
          |  |          | |                |   |
KI     ______|          | |________________|   |___________
          |             |
          |             |
         ___            |              ___
        | |             |             | |
        | |             |             | |
KTS   ___| |___________________________| |_____________________
          |             |
          ^             ^
          |             |
Calculate & schedule    Cancel window logic checked here, but window
next KTS                not cancelled because rpm is low
```

EXAMPLE 2: N > RMPCNL

```
          _______________          _______________
          |             |          |             |
          |             |          |             |
PIP    ___|             |__________|             |_______

          |             |
          |  _______________               _____
          |  |          | |                |   |
          |  |          | |                |   |
KI     ______|          | |________________|   |_______________
          |             |
          |             |
         ___            |
        | |             |
        | |             |
KTS   ___| |___________________________________________________
          |             |
          ^             ^
          |             |
Calculate & schedule    Cancel Window Logic Checked here, next KTS
next KTS.               cancelled bacause KNOCK sensed and rpm is high.
```

The following logic is checked every pip UP edge before calculating SPOUT.

```
KIHP = 0 -------------------------------| KNOCK_DETECTED = 0
                                        | KNOCK_OCCURRED = 0
                                        |
                                        | --- ELSE ---
KNOCK_DETECTED = 1 ---------------|     |
                                  |AND -| KNOCK_OCCURRED = 1
KNK_HIGH = 1 ---------------------|     |
 (KI currently indicating KNOCK)        | --- ELSE ---
                                        |
KNOCK_DETECTED = 1 ---------------|     | KNOCK_OCCURRED = 1
                                  |AND -| KNOCK_DETECTED = 0
MKNK_HIGH = 0 --------------------|     |
 (KI currently indicating no KNOCK)     | --- ELSE ---
                                        |
                                        | KNOCK_OCCURRED = 0
```

The following code is executed in real time (almost).

```
KNOCK_INTERRUPT ------------------|
 (HI or LOW transition)           |
                                  |AND -| KNOCK_DETECTED = 1
KIHP = 1 -------------------------|     | VIP_KNOCK = 1
(KNOCK sensor present)
```

KNOCK INTERRUPT ENABLE LOGIC

The following logic is executed upon Power-up, upon Re-Init, and every Background Loop.

```
KIHP = 1 ----------------------| ALLOW KNOCK INTERRUPTS TO OCCUR.
                               |
                               | --- ELSE ---
                               |
                               | PREVENT KNOCK INTERRUPTS FROM OCCURRING.
```

SPARK RETARD LOGIC

Whenever the Knock strategy is enabled, the software calculates RETINC as a funcion of rpm. RETINC is subtracted from each SPKAD that corresponds to a "knocking" cylinder. The software keeps track of the cylinders by means of a "PIP counter". The "PIP counter" is incremented once per PIP period and is set to 1 every time it exceeds KNKCYL. To prevent excessive retard (perhaps due to erroneous knock sense) each SPKAD is clipped to RETLIM.

RETINC = FN143(N)

During a particular PIP period ("PIP counter" = n), the software makes adjustments to SPKAD(n-1) based on whether Knock was sensed during the previous PIP period and uses SPKAD(n), calculated during the previous engine cycle (KNKCYL PIP periods ago) to determine the final value of spark advance for the next spark output.

```
KNOCK STRATEGY
 ENABLED -----------|
                    |
KNOCK_OCCURRED = 1 -|
 (knock sensed      |
  during last PIP   |AND --| SPKAD(n-1) = SPKAD(n-1) - RETINC
  Period)           |      | (Clip min. SPKAD(n-1) to RETLIM)
TIPRET = 0 ---------|
```

A separate part of the retard logic responds to Tip-in detonation, and even to potential Tip-in detonation, by retarding the spark TIPRET degrees.

Tip-in detonation is a result of the relatively slow response of both MAP and N, which are average values during a PIP period, to the sudden increase in manifold pressure and decrease in engine speed, respectively, which occur within a PIP period during a Tip-in. The result is that the delivered spark is over-advanced for the instantaneous conditions until the MAP calculation has updated to reflect the higher manifold pressure and the engine speed has recovered. The recovery from a Tip-in is normally complete within a few PIP periods.

The KNOCK STRATEGY is designed to anticipate detonation following a Tip-in from idle (the worst case Tip-in condition) and respond by retarding the spark before detonation occurs. Tip-in from part- throttle results in retarded spark only if knock is sensed. In both cases, Tip-in retard is applied to whichever cylinders follow the Tip-in, not to individual cylinders as is usually done in the individual cylinder knock strategy. Thus, there is no need to wait an entire engine cycle before responding to Tip-in detonation.

The Tip-in condition is recognized by comparing TP to a filtered TP, called TBART. If TCF, the difference between TP and TBART, exceeds KACRAT, and if either the Tip-in occurred from idle or if the knock is sensed following a Tip-in from part-throttle, then the spark for the next PIP is retarded by TIPMAX degrees. On the ensuing PIPS, the amount of retard is decremented by TIPINC degrees until all Tip-in retard is removed. The Tip-in logic can be disabled by setting KACRAT = 1023.

The following Tip-in logic is checked every background loop:

CTFLG LOGIC

```
APT = -1 -------------------------|
                                  |AND -|S Q--| CTFLG = 1
MAP < TIPMAP ---------------------|     |     |
                                        |     | --- ELSE ---
MAP > TIPMAP + TIPHYS ------------------|C    |
                                              | CTFLG = 0
```

TIPFLG LOGIC

```
ECT > ECTIP ----------------------------|
                                        |
TCF >= KACRAT --------------------------|AND -| TIPFLG = 1
                                        |     |
N_BYTE < NTIP --------------------------|     | --- ELSE ---
                                              |
                                              | TIPFLG = 0
```

Where,
TCF = TP - TBART
TBART = UROLAV(TP,TCTPT) (TBART is initialized to RATCH)
TCTPT = calibratable time constant
KACRAT = calibration constant
ECTIP = calibration constant
NTIP = calibration constant

NOTE: TIPMAP + TIPHYS is clipped to 31.875 in. Hg.

The following Tip-in logic is checked every rising edge of PIP:

```
TIPRET = 0 -----------------------|
                                  |
TIPFLG = 1 -----------------------|
                                  |AND -| TBART = TP
KNOCK_OCCURRED = 1 ---|           |     | TIPRET = TIPMAX
                      |           |     |
KNOCK_ENABLED = 1 ----|           |     | --- ELSE ---
                      |AND -|     |     |
KNKCYL <> 1 ----------|     |OR --|     | TIPRET = TIPRET + TIPINC
                            |           | (Clip MAX TIPRET to 0)
CTFLG = 1 ------------------|
```

Where,
TIPRET = Tip-in retard
TIPMAX = initial amount of retard following a Tip-in
(Must be a negative number; units are degrees)
TIPINC = advance per PIP following a TIP-in retard
(Must be a positive number; units are degrees)

NOTE: The final value of Spark advance is calculated by the EOS immediately prior to calculating the waiting time:

CALCULATED SPARK OUT = SAF + TIPRET

SPARK ADVANCE LOGIC

The following logic is checked every rising edge of PIP:

```
KNOCK STRATEGY ENABLED -|
                        |AND --| SPKAD(ALL) = SPKAD(ALL) +
TSLADV >= FN146A/4 -----|      |   0.25 deg.
                               | (Clip SPKAD(ALL) to FN190(N_BYTE)
                               |   as a maximum)
                               |
                               | TSLADV = TSLADV - FN146A/4
```

TSLADV is a free running millisecond timer which counts the time since the spark was last advanced by the KNOCK STRATEGY.

If the Knock Strategy is enabled, all of the spark adders, SPKAD1 through SPKADn are incremented 0.25 degrees every FN146A/4 seconds. FN146A is equivalent to 1/FN146A (used in previous strategies). Each of the SPKADn's is clipped to FN190. If FN190 = 0, the KNOCK STRATEGY will not advance the spark beyond SAF. The output of function FN190 is the maximum amount of advance beyond the SAF value that can be tolerated for a particular engine speed. The input to FN190 is N_BYTE, engine speed in 16 rpm increments, and the output is in degrees of spark, with a range of 0 to 31.875 and a resolution of 0.25 degrees.

NOTE: If the Knock Strategy is enabled and no cylinders are knocking, the spark to each cylinder will advance to SAF + FN190. If a particular cylinder is knocking, the Retard Strategy will tend to dominate the advancing mechanism. To insure that the spark to knocking cylinders is retarded more than the strategy can advance it, FN146A should be greater than or equal to 1/FN143. When FN146A is large, then the spark advance rate is small. For example, FN146A = 0.5 is equivalent to a spark advance rate of 2 degrees/sec. FN146A = 0.25 is equivalent to spark advance rate of 4 degrees/sec.

SUMMARY AND EXAMPLE

The final value of spark advance is calculated by the EOS immediately prior to calculating the waiting time:

CALCULATED SPARK OUT (n+1) = SAF + SPKAD(n+1) + TIPRET

The table shown below is included as an illustration of the Individual Cylinder Knock Control adjustment to the Spark Advance.

Example of Individual Cylinder Knock Control (4 cyl)

PIP Counter	1	2	3	4
Adjustment to Spark	TIPRET+ SPKAD1 = +2	TIPRET+ SPKAD2 = +4	TIPRET+ SPKAD3 = -6	TIPRET+ SPKAD4 = +6
Base Spark (SAF)	24	24	24	24
Calculated Spark Out	26	28	18	30
Actual Spark Out	26	28	18	28.8

Due to the physical time constraints for arming the coil and firing the next spark, the maximum spark advance increase between consecutive spark events must be no more than SPKLIM*360/ENGCYL degrees.

In this example, SPKLIM = .06 and ENGCYL = 2. Therefore, the largest spark advance increase allowed between cylinders is 10.8 deg. There is no limit on the amount of spark advance decrease allowed between cylinders.

CHAPTER 8

EGR STRATEGY

EGR SELECT LOGIC

```
PFEHP = 0 -----------------------------| DO SONIC EGR CONTROL
                                       |
                                       | --- ELSE ---
                                       |
                                       | EGR IS DISABLED, NO EGR CONTROL
                                       |  REQUIRED
                                       | EM = 0
                                       | EGRACT = 0
                                       | RETURN
```

DEFINTIONS

INPUTS

Registers:

- APT = Throttle mode flag; -1 -> closed throttle.
- ATMR1 = Time since startup (entering RUN mode), sec.
- ATMR2 = Time since Engine Coolant temperature exceeded TEMPFB, sec.
- EOFF = The EGR valve reading when the valve is full closed in A/D counts.
- EVP = EGR valve position (Sonic EGR) in A/D counts.
- MAP = Manifold Absolute Pressure, BIN 3.
- NACTMR = Time not at closed throttle, seconds.
- RATCH = Lowest closed throttle position, counts.
- TCSTRT = Engine Coolant Temperature at Start-up, deg F.
- TP = Throttle position, A/D counts.
- TP_REL = Relative Throttle Position, TP - RATCH.

Bit Flags:

- AFMFLG = Flag indicating ACT sensor has failed; 1 -> failure.
- CFMFLG = Flag indicating ECT sensor has failed; 1 -> failure.
- CRKFLG = Crank mode flag; 0 -> underspeed or run, 1 -> crank.
- EFMFLG = Flag indicating EVP sensor has failed; 1 -> failure.
- IMS = Inferred Mileage Sensor input; 0 -> low mileage, 1 -> high mileage or no sensor present.
- MFMFLG = Flag indicating ECT sensor has failed; 1 -> failure.

- TFMFLG = Flag indicating TP sensor has failed; 1 -> failure.

- WOT_EGR_FLG = Flag to enable WOT EGR.

Calibration Constants:

- EGRDED = EVP breakpoint above/below EOFF to determine open/closed EGR valve and if EOFF has learned closed valve position, counts.

- EGRTD1 = Hot Start Time Delay before enabling EGR, secs. Low mileage.

- EGRTD2 = Time delay before enabling EGR when the coolant temperature at Start-up was in the mid-range, sec. Low mileage.

- EGRTD3 = Cold Start Time delay before enabling EGR after the coolant temperature exceeds TEMPFB, sec. Low mileage.

- EGRTD4 = Hot Start Time delay before enabling EGR, secs. High mileage or no IMS.

- EGRTD5 = Time delay before enabling EGR when the coolant temp at start-up was in the mid-range, secs. High mileage or no IMS.

- EGRTD6 = Cold Start Time delay before enabling EGR after the coolant temp exceeds TEMPFB sec. High mileage or no IMS.

- EGRTD8 = Time delay at part throttle before EGR enabled.

- EGRTB1 = Throttle angle breakpoint to disable EGR, counts.

- EGTB1H = Hysteresis for EGR disable throttle angle, counts.

- CTHIGH = Hot Start Engine Coolant Temperature, deg F.

- CTLOW = Cold Start Engine Coolant Temperature, deg F.

- MAP_WOT_EGRC = MAP below which WOT EGR is disabled.

- MAP_WOT_EGRS = MAP above which WOT EGR is enabled.

- PFEHP = PFE hardware present; 0 -> Sonic EGR being used, 1 or 2 -> EGR not used.

OUTPUTS

Bit Flags:

- EGREN = Flag indicating of EGR is enabled; 0 -> disabled, 1 -> enabled.

PROCESS

STRATEGY MODULE: EGR_ENABLE_LH

EGR ENABLE/DISABLE LOGIC
APPLICATION: SONIC AND PFE WITH IMS AND OCTANE ADJUST

The following logic describes the operating conditions during which EGR is enabled. (When PFEHP = 1 or 2, no EGR is required and the EGR Strategy is always disabled)

```
MAP > MAP_WOT_EGRS --------------|S Q -| WOT_EGR_FLG
                                 |
MAP < MAP_WOT_EGRC --------------|C


"A" (See Below) -----|
                     |AND -|
IMS = 0 -------------|     |
                           |OR --|
"B" (See Below) -----|     |     |
                     |AND -|     |
IMS = 1 -------------|           |
                                 |
CRKFLG = 0 ----------------------|
                                 |
EFMFLG = 0 ----------------------|
 (EGR sensor is OK)              |
                                 |
MFMFLG = 0 ----------------------|
 (MAP sensor is OK)              |
                                 |
AFMFLG = 0 ----------------------|
 (ACT sensor is OK)              |AND -| egr_rdy = 1
                                 |     | V_EGR_RDY = 1
CFMFLG = 0 ----------------------|     |  (set flag to initiate
 (ECT sensor is OK)              |     |   EGR self test as
                                 |     |   applicable)
TFMFLG = 0 ----------------------|     |
 (TP sensor is OK)               |     |
                                 |     |
EVP >= (EOFF - EGRDED) ----------|     |
                                 |     |
NACTMR >= EGRTD8 ----------------|     |
 (EGR turn on delay time)              |
                                       | --- ELSE ---
                                       |
                                       | egr_rdy = 0
```

"A" AND "B" LOGIC

```
TCSTRT >= CTHIGH ----------|
                           |AND -|
ATMR1 >= EGRTD1 -----------|     |
                                 |
CTLOW < TCSTRT < CTHIGH ---|     |
                           |AND -|OR --| "A"
ATMR1 >= EGRTD2 -----------|     |
                                 |
TCSTRT <= CTLOW -----------|     |
                           |AND -|
ATMR2 >= EGRTD3 -----------|


TCSTRT >= CTHIGH ----------|
                           |AND -|
ATMR1 >= EGRTD4 -----------|     |
                                 |
CTLOW < TCSTRT < CTHIGH ---|     |
                           |AND -|OR --| "B"
ATMR1 >= EGRTD5 -----------|     |
                                 |
TCSTRT <= CTLOW -----------|     |
                           |AND -|
ATMR2 >= EGRTD6 -----------|



TP_REL < EGRTB1 - EGTB1H ----------------|S Q -| EGRTPQ
                                         |
TP_REL >= EGRTB1 ------------------------|C


egr_rdy = 1 -----------------------------|
 (Time, Temprature, etc)                 |AND -| EGREN = 1 (egr enabled)
                                         |     |
APT = 0 --------------------------|      |     |
 (part throttle)                  |OR --|     |
                                  |      |     |
APT = 1 --------------------|     |      |     |
 (At WOT)                   |AND -|      |     |
                            |            |     |
WOT_EGR_FLG = 1 ------------|            |     |
 (MAP high)                              |     |
                                         |     |
EGRTPQ = 1 ------------------------------|     |
 (tp within range)                             |
                                               | --- ELSE ---
                                               |
                                               | EGREN = 0 (egr disabled)
```

SONIC EGR VALVE STRATEGY (PFEHP = 0)

The Sonic Exhaust Gas Recirculation (EGR) system offers a high degree of flexibility. The chief benefit is improved drive and fuel economy. The abilities are:

- EGR flow can be precisely varied depending upon engine operating conditions.

- Spark advance can be precisely adjusted to compensate for the actual EGR flow.

The Sonic EGR system consists of:

_ Sonic EGR valve

_ EGR valve position (EVP) sensor

_ Electronic Vacuum Regulator (EVR)

The EGR valve controls the flow of exhaust gases to the intake manifold. The pintle valve and seat assembly are designed such that EGR flow is proportional to pintle position. Further, the output of the EVP sensor is directly proportional to the pintle position. This design allows direct calculation of EGR flow.

The EGR valve is operated by manifold vacuum.

The EVR:

- Applies more vacuum to the EGR valve (increases EGR flow).

- Maintains existing EGR valve vacuum (maintains EGR flow).

- Applies less vacuum (decreases EGR flow).

The strategy enables EGR during various engine operating modes. These modes are calibration items. Typical calibrations will enable EGR when these conditions are met:

- Time since start is greater than a calibration value.

- Engine is in part throttle mode.

- Current EGR valve position is not less than the fully closed position.

The sonic EGR strategy makes two sets of calculations:

1. Desired EGR rate = EGRATE (%)
   Desired EGR mass = DESEM (PPM)
   Desired EGR valve position = DELOPT (counts)

2. Actual EGR rate = EGRACT (%) (used in the spark equation)
   Actual EGR mass = EM (PPM) (subtracted from AMPEM for fuel)
   Actual EGR valve position = EVP (counts) (actual A/D reading)

The feedback for the sonic system is the difference between the desired and actual EGR valve position.

EGRERR = DELOPT - EVP

DEFINITIONS

INPUTS

Registers:

- AM = Air Mass flow, lb/min.
- AMT = Air Mass flow for torque calculation.
- AMPEM = Air Mass plus EGR Mass Flow, lb/min. (See SYSTEMS EQUATIONS Chapter).
- AMPEMT = Air Mass plus EGR Mass Flow for torque calculation.
- BP = Barometric pressure, in. Hg.
- BPCOR = Corrected BP = FN004(BP)
- DELOPT = Filtered desired EGR valve position.
- ECT = Engine Coolant Temperature.
- EGRACT = Filtered actual EGR Percentage = UROLAV(EM * 100/AMPEM, TCEACT).
- EGRDC = Desired EVR duty cycle, %/100.
- EGRERR = DELOPT - EVP.
- EGRTMR = Accumulated time EGR is enabled, sec. (See TIMER Chapter)
- EM = Actual EGR mass flow, lb/min.
- EOFF = The lowest EGR valve reading when the valve is fully closed, in A/D counts. See System Equations.
- EVP = EGR valve position reading in A/D counts.
- MAPOPE = MAP/PEXH
- PEXH = Absolute Exhaust Pressure, "Hg = FN074A(AM) * (29.875/BPCOR) + BP.
- WOTTMR = Time at WOT.

Bit Flags:

- AFMFLG = Flag indicating ACT sensor has failed; 1 -> failure.
- CFMFLG = Flag indicating ECT sensor has failed; 1 -> failure.
- EFMFLG = Flag indicating EVP sensor has failed; 1 -> failure.
- EGREN = EGR enable/disable flag: 0 -> disable EGR; 1 -> enable EGR.
- EGRFLG = Flag that indicates whether DCOFF has been added to EGRDC.
- ISCFLG = Idle speed control mode flag (see Idle Speed Control Chapter).

- MFMFLG = Flag indicating MAP sensor has failed; 1 -> failure.
- MPGFLG = Flag indicating Fuel Economy Mode.
- TFMFLG = Flag indicating TP sensor has failed; 1 -> failure.

Calibration Constants:

- DCOFF = Duty cycle required to start to open the valve equivalent to LGAOD in the Vent/Vac system.
- EGRDED = EVP breakpoint above/below EOFF to determine open/closed EGR valve and if EOFF has learned closed valve position, counts.
- EGRTD7 = Calibration time delay to ramp on EGR, sec.
- FN074A = Exhaust pressure as a function of AM. FN074A should be measured at sea level when mapping the data.
- FN211 = EGR rate multiplier as a function of Engine Coolant Temperature ECT.
- FN212A = EGR rate multiplier as a function of Barometric Pressure BP.
- FN218 = Ratio of mass flow to choked flow as a function of MAP/PEXH.
- FN219 = EGR mass flow as a function of EGR valve position EVP - EOFF.
- FN220 = EGR rate multiplier as a function of Air Charge Temperature, ACT.
- FN221 = Desired EGR valve position as a function of desired EM.
- FN239 = Change in EVR duty cycle as a function of the EGR valve position error, EGRERR.
- FN1220 = EGR rate table as a function of LOAD and N, percent.
- FN1222 = Fuel Economy EGR Rate Table, percent.
- KPEI = Constant EGR adder.
- TCDLOP = Time constant for DELOPT rolling average filter.
- TCEACT = Time constant for EGRACT rolling average filter.
- X = EGRATE multiplier for development.

OUTPUTS

Registers:

- DELOPT = See above.
- DESEM = Desired EM + EGRATE * AMPEM/100
- EGRATE = Desired EGR rate in percent.
- EGRDC = See above.

PROCESS

STRATEGY MODULE: EGR_SONIC_COM2

```
************************************
*                                  *
*  DESIRED EGR MASS FLOW EQUATIONS *
*                                  *
************************************
```

If EGR is enabled, the base amount of EGR to be added is determined from FN1220, or FN1222 if in MPG mode.

The table values are a function of engine speed (N) and load, where load = MAP.

The base amount of EGR can be adjusted by ECT, ACT, and BP to reflect special engine operating conditions.

```
EGREN = 0 ------| Turn off EGR, close EGR valve.  Exit DESIRED EGR MASS
 (EGR disabled) |  FLOW EQUATIONS.
                | EGRATE = 0
                | DESEM = 0
                | EGRDC = 0
                | DELOPT = 0
                |
                | --- ELSE ---
                |
                | Calculate EGRATE as shown below and continue on to
                |  DESEM CALCULATION.
```

```
Desired EGR rate = EGRATE = [A * FN211(ECT) * FN212A(BP) * FN220(ACT) *
                            (EGRTMR/EGRTD7) * FN240(WOTTMR) * X] + KPEI
```

Where,

```
MPGFLG = 0 -------------------------| A = FN1220(N,MAP)
                                    |  (Base EGR table)
                                    |
                                    | --- ELSE ---
                                    |
                                    | A = FN1222(N,MAP)
                                    |  (MPG mode table)
```

Desired EGR mass = DESEM = (EGRATE * AMPEM)/100 (lbm/min)

Clip DESEM to 1.99 ppm as a maximum.

The desired EGR mass is then converted into a desired position as:

Desired EGR valve position = DELOPT' = FN221 + EOFF

NOTE: The input to FN221 is corrected desired EGR mass flow = (DESEM/FN218) * (29.875/BP)

where: 29.875/BP corrects for density, and
FN218 corrects for unchoked (i.e. sub-sonic flow).

To prevent over control of the EGR valve, the valve position, DELOPT' filtered as DELOPT using the rolling average filter routine.

DELOPT = UROLAV(DELOPT',TCDLOP) (counts)

Clip DELOPT from 0 to 1023.99 counts.

```
***********************************
*                                 *
*  ACTUAL EGR MASS FLOW EQUATIONS *
*                                 *
***********************************
```

Actual EGR rate = EGRACT' = (EM/AMPEM) * 100

EGRACT is used to modify spark advance (see SAF calculations). To prevent large instantaneous changes in calculated spark, EGRACT' is filtered as EGRACT using the rolling average filter routine.

EGRACT = UROLAV(EGRACT',TCEACT) (percent EGR)

ACTUAL EGR MASS

```
EFMFLG = 1 ------------------|
 (EGR sensor is not OK)      |
                             |
MFMFLG = 1 ------------------|
 (LOAD sensor is not OK)     |
                             |
AFMFLG = 1 ------------------|
 (ACT sensor is not OK)      |
                             |OR --| EGR IS OFF
CFMFLG = 1 ------------------|     | EM = 0
 (ECT sensor is not OK)      |     |
                             |     |
TFMFLG = 1 ------------------|     |
 (TP sensor is not OK)       |     |
                             |     |
ISCFLG = 1 OR 2 -------|     |     |
 (RPM CONTROL)         |AND -|     |
                       |           |
DELOPT = 0 ------------|           |
 (Filtered desired EGR position)   |
                                   | --- ELSE ---
                                   |
                                   | EM = FN218(MAPOPE) * FN219(EVP-EOFF)
                                   |      * (BP/29.875)
                                   | Clip EM to 1.99 ppm as a maximum
```

AM = AMPEM - EM
 (AM is used to calculate fuel flow in the FUELPW calculation)

AMT = AMPEMT - EM
 (AMT is used to calculate ARCHG in the torque calculation)

```
***********************************
*                                 *
*         FEEDBACK CONTROL        *
*                                 *
***********************************
```

The error signal used to control the EGR valve is the difference between desired and actual EGR valve position.

EGR valve position error = EGRERR = DELOPT - EVP

EGRERR in turn is used to calculate the change in EVR duty cycle (plus or minus) via FN239(EGRERR).

NOTES:

1. When the desired EGR rate EGRATE equals zero, DELOPT is then set to zero. This action will close the EGR valve when zero EGR is requested.

2. When the desired EGR rate EGRATE is nonzero and DELOPT is zero, then DELOPT is set to EOFF before the DELOPT filter is run. This makes the DELOPT filter start at the closed EGR valve position when EGR is desired.

3. DELOPT is clipped to 922 counts (90% of VREF).

SONIC EGR VALVE OUTPUT CONTROL

The EGR valve is controlled in a closed loop manner using proportional control, and the EGR valve position, EVP, as the feedback variable. The valve is moved to the desired EGR position DELOPT through output commands to the Electronic Vacuum Regulator, EVR.

EGR FLOW	EVR OUTPUT
HOLD	MAINTAIN DUTY CYCLE
INCREASE	INCREASE DUTY CYCLE
DECREASE	DECREASE DUTY CYCLE
NONE (FULLY CLOSED)	DUTY CYCLE = 0

The change in the EVR duty cycle is a function of the sign and magnitude of the error in valve position according to the following logic.

```
DELOPT = 0 -------------------------| EGRDC = 0
                                    | Clear EGRFLG
                                    |
                                    | --- ELSE ---
EVP <= EOFF + EGRDED ---------|     |
                              |AND -| EGRDC = DCOFF + FN239(EGRERR)
EGRFLG CLEAR -----------------|     | EGRFLG
                                    |
                                    | --- ELSE ---
EVP > EOFF + EGRDED ----------|     |
                              |OR --| EGRDC = EGRDC + FN239(EGRERR)
EVP <= EOFF + EGRDED ---|     |
                        |AND -|
EGRFLG SET -------------|
```

NOTE: EGRDC is clipped to 0.90.

An EVR calibration method, EVR.MEM, is available in the Strategy group user area. Copies can be made by exercising the Xerox option as explained on the second page of this Strategy Book.

EVR CONTROL ALGORITHM
SONIC AND PFE

OVERVIEW

The EVR Control routine produces a variable frequency duty cycle signal to the EVR solenoid. To produce an 80-180 Hz signal on a low speed output while minimizing real execution time, a foreground repeater process is used. A repeater is a section of code that is executed approximately every millisecond as signaled by the internally-generated Output Interrupt #1.

Once a background loop, the EVR Control module converts the current EGR duty cycle (EGRDC) into "on time" (EGRCNT) and total period (EGRPER) for use by the foreground repeater. The foreground repeater module transfers these values into corresponding foreground registers, EGRCTF and EGRPRF when EGR is requested and the current value of EGRPRF < 1.0. The values of the foreground registers, which are decremented by one each time through the repeater, determine if the EVR is to be energized. If the foreground period (EGRPRF) is >= 1.0 and the on-count (EGRCTF) > 0, the EVR is energized. If the period >= 1 and the count is 0 ("on time" complete, period incomplete), the EVR is de-energized. When the period becomes < 1.0 (period complete), the foreground registers are updated with the current background counter and period, and the process is repeated. Any fractional part left in EGRPRF is included in the next period to produce duty cycles (on the average) not obtainable with integer on times and periods.

Calibration Guides for both PFE and EVR are available. Xerox copies may be obtained in the same manner as Strategy books. The file names are PFE1.MEM and EVR.MEM.

DEFINITIONS

INPUTS

Registers:

- EGRCNT = Background EVR "on" count, unitless.
- EGRCTF = Foreground EVR "on" count, unitless.
- EGRDC = Requested EVR duty cycle, unitless.
- EGRPER = Background EVR duty cycle period, unitless.
- EGRPRF = Foreground EVR duty cycle period, unitless.

Bit Flags:

- NO_START = Engine off VIP enable flag.

OUTPUTS

Registers:

- EGRCNT = See above.
- EGRCTF = See above.
- EGRPER = See above.
- EGRPRF = See above.

Bit Flags:

- EVR = Flag indicating state of EVR output, 0 = OFF, 1 = ON.

PROCESS

STRATEGY MODULE: EGR_EVR_CONTROL_COM1

BACKGROUND EVR CONTROL MODULE

```
EGRDC = 0 -----------------------------| EGRCNT = 0
                                       | EGRPER = 0
                                       |
                                       | --- ELSE ---
                                       |
EGRDC > 0.86 --------------------------| EGRCNT = 6
                                       |
                                       | --- ELSE ---
                                       |
0.69 < EGRDC <= 0.86 ------------------| EGRCNT = 5
                                       |
                                       | --- ELSE ---
                                       |
0.50 < EGRDC <= 0.69 ------------------| EGRCNT = 4
                                       |
                                       | --- ELSE ---
                                       |
0.35 < EGRDC <= 0.50 ------------------| EGRCNT = 3
                                       |
                                       | --- ELSE ---
                                       |
0.18 < EGRDC <= 0.35 ------------------| EGRCNT = 2
                                       |
                                       | --- ELSE ---
                                       |
0.08 < EGRDC <= 0.18 ------------------| EGRCNT = 1
                                       |
                                       | --- ELSE ---
                                       |
                                       | EGRCNT = 0

EGRDC <> 0 ----------------------------| EGRPER = (EGRCNT/EGRDC)
                                       |  (clip EGRPER to 12.0 as
                                       |   maximum)
```

EVR FOREGROUND REPEATER MODULE
(Performed on one-millisecond interrupt)

```
NO_START = 1 --------------------------| EVR IS UNDER VIP CONTROL
 (in KOEO VIP Test)                    | Do NOT Update EVR in this
                                       | Module
                                       |
                                       | --- ELSE ---
                                       |
EGRCNT = 0 ----------------------------| EVR = 0
 (EGR not requested)                   |  (turn EVR off)
                                       | EGRPRF = 0
                                       |
                                       | --- ELSE ---
                                       |
EGRPRF < 1.0 --------------------------| EVR = 1
 (last period complete,                |  (turn EVR on)
  get new data)                        | EGRCTF = EGRCNT - 1
                                       | EGRPRF = EGRPRF
                                       |  + EGRPER - 1.0
                                       |
                                       | --- ELSE ---
                                       |
EGRPRF >= 1.0 -------------------|     | EVR = 1
 (period incomplete)             |     |  (continue EVR on)
                                 |AND -| EGRCTF = EGRCTF - 1
EGRCTF > 0 ----------------------|     | EGRPRF = EGRPRF - 1.0
 (on time incomplete)                  |
                                       | --- ELSE ---
                                       |
EGRPRF >= 1.0 -------------------|     | EVR = 0
 (period incomplete)             |     |  (turn EVR off, or
                                 |AND -|   continue off)
EGRCTF = 0 ----------------------|     | EGRPRF = EGRPRF - 1.0
 (on time complete)
```

CHAPTER 9

IDLE SPEED CONTROL

GENERIC IDLE SPEED CONTROL

OVERVIEW

This chapter describes the adaptive air bypass idle speed control system. In general, the ISC system is designed to regulate the duty cycle to an air bypass solenoid as necessary to obtain the desired engine speed for all idle operating conditions (base idle; hi-cam; various accessory loads) and provide for a dashpot action. Predicted airflows for the different load states at idle are adaptively corrected to minimize the impact of hardware variability. Acceptable idling performance is achieved by a careful balance of bypass air solenoid control, feedback spark control, and idle fuel modulation.

Idle Fuel Modulation is used to eliminate the fueling errors common to speed density systems. Although Idle Fuel Modulation is important to overall idling performance, it should not be used to control idle rpm. Idle Fuel modulation is described in the fuel chapter.

Feedback Spark offers the fastest way to change engine torque. As such, its use is important in the control of load transitions like A/C, power steering, and neutral/drive changes. Aggressive use of Feedback Spark is very effective in limiting rpm changes during these conditions. The Feedback Spark strategy is described in the spark chapter.

The amount of airflow through the air bypass is controlled by the solenoid position, which is in turn determined by the solenoid duty cycle. The objective of the idle speed control strategy is to determine ISCDTY. As mentioned above, calibration of the bypass actuator control must be coordinated with that of Feedback Spark and Idle Fuel Modulation.

The overall bypass air ISC logic sets ISCDTY to one of the following:

I. CRANK

a) Engine Stopped:

ISCDTY = 0%

b) Engine Moving:

ISCDTY = FN884(TCSTRT)%

II. FMEM (MAP or TP sensor out of range):

ISCDTY = FMMISC

III. NORMAL RUNNING ISC:

ISCDTY = IDCMUL * FN800(DEBYMA) * FN820(VACUUM) + IDCOFS

NORMAL ISC

Under most operating conditions, ISCDTY is obtained from the Normal ISC logic. During Normal ISC, the strategy can operate in any one of four modes: DASHPOT PREPOSITION, DASHPOT, RPM CONTROL, or LOCKOUT. The modes will be described in more detail shortly.

Regardless of which of the four modes is active, the strategy first calculates a total desired idle airflow, DESMAF:

DESMAF = DESMAF_PRE + DASPOT + IPSIBR + ISCKAM

DESMAF represents the total engine airflow required for idle. There are slight differences in the calculation of DESMAF depending on which mode is active. These differences are summarized below and will be described in more detail in the discussion of each individual mode.

DASHPOT PREPOSITION MODE (ISCFLG = 0)

- DESMAF_PRE = Initial prediction, based on rpm, load, and temperature
- DASPOT = DASPTK * (DSTPBR - (RATCH + DELHYS)) + DASPTO
- IPSIBR = Fixed at last calculated value (not updated in this mode)
- ISCKAMn = KAM cell n, where n is selected by ISFLAG

DASHPOT MODE (ISCFLG = -1)

- DESMAF_PRE = Initial prediction, based on rpm, load and temperature
- DASPOT = old DASPOT - FN879(DASPOT)
- IPSIBR = Fixed at last calculated value (not updated in this mode)
- ISCKAMn = KAM cell n, where n is selected by ISFLAG

RPM CONTROL MODE (ISCFLG = 1)

- DESMAF_PRE = Initial prediction, based on rpm, load and temperature
- DASPOT <= DASCTL (DASPOT must be below DASCTL to enter rpm control)
- IPSIBR = old IPSIBR + ISCPSI
- ISCKAMn = KAM cell n, where n is selected by ISFLAG

RPM CONTROL LOCKOUT MODE (ISCFLG = 2)

- DESMAF_PRE = Initial prediction, based on rpm, load and temperature
- DASPOT <= DASCTL (DASPOT must be below DASCTL to enter Lockout Mode
- IPSIBR = old IPSIBR + ISCPSI
- ISCKAMn = KAM cell n, where n is selected by ISFLAG

During Normal ISC, the strategy executes a number of tasks in a specific order. These tasks perform the actual logic and calculations required for any of the four modes of operation. The order of execution of the tasks is listed below. The details of each task are described in separate sections which follow.

NORMAL ISC TASKS

1) DSDRPM_CALC - calculation of DSDRPM & DESMAF_PRE
2) RPMERR_CALC - calculation of RPMERR_A & RPMERR_S
3) DASPOT_CALC - calculation of DASPOT
4) MODE_SELECT - selection of mode & setting of ISCFLG
5) IPSIBR_CALC - IPSIBR update & calculation of DESMAF
6) ISCDTY_CALC - calculation of DEBYMA & ISCDTY
7) ISCKAM_UPDATE - adaptive update of ISCKAM

DEFINITIONS

INPUTS

Registers:

- APT = Throttle Mode flag.

- ATMR3 = Time since entering RUN mode, secs.

- N = Engine revolutions, RPM.

- N_RATCH = RPM value which only ratchets down. When not at closed throttle, N = N_RATCH. When at closed throttle, N_RATCH is only allowed to go down. N_RATCH is an input to the minimum daspot clip. N_RATCH ratchets down to prevent rpm flares after a declutch.

- RATCH = Closed throttle position, counts.

- TCSTRT = Temperature of ECT at Cold Start-up, deg F.

- TSLPIP = Time since last PIP, msecs.

- VACUUM = Intake manifold vacuum.

- VSBAR = Filtered vehicle speed for transmission.

Bit Flags:

- AFMFLG = ACT Failure flag; 1 -> ACT out of range.

- CFMFLG = ECT Failure flag; 1 -> ECT out of range.

- CRKFLG = Crank Mode flag; 1 -> Crank mode.

- DISABLE_ISC = Flag used by VIP to disable the ISC strategy and freeze ISCDTY at it's present value.

- DNDSUP = Delayed Neutral/Drive switch position. Set when drive engagement delay is exceeded; 1 -> Drive engaged.

- ISC_LATCH = ISC delay logic flag; 1 -> Enable delay logic.

- MFMFLG = MAP Failure flag; 1 -> MAP out of range.

- NDSFLG = 0 -> transmission in Neutral, 1 -> transmission in gear.

- REFLG = Re-initialization flag; 1 -> Re-init occurred.

- RUNUP_FLG = Flag indicating that initial runup is complete; 1 -> runup rpm exceeded.

- TFMFLG = TP Failure flag; 1 -> TP out of range.

Calibration Constants:

- CRKTIM = Time in run mode to clear 100% cranking duty cycle.

- DASMHYST = Hysteresis for DASMPH, mph.

- DASMPH = Minimum VSBAR for declutch DASPOT clip, mph.

- FMMDSD = Failure mode management default desired rpm.

- FMMISC = Default Duty Cycle to ISC, fraction.

- FN884(TCSTRT) = ISC Duty Cycle in Crank, deg.

- TRLOAD = Transmission Load.
  - 0 -> Manual Transmission, no clutch or gear switches, forced neutral state (NDSFLG = 0).
  - 1 -> Manual Transmission, no clutch or gear switch.
  - 2 -> Manual Transmission, one clutch or gear switch.
  - 3 -> Manual Transmission, both clutch and gear switches.
  - 4 -> Auto Transmission, non-electronic, neutral drive switch.
  - 5 -> Auto Transmission, non-electronic, neutral pressure switch, (AXOD).
  - 6 -> Auto Transmission, electronic, PRNDL sensor - park, reverse, neutral, overdrive, manual 1, manual 2.

OUTPUTS

Registers:

- DSDRPM = Desired engine speed.
- DSTPBR = Dashpot filtered throttle position.
- ISCDTY = Idle speed control duty cycle.
- N_RATCH = See above.

Bit Flags:

- FLG_DASMNQ = VSBAR flip-flop flag for minimum DASPOT clip.
- HCAMFG = Flag indicating the completion of Hi-Cam; 0 -> No desired engine speed adder exists, 1 -> an RPM adder above base idle is present. Flag is used in the ISC adaptive update routine to disable updates when HCAMFG = 1.
- ISCFLG = ISC mode flag; -1 -> Dashpot Mode, 0 -> Dashpot Preposition Mode, 1 -> Closed Loop rpm Control Mode, 2 -> Closed Loop rpm Control (Lock-out entry to rpm control).
- RUNUP_FLG = See above.

PROCESS

STRATEGY MODULE: ISC_OVERVIEW_COM3

The purpose of the ISC logic is to determine ISCDTY. Under certain conditions, ISCDTY is set to specific values. Under most operating conditions, ISCDTY is determined from the Normal ISC logic which is described later. The logic below describes, at highest level, how ISCDTY is determined.

OVERALL BYPASS AIR ISC LOGIC

```
DISABLE_ISC = 1 ------------------------|    (ISC logic disabled)
 (Flag set by VIP logic)                | freeze ISCDTY
                                        |
                                        | EXIT ISC LOGIC
                                        |
                                        | --- ELSE ---
                                        |
CRKFLG = 1 ----------------------|      |    (engine stopped)
 (in crank mode)                 |AND -| ISCDTY = 0
                                 |      |  (0% - actuator closed)
TSLPIP >= 2 sec. ----------------|      | DSTPBR = RATCH
 (no PIPs yet or stall)                 |
                                        | EXIT ISC LOGIC
                                        |
                                        | --- ELSE ---
CRKFLG = 1 ----------------------|      |
 (in crank mode)                 |      |   (engine moving)
                                 |OR --| ISCDTY = FN884(TCSTRT)%
CRANK DUTY CYCLE                 |      | (duty cycle function of
DELAY LOGIC TRUE ----------------|      |   temperature at start)
 (see logic below)                      | Do DSDRPM_CALC
                                        | Do RPMERR_CALC
                                        | (Update RUNUP_FLG)
                                        | DSTPBR = RATCH
                                        | EXIT ISC LOGIC
                                        |
                                        | --- ELSE ---
                                        |
MFMFLG = 1 ----------------------|      |   (FMEM fault present)
                                 |OR --| ISCFLG = 0
TFMFLG = 1 ----------------------|      | HCAMFG = 1
                                        | RUNUP_FLG = 1
                                        | DSDRPM = FMMDSD
                                        | ISCDTY = FMMISC
                                        |
                                        | EXIT ISC LOGIC
                                        |
                                        | --- ELSE ---
                                        |
                                        | (normal ISC)
                                        |
                                        | update FLG_DASMNQ
                                        | update N_RATCH
                                        | ISCDTY is calculated from the
                                        | Normal ISC logic described later
```

CRANK DUTY CYCLE DELAY LOGIC

This logic is part of the overall Bypass Air ISC logic described previously. The duty cycle used during engine cranking is 100%. The following logic (crank duty cycle delay) can be used to cause the 100% crank duty cycle to continue to be used for a calibratable time after entering run mode.

```
REFLG = 0 -------------------------|
 (Not a reinit)                    |
                                   |
RUNUP_FLG = 0 ---------------|     |AND -|    "TRUE"
 (runup not complete)        |     |     |
                             |     |     |
                             |     |     |
DNDSUP = 1 ------------|     |OR --|     |
 (in drive)            |     |           |
                       |     |           |
APT = -1 --------------|     |           |
 (closed throttle)     |AND -|           |
                       |                 |
ATMR3 < CRKTIM --------|                 |
 (time since run       |                 |
  mode)                |                 |
                       |                 |
ISC_LATCH = 1 ---------|                 |
                                         | --- ELSE ---
                                         |
                                         |    "FALSE"
```

ISC_LATCH LOGIC

ISC_LATCH is a software flag which is used to implement the above crank duty cycle delay logic. It is not displayable and is only included here to describe the above logic function.

```
DISABLE_ISC = 0 -------------|
 (not disabled by VIP)       |AND -|S  Q-| ISC_LATCH = 1
                             |     |     |
CRKFLG = 1 ------------------|     |     |
 (in crank mode)                   |     |
                                   |     |
DISABLE_ISC = 0 -------------|     |     |
 (not disabled by VIP)       |     |     |
                             |     |     |
CRKFLG = 0 ------------------|AND -|C    |
 (not closed throttle)       |           |
                             |           |
DUTY CYCLE DELAY ------------|           |
LOGIC "FALSE"                            |
                                         | --- ELSE ---
                                         |
                                         | ISC_LATCH = 0
```

FLG_DASMNQ and N_RATCH LOGIC

This logic is used by DASPOT. It is shown here because it is executed as part of the OVERALL ISC LOGIC. It is executed anytime the Normal Isc logic is done so it will be available, if required by DASPOT mode.

FLG_DASMNQ

```
VSBAR >= DASMPH + DASMHYST -------|S  Q-| FLG_DASMNQ = 1
                                  |     | (Prepare to add dashpot
VSBAR < DASMPH -------------------|C    |  to prevent declutch stall)
                                        |
                                        | --- ELSE ---
                                        |
                                        | FLG_DASMNQ = 0
```

N_RATCH

```
APT = 0 OR 1 ---------------------|
                                  |OR --| N_RATCH = N
NDSFLG = 1 -----------------|     |     |
                            |AND -|     | --- ELSE ---
TRLOAD = 3 -----------------|           |
                                        |
APT = -1 -------------------------|     |
                                  |AND -| N_RATCH = N
N <= N_RATCH ---------------------|     | (let N_RATCH come down)
                                        |
                                        | --- ELSE ---
                                        |
                                        | No change to N_RATCH
                                        | (RPM flare, keep N_RATCH
                                        |  the same)
```

DSDRPM CALCULATION

OVERVIEW

This section describes the calculation of desired idle rpm (DSDRPM) and the predicted mass air flow at idle (DESMAF_PRE). DSDRPM is used to calculate DESMAF_PRE, and also as a control input for closed loop and adaptive idle speed control.

The strategy calculates a desired value for engine speed, DSDRPM, which it attempts to maintain while in idle speed control. DSDRPM is composed of a Base portion, plus a Hicam portion, plus additional adders for A/C, Power Steering and Low ACT.

DSDRPM = Base + Hicam [+ RPMINC] [+ DNAC] [+ DNPOWS]

The square brackets indicate that the adders may or may not be present, depending on certain control logic. Whenever a square bracket appears in the text, there will be control logic which follows to indicate under what conditions the adder is used.

GPAS Requirements:

Automatic transmission vehicles operating in drive are limited by Ford GPAS requirements to a maximum desired idle RPM. Under these conditions, DSDRPM, including all adders, is clipped to the value ISCLPD as a maximum.

Base portion of DSDRPM:

The Base portion of DSDRPM is the part that does not go away after the engine warms up.

Base = NUBASE or DRBASE

Hicam portion of DSDRPM:

In normal (non-VIP) strategy, the Hicam portion of DSDRPM is composed of a variety of adders for special operating conditions (ECT, ACT, IDLTMR, etc).

Non-VIP: Hicam = pre_fn825a + FN825B [+pre_bzzrpm]
[+FN826A] [+ (FN880 + FN821A)]

In VIP, a special equation is used for DSDRPM:

VIP: DSDRPM = RVIPRPM [+RPMINC] [+DNAC] [+DNPOWS]

DSDRPM is allowed to rise instantaneously, but any decrease in value is filtered to prevent a sudden drop. This filtered value of DSDRPM is called DESNLO (time constant TCDESN for non-VIP, VTCDSN for VIP). If DSDRPM is decreasing, it is set to the filtered value, DESNLO. The flag, HCAMFG, is set if Hicam is non-zero, or if DSDRPM is decreasing. HCAMFG is used to prevent adaptive airflow updates (ISCKAM).

DESMAF_PRE CALCULATION

The predicted desired mass air flow (DESMAF_PRE) is the airflow which is expected to be required to provide a particular engine speed. The prediction is a function of DSDRPM, ECT, ATMR3 (time since entering run mode), power steering, A/C, and Cold Start Spark. This term is later added to an integration term (IPSIBR), a dashpot term (DASPOT), and an adaptive term (ISCKAM), to produce the total DESMAF.

DESMAF_PRE = (FN875D or FN875N) * FN1861 [+ AC_PPM] [+ PSPPM] [+ CSSMAF]

- DESMAF_PRE is a non-displayable parameter.
- FN875N and D are functions of DSDRPM. This means that, if a DSDRPM adder is used for power steering or A/C, the airflow to give the RPM increase is already accounted for in DESMAF_PRE. AC_PPM, PSPPM, and CSSMAF represent only the airflow needed for the increased load, not the increased RPM.

DEFINITIONS

INPUTS

Registers:

- AC_PPM = AC delta air mass, calculated.
- ACT = Air charge temperature, deg F.
- A3CTMR = A/C state transition timer. Timer is reset to 0 on every A/C state change.
- APT = Throttle mode flag.
- ATMR1 = Timer which counts up in run/underspeed mode.
- ATMR3 = Timer which counts up in run mode. (Reset to 0 only at powerup)
- Base = Symbol used to represent the base RPM portion of DSDRPM. Base is not displayable.
- DSDRPM = Desired engine speed. See Overview section for definition of the various uses of this register.
- DESNLO = Filtered value of DSDRPM. Applied only when DSDRPM is decreasing, using time constant TCDESN. Engine running VIP uses time constant VTCDSN.
- ECT = Engine coolant temperature, deg. F.
- Hicam = Symbol used to represent the hi-cam adders to DSDRPM. Hicam is not displayable.
- ISFLAG = Flag that indicates the degree of loading on the engine at Idle. See table at the end of the DSDRPM logic.
- ISLAST = Register which indicates the engine load state from the previous background pass.
- LOACT = Lowest value of ACT since startup.

Bit Flags:

- ACCFLG = A/C engaged flag: 1 -> A/C engaged; 0 -> A/C disengaged.
- ACIFLG = A/C engagement impending flag: 1 -> A/C about to engage - adjust airflow and fuel immediately; 0 -> A/C not about to engage.
- ALT_CAL_FLG = Flag to indicate use of alternate calibration.
- CSSFLG = Cold start spark flag; 0 -> no cold start spark required, 1 -> cold start spark required.
- DNDSUP = Delayed neutral/drive flag: 0 -> in neutral, no load; 1 -> in drive, loaded.

- POWSFG = Flag used to indicate that power steering load is high: 1 -> power steering on.

- PSFLAG = Flag to indicate last pass value of power steering to check for transitions: 1 -> power steering was on.

- PTSCR = Part throttle since crank mode flag: 0 -> driver has not tipped in since start; 1 -> driver tipped in, kick down desired RPM.

- VRUN_ISCFLG = Flag which indicates that idle speed is being controlled by Engine Running VIP: 1 -> in Engine Running VIP; 0 -> not in Engine Running VIP.

Calibration Constants:

- BZZRPM = RPM adder intended to provide a short increase in RPM for engine cleanout on start-up. The buzz-up function is not affected by the part throttle kickdown until BZZTM expires. ..Typical value - 300 RPM.

- BZZRPM_ALT = Alternate BZZRPM.

- BZZTM = Time for which BZZRPM adder is in effect. ..Typical value - 3 seconds.

- BZZTM_ALT = Alternate BZZTM.

- CHGRPM = Maximum RPM delta above base to enable battery charge, I/M logic. Also upper clip on DSDRPM adder due to battery charge and I/M. (FN821A + FN880), RPM.

- CSSMAF = Cold start spark DESMAF multiplier. Used to compensate for increased airflow requirement due to retarded spark.

- DACTM = Time to maintain A/C rpm adder after A/C has been disengaged. Used to prevent RPM changes when A/C cycles rapidly. ..Typical value - 30 sec.

- DNAC = RPM increment requested with the A/C on. ..Typical value - 75 RPM.

- DNPOWS = If a power steering pressure switch is used, this parameter increments the desired RPM when an increased load is sensed. ..Typical value - 75 RPM.

- DRBASE = Base desired engine speed in drive.

- DRBASE_ALT = Alternative Cal DRBASE.

- FN825A(ECT) = RPM adder as a function of ECT. Provides base Hi-Cam function.

- FN825A_ALT = Alternative FN825A.

- FN825B(ACT) = RPM adder as a function of ACT. Provides higher idle at very low ambients.

- FN826A(TCSTRT) = RPM adder as a function of ECT at start. This adder is not used when either the first part throttle transition since exiting crank is observed or the time since start exceeds a calibrated value (TKDTM).

- FN875D(DSDRPM) = Airflow required for closed throttle operation in drive. Input to this function is DSDRPM.

** Airflow requirements must be measured as accurately as possible over a representative population of vehicles. Data should be collected over a range of anticipated desired speeds on a stabilized engine for both neutral and drive (a temperature modulator (FN1861) will automatically adjust calibrated airflow to account for increased requirements at low ambients).

A hot wire airmeter can be remotely mounted to measure airflow directly over the desired speed range. Equipment is available at APTL to perform this procedure.

- FN875N(DSDRPM) = Airflow required for closed throttle operation in neutral. Input to this function is DSDRPM.

- FN880(IDLTMR) = DSDRPM adder vs. time at idle (IDLTMR). Used as part of the inspection/maintenance strategy. Remember that any RPM above base idle disables ISCKAM adaptive learning via HCAMFG. Also, IDLTMR requires RPM to be below IDLRPM, an absolute parameter which is not tied to DSDRPM. Too high an RPM adder in FN880 could disable IDLTMR.

- FN885 = A/C DESMAF adder based on N.

- FN887A(ACT) = A/C air flow correction based on ACT.

- FN1861(ECT,ATMR3) = Airflow multiplier vs. ECT and ATMR3. Used to compensate for additional friction at start-up as a function of time in addition to normal ECT compensation. Increased friction effects tend to go away after about one minute. Inputs are ECT normalizing by FN020C, ATMR3 normalizing by FN018B.

- ISCLPD = A clip on the maximum desired speed that can be requested with vehicle in drive. Usually the GPAS defined speed allowed at 0.2 miles on a cold start. ..Typical value - 1100 RPM.

- MINACT = Minimum ACT before adding RPMINC to desired RPM.

- NUBASE = Base desired engine speed in neutral.

- NUBASE_ALT = Alternative Cal NUBASE.

- PSPPM = Airflow increment required when power steering load is sensed. Value increments the desired flow through the ISC actuator to account for increased load. ..Typical value - 0.10 ppm.

- PSPSHP = Software switch used to indicate if Power Steering Pressure Switch is present; 1 -> switch used; 0 = no switch.

- TCDESN = Filter constant for the desired engine speed calculated value (DSDRPM). Used to slow changes in desired speed in the decreasing direction. ..Typical value - 3.5 sec

- TKDTM = Time since start after which FN826A is eliminated as a desired RPM adder. ..Typical value - 20 seconds.

- TRLOAD = Transmission Load switch - 0 -> Manual transmission, no clutch or gear switch, NDSFLG = 0; 1 -> Manual trans, no clutch or gear switch; 2 -> Manual trans, one clutch or gear switch; 3 -> Manual trans, both switches; 4 -> Automatic trans, NDS; 5 -> Automatic trans, NPS; 6 -> Automatic trans, 7 position PRNDL; 7 -> Automatic trans, PRNDL switches (4EAT).

- RVIPRPM = Desired RPM controlled by Engine Running VIP strategy.

- VTCDSN = Filter constant for ISC ramp down down, unitless.

OUTPUTS

Registers:

- DSDRPM = See above.

- DESMAF_PRE = Predicted desired idle air flow, ppm. This is the open loop air flow prediction which is required to idle, calculated as a function of ECT and time since start and including A/C, power steering and heated windshield adders. It does not include any closed loop or KAM corrections, and is NOT DISPLAYABLE.

- DESNLO = See above.

- LOACT = See above.

Bit Flags:

- HCAMFG = Flag indicating the completion of Hi-Cam; 0 -> no desired engine speed adder exists, 1 -> an rpm adder above base idle is present. Flag is used in the ISC adaptive update routine to disable updates when HCAMFG = 1.

- ISFLAG = See above.

- PSFLAG = Flag to indicate last pass value of power steering to check for transitions: 1 -> power steering was on.

PROCESS

STRATEGY MODULE: ISC_DSDRPM_COM1

For Normal Strategy: (VRUN_ISCFLG = 0)

```
    DSDRPM = (NUBASE or DRBASE) + Hicam  [+ RPMINC] [+ DNAC]  [+ DNPOWS]


ALT_CAL_FLG = 1 ------------------------| Hicam = FN825A_ALT + FN825B
                                        |         [+ FN821A] [+ FN880]
                                        |         [+ pre_bzzrpm] [+ FN826A]
                                        | pre_nubase = NUBASE_ALT
                                        | pre_drbase = DRBASE_ALT
                                        |
                                        | --- ELSE ---
                                        |
                                        | Hicam = FN825A + FN825B [+FN821A]
                                        | [+ FN880] [+ pre_bzzrpm] [+ FN826A]
                                        | pre_nubase = NUBASE
                                        | pre_drbase = DRBASE
```

For VIP strategy: (VRUN_ISCFLG = 1)

```
              DSDRPM = RVIPRPM [+ RPMINC] [+ DNAC]  [+ DNPOWS]
```

If DSDRPM is decreasing, the following filtering is done:

```
DSDRPM < DESNLO -----------------| (Filter DSDRPM - see notes below)
 (Decreasing)                    | DESNLO = UROLAV(DSDRPM,TCDESN)
                                 | DSDRPM = DESNLO
                                 |
                                 | --- ELSE ---
                                 |
                                 | (Do Not filter DSDRPM)
                                 | DESNLO = DSDRPM
```

NOTES:

- For VIP: The time constant VTCDSN is used in place of TCDESN. If VTCDSN = 0, the time constant defaults to TCDESN.

- DESNLO carries an extra byte of resolution which is not reflected in DSDRPM.

```
ALT_CAL_FLG = 1 -------------------------| pre_bzzrpm = BZZRPM_ALT
                                         | pre_fn825a = FN825A_ALT
                                         | pre_bzztm = BZZTM_ALT
                                         |
                                         | --- ELSE ---
                                         |
                                         | pre_fn825a = FN825A
                                         | pre_bzzrpm = BZZRPM
                                         | pre_bzztm = BZZTM
```

If automatic transmission and in drive, the GPAS clip is applied:

```
ALT_CAL_FLG = 1 -------------------------| pre_isclpd = ISCLPD_ALT
                                         |
                                         | --- ELSE ---
                                         |
                                         | pre_isclpd = ISCLPD


TRLOAD > 3 ----------------|
 (Not manual transmission) |
                           |
DNDSUP = 1 ----------------|
 (In drive)                |AND -| Clip DSDRPM to pre_isclpd
                           |     |  as a maximum
DSDRPM > ISCLPD -----------|     |
                                 | --- ELSE ---
                                 |
                                 | Do not clip DSDRPM
```

NOTE: The square brackets above,"[ ]", indicate that a term is optional. Anytime a square bracket appears in this chapter, there will be logic which follows to indicate under what conditions the optional term is used. The logic DOES NOT necessarily indicate the order in which software is executed.

(NUBASE or DRBASE) Logic:

```
TRLOAD > 3 ---------------|
 (Auto transmission)      |OR --| Use pre_drbase in DSDRPM calculation
                          |     |
DNDSUP = 1 ---------------|     |
 (In drive)                     |
                                | --- ELSE ---
                                |
                                | Use pre_nubase in DSDRPM calculation
```

[+RPMINC] Logic: (Low ACT and A/C panel switch adder)

```
ACT < LOACT --------------------| LOACT = ACT (lowest ACT since
                                |  startup)

LOACT <= MINACT ----------------|
 (ACT below normal)             |
                                |OR --| Add RPMINC to DSDRPM
ACD = 1 ------------------|     |     |
 (A/C Panel Switch on)    |     |     |
                          |AND -|     |
ACDHP = 1 ----------------|           |
 (A/C Panel Switch present)           |
                                      | --- ELSE ---
                                      |
                                      | Do NOT add RPMINC
```

[+DNAC] Logic: (Air Conditioning adder)

```
ACCFLG = 0 ---------------|
 (A/C Clutch off)         |
                          |
ACIFLG = 0 ---------------|AND -|
 (A/C Load off)           |     |
                          |     |
A3CTMR < DACTM -----------|     |OR --| Add DNAC to DSDRPM
 (Delay to turn                 |     |
  off A/C adder)                |     |
                                |     |
ACCFLG = 1 --------------------|     |
                                |     |
ACIFLG = 1 --------------------|     |
                                      | --- ELSE ---
                                      |
                                      | Do NOT add DNAC
```

[+DNPOWS] Logic: (Power Steering adder)

```
PSPSHP = 1 --------------------|
 (PSPS Present)                 |
                                |AND -| Add DNPOWS to DSDRPM
POWSFG = 1 --------------------|     |
 (PS On)                              | --- ELSE ---
                                      |
                                      | Do NOT add DNPOWS
```

OPTIONAL TERMS FOR Hicam CALCULATION

```
[+pre_bzzrpm] Logic: (Buzz up adder)

ATMR1 < pre_bzztm --------------------| Add pre_bzzrpm to Hicam
 (Buzz time not expired)              |
                                      | --- ELSE ---
                                      |
                                      | Do NOT add pre_bzzrpm


[+FN826A] Logic: (Engine Cleanout adder)

ATMR1 < TKDTM ------------------|
 (Kickdown time not up)         |
                                |AND -| Add FN826A to DSDRPM
PTSCR = 0 ----------------|     |     |
 (Not PT since Crank)     |OR --|     |
                          |           |
ATMR1 < pre_bzztm --------|           |
 (Buzz time not up)                   |
                                      | --- ELSE ---
                                      |
                                      | Do NOT add FN826A


[+ (FN880 + FN821A)] Logic:
(Battery charge control and Inspection Maintenance)

DSDRPM < CHGRPM ----------------------| Add (FN880 + FN821A) to DSDRPM
                                      | Clip DSDRPM to CHGRPM as a
                                      |  maximum
                                      |
                                      | --- ELSE ---
                                      |
                                      | Do NOT add (FN880 + FN821A)
```

DESMAF_PRE CALCULATION

DESMAF_PRE = ((FN875N or FN875D) * FN1861 [+AC_PPM] [+PSPPM]) [*CSSMAF]

```
(FN875N or FN875D) Logic:

TRLOAD <= 3 --------------------|
 (Manual transmission)          |
                                |OR --| Use FN875N in DESMAF_PRE
DNDSUP = 0 ---------------------|     |
 (In Neutral)                         | --- ELSE ---
                                      |
                                      | Use FN875D in DESMAF_PRE


[+AC_PPM] Logic:

ACCFLG = 1 ---------------------|
 (A/C on)                       |
                                |OR --| AC_PPM = FN885 * FN887A
ACIFLG = 1 ---------------------|     | Add AC_PPM to DESMAF_PRE
 (A/C Load impending)                 |
                                      | --- ELSE ---
                                      |
                                      | AC_PPM = 0
                                      | Add AC_PPM to DESMAF_PRE


[+PSPPM] Logic:

PSPSHP = 1 ---------------------|
 (PSPS present)                 |
                                |AND -| Add PSPPM to DESMAF_PRE
POWSFG = 1 ---------------------|     |
 (Power Steering on)                  | --- ELSE ---
                                      |
                                      | Do NOT add PSPPM


[*CSSMAF] Logic:

CSSFLG = 1 ---------------------------| Multiply DESMAF_PRE by CSSMAF
 (Cold Start spark in use)            |
                                      | --- ELSE ---
                                      |
                                      | Do NOT Multiply by CSSMAF
```

NOTE: DESMAF_PRE is a non-displayable parameter.

ISFLAG/ISLAST LOGIC

ISLAST reflects the state of ISFLAG on the last program pass. ISFLAG is set according to the following chart:

	AUTO IN DRIVE (DNDSUP = 1)	MANUAL OR AUTO IN NEUTRAL (DNDSUP = 0)
A/C Off	0	3
A/C Panel SW on or LOACT <= MINACT	1	4
A/C On	2	5

PSFLAG Logic: (Last state of POWSFG)

```
PSPSHP = 0 ----------------------------| Bypass PSFLAG logic
 (No pressure switch)                  |
                                       | --- ELSE ---
POWSFG = 0 ----------------------|     |
 (PS is off)                     |     |
                                 |AND -| PSFLAG = 0
PSFLAG = 1 ----------------------|     | IBGPSI = 0
                                       |
                                       | --- ELSE ---
PSFLAG = 0 ----------------------|     |
 (PS is Off->On transition)      |     |
                                 |AND -| PSFLAG = 1
POWSFG = 1 ----------------------|     | IBGPSI = 0
                                       |
                                       | --- ELSE ---
                                       |
                                       | No Change to PSFLAG
```

HCAMFG Logic: (Enable/Disable Adaptive airflow updates)

```
VRUN_ISCFLG = 1 -----------|
 (In running VIP)          |
                           |AND -|
RVIPRPM <> NUBASE ---------|     |
                                 |
VRUN_ISCFLG = 0 -----------|     |
 (In Normal Strategy)      |     |
                           |AND -|OR --| (Disable Adaptive airflow
Hicam <> 0 ----------------|     |     |  update)
 (See Hicam Equation)            |     | HCAMFG = 1
                                 |     |
DSDRPM < DESNLO -----------------|     |
 (DSDRPM decreasing)             |     |
                                 |     |
PSPSHP = 1 ----------------|     |     |
 (PSPS Switch present)     |     |     |
                           |AND -|     |
POWSFG = 1 ----------------|           |
 (PS is on)                            |
                                       | --- ELSE ---
                                       |
                                       | (Allow Adaptive airflow
                                       |  update)
                                       | HCAMFG = 0
```

RPM ERROR CALCULATION

OVERVIEW

Two separate RPM error calculations are executed in RPMERR_CALC. An instantaneous value (RPMERR) is calculated as the difference between the desired and actual RPM -- positive value of RPMERR is an engine speed below desired, a negative value indicates engine speed above desired.

The instantaneous value of RPMERR is then filtered using time constants TCBPA (for bypass air) and TCFBS (for feedback spark). The two filtered values. RPMERR_A and RPMERR_S are used in bypass air RPM control and feedback spark, respectively.

In addition, RPMERR_CALC contains the set logic for RUNUP_FLG, which is used to disable IPSIBR updates, idle fuel modulation and feedback spark during the initial runup. The flag is cleared in CRANK/UNDERSPEED/RUN mode selection.

DEFINITIONS

INPUTS

Registers:

- DSDRPM = Desired engine speed. See overview section for definition of the various uses of this register.

- ECTCNT = Number of times the ECT sensor input was read.

- ISCFLG = ISC mode indicator flag; -1 -> Dashpot mode, 0 -> Dashpot Preposition Mode, 1 -> Closed Loop RPM Control Mode, 2 -> Closed Loop RPM Control (Lock-out entry to RPM control).

- N = Engine RPM.

- RPMERR_A = Filtered rpm error for bypass air calculations.

- RPMERR_S = Filtered rpm error for feedback spark.

Bit Flags:

- ALT_CAL_FLG = Flag to indicate use of alternate calibration.

- RUNUP_FLG = Flag indicating initial runup is complete; 1 -> runup complete.

- VRUN_ISCFLG = RVIP idle speed control flag.

Calibration Constants:

- RUNUP_DIFF = RPM difference from DSDRPM to set RUNUP_FLG = 1.

- RUNUP_DIFF_A = Pre-delivery RPM difference from DSDRPM to set RUNUP_FLG = 1.

- TCBPA = Time constant for RPMERR_A.
- TCFBS = Time constant for RPMERR_S.
- V_RUNUP_DIFF = RPM difference from DSDRPM to set RUNUP_FLG = 1 when in VIP.

OUTPUTS

Registers:

- RPMERR = Instantaneous rpm error (DSDRPM - N).
- RPMERR_A = See above.
- RPMERR_S = See above.

Bit Flags:

- RUNUP_FLG = See above.

PROCESS

STRATEGY MODULE: ISC_RPMERR_COM3

```
always -------------------------| RPMERR = DSDRPM - N


ISCFLG > 0 ---------------------| RPMERR_A = ROLAV(RPMERR,TCBPA)
 (RPM control or lockout)       |  (calculate RPM error for airflow control)
                                | RPMERR_S = ROLAV(RPMERR,TCFBS)
                                |  (calculate RPM error for spark control)
                                |
                                | --- ELSE ---
                                |
                                | RPMERR_A = RPMERR
                                | RPMERR_S = RPMERR


VRUN_ISCFLG = 1 ----------------| pre_runup = V_RUNUP_DIFF
                                |
                                | --- ELSE ---
                                |
ALT_CAL_FLG = 1 ----------------| pre_runup = RUNUP_DIFF_A
                                |
                                | --- ELSE ---
                                |
                                | pre_runup = RUNUP_DIFF


ECTCNT >= 8 --------------|
 (TCSTRT OK to use)       |
                          |
N > DSDRPM + pre_runup ---|AND -| RUNUP_FLG = 1
 (runup RPM exceeded)     |     |  (initial runup complete)
                          |     |
RUNUP_FLG = 0 ------------|     |
 (first time this start)        |
                                | --- ELSE ---
                                |
                                | No change to RUNUP_FLG
```

DASHPOT CALCULATIONS

OVERVIEW

Dashpot Pre-position

Logic controlling the dashpot pre-position airflow is intended to increase the ISC duty cycle during part/WOT operation. Strategy determines the rate at which ISC valve flow increases/decreases in part/WOT operation as well as the maximum allowed pre-position airflow. Adequate pre-position airflow (DASPOT) is essential prior to entering the dashpot control mode in order to avoid HC spiking and/or deceleration stalls. The calculated pre-position airflow increment is added to an adaptively corrected idle flow requirement (DESMAF) prior to output of the ISC duty cycle. Pre-position airflow (DASPOT) is a function of the difference between a filtered throttle position (DSTPBR) and the lowest recorded throttle position (RATCH).

DSTPBR is a time dependent rolling average filter of Throttle position. It is updated once per background loop while in RUN or Underspeed Mode. The two time constants, TCDASU and TCDASD are calibratable. TCDASU is used when DSTPBR is filtering UP to TP. TCDASD is used to filter DSTPBR DOWN to TP.

The DASPOT value is adjusted as TP changes to provide the desired dashpot action to decelerations as initiated over the range of possible engine operating conditions, using separate time constants (TCDASU/TCDASD) to control the response of DSTPBR.

Dashpot Bleed

During Closed Throttle Mode, the DASPOT airflow is "bled off" by decrementing it. This action smooths the transition into RPM control by gradually eliminating the DASHPOT contribution to the Idle airflow, DESMAF. The bleed rate is determined by FN879. This allows a more aggressive daspot calibration to eliminate clunk in gear without affecting neutral.

DEFINITIONS

INPUTS

Registers:

- APT = Throttle Mode Flag; -1 -> Closed Throttle, 0 -> Part Throttle, 1 -> Wide Open Throttle.

- DSTPBR = Time dependent rolling average filter of throttle position. Filtered using TCDASU when filtering UP to TP, and TCDASD when filtering DOWN to TP.

- ISCFLG = ISC mode indicator flag; -1 -> Dashpot Mode; 0 -> Dashpot Preposition Mode; 1 -> Closed Loop RPM Control Mode; 2 -> Closed Loop RPM Control (Lock-out entry to RPM control)

- N = Engine revolutions, RPM.

- N_RATCH = RPM value which only ratchets down. When not at closed throttle, N = N_RATCH. When at closed throttle, N_RATCH is only allowed to go down. N_RATCH is an input to the minimum daspot clip. N_RATCH ratchets down to prevent rpm flares after a declutch.

- RATCH = Closed throttle position, counts

- TP = Throttle position sensor.

- VSBAR = Vehicle speed, MPH.

Bit Flags:

- FLG_DASMNQ = VSBAR flip-flop flag for minimum DASPOT clip.

- VRUN_ISCFLG = RVIP Idle Speed Control flag; 1 -> running VIP ISC action, 0 -> normal ISC action.

Calibration Constants:

- DASPTK = Gain associated with the desired DASPOT airflow. To calibrate this value first determine the throttle position above RATCH at which maximum DASPOT airflow is desired. Subtract DASPTO from DASMAX and divide the result by the throttle delta between RATCH and this maximum dashpot airflow to determine the DASPTK value.

- DASPTO = An offset term applied to the DASPOT calculation. Insures at least some dashpot airflow on rapid tip-in/tip-outs.

- DELHYS = Closed to part throttle hysteresis in TP counts. DELHYS should be set equal to DELTA + HYSTS (closed throttle breakpoint). This starts the dashpot calculation relative to the C.T./P.T. breakpoint to prevent changes in dashpot when leaving closed throttle.

- DPNEU_MUL = Neutral daspot multiplier, unitless.

- FN879 = A background driven decrement to the dashpot preposition airflow register (DASPOT) as a function of DASPOT. FN879 can be calibrated to achieve an exponentially decaying dashpot which is useful in decaying the large DASPOT values used to control over-rich tip out conditions.

- FN882A(N - DSDRPM) = Maximum dashpot clip as a function of the RPM delta above desired rpm. (N - DSDRPM) is clipped to 0 as a minimum.

- FN891(VSBAR) = Dashpot maximum clip as a function of vehicle speed. Used in automatic transmission vehicles at higher vehicle speeds to prevent harsh backout shifts by limiting large values of dashpot.

- FN894(N_RATCH - DSDRPM) = Minimum DASPOT airflow clip for 1st, 2nd, and 3rd gears, ppm. For manual transmissions, FN894 can be used to prevent declutch stalls. The input is N_RATCH which only ratchets down after the declutch. This is done to prevent the rpm from hanging up if it flares after the declutch.

- TCDASD = Time constant used when TP is less than or equal to the filtered TP value. Should be calibrated such that part throttle backouts where closed throttle is not entered do not exhibit a run-on feel. Too fast a filter can have the effect of greatly reducing dashpot airflow prior to entry into dashpot control.

- TCDASU = Filter constant used when TP is greater than the filtered TP value (DSTPBR). The larger the time constant the more slowly pre-position airflow will be available to respond to tip in/tip out actions. Fast response can also be obtained by use of the offset value DASPTO without the potential runaway feel that may come with too fast a filter constant/airflow gain (DASPTK) combination.

- V_879_MULT = VIP multipler for DASPOT function FN879.

OUTPUTS

Registers:

- DASPOT = Dashpot contribution to idle air flow. Used to provide a preposition air flow in Part Throttle and Wide Open Throttle Modes, which is "bled off" after a transition to Closed Throttle. This gradual air decrement allows a smooth transition to RPM control.

- DSTPBR = See above.

CALIBRATION INFORMATION

Typical values are provided for the following calibration constants:

- DASPTK = 0.002 ppm/TP count
- DASPTO = 0.10 ppm
- FN879 = (0,0.001) (0.1,0.002) (0.3,0.006) (0.75,0.05) (2.00,0.10)
- TCDASD = 0.75 sec.
- TCDASU = 3.3 sec.

PROCESS

STRATEGY MODULE: ISC_DASPOT_COM1

DASHPOT-PREPOSITION MODE (APT >= 0)
(Mode select logic will set ISCFLG to 0)

While at PT or WOT, airflow is added to the DASPOT term to prepare for a deceleration:

DASPOT = DASPTK * (DSTPBR - (RATCH + DELHYS)) + DASPTO

Where, DSTPBR = ROLAV(TP,time constant)

```
time constant = TCDASU for increasing TP (TP > old DSTPBR)
              = TCDASD for decreasing TP (TP <= old DSTPBR)
```

Clips:

- DASPOT is clipped to the smaller of FN882A(N - DSDRPM)*FN891(VSBAR) or 2.99 as a maximum; clip (N - DSDRPM) to zero as a minimum.

- minimum clip may be in effect.

FN894 minimum DASPOT clip logic

```
FLG_DASMNQ = 1 ---------------------|
 (speed above threshold)            |
                                    |
DASPOT < FN894(N_RATCH - DSDRPM) ---|AND -| Clip DASPOT to
 (airflow too low)                  |     |  FN894(N_RATCH - DSDRPM) *
                                    |     |  DPNEU_MUL as a minimum
DNDSUP = 0 -------------------------|     |
                                          | --- ELSE ---
FLG_DASMNQ = 1 ---------------------|     |
 (speed above threshold)            |     |
                                    |     |
DASPOT < FN894(N_RATCH - DSDRPM) ---|AND -| Clip DASPOT to
 (airflow too low)                  |     |  FN894(N_RATCH - DSDRPM)
                                    |     |  as a minimum
DNDSUP = 1 -------------------------|     |
                                          | --- ELSE ---
                                          |
                                          | Do not clip to FN894
```

DASHPOT MODE (APT = -1)
(Mode select logic will set ISCFLG to -1)

At closed throttle, the airflow which was previously added to the DASPOT term is bled away. DASPOT is calculated from the equation below either during non-VIP vehicle operation, or in running VIP when VRUN_ISCFLG = 0.

DASPOT = DASPOT - FN879(DASPOT)

clips:
- no maximum clip
- DASPOT is clipped to either 0, FN894 * DPNEU_MUL, or FN894 as a minimum (see FN894 on the previous page)

DASPOT is calculated from the equation below
when in running VIP, and VRUN_ISCFLG = 1:

DASPOT = DASPOT - (FN879(DASPOT) * V_879_MULT)

clips:
- no maximum clip
- DASPOT is clipped to either 0, FN894 * DPNEU_MUL, or FN894 as a minimum (see FN894 on the previous page)

MODE SELECT (MODE_SELECT)

OVERVIEW

Bypass idle speed control has four modes of operation; dashpot, dashpot preposition, RPM control, and RPM control lockout. A flag is used to identify these modes for both calibrator convenience and required interaction with fuel modulation and spark feedback strategies.

The mode select logic selects the mode of operation and sets a flag (ISCFLG) which is used to adjust the total desired airflow through the air bypass valve.

* ISCFLG = -1 DASHPOT CONTROL
* ISCFLG = 0 DASHPOT PRE-POSITION
* ISCFLG = 1 CLOSED LOOP RPM CONTROL
* ISCFLG = 2 CLOSED LOOP RPM CONTROL (Lock-out entry to RPM control)

- DASHPOT PRE-POSITION MODE (ISCFLG = 0)

  In engine run/underspeed mode and when operating at part or wide open throttle the ISC system is placed in dashpot pre-position mode. In this mode the ISC duty cycle is incremented a calibratable amount in anticipation of a required dashpot action. Proper dashpot operation is essential on systems having speed density fuel controls in order to avoid tip in/tip out stalls and HC spiking on decels.

- DASHPOT MODE (ISCFLG = -1)

  In engine run/underspeed mode and having just transitioned from part to closed throttle the system is placed in ISC dashpot control mode. The length of time the ISC system will remain in dashpot control is both hardware/strategy dependent (some applications have VSS; some manual transmission applications have gear and clutch switches). Regardless of the length of time required to enter RPM control, as long as closed throttle operation is maintained the amount of airflow specified by the dashpot pre-position (see dashpot pre- position logic) is decremented at a constant rate until exhausted (until DASPOT = 0).

- CLOSED LOOP RPM CONTROL (ISCFLG = 1 OR 2)

  For normal entry into C/L RPM control the following conditions must be satisfied:

  . If VSS hardware is used it must indicate a speed less than MINMPH

  . If a manual trans. with gear/clutch switches; must indicate neutral

* Note: Although the system can provide acceptable function without the above mentioned hardware either item will increase reliability in production. The vehicle speed sensor has calibration benefits outside of ISC (lean cruise control, etc.) and should be considered when specifying system assumptions for future applications utilizing ISC.

. Regardless whether the above hardware is used, normal entry into RPM control requires that actual engine speed be less than or equal to (DSDRPM + RPMCTL) and that closed throttle is indicated.

The following discussion will attempt to describe entry into C/L RPM control through the lock-out logic (ISCFLG = 2). In a normal deceleration the dashpot bleed time will be short relative to the vehicle coastdown time. As soon as engine speed drops low enough the ISC system should enter RPM control. However, due to hysteresis in the bypass valve, overspecification of idle airflow requirements prior to adaptive ISC learning, ISC learning in an unusually high state of engine load (400 psi A/C head pressure), etc. the ISC actuator may flow too much air at the specified idle duty cycle to allow normal entry in RPM control. When this condition occurs the system will remain in dashpot control until it can recognize that it should in fact be in RPM control.

Obviously this task is easy if you happen to have a VSS or have a manual calibration with gear/clutch switches. The problem without this hardware is to differentiate between a deceleration condition (especially a constant rate of speed deceleration -- as in a coast down a mountain) and a true locked-out-of-idle condition. Most of the logic in the above-mentioned attachment deals with this lock-out feature.

To differentiate between deceleration and idle the rate of change in RPM is first evaluated over a calibrated period of time (ISCTM). If the speed has remained within a specified deadband (NDIF) for this time period a second check is performed to compare MAP with a calibrated value (FN862(BP) for A/C off; FN862(BP) + ACMAP for A/C on). The assumption is that all idle MAP values, including green engine/altitude effects etc., will be greater than this value and all true deceleration conditions, including the same variabilities, will yield lower MAP. It goes without saying that great care must be taken in selecting the correct calibration for FN862(BP).

If the ISC system were locked in dashpot control and both the rate of engine speed change and MAP criteria were satisfied the strategy would be forced into C/L RPM control with ISCFLG indicating 2. This state would be present until the speed fell below the normal entry point. The adaptive ISC would learn the required correction, assuming sufficient time at idle, and subsequent dashpot to RPM control transitions should follow a normal entry path.

DEFINITIONS

INPUTS

Registers:

- APT = Throttle Mode flag.

- BP = Barometric Pressure.

- DASCTL = Value of DASPOT, below which RPM control can begin, with any remaining DASPOT airflow being rolled into IPSIBR.

- DASPOT = Dashpot contribution to idle air flow. Used to provide a preposition air flow in Part Throttle and Wide Open Throttle Modes, which is "bled off" after a transition to Closed Throttle. This gradual air decrement allows a smooth transition to RPM control.

- DSDRPM = Desired engine speed.

- ISCFLG = ISC mode indicator flag; -1 -> Dashpot Mode, 0 -> Dashpot Preposition Mode, 1 -> Closed Loop RPM Control Mode, 2 -> Closed Loop RPM Control (Lock-out entry to RPM control).

- ISCTMR = RPM sample timer for lockout logic, secs. Timer is cleared on each RPM sample.

- MAP = Manifold Absolute Pressure.

- N = Engine speed, RPM.

- NLAST = Last sampled RPM for lockout logic. NLAST is re-calculated when ISCTMR exceeds ISCTM.

- SETTMR = RPM control entry delay timer, secs. Used to delay entry into RPM control until manifold stabilizes.

- VSBAR = Filtered vehicle speed, mph.

Bit Flags:

- ACCFLG = A/C engaged flag; 1 -> A/C engaged, 0 -> A/C disengaged.

- DNDSUP = Delayed neutral/drive flag; 1 -> in drive, loaded.

- VRUN_ISCFLG = RVIP Idle Speed Control flag; 1 -> Running VIP ISC action, 0 -> normal ISC action.

Calibration Constants:

- ACMAP = An adder to FN862(BP) when A/C is on. Should be based on observed differences between A/C on & A/C off idle MAP readings.

- FN862(BP) = Decel MAP value as a function of BP. This value takes the place of LOWMAP and is used to vary decel MAP as a function of altitude.

- ISCTM = Time interval over which the rate of change in engine speed is evaluated. Value should be small enough to avoid prolonged speed hang-ups if the ISC system were locked out of C/L speed control but not too short such that the rate of speed change check becomes meaningless.

- MINMPH = Minimum speed to enter C/L RPM control. Applies to systems having VSS. Should be set below the speed at which an automatic trans. vehicle rolls along in drive without the brakes. This is to prevent going into RPM control during parking lot maneuvers.

- NDIF = The deviation in engine speed allowed over the ISCTM specified time interval. Values too small could lock the ISC system out of C/L speed control indefinitely. Values too large invalidate the check.

- RPMCTL = Added to DSDRPM. The total defines the engine speed threshold below which entry into C/L RPM control is allowed. This value should be reasonably small to avoid inadvertent entry into C/L ISC.

- SETLNG_TM = Manifold stabilization time. Used to delay entry into RPM control.

- TRLOAD = Transmission Load switch;
  - 0 -> Manual Transmission, no clutch or gear switches, forced neutral state
  - 1 -> Manual Transmission, no clutch or gear switch.
  - 2 -> Manual Transmission, one clutch or gear switch.
  - 3 -> Manual Transmission, both clutch and gear switches.
  - 4 -> Auto Transmission, non-electronic, neutral drive switch.
  - 5 -> Auto Transmission, non-electronic, neutral pressure switch, (AXOD).
  - 6 -> Auto Transmission, electronic, PRNDL sensor - park, reverse, neutral, overdrive, manual 1, manual 2.

- V_SETLNG_TM = VIP delay to enter ISC rpm control.

OUTPUTS

Registers:

- ISCFLG = See above.

- ISCTMR = See above.

- NLAST = See above.

CALIBRATION INFORMATION

Typical values are supplied for the following calibration constants:

- ACMAP = 2 " Hg (engine specific parameter).
- FN862(BP) = (0,6.5) (19.5,6.5) (31.875,8.5)
- ISCTM = 4 sec.
- MINMPH = 3 MPH
- NDIF = 32 RPM
- RPMCTL = 90 RPM

PROCESS

STRATEGY MODULE: ISC_MODE_SELECT_COM1

```
                                         | (In Dashpot Preposition mode)
APT >= 0 --------------------------------| ISCFLG = 0
 (Not closed throttle)                   | NLAST = N
                                         | ISCTMR = 0
                                         |
                                         | --- ELSE ---
VSBAR <= MINMPH ------------|            |
 (Vehicle stopped)          |            | (In RPM Control Mode)
                            |            |
DASPOT <= DASCTL -----------|            |
 (Dashpot complete)         |            |
                            |            |
TRLOAD <> 3 ----------|     |            |
 (Not man trans       |     |            |
  w/both switches)    |OR --|AND -------| ISCFLG = 1
                      |     |            |
DNDSUP = 0 -----------|     |            |
 (Neutral)                  |            |
                            |            |
SETTMR > SETLNG_TM ---------|            |
 (Manifold stable)          |            |
                            |            |
N <= (DSDRPM + RPMCTL) -----|            |
                                         |
                                         | --- ELSE ---
VSBAR <= MINMPH ------------------|      |
 (Vehicle stopped)                |      | (In RPM Control Lockout mode;
                                  |      | same action as RPM Control mode)
DASPOT <= DASCTL -----------------|      |
 (Dashpot complete)               |      |
                                  |      |
TRLOAD <> 3 ----------------|     |      |
 (Not manual transmission   |     |      |
  w/ both switches)         |OR --|AND -| ISCFLG = 2
                            |     |      |
DNDSUP = 0 -----------------|     |      |
 (Neutral)                        |      |
                                  |      |
SETTMR > SETLNG_TM ---------------|      | --- ELSE ---
 (Manifold stable)                |      |
                                  |      | (In Dashpot mode)
LOCKOUT LOGIC TRUE ---------------|      |
 (Locked out of RPM control)             | ISCFLG = -1
```

NOTE: SETTMR and ISCTMR are cleared in MODE_SELECT logic. See TIMERS chapter for complete timer logic.

LOCKOUT LOGIC

```
ISCFLG >= 1 -------------------|
                               |
                               |OR ---| LOCKOUT LOGIC TRUE
VRUN_ISCFLG = 1 --------|      |      |  (Stay in lockout until RPM falls)
 (VIP modifying ISC)    |AND --|      | EXIT LOCKOUT LOGIC
                        |             |
SETTMR > V_SETLNG_TM ---|             |
 (manifold stable)                    |
                                      | --- ELSE ---
                                      |
ISCTMR < ISCTM -----------------------| LOCKOUT LOGIC FALSE
                                      |  (Not time to sample RPM yet)
                                      | EXIT LOCKOUT LOGIC
                                      |
                                      | --- ELSE ---
                                      |
|N - NLAST| > NDIF -------------------| LOCKOUT LOGIC FALSE
                                      |  (RPM changing quickly, must be
decel)
                                      | NLAST = N
                                      | ISCTMR = 0
                                      | EXIT LOCKOUT LOGIC
                                      |
TRLOAD >= 3 --------------------|     | --- ELSE ---
 (Auto trans)                   |     |
                                |AND -| LOCKOUT LOGIC TRUE
DNDSUP = 0 ---------------------|     | (Can't be decel when in neutral)
 (In neutral)                         | EXIT LOCKOUT LOGIC
                                      |
ACCFLG = 1 ---------------|           | --- ELSE ---
 (A/C on)                 |AND -|     |
MAP < FN862(BP) + ACMAP --|     |OR --| LOCKOUT LOGIC FALSE
 (Decel MAP)                    |     |  (MAP indicates decel)
                                |     | EXIT LOCKOUT LOGIC
ACCFLG = 0 ---------------|     |     |
 (A/C off)                |AND -|     | --- ELSE ---
MAP < FN862(BP) ----------|           |
 (Decel MAP)                          | LOCKOUT LOGIC TRUE
                                      | EXIT LOCKOUT LOGIC
```

KAM UPDATE (ISCKAM_UPDATE)

OVERVIEW

This section describes the adaptive ISC update routine. In general, under steady state conditions on a stabilized engine at idle, the adaptive ISC logic will evaluate whether the open loop prediction of airflow requires correction. If a correction factor was applied, IPSIBR has a non-zero value, the adaptive ISC strategy will roll this correction value into KAM and drive the IPSIBR term back to zero. Control of the rate at which the IPSIBR value is driven to zero is calibration-dependent.

There are six ISCKAM cells designated for idle corrections. The appropriate cell is pointed to by the flag ISFLAG which tracks the load state at idle. The following logic must be satisfied to update KAM:

* In RPM control
* Within the RPM deadband for a calibrated time interval (UPDISC)
* No hi-cam adder present (HCAMFG = 0)
* IPSIBR non zero
* No kam errors
* IBGPSI >= UPDATM

ISCKAM corrections are clipped to the same maximum and minimum limits as the C/L RPM integrator (PSIBRM/PSIBRN). Each time the update criteria are satisfied both IPSIBR and ISCKAM are adjusted one bit (0.00024 ppm) in opposite directions until IPSIBR = 0.

DEFINITIONS

INPUTS

Registers:

- IBGPSI = Background loop counter, used to pace ISCKAMn update.

- ISCKAMn = Adaptive ISC correction for each load condition n, where n is the value of ISFLAG. The calculated value of ISCKAMn is added to the total desired idle air flow (DESMAF).

- ISCTMR = RPM sample timer for adaptive ISC, secs. Timer is cleared if |RPMERR_A| exceeds the rpm deadband, RPMDED.

- IPSIBR = The closed loop integration component of total DESMAF, ppm. Designed to provide integral feedback, IPSIBR adjusts the value of DESMAF to correct for sustained changes in idle load. An increase or decrease in IPSIBR results in a corresponding change to bypass valve duty cycle.

Bit Flags:

- HCAMFG = Flag indicating the completion of Hi-Cam; 0 -> no desired engine speed adder exists, 1 -> an rpm adder above base idle is present. Flag is used in the ISC adaptive update routine to disable updates when HCAMFG = 1.

- ISCFLG = ISC mode indicator flag; -1 -> Dashpot Mode; 0 -> Dashpot Preposition Mode; 1 -> Closed Loop RPM Control Mode; 2 -> Closed Loop RPM Control (Lock-out entry to RPM control)

- ISFLAG = Idle load state indicator used to select the ISCKAM cell.

- KAM_ERROR = KAM error flag; 1 -> KAM data invalid.

Calibration Constants:

- PSIBRM = Maximum allowed value for ISCKAMn. ISCKAMn is clipped to this value.

- PSIBRN = Minimum allowed value for ISCKAMn. ISCKAMn is clipped to this value.

- RPMDED = Adaptive ISC learning deadband. Learning is disabled if RPMERR_A exceeds this deadband.

- UPDATM = Pacing at which the IPSIBR correction factor is rolled into KAM. Value is in terms of background loop counts.

- UPDISC = Time that engine speed must be within the specified deadband (RPMDED) prior to KAM update.

OUTPUTS

Registers:

- IBGPSI = See above.

- ISCKAMn = See above.

- ISCTMR = See above.

- ISKSUM = CHECKSUM for adaptive idle speed KAM cells, used in KAM initialization strategy.

- IPSIBR = See above.

CALIBRATION INFORMATION

The following typical values are provided for calibration constants:

- RPMDED = 50 rpm.
- UPDATM = 5 background passes.
- UPDISC = 2 seconds.

PROCESS

STRATEGY MODULE: ISC_ISCKAM_COM1

```
always ---------------------------------| IBGPSI = IBGPSI + 1
                                        | Clip to Maximum


ISCFLG <> 1 ----------------------------| IBGPSI = 0
                                        | Exit ISCKAM_UPDATE logic


|RPMERR_A| > RPMDED --------------------| ISCTMR = 0


ISCTMR < UPDISC ------------------|
                                  |
HCAMFG = 1 -----------------------|
                                  |OR --| IBGPSI = 0
IPSIBR = 0 -----------------------|     | Exit ISCKAM_UPDATE logic
                                  |
KAM_ERROR = 1 --------------------|


IBGPSI < UPDATM ------------------------| Exit ISCKAM_UPDATE logic
                                        |
                                        | --- ELSE ---
                                        |
                                        | IBGPSI = 0


IPSIBR > 0 -----------------------|
                                  |AND -| Increment ISCKAMn
ISCKAMn < PSIBRM -----------------|     | Increment ISKSUM
                                        | Decrement IPSIBR
                                        |
                                        | --- ELSE ---
IPSIBR <= 0 ----------------------|     |
                                  |AND -| Decrement ISCKAMn
ISCKAMn > PSIBRN -----------------|     | Decrement ISKSUM
                                        | Increment IPSIBR
```

DUTY CYCLE CALCULATION

OVERVIEW

The ISC duty cycle is calculated in ISCDTY_CALC. The mass air flow through the ISC actuator (DEBYMA) is calculated as the desired mass air flow at idle (DESMAF) less the flow through the throttle plate etc, corrected for altitude.

The desired duty cycle is calculated as follows:

Once the desired mass flow value is finalized, the appropriate duty cycle is calculated and output. The final DESMAF value is calculated in IPSIBR_CALC.

The calibrated leakage term (ITHBMA) is subtracted from DESMAF to obtain the actual flow required from the ISC actuator (DEBYMA). If BPCOR_SW is set, DEBYMA is adjusted for altitude. This value, clipped at DEBYCP as a minimum allowed actuator airflow, becomes the input to the ISC duty cycle transfer function (FN800). Output from FN800 is the specified ISC duty cycle. The nature of the bypass air solenoid is such that at high manifold vacuum the device flows less air than at idle vacuum levels assuming a constant duty cycle. To account for this a modulator (FN820A) is available to increase the duty cycle as necessary to hold constant flow.

The final value of ISCDTY includes an offset and a multiplier (IDCOFS and IDCMUL) used primarily as calibration tools.

DEFINITIONS

INPUTS

Registers:

- BP = Barometric pressure, " Hg.

- DEBYMA = Desired mass air flow through the ISC valve, ppm. This quantity is calculated as the total desired idle air flow (DESMAF) less the air flow through the throttle plate, and is corrected for altitide using FN890(BP).

- DESMAF = Total desired idle air flow, ppm. Calculated as the sum of predicted air flow (DESMAF_PRE), dashpot air flow (DASPOT), integral air flow (IPSIBR) and KAM correction (ISCKAMn).

- VACUUM = Intake manifold vacuum.

Bit Flags:

- VRUN_ISCFLG = RVIP Idle Speed Control flag; 1 -> Running VIP ISC action, 0 -> normal ISC action.

Calibration Constants:

- BPCOR_SW = Calibration switch for BP correction to DEBYMA; 0 -> no correction, 1 -> use correction.

- DEBYCP = Minimum allowed airflow through the ISC actuator. This is a clip on DEBYMA.

- FN800(DEBYMA) = Transfer function for the ISC actuator. Initial values for this function should come directly from flow data provided by fuel systems. Data must be generated at the expected idle vacuum setting for each particular application. The best way to get actual vehicle data is to connect a duty cycle box to the ISC actuator, vary the duty cycle and plot actual airflow using a hot wire type air meter. ..Typical values vary based on engine application and flow capacity of the ISC actuator.

- FN820A(VACUUM) = ISC duty cycle multiplier versus VACUUM. Used to hold constant actuator airflow on a decel after dashpot action is complete.

- FN890(BP) = ISC duty cycle altitude compensation subtractor. Required to offset the effect with altitude of a varying pressure drop across the ISC actuator, which affects its air flow.

- IDCMUL = ISCDTY multiplier, no units.

- IDCOFS = ISCDTY adder, fraction of fully open.

- ITHBMA = Throttle body idle mass air flow with throttle plate at idle screw stop and 0% ISC duty cycle. This is any airflow which does not go through the bypass air solenoid, i.e. throttle plate, PCV system, intake leakage, etc.

- V820A = ISC duty cycle multiplier, used for VIP only.

OUTPUTS

Registers:

- DEBYMA = See above.
- ISCDTY = Idle speed control valve duty cycle, fraction of fully open. Calculated as a transfer function (FN800) of the desired mass flow through the ISC valve.

PROCESS

STRATEGY MODULE: ISC_ISCDTY_COM1

DEBYMA AND ISCDTY CALCULATION

```
ISCDTY = IDCMUL * FN800(DEBYMA) * FN820A(VACUUM) + IDCOFS
```

where:

```
         /                 \
DEBYMA = ( DESMAF - ITHBMA ) * [29.92/BP] - FN890(BP)
         \                 /
```

BP correction logic:

```
BPCOR_SW = 1 -----------------| Use 29.92/BP in DEBYMA calculation
                              |
                              | --- ELSE ---
                              |
                              | Do not use 29.92/BP
```

- DEBYMA is clipped to DEBYCP as a minimum.
- ISCDTY is clipped to 1.0 as a maximum.

NOTE: During certain times, Running VIP may need to control ISC. When this is the case, Running VIP will set VRUN_ISCFLG = 1. This causes V820A to be used in the ISCDTY equation above, instead of FN820A.

IPSIBR CALCULATION (IPSIBR_CALC)

OVERVIEW

IPSIBR is the closed loop integration component of total DESMAF. Designed to provide integral feedback, IPSIBR adjusts the value of DESMAF to correct for sustained changes in idle load. An increase or decrease in IPSIBR results in a corresponding change to bypass valve duty cycle.

The IPSIBR calculation has the following characteristics:

- IPSIBR is only updated in RPM control or lockout mode.
- Different time constants are used depending on whether RPM is too high or too low.
- IPSIBR is not updated if the ISC valve is already at its maximum or minimum position (PSIBRM and PSIBRN, respectively).
- On load state transitions (A/C, N/DR and CT/PT), IPSIBR is always clipped to zero as a minimum.
- The IPSIBR update has no deadband. Instead, it is driven by an RPM error term RPMERR_A (A for "air"). The time constant is TCBPA.
- Calibration of gain is not required. The term (DESMAF_PRE / DSDRPM) in the ISCPSI calculation automatically adjusts the gain for RPM, temperature and accessory loads.
- The IPSIBR pacing calibration is controlled by the two scalars TC_OVER and TC_UNDER for speed higher and lower than desired, respectively. These scalars represent time constants for the engine to respond to changes in duty cycle.

DEFINITIONS

INPUTS

Registers:

- ATMR1 = Time since start (time since exiting crank mode).
- BG_TMR = Background loop time, secs.
- DASPOT = Dashpot contribution to idle air flow. Used to provide a preposition air flow in Part Throttle and Wide Open Throttle Modes, which is "bled off" after a transition to Closed Throttle. This gradual air decrement allows a smooth transition to RPM control.

- DEBYMA = Desired mass air flow through the ISC valve, ppm. This quantity is calculated as the total desired idle air flow (DESMAF) less the air flow through the throttle plate and is corrected for altitude using FN890(BP).

- DSDRPM = Desired engine speed.

- ECT = Engine Coolant Temperature.

- ISCDTY = Idle speed control valve duty cycle, fraction of fully open. Calculated as a transfer function (FN800) of the desired mass flow through the ISC valve.

- ISCFLG = ISC mode indicator flag: -1 -> Dashpot Mode; 0 -> Dashpot Preposition Mode; 1 -> Closed Loop RPM Control Mode; 2 -> Closed Loop RPM Control (Lock-out entry to RPM control).

- ISCPSI = The quantity of air which is added to IPSIBR each background pass.

- ISLAST = Register which tracks the state of engine load from the previous background pass.

- IPSIBR = The closed loop integration component of total DESMAF. Designed to provide integral feedback, IPSIBR adjusts the value of DESMAF to correct for sustained changes in idle load. An increase or decrease in IPSIBR results in a corresponding change to bypass valve duty cycle.

- ISCKAM(ISFLAG) = Adaptive correction for each load condition.

- ISFLAG = Idle load indicator, set according to the current state of the A/C (on or off) and of the transmission (neutral or drive).

- KAM_ERROR = KAM error flag; 1 -> KAM invalid.

- RPMERR = Unfiltered RPM error, DSDRPM - N.

- RPMERR_A = Filtered RPMERR for airflow control, time constant TCBPA.

- RUNUPTMR = Time since runup RPM exceeded (0.125 sec).

Bit Flags:

- ACCFLG = A/C engaged flag; 1 = A/C engaged, 0 -> A/C disengaged.

- CTPTFG = Closed throttle to part / wide open throttle transition flag; 1 -> transition occurred.

- KAM_ERROR = KAM error flag; 1 -> KAM invalid.

- V_MODE_SETUP = VIP throttle adjust mode enabled flag; 1 -> enabled.

- VRUN_ISCFLG = Self test engine running flag; 1 -> Engine running self test in progress.

- RUNUP_FLG = Flag indicating the initial runup is complete; 1 -> Runup RPM exceeded.

Calibration Constants:

- ACHPTM = Calibrated time since start to recognize high AC head pressure.

- DEBYCP = Minimum value of ISC valve airflow, ppm.

- ECT_HP = ECT temperature where AC head pressure is capable of switching the A/C off.

- FN852(RPMERR) = Proportional control function used to modify DESMAF.

- IPSIDLY = Time delay to disable IPSIBR Update, sec.

- PSIBRM = Maximum allowed value for IPSIBR when in normal strategy.

- PSIBRN = Minimum allowed value for IPSIBR when in normal strategy.

- TC_OVER = Time constant used to control the integral gain (or pacing) of the term, IPSIBR. TC_OVER is used when RPMERR_A <= 0, i.e., when the actual speed is higher than desired. A large value of TC_OVER corresponds to a small integral gain, a small value corresponds to a high gain.

- TC_UNDER = Time constant used to control the integral gain (or pacing) of the term, IPSIBR. TC_UNDER is used when RPMERR_A > 0, i.e., when the actual speed is lower than desired. A large value of TC_UNDER corresponds to a small integral gain, a small value corresponds to a high gain.

- VSIBRM = Maximum allowed value for IPSIBR when in Running VIP.

- VSIBRN = Minimum allowed value for IPSIBR when in Running VIP.

- VTC_OVER = Time constant used to control the integral gain (or pacing) of the term, IPSIBR, when in Engine Running VIP. VTC_OVER is used when RPMERR_A <= 0, i.e. when the actual speed is higher than desired. Corresponds to TC_OVER in normal strategy.

- VTC_UNDER = Time constant used to control the integral gain (or pacing) of the term, IPSIBR, when in Engine Running VIP. VTC_UNDER is used when RPMERR_A > 0, i.e. when the actual speed is lower than desired. Corresponds to TC_UNDER in normal strategy.

Non-displayable Parameters:

- DESMAF_PRE = Predicted desired idle air flow, ppm. This is the open loop air flow prediction which is required to idle, calculated as a function of ECT and time since start and including A/C, power steering and heated windshield adders. It does not include any closed loop or KAM corrections.

OUTPUTS

Registers:

- DESMAF = Total desired idle air flow, ppm. Calculated as the sum of predicted air flow (DESMAF_PRE), dashpot air flow (DASPOT), integral air flow (IPSIBR) and KAM correction (ISCKAM).

- IBGPSI = Background counter used to control pacing of ISC KAM learning.

- IPSIBR = See above.

- ISCPSI = See above.

- ISKSUM = Checksum for adaptive idle speed KAM cells, used in KAM initialization strategy.

PROCESS

STRATEGY MODULE: ISC_IPSIBR_COM1

```
ISCFLG <= 0 ------------------------|
 (Dashpot or dashpot preposition)   |
                                    |OR --| (Do not update IPSIBR unless
RUNUPTMR < IPSIDLY -----------------|     |  in RPM Control or RPM Control
 (Delay IPSIBR update after start)        |  lockout)
                                          | No change to IPSIBR
                                          |
                                          | --- ELSE ---
RPMERR_A < 0 -----------------|           |
 (Overspeed error)            |           |
                              |AND -|     |
DEBYMA <= DEBYCP -------------|     |     | (Actuator cannot respond
 (Actuator airflow at minimum)      |     |  to a change in IPSIBR)
                                    |OR --| ISCPSI = 0
                                    |     | No change to IPSIBR
RPMERR_A > 0 -----------------|     |     |
 (Underspeed error)           |     |     |
                              |AND -|     |
ISCDTY >= 98% ----------------|           |
                                          | --- ELSE ---
                                          |
                                          | (OK to adjust IPSIBR)
                                          | IPSIBR = IPSIBR + ISCPSI
                                          | Clip IPSIBR to PSIBRN as a
                                          |  minimum and PSIBRM as a maximum
```

NOTE: In Engine Running VIP (VRUN_ISCFLG = 1), IPSIBR is clipped to VSIBRM as a maximum and VSIBRN as a minimum.

Where ISCPSI is calculated as follows:

Normal Conditions

```
ISCPSI = RPMERR_A * (DESMAF_PRE)/DSDRPM * BG_TMR /(TC_UNDER | TC_OVER)
```

DECISION LOGIC

```
RPMERR_A > 0 -----------------------------| Use TC_UNDER in ISCPSI
                                          |  calculation
                                          |
                                          | --- ELSE ---
                                          |
RPMERR_A <= 0 ----------------------------| Use TC_OVER in ISCPSI
                                          |  calculation
```

In Engine Running VIP (VRUN_ISCFLG = 1)

```
ISCPSI = RPMERR_A * (DESMAF_PRE)/DSDRPM * BG_TMR /(VTC_UNDER | VTC_OVER)
```

DECISION LOGIC

```
RPMERR_A > 0 ----------------------------| Use VTC_UNDER in ISCPSI
                                         |  calculation
                                         |
                                         | --- ELSE ---
                                         |
RPMERR_A <= 0 ---------------------------| Use VTC_OVER in ISCPSI
                                         |  calculation
```

Where: "|" means "or"

TOTAL DESIRED IDLE AIR FLOW (DESMAF) CALCULATION

```
ISFLAG <> ISLAST ------------------|
                                   |
CTPTFG = 1 ------------------|     |OR --| Clip IPSIBR at 0 as a minimum
 (Closed throttle to         |     |     |  (Reset IPSIBR for a new load)
  part/WOT transition)       |AND -|     | IBGPSI = 0
                             |           |  (Reset C/L correction pacer)
VRUN_ISCFLG = 0 -------------|
 (Not in running VIP)
```

Total DESMAF CALCULATION

```
KAM_ERROR = 1 ---------------------|
 (KAM qualify error)               |
                                   |
ISCKAM(ISFLAG) > PSIBRM -----------|OR --| Assume ISCKAM cells are invalid
 (greater than maximum)            |     | Reinitialize all ISCKAM cells:
                                   |     | All ISCKAM cells = 0
ISCKAM(ISFLAG) < PSIBRN -----------|     | ISKSUM = 0
 (less than minimum)


V_MODE_SETUP = 1 ------------------------| DESMAF = DESMAF_PRE
 (VIP Throttle Adjust Mode)              |
                                         | --- ELSE ---
ACCFLG = 1 ------------------------|     |
                                   |     |
ECT > ECT_HP ----------------------|     |
                                   |AND -| DESMAF = DESMAF_PRE + IPSIBR +
ATMR1 > ACHPTM --------------------|     |          DASPOT + ISCKAM(ISFLAG) +
                                   |     |          FN852
ISCFLG > 0 ------------------------|     |
                                         | --- ELSE ---
                                         |
                                         |  DESMAF = DESMAF_PRE + IPSIBR
                                         |      + DASPOT + ISCKAM(ISFLAG)
```

NOTE: If KAM_ERROR = 1 (KAM data invalid), the adaptive ISC cells (ISCKAM) are initialized to zero.

CHAPTER 10

7.0L GOVERNOR MAP SIGNAL

7.0L GOVERNOR MAP SIGNAL (GOVHP = 1)

OVERVIEW

The DPI output, HSO-0, is used on the 7.0L heavy duty truck application as an input to the stand alone governor. It is used to provide information about engine load, compensated for altitude, so that high governor gains can be used to achieve a quick response on an unloaded engine, while lower gains can be used on a loaded engine for normal throttle control. The value MAPPA is used as a load parameter compensated for altitude.

The signal is a 100 Hz pulse width modulated signal.

TP_REL is output to the Governor via the DOL. The strategy associated with this is located in the DOL chapter.

CALIBRATION PHILOSOPHY

Set GOVHP = 1 to activate the logic.

DEFINITIONS

INPUTS

Registers:

- BP = Barometric pressure, " Hg.
- MAP = Manifold Absolute Pressure, " Hg.
- MAPPA = MAP/BP.

Bit Flags:

- MFMFLG = MAP sensor failure; 1 -> failure.
- TFMFLG = Flag indicating TP sensor has failed; 1 -> failure.

Calibration Constants:

- FN501(MAPPA) = MAPPA to duty cycle ratio transfer function.
- GOVHP = 7.0L Governor hardware present switch; set GOVHP = 1 to activate this logic.

OUTPUTS

Registers:

- GMAPDC = Time of next high edge relative to master timer, ticks.

PROCESS

STRATEGY MODULE: ACC_GOV_MAP_COM2

(follows calculation of MAPPA)

```
GOVHP = 1 ----------------|
                          |AND -|
MFMFLG = 0 ---------------|     |
 (MAP ok)                       |
                                |OR --| GMAPDC = FN501(MAPPA)
GOVHP = 1 ----------------|     |     |  (GMAPDC is duty cycle)
                          |     |     |
MFMFLG = 1 ---------------|AND -|     |
 (MAP not ok)             |           |
                          |           |
TFMFLG = 0 ---------------|           |
 (TP ok)                              |
                                      | --- ELSE ---
                                      |
                                      | GMAPDC = 0
                                      |  (duty cycle is zero, no output)


GOVHP = 1 ----------------------------| Output duty cycle GMAPDC at
                                      |  100 Hz on HSO-0
```

NOTE:

- In LOS, the output will de-energize, giving no transitions. The governor will detect this state and act accordingly.

- The DPI strategy is disabled when GOVHP = 1.

- Should the MAP sensor fail, MAP is simulated from TP and output. Should the TP sensor also fail, no signal is output.

CHAPTER 11

CANISTER PURGE STRATEGY

CANISTER PURGE

OVERVIEW

Canister Purge refers to the solenoid and valve combination that is located in the line between the intake manifold and the carbon canister. When the solenoid is energized the valve opens, allowing the flow of vapors from the canister to the intake manifold.

The strategy enables canister purge during various engine operating modes. These modes are calibration items. Typical calibrations will enable purge when these conditions are met:

1. Fuel control is in the desired mode. The calibrator can choose between purging during closed loop only or during both open loop and closed loop.

2. The EGO sensor has been warm at least 1 time or ECT indicates that the engine is warm. Set PURECT to 254 to calibrate out ECT gate.

3. The not at closed throttle delay has been met.

4. The current value of the air mass modulator function, FN605A, is non-zero.

The strategy includes features to prevent the rich surge which may occur on initial purge turn-on and to prevent purge vapors from driving the Closed Loop Fuel Control beyond its control limit. When purge is enabled, the output is a 10HZ variable duty cycle, determined from the product of two fox functions. FN600 determines the duty cycle (and purge flow) based on the total accumulated time that purge has been enabled. This allows the purge flow to be slowly introduced when the fuel vapor concentration may be high. FN605A further modulates the output of FN600 versus total air flow. This permits limiting the purge flow to a small percentage of total engine air flow.

Certain adverse conditions, such as extended idles at elevated engine speed, may cause the purge vapors to contribute a significant amount of the fuel required to run the engine. When this condition, defined as LAMBSE at a large value while in Closed Loop Fuel, is encountered, the timer input to FN600(PRGTMR) is decremented to reduce purge flow and allow proper control of the air/fuel ratio. PRGTMR is also decremented if the engine is above normal operating temperature and is operating in Open Loop Fuel Control because of a condition that can exist when the vehicle is stopped (i.e., WRMEGO = 0, LESFLG = 1, OFMFLG = 1). This is to prevent the purge vapors from causing an excessively rich condition in open loop. In order for the purge duty cycle to be reduced immediately when PRGTMR begins decrementing, PRGTMR is clipped to FULPRGTM as a maximum. FULPRGTM should be calibrated to the time when output of FN600 equals 1.0 (i.e., the end of the ramp-in).

DEFINITIONS

INPUTS

Registers:

- ACT = Air charge temperature, deg. F.
- AM = Air mass flow, ppm.
- ECT = Engine coolant temperature, deg F.
- KAMREF = Adaptive fuel correction.
- LAMBSE = Air/fuel equivalence ratio.
- N = Engine RPM.
- NACTMR = Not at Closed Throttle timer, sec.
- PRGTMR = Purge ramp up/down timer, sec.
- PURG_ADP_SF = Adaptive learning safety factor; delta from minadp at which time purge is disabled.
- PURGDC = Canister Purge Duty Cycle.

Bit Flags:

- ADT1FMFLG = Adaptive table 1 failure mode.
- CRKFLG = CRANK mode flag; 1 -> closed loop.
- LESFLG = Lack of EGO switching flag; 1 -> EGO not switching.
- OFMFLG = ETV solenoid shorted flag; 1 -> ETV solenoid circuit shorted to ground.
- OLFLG = Open Loop Fuel Control flag; 1 -> Open Loop.
- PURGING = Purge enabled flag; 1 -> purge enabled.
- WRMEGO = Warm EGO flag; 1 -> EGO is currently warm.

Calibration Constants:

- CANPHP = Canister purge hardware present switch; 1 -> EEC controlled purge hardware present.

- EVTDOT = Minimum time not at Closed throttle to enable purge, sec. Set to "zero" to purge at Closed Throttle.

- FN600(PRGTMR) = Purge duty cycle as a function of accumulated purge enable time.

- FN603 = Maximum LAMBSE allowed before purge duty cycle is decremented.

- FN605A(AM) = Purge duty cycle modulator as a function of air mass.

- FULPRGTM = Maximum clip for PRGTMR, sec. Should be set to the time when the output of FN600 first reaches its maximum.

- MINADP = Minimum allowable correction.

- NLMT = Maximum engine RPM.

- PRG_DEC = PURGDC decrement value.

- PURECT = Minimum ECT to enable purge if WMEGOL is not set, deg F.

- PURECT1 = Minimum ECT to decrement PRGTMR when EGO is cold or not switching, deg F.

- PURGSW = Calibration switch to enable purge in Open Loop.

OUTPUTS

Registers:

- PRGTMR = See above.

- PURGDC = See above.

Bit Flags:

- LIMIT_PURGE = Flag which indicates Purge Duty Cycle is being limited due to LAMBSE being clipped; 1 -> limited Purge.

- PURGING = See above.

PROCESS

STRATEGY MODULE: CANP_COM4

CANISTER PURGE CONTROL LOGIC

```
CRKFLG = 0 ------------------------------|
                                         |
CANPHP = 1 ------------------------------|
                                         |
NACTMR >= EVTDOT ------------------------|
                                         |
OLFLG = 0 ------------------------|      |       (enable purge)
                                  |OR --|AND -|  PURGING = 1
PURGSW = 1 -----------------------|      |      | purgdc = FN600(PRGTMR)
                                         |      |            * FN605A(AM)
WRMEGO = 1 -----------------------|      |      |
                                  |OR --|       |
ECT > PURECT ----------------------|     |      |
                                         |      |
FN605A(AM) > 0 --------------------------|      |
                                         |      |
N < NLMT --------------------------------|      |
                                         |      |
OFMFLG = 0 ------------------------------|      |
                                                | --- ELSE ---
                                                |
                                                | (disable purge)
                                                | PURGING = 0
                                                | purgdc = 0
                                                | PURGDC = 0
```

PURGE DUTY CYCLE DECREMENT/INCREMENT LOGIC

```
LIMIT_PURGE = 0 -------------------------------| PURGDC = purgdc
                                               |
                                               | --- ELSE ---
                                               |
                                               | Continue to Increment/
                                               |  Decrement with old
                                               |  value of PURGDC
```

The following logic allows the reduction of the purge duty cycle (PURGDC) when conditions indicate that the fuel system may be failing. The purge is backed out to prevent the continuous EGO TESTS and adaptive fuel test from indicating failure due to purge overload. Also, the purge is backed out when the engine temperature is extremely hot and the EGOs are not switching or are not warmed-up yet. Once a failure has been indicated or control has been restored for one of the areas indicated by the logic show below, purging is resumed. If subsequently another area begins indicating failure, purging will again be backed out until the appropriate failure is verified or until the system comes back into full control.

```
OLFLG = 0 ------------------------|
 (closed loop)                    |
                                  |
LAMBSE => FN603(ACT) -------------|AND -|
 (ego failure condition present)  |     |
                                  |     |
LESFLG = 0 -----------------------|     |
 (ego failure recognition not latched)  |
                                        |
KAMRF <= MINADP + 0.5 +                 |
                   PURG_ADP_SF ----|    |
 (adaptive limit imminent)         |AND -|OR --| PURGDC = PURGDC -
                                   |     |     |   PRG_DEC
ADT1FMFLG = 0 ---------------------|     |     | Clip to zero as a minimum
 (adaptive failure not yet               |     | Clip to purgdc as a
  recognized)                            |     |  maximum
                                         |     |
ECT > PURECT1 ---------------------|     |     |
 (extreme hot temperatures)        |AND -|     |
                                   |     |     |
WRMEGO = 0 ------------------------|     |     |
 (EGO not warm)                          |     |
                                         |     |
LESFLG = 1 ------------------------------|     |
 (lack of EGO switching)                       |
                                               | --- ELSE ---
                                               |
                                               | PURGDC = PURGDC +
                                               |          PRG_DEC
                                               | Clip to purgdc as a maximum
```

```
purgdc > PURGDC -------------------------------| LIMIT_PURGE = 1
                                               |  (purge is being limited
                                               |   due to LAMBSE being
                                               |   at its clip)
                                               |
                                               | --- ELSE ---
                                               |
                                               | LIMIT_PURGE = 0
```

PRGTMR LOGIC

```
CRKFLG = 1 ------------------------------------| PRGTMR = 0
 (CRANK mode)                                  |
                                               | --- ELSE ---
                                               |
PURGING = 1 -----------------------------------| Increment PRGTMR
                                               |  (clip to FULPRGTM as
                                               |   a maximum)
                                               |
                                               | --- ELSE ---
                                               |
                                               | Freeze PRGTMR
```

CHAPTER 12

THERMACTOR AIR STRATEGY

THERMACTOR AIR STRATEGY

Thermactor air refers to air added to the exhaust gas mixture from the belt-driven thermactor air pump.

The computer controls two solenoids to create three mutually exclusive air states:

Thermactor Air State	AM1 Solenoid	AM2 Solenoid
Upstream	on	on
Downstream	on	off
Bypass	off	off

Upstream refers to air added at or near the exhaust ports. This is done to provide better oxidation of the exhaust gas mixture when a richer exhaust gas mixture is anticipated. It is not possible to operate in closed loop fuel control while air is introduced upstream (the EGO sensor may always indicate a lean condition).

Downstream refers to air added to the catalyst mid-bed. Downstream air is compatible with closed loop fuel control and is the normal thermactor air state.

Bypass refers to the condition in which no thermactor air is added to the exhaust gas mixture. This feature is used primarily to protect the catalyst from over-temperature conditions.

NOTE: THRMHP must be set to 1 to enable the Thermactor Air logic.

DEFINITIONS

INPUTS

Registers:

- ACT = Air charge temperature deg. F.
- APT = Throttle mode indicator; -1 -> closed throttle, 0 -> part throttle, 1 -> wide open throttle.
- ATMR1 = Time since start, sec.
- ATMR2 = Time after coolant temperature exceeded TEMPBF, sec.
- AWOTMR = Time at wide open throttle, sec.
- BYPTMR = Thermactor bypass timer, sec.
- DFSO_A_TMR = Free running, down counting, TP based thermactor air shut off timer.
- ECT = Engine coolant temperature, deg F.
- HMUTMR = High MAPPA upstream air timer, sec.
- HTPTMR = Heat protection timer (bankline timer), sec.
- LMBTMR = Low MAP bypass timer, sec.
- MAP = Manifold absolute pressure BIN 3.
- N = Engine speed RPM.
- TCSTRT = Temperature of engine coolant at startup, deg F.
- TP_REL = Relative TP (TP - RATCH).
- WOTTMR = Time at wide open throttle.

Bit Flags:

- AFMFLG = ACT FMEM flag.
- CFMFLG = ECT FMEM flag.
- CRKFLG = Crank mode flag; 1 -> in crank mode.
- DFSFLG = Flag indicating status of Decel Fuel Shut-off; 0 -> Fuel not shut-off for decel, 1 -> Fuel shut-off for decel.
- EGO1FMFLG = EGO #1 FMEM flag.
- HMUTMR_FLG = High Mappa Upstream Air timer control flag.
- IMS = Inferred milage sensor flag; 0 -> low milage, 1 -> high milage.

- LESFLG = Lack of EGO switching flag.
- MFMFLG = MAP FHEM flag.
- MPGFLG = MPG mode flag.
- OLFLG = Open loop flag.
- OFMFLG = ETV overcurrent monitor failure flag; 0 -> ETV O.K., 1 -> ETV failure mode.
- TAQ1 = Thermactor air latch flag (Based on ECT)
- TFMFLG = TP FMEM flag.
- TP_AIR_OFF_F = flag to indicate state of FNTP_AIR_OFF Flip Flop (dump thermactor air at low TP_REL as a f'n of RPM)

Calibration Constants:

- BYPMAP = Minimum value of BYPTMR to bypass thermactor, sec.
- BYPWOT = Minumum time to wide open throttle to bypass air, sec.
- BYSTM1 = Maximum time to bypass thermactor after a high ECT startup, sec.
- BYSTM2 = Maximum time to bypass thermactor after an intermediate ECT startup, sec.
- BYSTM3 = Maximum time to bypass thermactor after ECT > TEMPFB for a low ECT startup, sec.
- BYSTM4 = Minimum time at closed throttle idle to bypass thermactor, sec.
- BYSTM8 = Minimum time at low MAP to bypass thermactor during decel, sec.
- CHKASW = Control switch for CHKAIR.
- CTBYS = Minimum coolant temperature to bypass thermactor air, deg F.
- CTBYSH = Hysteresis term for CTBYS, deg F.
- CTHIGH = Hot start minimum ECT, deg F.
- CTLOW = Cold start maximum ECT, deg F.
- DSFTSW = Control switch for thermactor air decel fuel.
- FNTP_AIR_OFF = TP_REL at which to dump air below.
- SW_MPD = Control switch for Thermactor air bypass MPG mode; 0 -> Do no bypass in MPG Mode, 1 -> bypass in MPG mode.
- TP_AIR_OFF_H = Hystersis for FNTP_AIR_OFF (bypass air at low TP_REL as a f'n of ROM)
- T70LSW = 7.0L thermactor application switch.

- T75LSW = Switch to dump air during closed loop for 7.5L applications.
- UPSMAP = Maximum time for upstream air at high MAPPA, sec.
- UPSTM1 = Maximum time for upstream air after a low milage, high ECT startup, sec.
- UPSTM2 = Maximum time for upstream air after a low milage, intermediate ECT startup, sec.
- UPSTM3 = Maximum time for upstream air after ECT > TEMPFB for a low milage, low ECT startup, sec.
- UPSTM4 = Maximum time for upstream air after a high milage, high ECT startup, sec.
- UPSTM5 = Maximum time for upstream air after a high milage, intermediate ECT startup, sec.
- UPSTM6 = Maximum time for upstream air after ECT > TEMPFB for a high milage, low ECT startup, sec.
- UPSWOT = Maximum time for upstream air at W.O.T., sec.

OUTPUTS

Bit Flags:

- AM1 = Flag that controls the state of the AM1 output; 0 -> output off, 1 -> output on.
- AM2 = Flag that controls the state of the AM2 output; 0 -> output off, 1 -> output on.
- CHKAIR = Thermactor forced open loop flag; 1 -> not forced Open loop, 0 -> Force Open loop.
- TAQ1 = Thermactor air latch flag (Based on ECT)
- TP_AIR_OFF_F = See above.

PROCESS

STRATEGY MODULE: THERM_LH

THERMACTOR AIR CONTROL LOGIC

```
ECT > CTBYS -----------------------|S Q -| TAQ1
                                   |
ECT < CTBYS - CTBYSH --------------|C


FNTP_AIR_OFF(N) = 0 ---------------|
 (air dump not requested)          |
                                   |OR --| TP_AIR_OFF_F = 0
TP_REL >= FNTP_AIR_OFF(N) +        |     |  (allow air)
                   TP_AIR_OFF_H ---|     |
 (tp_rel above hysterysis band)          |
                                         | --- ELSE ---
                                         |
TP_REL < FNTP_AIR_OFF(N) ----------------| TP_AIR_OFF_F = 1
 (dump air at low tp_rel)                |  (dump air)
                                         |
                                         | --- ELSE ---
                                         |
                                         | no action
```

```
DFSO_A_TMR <> 0 -------------------------|
 (turn or keep air off)                  |
                                         |
EGO1FMFLG = 1 (EGO failure) -------------|
                                         |
CRKFLG = 1 (in crank mode) --------------|
                                         |
OFMFLG = 1 (ETV solenoid shorted) -------|
                                         |
AFMFLG = 1 (ACT failure) ----------------|
                                         |
CFMFLG = 1 (ECT failure) ----------------|
                                         |
TFMFLG = 1 (TPS failure) ----------------|
                                         |
MFMFLG = 1 (MAP failure) ----------------|
                                         |
HTPTMR >= BYSTM4 ------------------------|
 (Bankline timed out)                    |
                                         |
LESFLG = 1 (EGO not switching) ----------|
                                         |
"A" (See next page) ---------------------|
 (startup bypass)                        |
                                         |
TAQ1 = 1 --------------------------------|OR --|  (bypass air)
 (ECT meets CTBYS criteria)              |     | AM1 = 0
                                         |     | AM2 = 0
MPGFLG = 1 (MPG mode) -------------|     |     | CHKAIR = 1
                                   |AND -|     |  (not forced O.L.)
SW_MPD = 1 ------------------------|     |     |
                                         |     |
DFSFLG = 1 (in DFSO) --------------|     |     |
                                   |AND -|     |
DSFTSW = 1 ------------------------|     |     |
 (select bypass if in DFSO)        |     |     |
                                         |     |
APT = 1 (WOT) ---------------------|     |     |
                                   |AND -|     |
WOTTMR >= BYPWOT ------------------|     |     |
 (WOT bypass)                            |     |
                                         |     |
CHKASW = 1 (Calib. Sw) ------------|     |     |
                                   |AND -|     |
BYPTMR = BYPMAP -------------|     |     |     |
 (cruise bypass)             |OR --|     |     |
                             |           |     |
LMBTMR >= BYSTM8 ------------|           |     |
 (low MAP bypass)                        |     |
                                         |     |
TP_AIR_OFF_F = 1 ------------------------|     |
 (dump air at low tp_rel)                |     |
                                         |     |
T75LSW = 1 ------------------------|     |     |
  (7.5L application)               |AND -|     |
                                   |           |
OLFLG = 0 -------------------------|           |
 (dump air during closed loop)                 |
                                               | --- ELSE ---
```

(continued on next page)

(continued from previous page)

```
CHKASW = 0 (Calib. Sw) ------------------|      |
                                         |AND -|  (bypass air)
BYPTMR = BYPMAP -------------------|      |      | AM1 = 0
 (cruise bypass)                   |OR --|      | AM2 = 0
                                   |            | CHKAIR = 0
LMBTMR >= BYSTM8 ------------------|            |  (forced O.L.)
 (low MAP bypass)                               |
                                                | --- ELSE ---
"B" (see next page) ---------------|            |
 (startup upstream)                |AND -|      |
                                   |      |      |
IMS = 0 ---------------------------|      |      |
                                          |      |
"C" (see next page) ---------------|      |      |
 (startup upstream)                |AND -|      |
                                   |      |      |
IMS = 1 ---------------------------|      |      |
                                          |      |
AWOTMR < UPSWOT --------------------|     |      |  (upstream air)
 (WOT upstream time)               |AND -|OR --| AM1 = 1
                                   |      |      | AM2 = 1
APT = 1 (WOT) ---------------------|      |      | CHKAIR = 0
                                          |      |  (force O.L.)
HMUTMR < UPSMAP -------------------|      |      |
 (high load upstream time)         |AND -|      |
                                   |      |      |
HMUTMR_FLG = 1 --------------------|      |      |
                                          |      |
OLFLG = 1 -------------------------|      |      |
 (open loop fuel)                  |AND -|      |
                                   |            |
T70LSW = 1 ------------------------|            |
 (7.0L application)                             |
                                                | --- ELSE ---
                                                |
                                                |  (downstream air)
                                                | AM1 = 1
                                                | AM2 = 0
                                                | CHKAIR = 1
                                                |  (not forced O.L.)
```

```
TCSTRT >= CTHIGH ------------------------|
                                         |AND -|
ATMR1 < BYSTM1 --------------------------|     |
                                               |
CTLOW < TCSTRT < CTHIGH -----------------|     |
                                         |AND -|OR --| "A"
ATMR1 < BYSTM2 --------------------------|     |
                                               |
TCSTRT <= CTLOW -------------------------|     |
                                         |AND -|
ATMR2 < BYSTM3 --------------------------|
```

```
TCSTRT >= CTHIGH ------------------------|
                                         |AND -|
ATMR1 < UPSTM1 --------------------------|     |
                                               |
CTLOW < TCSTRT < CTHIGH -----------------|     |
                                         |AND -|OR --| "B"
ATMR1 < UPSTM2 --------------------------|     |
                                               |
TCSTRT <= CTLOW -------------------------|     |
                                         |AND -|
ATMR2 < UPSTM3 --------------------------|
```

```
TCSTRT >= CTHIGH ------------------------|
                                         |AND -|
ATMR1 < UPSTM4 --------------------------|     |
                                               |
CTLOW < TCSTRT < CTHIGH -----------------|     |
                                         |AND -|OR --| "C"
ATMR1 < UPSTM5 --------------------------|     |
                                               |
TCSTRT <= CTLOW -------------------------|     |
                                         |AND -|
ATMR2 < UPSTM6 --------------------------|
```

CHAPTER 13

DATA COMMUNICATIONS LINK

DATA COMMUNICATION LINK (DCL)
(LINK_SW = 2, 3, 6 or 7)

OVERVIEW

The EEC communicates with other vehicle microcomputers through the 81C62 RAM/CART chip. The RAM/CART is a special microchip which contains a serial I/O port. This port performs serial communications on the data communication link (DCL). The communication link consists of a twisted pair (DATA+ and DATA-). The EEC treats the RAM/CART as read/write memory. The RAM/CART is identical to the 81C61 RAM-I/O chip, with exception of the serial port.

The RAM/CART chip may operate as an UART (Universal Asynchronous Receiver Transmitter), or a CART (Custom Asynchronous Receiver Transmitter). In UART mode, information is sent character by character, requiring large software overhead. In CART mode, information is sent in frames with minimal software intervention.

The strategy initializes to UART mode. Under most vehicle operating conditions, there is no communication on the DCL and the strategy remains in UART mode. Under these conditions, the EEC waits for a message from SBDS and does not transmit. The SBDS can tell the EEC to either go permanently off-line or to wake-up in CART mode to perform diagnostics.

```
  - - - - - - - - - - - - - - - -
 |  ________                      |
 | |        |      ______         |     |      ______
 | |        |<--->|      |<----|---->|<--->| SBDS |
 | |        |     | RAM/ |         |     |      ------
 | |  EEC   |     | CART |         |     |
 | |  CPU   |     |      |         |
 | |        |     |______|         |
 | |        |                      |
 | |________|                      |
 |                                 |
 |        EEC IV                   |
 |_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _|
```

DCL STATE DIAGRAM
(UART Mode Enabled)
(Link_sw = 4,5,6 or 7)

```
                                       Power-Up
                                          |
                                          |
                                          V
                              -------------------------
                             |         Standby         |
          Wake-up Message    |                         |
           from SBDS         | - Configured as UART    |
       ----------------------| - No Link Master        |
      |                      | - EEC monitors Link for |
      |                      |   SBDS message          |
      |                      | - Link not active except|
      V                      |   for SBDS              |
 -------------------------    -------------------------
|   Diagnostics           |               |
|                         |               |
| - Configured as CART    |               |
| - EEC is Link Master    |               | Go off-line
| - EEC communicates with |               | Message
|   SBDS                  |               | from SBDS
 -------------------------                |
     |                                    |
     |                                    |
     |                                    V
     |                         -------------------------
     |                        |       Off-Line          |
     |                        |                         |
      ----------------------> | - No communication      |
          Go off-line         | - No return to other    |
           from SBDS          |   states                |
                               -------------------------
```

DIAGNOSTIC MODE

The intent of the Data Communications Link (DCL) usage for diagnostics is to provide on-board, Electronic Engine Control Module (EEC-IV) to off-board, Service Bay Diagnostic System (SBDS), communications for enhanced diagnostic capabilities. To accomplish this task, the SBDS will specify various tasks which will be dependent on the level of diagnostics required.

If the strategy is in Standby (UART mode), the SBDS will send a message instructing the EEC to enter Diagnostics (CART mode), then monitor the DPS for further commands.

Once the EEC is in diagnostic mode, the SBDS designates the task by sending commands in the DPS triggering the vehicle subsystems to perform the appropriate actions. The diagnostic task specified by the contents of the DPS takes effect at the start of the next block. All remaining link devices will continue to monitor the DPS unless they have been sent off-line by the SBDS. Non-active devices must refrain from transmitting or receiving non-pertinent information until the DPS specifies a new diagnostic mode or returns to the idle value.

DEFINITIONS

- Block = A group of sixteen sequential frames, transmitted in ascending numerical order from 0 to 15.

- Byte = Eight bits of data.

- Cart_Mode = Bit flag that indicates the current RAM/CART operating mode; 0 -> UART mode; 1 -> CART mode.

- DCLCT_START = Flag to cause VIP to load continuous Self Test codes into a RAM table for transmission over DCL, unitless.

- DCLST_DONE = Handshaking flag which indicates VIP has finished loading the RAM table, commands 25H and 26H.

- DCLST_START = Handshaking flag which causes VIP to load the continuous codes into a RAM table for subsequent transmission over DCL. Used in DCL 25H.

- DEFAULT_LVF = Default value for the last valid frame in CART mode, unitless.

- Diagnostic Parameter Slots = The diagnostic parameter slots are a sequence of four information words, contained in the first four frames of IS1, which define the diagnostic mode, beginning at the start of the next block. The sequence is defined as IW1:F0, IW1:F1, IW1:F2, IW2:F3.

- Frame = A defined number of slots (1-16) preceded by an idle time. The first slots are used for link control (sync and ID) and status information, and the remaining slots are used for information transfer.

- ID Slot = The slot immediately following the sync slot, also referred to as IW0, which contains the 4 bit frame identification number.

- Idle Slot = A slot where no message is transmitted (all logic ones expected) (i.e. empty slot, blank slot)

- Information Slot = All slots except for sync slot and ID slot (IW1-IWE).

- Link Master = The control module which generates the sync sequence and ID slot.

- Link_sw = Hardware present switch for communication link configuration. See EEC Overview Chapter.

- Message = A serial information flow consisting of two bytes of data. The two bytes of data are also referred to as a word.

- Nibble = The upper or lower four bits of data in a byte.

- Slot = A dedicated time period relative to the sync sequence which is used for transmission of a message.

- Sync Message = The message which contains the sync word 0000H (H denotes hexadecimal).

- UART_State = State counter for UART message check.
- Word = The 16 data bits of a message.

OVERVIEW

The intent of the Data Communications Link (DCL) usage for diagnostics is to provide on-board, Electronic Engine Control Module (EEC-IV), to off-board, Service Bay Diagnostic System (SBDS), communications for enhanced diagnostic capabilities.

In order to accomplish this task, SBDS will provide various modes of operation which will be dependent upon the level of diagnostics required.

The desired mode of operation will be determined by the SBDS and broadcast on the DCL in the Diagnostic Parameter Slots, (DPS), which is defined to be Information Word 1, Frames 0 through 3 inclusive. For normal (non-diagnostic mode) operation, the Diagnostic Parameter Slots will be idle.

Once the SBDS is connected to the DCL, the operational mode may change at which time the predetermined vehicle subsystems will recognize the change and take the appropriate action.

The diagnostic mode specified by the contents of the Diagnostic Parameter Slots will take effect at the start of the next Block, (ie., Frame 0 of the next Block).

All remaining link devices will continue to monitor the DPS unless they have been permanently disconnected by the SBDS. Non-active devices must refrain from transmitting or receiving non-pertinent information until the DPS specifies a new diagnostic mode or returns to the idle value.

DEFINED EEC-IV DIAGNOSTIC MODES

An eight (8) bit value is used to encode the various diagnostic modes for each device. The following diagnostic modes are defined in this document:

DIAGMODE	Value	Function
00000001	1H	clear DCL error/flag bits
00000010	2H	clear Continuous Self Test codes
00000011	3H	display status information only
00100001	21H	transmit PID values
00100010	22H	transmit DMR values
00100011	23H	transmit PID map
00100100	24H	transmit DMR map
00100101	25H	run Self Test (K0E0 or K0ER)
00100110	26H	transmit Continuous Self Test codes
00100111	27H	transmit PID and DMR values
01000001	41H	read Parameter Identification (PID) map
01000010	42H	read Direct Memory Reference (DMR) map
01000011	43H	read program/data bytes
01000101	45H	read program execution vector
01000110	46H	read A/D substitution values

In addition, one two diagnostic codes are reserved for all link devices.

DIAGMODE	Value	Function
10000000	80H	go permanently offline, disable DCL function
10000001	81H	Set DCL baud rate

These functions are described in the text below.

DCL DIAGNOSTIC STATUS INFORMATION

The EEC-IV Status Display

When the EEC-IV module has been selected with a DIAGMODE command in the range 00H through 7FH inclusive, the EEC will write the following status information in Slot IW1, Frames 04H through 0FH inclusive:

Frame	Description
4	Display the current diagnostic mode
5	Display the diagnostic mode for the next Block
6	DCL Error/Flag bits - Low byte
7	DCL Error/Flag bits - High byte
8	Direct Memory Reference (DMR) Base register - Low byte
9	Direct Memory Reference (DMR) Base register - High byte
A	ROM Calibration ID - Low byte
B	ROM Calibration ID - High byte
C	Idle
D	Idle
E	Idle
F	Idle

The DCL Error/Flag register, (16 bit), is used to maintain status information regarding the actual DCL system. The SBDS will use this information to insure error free DCL communications and the determination of system status.

The Error/Flag bits are assigned as follows:

Bit	Function	Condition
0	Read Bytes	Load Address (low byte) Parity
1	Read Bytes	Load Address (high byte) Bad Value
2	Read Bytes	Data or Checksum Parity
3	Read Bytes	Incorrect Checksum
4	Read A/D Values	Parity Error
5	Read PID Map	Parity Error
6	Read (DMR) Offset	Parity Error
7	*** not used ***	
8	*** not used ***	
9	*** not used ***	
10	Read Execute Vector	Parity Error
11	Read Execute Vector	Incorrect Checksum
12	DCL Mode Scheduler	Bad Diagnostic Parameter Slot
13	EEC Reset	Set if EEC Resets
14	Self Test	Self Test Complete
15	Background	Set to Disable Program* Execution

* Refers to execution of diagnostic programs downloaded into the EEC-IV by the SBDS.

This error register is initialized to BC1FH on EEC power-up/initialize.

The Direct Memory Reference, (DMR), Base Register is a 16 bit value which is used by the EEC-IV DCL software for information requested by address. The SBDS will request data by sending 8-bit offset values which are added to the base register to compute the absolute address of the parameters requested. All parameters requested in this manner are to be returned as unscaled byte values.

DCL DIAGNOSTIC MODE CHANGE PROTOCOL

All devices connected to the DCL will change operational modes based on the contents of the Diagnostic Parameter Slots, (DPS), which are defined to be Information Word 1, Frames 0 through 3 inclusive. Normal mode is indicated by all Diagnostic Parameter Slots being Idle.

The Diagnostic Parameter Slots

The DPS may be divided into five (5) fields as follows:

Frame	Bits	Description
0	0-11	DCL Module Select Bit Map
1	0-11	DCL Module Offline Bit Map
2	0-7	DIAGMODE Command Code
2	8-11	Frame Length Specifier
3	0-11	DIAGMODE Command Qualifier

The Module Select Bit Map [IW:1 F:0 b:0-11]

The first field is the Module Select Bit Map which is defined as follows:

Bit	Selected Module
0	Electronic Engine Control Module (EEC-IV)
1	Cluster Control Assembly (CCA)
2	Message Center Control Assembly (MCCA)
3-11	reserved for future expansion

If bit 0 is set (1), the EEC-IV should execute the specified DIAGMODE command. If bit 0 is clear (0), the EEC-IV should ignore the DIAGMODE command.

The Module Offline Bit Map [IW:1 F:1 b:0-11]

The second field is the Module Select Bit Map which is defined as follows:

Bit	Selected Module
0	Electronic Engine Control Module (EEC-IV)
1	Cluster Control Assembly (CCA)
2	Message Center Control Assembly (MCCA)
3-11	reserved for future expansion

If bit 0 is set (1), the EEC-IV should not transmit or receive any normal map information. If bit 0 is clear (0), the EEC-IV MUST transmit its normal map information.

The DIAGMODE Command Code [IW:1 F:2 b:0-7]

The third field is the DIAGMODE Command Code. This is a byte value which identifies the requested diagnostic mode. DIAGMODE codes in the range 00H through 7FH inclusive are to be considered as specific, (or private), diagnostic modes. DIAGMODE codes in the range 80H through FFH inclusive are used to designate common diagnostic modes. Common diagnostic command codes MUST NOT be assigned without approval by all link devices.

The Frame Length Specifier [IW:1 F:2 b:8-11]

The fourth field is the Frame Length Specifier for the next Block. This is a four bit value in the range 1H through EH inclusive. This value specifies the Frame Length as the last valid Slot number. Therefore, this value is one (1) less than the value to be written to the CART Frame Length Register.

The DIAGMODE Command Qualifier [IW:1 F:3 b:0-11]

The fifth and last field is the DIAGMODE Command Qualifier. This is a 12-bit value which may contain additional information required to process the requested diagnostic procedure. Currently, only one defined EEC-IV diagnostic mode uses this information, (see DIAGMODE CODE 80H, Go Permanently Offline).

Diagnostic Parameter Slots Error Processing

Unless normal mode is specified, (ie., all DPS idle), all DPS fields MUST contain valid information. That is to say, all DPS MUST not be idle and all Slots MUST have the correct Vertical Parity value.

If any of the above conditions are not met, then the entire DPS is to be considered in error and disregarded. Normal map information must then be transmitted at the start of the next Block.

DCL DIAGNOSTIC MODE DESCRIPTIONS

The diagnostic modes used by the SBDS/EEC-IV interface may be divided into the following categories:

I. Read information from the EEC-IV module
II. Parameter Substitution
III. Download programs/data
IV. Run Self Test (KOEO and KOER)
V. Housekeeping functions

Each of these functions are described, in detail, below.

Read Information from the EEC-IV Module

All diagnostic modes in this category are used by the SBDS to request information from the EEC-IV module. Information may be requested in one of two ways depending on the mode selected: by name, (PID index code), or by address, (DMR offset).

Parameter Reference by Name

Certain key engine parameters will be referenced by a unique index code (ie., name). This mode is intended to provide a fast strategy independent means of requesting parameter values which are considered to be vital to most diagnostic procedures. It is our intent to keep this list as small as possible to minimize the index table storage requirements.

The SBDS will specify these parameters in the following way:

I. The SBDS will specify this diagnostic mode by transmitting the following DPS:

DPS Field	Value
Module Select	001H
Module Offline	FFFH
DIAGMODE Code	41H
Frame Length	X
DIAGMODE Qualifier	000H

The EEC-IV will change to this mode at the start of the next Block. All other link devices will go offline. Frame length can vary from 2 to E slots.

II. The SBDS will then transmit the parameter reference index codes in the selected Slots when mode 41H has taken effect. These index codes are single byte values in the range 1 through 255 inclusive. An IDLE Slot or a Slot containing the value 0 indicates that the EEC-IV should not transmit in that Slot during the response phase, (mode 21H).

III. The EEC-IV will read these index codes and store them in a RAM table for later usage. This table needs to be 208 bytes in length, (16 Frames by 13 Slots; Slots IW0 and IW1 are not represented in this table). This table will begin at the end of reference by address table i.e., DCL_RAM_START + 208.

IV. While in mode 041H, the EEC-IV will transmit only the Sync and ID Slots along with status information in the Diagnostic Slot, (IW1).

After the Parameter Identification, (PID), map has been sent to the EEC-IV, the SBDS will change modes to allow the EEC-IV to transmit the requested parameters.

I. Once the table has been loaded by the SBDS, (a minimum of one Block time), the SBDS will specify mode 21H by transmitting the following DPS:

DPS Field	Value
Module Select	001H
Module Offline	FFFH
DIAGMODE Code	21H
Frame Length	X
DIAGMODE Qualifier	000H

The EEC-IV will change to this mode at the start of the next Block. All other link devices will go offline. Frame length can vary from 2 to E slots.

When mode 21H takes effect, the EEC-IV will transmit the requested parameter values in the same Slots that their index codes were transmitted in by the SBDS in mode 41H. The parameter values may be up to 12 bits in length and are to be scaled appropriately. If an unused PID code is requested, a value of zero will be returned in that slot.

The parameter names, indices (PIDs), and scaling factors are included towards the end of this chapter.

Parameter Reference by Address Offset

In order to allow the SBDS to access any strategy parameter, another method of requesting data will be provided by the EEC-IV module. In this mode, the SBDS will request information by transmitting an 8 bit offset to a base address. The sum of the base address value and offsets will form the absolute addresses used to reference the required engine parameters.

The offset values are specified in the following way:

I. The SBDS will specify this diagnostic mode by transmitting the following DPS:

DPS Field	Value
Module Select	001H
Module Offline	FFFH
DIAGMODE Code	42H
Frame Length	X
DIAGMODE Qualifier	000H

The EEC-IV will change to this mode at the start of the next Block. All other link devices will go offline.

II. At the start of the next Block, the SBDS will transmit the the desired offset values in the appropriate Slots. The EEC-IV must read these values and store them in a RAM table, called the Direct Memory Reference (DMR) map, for future use. The table must be 208 bytes in length, (16 Frames by 13 Slots; Slots IW0 and IW1 are not represented in this table). This table space cannot be shared with the PID RAM table as the SBDS may have loaded valid data into both tables at the same time. This table is located at DCL_RAM_START.

III. While in mode 42H, the EEC-IV will transmit only the Sync and ID Slots along with the status information in the Diagnostic Slot, IW1.

IV. Slots which are IDLE or contain the value 0 indicate that the EEC-IV should not transmit any information in these Slots during the response phase (mode 22H).

After the byte offset values have been loaded into the EEC-IV, the SBDS may then command the requested information be returned as follows:

I. The SBDS will specify this diagnostic mode by transmitting the following DPS:

DPS Field	Value
Module Select	001H
Module Offline	FFFH
DIAGMODE Code	22H
Frame Length	X
DIAGMODE Qualifier	XXXH

The EEC-IV will change to this mode at the start of the next Block. All other link devices will go offline.

II. The base address is read from the command qualifier slot and is normalized by shifting left 4 bit positions to obtain a 16-bit address aligned on a 16-byte boundary in memory. While mode 22H is in effect, the EEC-IV will transmit the requested information in the specified Slots. Each parameter to be returned is an unscaled byte value whose address is obtained by adding the 8 bit offset to the 16 bit base address value (obtained as above, or the default value of 0000H).

Parameter Reference by both PID MAP and Address Offset:

After both the PID Map and the DMR byte offset values have been loaded into the EEC-IV, the SBDS may then command the requested information to be returned using a combination of PID and DMR as follows:

I. The SBDS will specify this diagnostic mode by transmitting the following DPS:

DPS Field	Value
Module Select	001H
Module Offline	FFFH
DIAGMODE Code	27H
Frame Length	X
DIAGMODE Qualifier	000H

The EEC-IV will change to this mode at the start of the next Block. All other link devices will go offline.

II. While mode 27H is in effect, the EEC-IV will transmit the requested information in the specified Slots. The transmission will be done via either PID or DMR, depending upon whether PID codes or address offsets have been specified by SBDS. If a conflict arises, where both a PID code and an address offset have been specified for the same slot, the PID transmission will have priority. The parameters will be returned according to the descriptions in commands 21H and 22H, above.

EEC-IV Transmission of Parameter Identification (PID) Map

The SBDS May request that the EEC-IV module transmit the current PID map to verify the information which was previously requested. The PID map is requested as follows:

I. The SBDS will specify this diagnostic mode by transmitting the following DPS:

DPS Field	Value
Module Select	001H
Module Offline	FFFH
DIAGMODE Code	23H
Frame Length	E
DIAGMODE Qualifier	000H

The EEC-IV will change to this mode at the start of the next Block. All other link devices will go offline.

II. The EEC-IV will transmit a single byte value in all data Slots (Slots IW2 through IWE). This value will be the parameter index for that Slot. A value of 0 will indicate that no request was made for that information slot.

III. The EEC-IV will also transmit the Sync and ID Slots as well the status information in the Diagnostic Slot, IW1, as usual.

EEC-IV Transmission of Direct Memory Reference (DMR) Map

The SBDS May request that the EEC-IV module transmit the current base address offset map to verify the information which was previously requested. The base address offset map is requested as follows:

I. The SBDS will specify this diagnostic mode by transmitting the following DPS:

DPS Field	Value
Module Select	001H
Module Offline	FFFH
DIAGMODE Code	24H
Frame Length	E
DIAGMODE Qualifier	000H

The EEC-IV will change to this mode at the start of the next Block. All other link devices will go offline.

II. The EEC-IV will transmit a single byte value in all data Slots (Slots IW2 through IWE). This value will be the address offset for that slot. A value of 0 will indicate that no request was made for that information slot.

III. The EEC-IV will also transmit the Sync and ID Slots as well as the status information in the Diagnostic Slot, IW1, as usual

Parameter Substitution

The only diagnostic mode of this type defined thus far is the substitution of the A/D sensor values. The SBDS will transmit the values to be used in place of the normal strategy values.

A/D Sensor Value Substitution

The instantaneous A/D sensor values will be substituted in the following way:

I. The SBDS will specify this diagnostic mode by transmitting the following DPS:

DPS Field	Value
Module Select	001H
Module Offline	FFFH
DIAGMODE Code	46H
Frame Length	E
DIAGMODE Qualifier	000H

The EEC-IV will change to this mode at the start of the next Block. All other link devices will go offline.

II. When mode 46H has taken effect, the SBDS will write up to 13 A/D substitution values in Slot IW2, Frames 0 through C, inclusive. These 10 bit binary 0 values will be substituted for their respective A/D sensor values according to the following table:

A/D Register	Slot
ACH-0	IW2
ACH-1	IW3
ACH-2	IW4
ACH-3	IW5
ACH-4	IW6
ACH-5	IW7
ACH-6	IW8
ACH-7	IW9
ACH-8	IWA
ACH-9	IWB
ACH-10	IWC
ACH-11	IWD
ACH-12	IWE

III. The EEC-IV will transmit only the Sync and ID Slots and the status information in the Diagnostic Slot, IW1, as usual.

IV. The values substitution should begin during the next background loop and continue until changed later by the SBDS.

V. A/D parameter substitution is disabled for a particular A/D channel when its associated Slot is left IDLE during mode 46H.

VI. When A/D parameter substitution is no longer requested all RAM must be re-initialized.

VII. Before substituting A/D values all KAM must be read and stored by the SBDC using parameter reference by address offset. At the completion of the A/D substitutions the KAM data stored in the SBDC must be read back to the EEC using data download.

Downloading Programs and Data

The next class of operations involves the transmission of data and diagnostic programs from the SBDS to the EEC-IV module. Up to 10 bytes of data may be transmitted during each Frame. An Exclusive-Or checksum is provided to ensure data integrity.

Program / Data Download

There is no real difference between the transmission of data and programs. In fact, the two may be transmitted at the same time. To keep processing to a minimum, only 10 bytes may be transmitted in a given Frame in Slots IW2 through IWE inclusive. This information is downloaded as follows:

I. The SBDS will specify this diagnostic mode by transmitting the following DPS:

DPS Field	Value
Module Select	001H
Module Offline	FFFH
DIAGMODE Code	43H
Frame Length	E
DIAGMODE Qualifier	000H

The EEC-IV will change to this mode at the start of the next Block. All other link devices will go offline.

II. The SBDS will then transmit the program/data using the following format for each Frame in the Block:

Slot	Contents
IW2	load address - low byte
IW3	load address - high byte
IW4	data byte 1
IW5	data byte 2
IW6	data byte 3
IW7	data byte 4
IW8	data byte 5
IW9	data byte 6
IWA	data byte 7
IWB	data byte 8
IWC	data byte 9
IWD	data byte 10
IWE	Exclusive-Or checksum

III. If Slot IW2 is IDLE, the EEC-IV should ignore the rest of the data in that Frame. If Slot IW2 contains the low byte of the load address, then Slot IW3 MUST contain the load address high byte. The data Slots may contain byte values or be IDLE. An IDLE Slot indicates that no data should be written into the address associated with that Slot. Data byte #1 will be loaded into the address specified in Slots IW2 and IW3. Data byte #2 will be loaded into the next sequential address and so on. A new load address is provided in every Frame. The load address MUST be between 'DCL_RAM_START' and 'DCL_RAM_END', or KAM_START and KAM_END. The only data written between KAM_START and KAM_END is previously saved KAM data. DCL_RAM_END is equal to DCL_RAM_START + 600.

IV. Slot IWE will contain an 8 bit Exclusive-Or checksum of Slots IW2, IW3 and all data Slots which are not IDLE. This checksum must be used by the EEC-IV to validate the received data bytes.

V. The EEC-IV will transmit only the Sync and ID Slots as well as the status information in the Diagnostic Slot, IW1.

VI. This mode will remain in effect for as many Blocks as necessary to transmit all of the required data/program bytes.

VII. Before downloading a program to execute all KAM must be read and stored by the SBDC using Parameter Reference by Address Offset. At the completion of running a downloaded program the KAM data must be written back to the EEC using data down load.

After a program has been downloaded, the SBDS may enable execution of this program by sending the start address of the desired program. The requested routine is to be executed once per background loop until the request is revoked by the SBDS transmitting an execution address of 0000H, or when bit 15 of the DCL Error/Flag register is set, or when the EEC-IV is reset. When the request is revoked all RAM must be re-initialized.

The program execution vector will be specified as follows:

I. The SBDS will specify this diagnostic mode by transmitting the following DPS:

DPS Field	Value
Module Select	001H
Module Offline	FFFH
DIAGMODE Code	45H
Frame Length	4
DIAGMODE Qualifier	000H

The EEC-IV will change to this mode at the start of the next Block. All other link devices will go offline.

II. As soon as this mode takes effect, the SBDS will transmit the 16 bit start address by writing the low byte in Slot IW2 and the high byte in Slot IW3. An 8 bit checksum of the address will be transmitted in Slot IW4, all in Frame 0.

III. The program should begin execution within the next background loop time and be repeated during each background loop.

IV. The program execution must be inhibited if any of the associated error bits in the DCL Error/Flag register are set (bits 0-3 and bits 10-11). In addition, the program may disable itself by setting bit 15 of the DCL Error/Flag register.

Running On Demand Self Test (Key-On-Engine-Off and Key-On-Engine-Running)

The next class of diagnostic operational modes are used to initiate on demand self test and to return the fault (service) codes. A single DCL mode command is used to initiate both Key-On-Engine-Off and Key-On-Engine- Running self tests. The PIP signal will be used to determine which test sequence is performed.

Continuous Self Test codes are not to be transmitted along with the on-demand codes and must not be cleared during this time.

The EEC will set a flag DCLST_START after receiving diagnostic mode code 25H. This flag is continuously read by VIP to initiate self test.

```
DIAGMODE = 25H ---------------| DCLST_START = 1
 (SBDC request for VIP)       | (initiate self test)
                              |
                              | --- ELSE ---
                              |
                              | DCLST_START = 0
                              | (clear request for self test)
```

Key On Engine Off Self Test

One of these tests will be performed when the SBDS has requested on demand Self Test. PIP is used to determine which Self Test is performed.

I. The SBDS will specify this diagnostic mode by transmitting the following DPS:

DPS Field	Value
Module Select	001H
Module Offline	FFFH
DIAGMODE Code	25H
Frame Length	E
DIAGMODE Qualifier	000H

The EEC-IV will change to this mode at the start of the next Block. All other link devices will go offline.

II. The EEC-IV will transmit only the Sync and ID Slots and the status information in the Diagnostic Slot, IW1, while self test is being run. All other link devices will be offline.

III. When the self test has completed, the EEC-IV will transmit the service/fault codes in the last two slots. The codes will begin in the last slot of frame 0, and continue in the last slot of each successive frame up to frame 15. If there are more than 16 codes, the 17th code will be transmitted in slot (last slot - 1) of frame 0. The rest will be transmitted in successive frames, as above. The maximum number of codes is presently 20. These codes are to be transmitted every Block until the diagnostic mode has been changed by the SBDS. The EEC-IV will inform the SBDS that all codes have been transmitted, at least once, by setting bit 14 in the DCL Error/Flag register.

Key On Engine Running Self Test

This self test will be performed when the engine is running, (normal engine running strategy) and the SBDS has requested on demand self test.

I. The SBDS will specify this diagnostic mode by transmitting the following DPS:

DPS Field	Value
Module Select	001H
Module Offline	FFFH
DIAGMODE Code	25H
Frame Length	E
DIAGMODE Qualifier	000H

The EEC-IV will change to this mode at the start of the next Block. All other link devices will go offline.

II. The EEC-IV will transmit only the Sync and ID Slots and the status information in the Diagnostic Slot, IW1, while self test is being run. All other link devices will be offline.

III. When the self test has completed, the EEC-IV will transmit the service/fault codes in Slots IW2 through IWE, starting in Frame 0 and using as many Frames as necessary to transmit all of the service code table. (In strategy MUN1, this table is called SERV_CODE_TAB and is 20 bytes in length; therefore, only Frames 0 and 1 are required to transmit all possible service/ fault codes). These codes are to be transmitted every Block until the diagnostic mode has been changed by the SBDS. The EEC-IV will inform the SBDS that all codes have been transmitted at least once, by setting bit 14 in the DCL Error/Flag register.

Transmit Continuous Self Test Codes

I. The SBDS will specify this diagnostic mode by transmitting the following DPS:

DPS Field	Value
Module Select	001H
Module Offline	FFFH
DIAGMODE Code	26H
Frame Length	X
DIAGMODE Qualifier	000H

The EEC-IV will change to this mode at the start of the next block.

II. The EEC-IV will transmit only the Sync and ID Slots and the status information in the Diagnostic Slot, IW1.

III. When the self test is complete, the EEC-IV will transmit the service/fault codes in the last two slots. The codes will begain in the last slot of frame 0, and continue in the last slot of each successive frame up to frame 15. If there are more than 16 codes, the 17th code will be transmitted in slot (last slot - 1) of frame 0. The rest will be transmitted in successive frames as above. The maximum number of codes is presently 32 (2 slots in 16 frames). Each time the codes are transmitted, the flag DCLST_DONE is cleared causing VIP to reload the RAM table with the latest continuous codes. The latest codes will be transmitted every block until the diagnostic mode has been changed by the SBDS. If the diagnostic command remains 26H, and VIP does not have time to load the RAM table before the next block begins, that block will be idle. The EEC-IV will inform the SBDS that a given block contains codes by setting bit 14 of the DCL Error/Flag register.

Miscellaneous Housekeeping Functions

The following operational modes are used by the SBDS System to provide housekeeping functions. Currently, three of these diagnostic modes have been defined.

Clear DCL Error/Flag Register

This diagnostic mode is specified by the SBDS to clear the EEC-IV DCL Error/Flag Register. The sequence of events proceeds as follows:

I. The SBDS will specify this diagnostic mode by transmitting the following DPS:

DPS Field	Value
Module Select	001H
Module Offline	FFFH
DIAGMODE Code	01H
Frame Length	1
DIAGMODE Qualifier	000H

The EEC-IV will change to this mode at the start of the next Block. All other link devices will go offline.

II. While this mode is in effect, the EEC-IV will clear all bits except bit 12 of the DCL Error/Flag Register. Bit 12 will remain at its previous value.

III. During this mode, the EEC-IV will continue to transmit the Sync and ID Slots and the status information in the Diagnostic Slot, IW1. All other link devices will be offline.

Clear Continuous VIP Codes

This diagnostic mode is specified by the SBDS to clear the EEC-IV Continuous Self Test Codes. The sequence of events proceeds as follows:

I. The SBDS will specify this diagnostic mode by transmitting the following DPS:

DPS Field	Value
Module Select	001H
Module Offline	FFFH
DIAGMODE Code	02H
Frame Length	1
DIAGMODE Qualifier	000H

The EEC-IV will change to this mode at the start of the next Block. All other link devices will go offline.

II. While this mode is in effect, the EEC-IV will clear the four KAM bytes which hold the Continuous Self Test Codes.

III. During this mode, the EEC-IV will continue to transmit the Sync and ID Slots and the status information in the Diagnostic Slot, IW1. All other link devices will be offline.

Display DCL Status Information

This diagnostic mode is used by the SBDS to read the DCL status bytes which are transmitted in the Diagnostic Slot, IW1, without having the EEC-IV reading from or writing to any other Slots, (except the usual Slots in II below). This mode may also be used by the SBDS for certain critical mode timing change requirements. The sequence of events proceeds as follows:

I. The SBDS will specify this diagnostic mode by transmitting the following DPS:

DPS Field	Value
Module Select	001H
Module Offline	FFFH
DIAGMODE Code	03H
Frame Length	1
DIAGMODE Qualifier	000H

The EEC-IV will change to this mode at the start of the next Block. All other link devices will go offline.

II. During this mode, the EEC-IV will continue to transmit the Sync and ID Slots and the status information in the Diagnostic Slot, IW1. All other link devices will be offline.

COMMON DCL DIAGNOSTIC MODES

One diagnostic mode has been defined which is common to all link devices. This mode is defined below.

All DCL Devices Go Offline

Permanent Device Disconnect

This mode is used by the SBDS to command all link devices to go offline. This mode may be used to diagnose the link itself, or when external diagnostic devices may be attached to the link.

This mode will be specified by the SBDS as follows:

I. The SBDS will specify this diagnostic mode by transmitting the following DPS:

DPS Field	Value
Module Select	001H
Module Offline	FFFH
DIAGMODE Code	80H
Frame Length	X
DIAGMODE Qualifier	07FH

The EEC-IV will change to this mode at the start of the next Block. All other link devices will go offline.

II. The EEC-IV will completely disable all DCL functions. The CART circuit should be placed into PAUSE mode. The EEC-IV must not generate Sync Words in this mode.

III. This mode will remain in effect until the EEC-IV has been reset, (ignition key OFF-ON).

Note that this mode makes use of the DIAGMODE Command Qualifier Field. The EEC-IV MUST verify this command by taking the 1's complement of the DIAGMODE Command Qualifier and comparing it to the DIAGMODE Code. If the comparison fails, the EEC-IV MUST NOT go offline.

ERROR PROCESSING

I. Vertical Nibble Parity Error

In general, ignore data and retain previous valid data. Exceptions have been noted in this chapter.

II. Non-EEC Diagnostic Codes

EEC-IV DCL is idle. No intervention until EEC-IV Diagnostic Modes or Normal Mode are in effect. The EEC-IV will continue to transmit the Sync and ID Slots. No status information should be transmitted in a non-EEC-IV diagnostic mode.

SET DATA COMMUNICATIONS LINK BAUD RATE

This diagnostic mode is used by the diagnostic computer system to command all link devices to change baud rate. If any module is not capable of supporting the new desired baud rate, it must first be placed permanently offline.

I. The diagnostic computer will specify this diagnostic mode by transmitting the following DPS:

DPS Field	Value	Comment
Module Select	FFFH	Select all active modules
Module Offline	XXXH	as required, FFF in bay 0 on road
DIAGMODE Code	81H	command code
Frame Length	2H	minimum required frame length
DIAGMODE qualifier	000H	ignored, 0 recommended

The EEC-IV will change to this mode at the start of the next Block.

II. The diagnostic computer will then transmit the baud rate specifier information in Frame F, IW2 according to the following table:

Specifier	Requested Baud Rate
00H	2400 baud
01H	4800 baud
02H	9600 baud
03H	19200 baud

III. The EEC-IV will set the new baud rate immediately upon reading and verifying the baud rate specifier, (during the CART service routine for receive frame F). Therefore, the EEC-IV will use the new baud rate beginning at the next Block.

IV. If the baud rate specifier is missing or invalid, the default baud rate of 2400 will be used and the corresponding error bit must be set. The error bit must be clear before the baud rate can be set to any value other than 2400 baud.

V. The EEC-IV will transmit the Sync and ID Slots during mode OCOM. Status information should also be transmitted in the Diagnostic Slot, IW1, as usual. Normal mode information may also be transmitted depending upon the state of the Module Offline Bit Mask.

A sample information map of this diagnostic mode is presented in Appendix P of this document.

9.0 Parameter Identification (PID) Code Tables:

Parameter Name	PID Code	Data Type	Output Scaling	Output Resolution
N	01	word	bin -2	4 RPM
MAP	02	byte	bin 3	0.125 "Hg
BP	03	byte	bin 3	0.125 "Hg
SAFTOT	04	byte	bin 2	0.25 degrees S.A.
IACT	05	word	bin 0	1 A/D count
IECT	06	word	bin 0	1 A/D count
IEGR	07	word	bin 0	1 A/D count
IEGO1	08	word	bin 0	1 A/D count
ITP	09	word	bin 0	1 A/D count
NOT USED	0A			
IVCAL	0B	word	bin 0	1 A/D count
FUELPW1	0C	word	bin -5	32 clock ticks
LAMBSE1	0D	word	bin 11	1/2048 unitless
APT	0E	byte	bin 0	-1,0,1 unitless
ACT	0F	byte	bin -1	2 degrees F
ECT	10	byte	bin -1	2 degrees F
VBAT	11	byte	bin 4	0.0625 volts
MAP_FREQ	12	word	bin 4	0.0625 Hz
EGRDC	13	word	bin 11	1 EEC-IV count
NOT USED	14			
ISCDTY	15	word	bin 11	1 EEC-IV count
NOT USED	16			
VSBAR	17	byte	bin 1	0.5 MPH
VS	18	word	bin 5	0.03125 MPH
NOT USED	19			
BITMAP_0	1A	word	N/A	N/A (see def. below)
BITMAP_1	1B	word	N/A	N/A (see def. below)
NOT USED	1C			
NOT USED	1D			
SBDS01	1E	byte	bin 0	N/A
SBDS02	1F	byte	bin 0	N/A
SBDS03	20	byte	bin 0	N/A
SBDS04	21	byte	bin 0	N/A
SBDS05	22	byte	bin 0	N/A
SBDS06	23	byte	bin 0	N/A

9.0 Parameter Identification (PID) Code Tables (continued):

Parameter Name	PID Code	Data Type	Output Scaling	Output Resolution
FMEM_FLAGS	24	byte	bin 0	N/A
FMEM_FLAG2	25	byte	bin 0	N/A
NOT USED	26			
LOAD	27	word	bin 11	0.0488% of standard air charge
KAMRF1	28	word	bin 4	0.0625 A/F
NOT USED	29			
DSDRPM	2A	byte	bin -4	16 r.p.m.
RATCH	2B	word	bin 2	0.25 count
NOT USED	2C			
ATMR1	2D	word	bin 0	1 second
IOCC	2E	word	bin 0	1 A/D count
INDS	2F	word	bin 0	1 A/D count
BCSDC	30	word	bin 11	0.0488% duty cycle on time
NOT USED	31			
GR_CM	32	byte	bin 1	N/A
NOT USED	33			
NOT USED	34			
ETVOCM	35	word	bin 2	0.25 volts
TV_PRES	36	byte	bin 1	0.5 p.s.i
ITOT	37	word	bin 0	1 A/D count
PDL	38	byte	bin 1	N/A
NOT USED	39			

Where:

```
FLG_LK_CM = 1 ---------------| BCSDC = 1
                             |
                             | --- ELSE ---
                             |
                             | BCSDC = 0
```

BITMAP REGISTER DEFINITIONS

9.1 BITMAP_0:

```
      +-------------------------------+
HIGH: | v | v | v | v | 3 | 2 | 1 | 0 |
      +-------------------------------+

0:    *** not used ***, always 0.
1:    *** not used ***, always 0.
2:    1 if canister purge has non-zero duty cycle.
3:    1 if A/C clutch is disengaged.
4-7:  Vertical Nibble Parity (VNP)


      +-------------------------------+
LOW:  | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
      +-------------------------------+

0:    *** not used ***, always 0.
1:    *** not used ***, always 0.
2:    *** not used ***, always 0.
3:    *** not used ***, always 0.
4:    *** not used ***, always 0.
5:    *** not used ***, always 0.
6:    1 if not in neutral or park.
7:    1 if fuel pump is on.
```

9.2 BITMAP_1:

```
      +-------------------------------+
HIGH: | v | v | v | v | 3 | 2 | 1 | 0 |
      +-------------------------------+

0:    1 if alternate shift mode/overdrive cancel is selected.
1:    *** not used ***, always 0.
2:    *** not used ***, always 0.
3:    *** not used ***, always 0.
4-7:  Vertical Nibble Parity (VNP)


      +-------------------------------+
LOW:  | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
      +-------------------------------+

0:    1 if in closed loop fuel control.
1:    1 if power sterring pressure switch is closed.
2:    1 if driver has selected A/C.
3:    *** not used ***, always 0.
4:    1 if Ignition Diagnostic Monitor EEC module input is high.
5:    1 if output AM1 is on.
6:    1 if output AM2 is on.
7:    *** not used ***, always 0.
```

PLEASE NOTE: If the hardware related to a certain BITMAP bit is not present on a specific application then that bit will always be zero for that application.

UART MESSAGE CHECK

OVERVIEW

The message check logic first checks to see if the UART receive buffer is full (bit 0 of CART_STATUS = 1). If the buffer is not full, then reception of data did not cause the interrupt. BYTE_NUM and EEC_CHKSUM are set to 0, CART_STATUS is set to 1818H and the message check logic is exited. If buffer is full, then the new byte is read from UART receive buffer and following logic is executed.

DEFINITIONS

INPUTS

Registers:

- BYTE_NUM = Indicates which byte of UART message.
- CMD_CODE = Command from SBDS.
- EEC_CHKSUM = XOR checksum of bytes 6 thru 10 of UART message.
- SBDS_CHKSUM = CHECKSUM computed by SBDS.

Bit Flags:

- MODULE_ID = 1 -> EEC selected.

OUTPUTS Registers:

- BYTE_NUM = See above.
- CART_STATUS = CART status register.
- EEC_CHKSUM = See above.
- NO_OF_STARTS = Number of starts using alternative calibration.
- NO_START_CHK = Number of starts, check byte.
- XDCL_BAUD = Current DCL baud rate.

Bit Flags:

- CART_MODE = 1 -> CART mode, 0 -> UART mode.
- XDCL_ERR0 = Error/flag register 0.

PROCESS

STRATEGY MODULE: DCL_UART_COM1

```
RECEIVE OVERRUN ERROR ---------------|
 (bit 5 of CART_STATUS = 1)          |
                                     |OR --|  (bad sync sequence or ...
BYTE_NUM < 5 ------------------|     |     |   byte lost in transmission)
 (5 zeros not received)        |AND -|     | CART_STATUS = 1818H
                               |           | BYTE_NUM = 0
new byte <> 0 -----------------|           | EEC_CHKSUM = 0
 (non-zero received)                       | Exit message check logic
                                           |
                                           | --- ELSE ---
BYTE_NUM < 5 ------------------------|     |
 (5 zeros not received)              |AND -|  (normal sync sequence)
                                     |     | CART_STATUS = 1818H
new byte = 0 ------------------------|     | Increment BYTE_NUM
 (zero received)                           | EEC_CHKSUM = 0
                                           | Exit message check logic
                                           |
                                           | --- ELSE ---
BYTE_NUM = 5 ------------------------|     |
 (5 zeros received)                  |AND -|  (sync sequence complete)
                                     |     | CART_STATUS = 1818H
new byte = 0 ------------------------|     | EEC_CHKSUM = 0
 (zero received)                           | Exit message check logic
                                           |
                                           | --- ELSE ---
                                           |
                                           |  (first non-zero received)
                                           | Increment BYTE_NUM
                                           | Store new byte in uart_msg
                                           | DO SBDS COMMAND LOGIC
```

Byte numbers 6 through 11 of the message are stored in RAM as follows:

```
                     _____________
      uart_msg:    6 | MODULE_ID   |    (1 -> EEC-IV selected)
                     -------------
                   7 | CMD_CODE    |    (Command from SBDS)
                     -------------
                   8 |  BYTE_8     |    (Not presently used)
                     -------------
                   9 |  BYTE_9     |    (Not presently used)
                     -------------
                  10 |  BYTE_10    |    (Not presently used)
                     -------------
                  11 | SBDS_CHKSUM |    (Checksum sent by SBDS)
                     -------------
```

SBDS COMMAND LOGIC

```
BYTE_NUM < 11 -----------------------------| CART_STATUS = 1818H
                                           |  (update the check sum)
                                           | EEC_CHKSUM = EEC_CHKSUM
                                           |  XOR new byte
                                           | Exit message check logic
                                           |
                                           | --- ELSE ---
                                           |
EEC_CHKSUM <> SBDS_CHKSUM -----------|     |  (ignore message)
 (bad data in buffer)                |OR --| CART_STATUS = 1818H
                                     |     | BYTE_NUM = 0
MODULE_ID <> 01 ---------------------|     | EEC_CHKSUM = 0
 (EEC not selected)                        | Exit message check logic
                                           |
                                           | --- ELSE ---
                                           |
CMD_CODE = 82H ----------------------------|  (echo MODULE_ID to SBDS)
 (valid command)                           | CART_STATUS = 1818H
                                           | uart transmit buffer = MODULE_ID
                                           | BYTE_NUM = 0
                                           | EEC_CHKSUM = 0
                                           | Exit message check logic
                                           |
                                           | --- ELSE ---
                                           |
CMD_CODE = 80H ----------------------------|  (go permanently off-line)
 (valid command)                           | CART_STATUS = 4040H
                                           | BYTE_NUM = 0
                                           | EEC_CHKSUM = 0
                                           | Exit message check logic
                                           |
                                           | --- ELSE ---
                                           |
CMD_CODE = 04H ----------------------------|  (go to CART Mode)
 (valid command)                           | XDCL_BAUD = 03H
                                           | CART_MODE = 1
                                           | CART_STATUS = 9292H
                                           | XFRAME = 0FH
                                           | XDCL_ERRO = BC1FH
                                           | BYTE_NUM = 0
                                           | EEC_CHKSUM = 0
                                           | Exit message check logic
                                           |
                                           | --- ELSE ---
                                           |
```

(continued on next page)

(continued from previous page)

```
CMD_CODE = 05H ----------------------------|  (load NO_OF_STARTS)
                                           | NO_OF_STARTS = BYTE_8
                                           | NO_START_CHK = BYTE_9
                                           | CART_STATUS = 1818H
                                           | BYTE_NUM = 0
                                           | EEC_CHKSUM = 0
                                           | Exit message check logic
                                           |
                                           | --- ELSE ---
                                           |
                                           |  (bad command)
                                           | CART_STATUS = 1818H
                                           | BYTE_NUM = 0
                                           | EEC_CHKSUM = 0
                                           | Exit message check logic
```

CHAPTER 14

DATA OUTPUT LINK

DATA OUTPUT LINK

OVERVIEW

In this configuration, the DOL provides relative TP information to the stand alone governor. DOL_DUTY is a duty cycle with transfer function FN500(TPREL). DOL_DUTY is sent to the governor as a PWM signal with constant frequency of 300 Hz (+/- 0.1%).

DEFINITIONS

INPUTS

Registers:

- RATCH = Lowest filtered throttle position (see SYSTEM EQUATIONS Chapter).

- TP = Throttle position, counts.

- TP_REL = Relative TP (TP - RATCH).

Bit Flags:

- MFMFLG = Flag indicating MAP sensor failure; 1 -> failure.

- TFMFLG = Flag indicating a TP sensor failure; 1 -> failure.

Calibration Constants:

- GOVHP = Governor hardware present switch; 0 -> no governor, 1 -> governor present.

- FN500 = Transfer function to convert TPREL counts to percent duty cycle for transmission on the DOL.

OUTPUTS

Registers:

- DOL_DUTY = Duty cycle to be output to the stand alone governor.

PROCESS

STRATEGY MODULE: DOL_LH

```
GOVHP = 1 --------------|
                        |AND -|
TFMFLG = 0 -------------|     |
 (TP sensor ok)               |
                              |OR --| DOL_DUTY = FN500(TP_REL)
GOVHP = 1 --------------|     |     |  (GMAPDC is duty cycle)
                        |     |     |  (output duty cycle DOL_DUTY
TFMFLG = 1 -------------|AND -|     |   at 300 Hz on HSO-2)
 (TP sensor failed)     |           |
                        |           |
MFMFLG = 0 -------------|           |
                                    | --- ELSE ---
                                    |
                                    | No action
                                    | Disable DOL outputs
```

NOTE:

- Should the TP sensor fail, TP is simulated from MAP and is output to the governor. Should the MAP sensor also fail, no value for TP is output.

PULSE CALCULATION

OVERVIEW

The Data Output Link (DOL) provides fuel consumption information to the vehicle dashboard fuel economy display products (Tripminder or Message Center). The output, in the form of pulses, represents the amount of fuel used since the last update. This information is used in calculating fuel economy and distance-to-empty for display to the driver.

The injector fuel flow (in lbmf/(injector/port)) is accumulated in the register FUEL_SUM every time an injector port is energized. Both normal and AE pulses are accumulated in FUEL_SUM. Once a background loop, FUEL_SUM is converted to the appropriate integer number of DOL pulses, and DOL_COUNT is updated according to the equation shown below. The amount of FUEL_SUM which cannot be converted to an integer count remains in FUEL_SUM for the next conversion. For some strategies, the injector fuel flow sum is stored in ticks (register FUEL_SUM_TICKS) instead of lbmf. The ticks are then converted to lbmf when the DOL pulses are calculated.

Approximately every two milliseconds, the value of DOL_COUNT is checked. If DOL_COUNT is greater than 1.0, the DOL output is energized for about one msec, and DOL_COUNT is reduced by 1.0.

DEFINITIONS

INPUTS

Registers:

- A0COR = Corrected fuel flow rate of injectors, lb/sec.

- DOL_COUNT = Number of pulses to be output to the Fuel Economy display device.

- FUEL_SUM_TKS = Register for DOL summer, ticks.

- INJOUT = Number of injectors per output port, unitless.

- stcf = Seconds to clock ticks conversion factor, ticks/second.

Calibration Constants:

- PUL_PER_GAL = Number of DOL pulses to be issued for each gallon of fuel used, pulse per gal.

  NOTE: THE VALUE FOR PUL_PER_GAL MUST BE OBTAINED FROM EED/INSTRUMENT SYSTEMS FOR EACH APPLICATION.

OUTPUTS

Registers:

- DOL_COUNT = See above.

- FUEL_SUM_TKS = See above.

PROCESS

STRATEGY MODULE: DOL_PULSE_CALC_COM3

Once per background loop execute the following:

```
bit 0 of LINK_SW = 1 ----------| [pulses] = FUEL_SUM_TKS * A0COR *
                               |             INJOUT * PUL_PER_GAL
                               |   --------------------------------
                               |   6.15 (lbm/gallon) * stcf
                               |
                               | DOL_COUNT = DOL_COUNT + integer([pulses])
                               |
                               | FUEL_SUM_TKS = remainder([pulses]) * stcf *
                               |           6.15 (lbm/gallon) / ( A0COR *
                               |                INJOUT * PUL_PER_GAL )
```

Once per millisecond execute the following:

```
bit 0 of LINK_SW = 1 ----|
                         |AND -| Toggle DOL output
DOL_COUNT > 0.5 ---------|     |
                               | DOL_COUNT = DOL_COUNT - 1
                               |
                               | --- ELSE ---
                               |
bit 0 of LINK_SW = 1 ----------| DOL_COUNT = 0.5
```

CHAPTER 15

ALTERNATIVE CALIBRATION

CALIBRATION INITIALIZE LOGIC

OVERVIEW

This module sets the flag which will invoke the alternate calibration. To do this, it looks at the value of the parameter NO_OF_STARTS which may have been downloaded in DCL, and it also checks the check byte which is the one's complement of NO_OF_STARTS. The value of NO_OF_STARTS is subsequently decremented. These checks are performed during the RAM initialization process only.

DEFINITIONS

INPUTS

Registers:

- NO_OF_STARTS = Number of starts using alternative calibration.
- NO_START_CHK = Number of starts, check byte.

Bit Flags:

- ALT_CAL_FLG = Flag to indicate use of alternate calibration.

OUTPUTS

Registers:

- NO_OF_STARTS = See above.
- NO_START_CHK = See above.

Bit Flags:

- ALT_CAL_FLG = See above.

PROCESS

STRATEGY MODULE: ALTR_CAL_INIT_COM1

Performed during RAM Initialization only.

```
NO_OF_STARTS > 0 ------------------|
                                   |AND -| ALT_CAL_FLG = 1
NO_START_CHK = 255 - NO_OF_STARTS -|     |
                                         | --- ELSE ---
                                         |
                                         | ALT_CAL_FLG = 0


ALT_CAL_FLG = 1 -------------------------| Decrement NO_OF_STARTS
                                         | Increment NO_START_CHK
                                         |
                                         | --- ELSE ---
                                         |
                                         | Do nothing
```

CALIBRATION CLEAR LOGIC

OVERVIEW

This module erases any remaining alternate calibration starts should the vehicle travel further than ALT_CAL_DIST in any one journey. The value of ALT_CAL_FLG is also set each background loop to reduce the effect of any possible corruption of the register that may occur.

DEFINITIONS

INPUTS

Registers:

- BG_TMR = Background loop timer.
- DISTANCE = Distance traveled since start, miles.
- NO_OF_STARTS = Number of starts using alternative calibration.
- NO_START_CHK = Number of starts, check byte.
- VSBAR = Filtered vehicle speed.

Calibration Constants:

- ALT_CAL_DIST = Distance traveled before alternate calibration is revoked, miles.

OUTPUTS

Registers:

- DISTANCE = See above.
- NO_OF_STARTS = See above.
- NO_START_CHK = See above.

Bit Flags:

- ALT_CAL_FLG = Flag to indicate use of alternate calibration.

PROCESS

STRATEGY MODULE: ALTR_CAL_CLR_COM1

```
                DISTANCE = DISTANCE + (VSBAR * BG_TMR) / 3600

DISTANCE > ALT_CAL_DIST -----------------| NO_OF_STARTS = 0
                                         | NO_START_CHK = 0
                                         | ALT_CAL_FLG = 0
                                         |
                                         | --- ELSE ---
NO_OF_STARTS > 0 ------------------|     |
                                   |AND -| ALT_CAL_FLG = 1
NO_START_CHK = 255 - NO_OF_STARTS -|     |
                                         | --- ELSE ---
                                         |
                                         | ALT_CAL_FLG = 0
```

CHAPTER 16

SHIFT CONTROL

E4OD TRANSMISSION STRATEGY OVERVIEW

The E4OD strategy and software are comprised of a set of distinct, independent modules, each with a specific function. The modules are designed to minimize the software impact of different transmission hardware and can thus be re-used in other future transmission strategies. The main modules and subroutines are show on the following page. They are executed in the order shown except for System Equations which is done immediately after input conversion. Also shown are the main output parameters of the modules.

```
*******************************************************************
*******************************************************************
***                                                             ***
***                                                             ***
***     All references to "PDL", "PRNDL", or variations thereof ***
*** with respect to electronic transmission controls are        ***
*** synonymous with "Manual Level Indicated Position" as deter- ***
*** mined from the manual lever position sensor.                ***
***                                                             ***
***     All references to "TV", "ETV", or terms containing "TV" ***
*** with respect to electronic transmission controls are synon- ***
*** ymous with "Electronic Pressure Control", and are not       ***
*** associated with any control function of the engine throttle.***
***                                                             ***
***                                                             ***
*******************************************************************
*******************************************************************
```

STRUCTURE

```
MAIN ROUTINE
         SUBROUTINES                                         OUTPUT PARAMETERS
-----------------------------------------------------------------------------

E4OD_SYS_EQU_COM1                                            NEBART
E4OD_INPUT_PROCESSING_COM1                                   VSBART
                                                             VSBART_RT
                                                             NOBART
                                                             PDL
                                                             SPD_RATIO
                                                             TP_REL
                                                             FLG_4X4L
                                                             FLG_OCS
                                                             FLG_PWR
                                                             TQ_NET

DESRD_GR_DETR_COM1                                           GR_DS
                                                             FLG_FRST_DS
         GR_DS_AUTO_COM1                                     FLG_SFT_UP
         VER_AUTO_SHFT_COM1                                  FLG_SFT_DN
                                                             FLG_SF_AUTO

CM_GR_DETR_COM1                                              GR_CM
                                                             GR_OLD
         CM_GR_MAN1_COM1                                     RT_GR_OLD
         CM_GR_MAN2_COM1                                     FLG_FRST_CM
         CM_GR_AUTO_DWN_COM1

SHFT_TIMER_COM1                                              TM_SFT_IN
                                                             FLG_SFT_IN
                                                             FLG_SFT_MDN

SHFT_SOL_CTL_COM1                                            FLG_SS_1
                                                             FLG_SS_2
                                                             RT_GR_CUR
                                                             GEAR_CUR

SHIFT_VALID_COM1                                             SFT_ERROR

CST_CLTCH_CTL_COM1                                           FLG_CS_CM
                                                             FLG_CS_ENG

TV_GUIDE_COM1                                                TV_PRES
         TV_STARTUP_COM1                                     TV_COUNTS
         TV_CST_BOOST_COM4                                   OFMFLG
         TV_ENGMT_STALL_COM1
         TV_NORM_COM1                                        TV_STAT
             TV_TQ_IALPHA_COM4                               TQ_IALPHA
             TV_DYNAMIC_COM1                                 TV_DYN
```

```
MAIN ROUTINE
         SUBROUTINES                                              OUTPUT PARAMETERS
---------------------------------------------------------------------------

ET_EPC_OFM_COM2
ET_TV_VFS_OUT_COM2

CNVRTR_CLUTCH_CTL_COM1                                            FLG_LK_CM
         UNCOND_UNLCK_COM1                                        FLG_UNC_UNLK
         SHFT_UNLCK_COM1                                          FLG_SFT_UNLK
             INI_DWN_CNVR_CLCH_COM1                               TM_SFT_CCO
             DWN_CNVR_CLCH_COM1
             INI_UP_CNVR_CLCH_COM1
             UP_CNVR_CLCH_COM2
         SCHLD_LCK_UP_COM1                                        FLG_CRV_LK
         WOT_LCK_UP_COM2                                          FLG_WOT_LK

RT_NOVS_KAM_CALC_COM1                                             RT_NOVS

CONV_CLCH_VALID_COM1                                              CC_ERROR

OD_CANCEL_SW_COM1                                                 FLG_OCS

OCIL_STATE_COM1                                                   OCIL_STATE
                                                                  OCILTMR
                                                                  OCIL_FLASH_TMR

OCIL_REPEAT_COM1

CST_OUT_REPEAT_COM1

CONVERTER_CLUTCH_REPEAT_COM3

TV_VFS_OUT_REPEAT_COM1
```

PRNDL BASED DESIRED GEAR DETERMINATION

OVERVIEW

The desired transmission gear is calculated based on PRNDL position. Possible gears are:

GR_CM	Transmission State	GEAR_CUR	RT_GR_CUR
1	1ST	1	2.710
2 (PDL < 3)	2ND, interm. band ON	2	1.538
2 (PDL > 2)	2ND, interm. band OFF	2	1.538
3	3RD	3	1.0
4	4TH	4	0.742

In the normal Drive/Overdrive position, the desired gear is calculated based on a maximum WOT RPM shift point or as a function of throttle position versus vehicle speed. All shift points are adjusted for altitude. There are no excluded shifts in automatic mode, that is 1 - 4 shifts or 3 - 1 shifts are permitted if the calibration calls for it.

In manual 2 or 1, desired gear is set to the ultimate desired gear: 2 in manual 2 and 1 in manual 1. Sequencing through the downshift routine is left to the commanded gear routine.

The main outputs of the desired gear routine are:

- GR_DS, the desired gear;
- FLG_FRST_DS, global flag to indicate a shift is desired this background pass.

DEFINITIONS

INPUTS

Registers:

- GEAR_CUR = Current transmission gear.
- GR_DS = Desired transmission gear.
- GR_DS_LST = Desired gear in the last background pass
- GR_DS_TV = Desired gear used to compute TV pressure.
- GEAR_OLD = Last commanded gear (global register).
- PDL = Current PRNDL position.
- PDL_LST = Last PRNDL position.

- TM_DEL_SFT = Time during which a shift is delayed.

OUTPUTS

Registers:

- GR_DS = See above.
- GR_DS_LST = See above.
- GR_DS_TV = Desired gear used to compute TV pressure.
- TM_DEL_SFT = See above.
- TM_SFT_IN = Shift in progress timer.

Bit Flags:

- FLG_DE_DSGR = Delay desired gear 1st pass flag.
- FLG_DEL_MDN = Flag indicating a manual downshift is being delayed: 0 -> no manual downshift is being delayed, 1 -> a manual downshift is being delayed.
- FLG_FRST_DS = First time a shift is desired flag; 0 -> no shift desired, 1 -> shift desired this background pass.
- FLG_SF_AUTO = Automatic upshift/downshift flag; 1 -> automatic shift (PRNDL = 3 or 4), 0 -> manual shift (PRNDL = 2 or 1).
- FLG_SFT_DN = Downshift flag; 1 -> indicates current or last shift is/was a downshift.
- FLG_SFT_IN = Shift in progress flag; 1 -> shift in progress, 0 -> no shift in progress.
- FLG_SFT_UP = Upshift flag; 1 -> indicates current or last shift is/was an upshift.

PROCESS

STRATEGY MODULE: SC_DESRD_GR_DETR_COM1

```
PDL = 3 OR 4 ---------------------|
                                  |AND -| TM_SFT_IN = 0
PDL_LST <> 3 OR 4 ----------------|     | FLG_SFT_IN = 0


always ---------------------------------| GR_DS_LST = GR_DS


PDL = 1 --------------------------------| GR_DS = 1
 (PRNDL in manual 1)                    |
                                        | --- ELSE ---
                                        |
PDL = 2 --------------------------------| GR_DS = 2
 (PRNDL in manual 2)                    |
                                        | --- ELSE ---
PDL = 3 --------------------------|     |
 (PRNDL in overdrive cancel)      |OR --| Do "GR_DS, PRNDL = 3 OR 4" Logic
                                  |     |
PDL = 4 --------------------------|     |
 (PRNDL in overdrive)                   |
                                        | --- ELSE ---
PDL = 5 --------------------------|     |
 (PRNDL in neutral)               |     |
                                  |     |
PDL = 6 --------------------------|OR --| GR_DS = 1
 (PRNDL in reverse)               |
                                  |
PDL = 7 --------------------------|
 (PRNDL in park)


always ---------------------------------| GR_DS_TV = GR_DS
```

```
PDL <> 3 OR 4 --------------------|
 (manual gear)                    |
                                  |OR --| TM_DEL_SFT = 0
PDL_LST <> 3 OR 4 ----------|     |     | FLG_DE_DSGR = 0
                            |AND -|     | FLG_SF_AUTO = 0
PDL = 3 OR 4 ---------------|           |
 (man-to-auto shift)                    | --- ELSE ---
                                        |
                                        | Do "Delay Shift Logic"


TM_DEL_SFT = 0 -------------------------| FLG_DEL_MDN = 0


GR_DS_TV > GEAR_CUR --------------|
 (upshift is being verified or    |
  will be commanded this pass)    |OR --| FLG_SFT_UP = 1
                                  |     | FLG_SFT_DN = 0
GR_DS_TV = GEAR_CUR --------|     |     |  (indicate upshift)
 (no shift pending)         |AND -|     |
                            |           |
GEAR_CUR > GEAR_OLD --------|           | --- ELSE ---
 (last shift was an upshift)            |
                                        | FLG_SFT_UP = 0
                                        | FLG_SFT_DN = 1
                                        |  (indicate downshift)


GR_DS <> GR_DS_LST ---------------------| FLG_FRST_DS = 1
 (desired gear has changed)             |  (new desired gear for this
                                        |   program pass only)
                                        |
                                        | --- ELSE ---
                                        |
                                        | FLG_FRST_DS = 0
```

GR_DS, PRNDL = 3 OR 4 LOGIC

OVERVIEW

This module handles the Desired Gear computation when PRNDL = 3 or 4.

DEFINITIONS

INPUTS

Registers:

- BP_INTR = BP interpolation factor.
- CS_SFT_MULT = Cold start shift multiplier.
- GR_DS = Desired transmission gear.
- GR_DS_LST = Desired gear, last background pass.
- GEAR_CUR = Current transmission gear
- NEBART = Filtered engine RPM for transmission.
- PDL = Current PRNDL position.
- RT_NOVS = Ratio of actual N/V to base N/V in KAM.
- TP_REL = Relative throttle position, counts.
- TP_REL_H = Relative TP (TP - RATCH) high byte only.
- VS_RATEPH = Vehicle accel rate for Powertrain Hunting.
- VSBART_RT = Filtered vehicle speed adjusted for RT_NOVS for transmission.
- VSCTR = Counter for unrealistic changes in vehicle speed.

Bit Flags:

- FLG_4X4L = Flag indicating 4X4 low mode; 1 -> in 4X4 low mode.
- FLG_SFT_IN = Shift in progress flag; 1 -> shift in progress, 0 -> no shift in progress.
- VSFMFLG = Vehicle speed sensor failure flag; 1 -> VSS failure, 0 -> no VSS failure.

Calibration Constants:

- FN12A(TP_REL_H) = Vehicle speed for 1 - 2 upshift at altitude.
- FN12S(TP_REL_H) = Vehicle speed for 1 - 2 upshift at sea level.
- FN21A(TP_REL_H) = Vehicle speed for 2 - 1 downshift at altitude.
- FN21S(TP_REL_H) = Vehicle speed for 2 - 1 downshift at sea level.
- FN23A(TP_REL_H) = Vehicle speed for 2 - 3 upshift at altitude.
- FN23PPH(TP_REL) = Min VS_RATEPH to allow 2 - 3 upshift.
- FN23S(TP_REL_H) = Vehicle speed for 2 - 3 upshift at sea level.
- FN32A(TP_REL_H) = Vehicle speed for 3 - 2 downshift at altitude.
- FN32S(TP_REL_H) = Vehicle speed for 3 - 2 downshift at sea level.
- FN34A(TP_REL_H) = Vehicle speed for 3 - 4 upshift at altitude.
- FN34PPH(TP_REL) = Minimum VS_RATEPH to allow 3 - 4 upshift.
- FN34S(TP_REL_H) = Vehicle speed for 3 - 4 upshift at sea level.
- FN43A(TP_REL_H) = Vehicle speed for 4 - 3 downshift at altitude.
- FN43S(TP_REL_H) = Vehicle speed for 4 - 3 downshift at sea level.
- FN689D(TP_REL) = Engine speed for downshifts during VSS failure.
- FN689U(TP_REL) = Engine speed for upshifts during VSS failure.
- NE12A = WOT RPM 1 - 2 shift point, altitude.
- NE12S = WOT RPM 1 - 2 shift point, sea level.
- NE23A = WOT RPM 2 - 3 shift point, altitude.
- NE23S = WOT RPM 2 - 3 shift point, sea level.
- NE34A = WOT RPM 3 - 4 shift point, altitude.
- NE34S = WOT RPM 3 - 4 shift point, sea level.

OUTPUTS

Registers:

- GR_DS = See above.

Bit Flags:

- FLG_UP_NE = WOT engine RPM upshift flag; 1 -> upshift due to WOT RPM, 0 -> upshift due to shift curves.

PROCESS

STRATEGY MODULE: SC_GR_DS_AUTO_COM1

```
Always ---------------------------------------| FLG_UP_NE = 0
                                              |  (Start fresh every time)


VSFMFLG = 0 ----------------------------|
 (VS sensor OK)                         |AND -| Do "VEHICLE SPEED SENSOR
                                        |     |  OK SHIFT LOGIC"
VSCTR = 0 ------------------------------|     |
 (stable VS values)                           |
                                              | --- ELSE ---
                                              |
                                              | Do "VEHICLE SPEED SENSOR
                                              |  FAILURE SHIFT LOGIC"


GR_DS > GR_DS_LST ----------------------|
                                        |AND -| GR_DS = GR_DS_LST
FLG_SFT_IN = 1 -------------------------|     |  (Do not allow an upshift
                                              |   if a previous shift is
                                              |   still in progress)
                                              | FLG_UP_NE = 0
```

VEHICLE SPEED SENSOR OK SHIFT LOGIC

```
GEAR_CUR = 1 ---------------------------|
                                        |
NEBART > NE12S + [BP_INTR * NE12A] -----|AND -| GR_DS = 2
 (1-2 WOT RPM upshift)                  |     | FLG_UP_NE = 1
                                        |     |
FLG_4X4L = 0 ---------------------------|     |
 (not in 4X4 low)                             |
                                              | --- ELSE ---
GEAR_CUR = 2 ---------------------------|     |
                                        |     |
NEBART > NE23S + [BP_INTR * NE23A] -----|AND -| GR_DS = 3
 (2-3 WOT RPM upshift)                  |     | FLG_UP_NE = 1
                                        |     |
FLG_4X4L = 0 ---------------------------|     |
 (not in 4X4 low)                             |
                                              | --- ELSE ---
GEAR_CUR >= 3 --------------------------|     |
                                        |     |
NEBART > NE34S + [BP_INTR * NE34A] -----|     |
                                        |AND -| GR_DS = 4
PDL = 4 --------------------------------|     | FLG_UP_NE = 1
 (3-4 WOT RPM upshift, if PRNDL = 4)    |     |
                                        |     |
FLG_4X4L = 0 ---------------------------|     |
 (not in 4X4 low)                             |
                                              | --- ELSE ---
GEAR_CUR < 4 ---------------------------|     |
                                        |     |
VSBART_RT >  [FN34S +  (BP_INTR *       |     |
  FN34A)] * CS_SFT_MULT ----------------|     |
                                        |AND -| GR_DS = 4
PDL = 4 --------------------------------|     |
 (VS vs. TP_REL upshift to 4th gear,    |     |
  if PRNDL = 4)                         |     |
                                        |     |
VS_RATEPH > FN34PPH * RT_NOVS ----------|     |
 (VS_RATEPH vs. TP_REL upshift to 4th gear,   |
  if minimum accel rate is satisfied)         |
                                              | --- ELSE ---
GEAR_CUR < 3 ---------------------------|     |
                                        |     |
VSBART_RT > [FN23S + (BP_INTR *         |     |
  FN23A)] * CS_SFT_MULT ----------------|AND -| GR_DS = 3
 (VS vs. TP_REL upshift to 3rd gear)    |     |
                                        |     |
VS_RATEPH > FN23PPH * RT_NOVS ----------|     |
 (VS_RATEPH vs. TP_REL upshift to 3rd gear,   |
  if minimum accel rate is satisfied)         |
                                              | --- ELSE ---
                                              |
```

(continued on next page)

(continued from previous page)

```
GEAR_CUR < 2 ---------------------------|      |
                                        |AND -| GR_DS = 2
VSBART_RT > [FN12S + (BP_INTR *         |      |
  FN12A)] * CS_SFT_MULT ----------------|      |
 (VS vs. TP_REL upshift to 2nd gear)           |
                                               | --- ELSE ---
GEAR_CUR > 1 ---------------------------|      |
                                        |AND -| GR_DS = 1
VSBART_RT < [FN21S + (BP_INTR *         |      |
  FN21A)] * CS_SFT_MULT ----------------|      |
 (VS vs. TP_REL downshift to 1st gear)         |
                                               | --- ELSE ---
GEAR_CUR > 2 ---------------------------|      |
                                        |AND -| GR_DS = 2
VSBART_RT < [FN32S + (BP_INTR *         |      |
  FN32A)] * CS_SFT_MULT ----------------|      |
 (VS vs. TP_REL downshift to 2nd gear)         |
                                               | ---ELSE ---
GEAR_CUR > 3 ---------------------|            |
                                  |AND -|      |
VSBART_RT < [FN43S + (BP_INTR *   |     |      |
  FN43A)] * CS_SFT_MULT ----------|     |      |
 (VS vs. TP_REL downshift to 3rd gear)  |OR --| GR_DS = 3
                                        |      |
GEAR_CUR > 3 ---------------------|     |      |
                                  |AND -|      |
PDL = 3 --------------------------|            |
 (manual 4-3 downshift)                        |
                                               | --- ELSE ---
                                               |
                                               | GR_DS = GEAR_CUR
```

VEHICLE SPEED SENSOR FAILURE SHIFT LOGIC

```
NEBART > FN689U(TP_REL) ----------------|
                                        |AND -| GR_DS = GEAR_CUR + 1
GEAR_CUR <> 4 --------------------------|      |
                                               | --- ELSE ---
NEBART < FN689D(TP_REL) ----------------|      |
                                        |      |
GEAR_CUR <> 1 --------------------------|AND -| GR_DS = GEAR_CUR - 1
                                        |
FLG_SFT_IN = 0 -------------------------|


GR_DS = 4 ------------------------------|
                                        |AND -| GR_DS = 3
PDL = 3 --------------------------------|
```

DELAY/VERIFY SHIFT LOGIC

OVERVIEW

This logic delays the commanding of the desired gears a calibratable amount of time to:

- allow for the TV pressure to ramp-up prior to commanding the shift. TV is increased tremendously during a shift. It is necessary to start commanding the extra TV prior to the shift to overcome the delays associated with the time constant of the TV solenoid.

- allow the RPM to decrease before commanding a tip-out upshift. Commanding an upshift immediately after a tip-out feels harsh, since the clutches will be used to slow the engine down. Waiting a period of time before commanding this shift, smoothes the shift significantly.

- verify a shift is absolutely necessary. Noise perturbations may result in incorrect gears being desired for one or two background loops. Delaying the shift, verifies that a gear is truly desired, and that it is not just noise.

Entering into this logic is the parameter, GR_DS, the desired gear. The logic first transfers this gear value into another parameter, GR_DS_TV, which is used to calculate the additional amount of TV required for the upcoming shift. The delay logic then proceeds to delay the shift by setting the GR_DS to its previous value until the delay timer runs out. When the timer expires, the GR_DS is no longer set to its previous value, but is allowed to pass through to the Commanded Gear Determination logic and Shift Solenoid logic which actually processes the GR_DS into shift solenoid commands; i.e., the shift is commanded.

By the time the shift is commanded, the necessary TV required for the shift will have been commanded, the engine RPM will have slowed some to make the tip-out upshifts feel smooth, and the shift will have been verified.

If the GR_DS changes prior to the timer expiring, the timer continues to count down. The timer does not affect the amount of TV which is commanded, because GR_DS_TV will always be set to the latest desired gear, and the TV will be commanded based on that value. Worst case, the TV pressure will be unnecessarily increased for a short period of time if the GR_DS/GR_DS_TV fluctuates due to a noise spike. When the timer becomes zero, the latest GR_DS is passed through.

NOTES:

The delay timer is set to one of three values. These are:

- delay to allow TV to ramp up for an upshift, and/or the delay to verify a gear; TM_DEL_UP

- delay to allow TV to ramp up for a downshift, and/or the delay to verify a gear; TM_DEL_DOWN

- delay for a tip-out upshift to allow the RPM decrease, TM_DEL_TO_UP

DEFINITIONS

INPUTS

Registers:

- GEAR_CUR = Current transmission gear.

- GR_DS = Desired transmission gear.

- GR_DS_LST = Desired gear in last background pass.

- PDL = PRNDL position.

- TM_DEL_SFT = Time to delay automatic desired shift.

- TP_RATE = Throttle rate = TP - TBART.

Bit Flags:

- FLG_DE_DSGR = Delay desired gear first pass flag; 0 -> First pass through DELAY DESIRED GEAR, 1 -> Delay desired gear in process.

Calibration Constants:

- TM_DEL_DOWN = Time to delay/verify a downshift.

- TM_DEL_TO_UP = Time to delay a tip-out upshift.

- TM_DEL_UP = Time to delay/verify an upshift.

- TO_TP_RATE = TP_RATE required to recognize a tip-out.

OUTPUTS

Registers:

- GR_DS = See above.

- GR_DS_TV = Desired gear used to compute TV pressure.

- TM_DEL_SFT = See above.

Bit Flags:

- FLG_DE_DSGR = See above.

- FLG_DEL_MDN = Flag indicating a manual downshift is being delayed: 0 -> no manual downshift is being delayed; 1 -> a manual downshift is being delayed.

- FLG_SF_AUTO = Automatic upshift/downshift flag: 1 -> Automatic shift (PRNDL = 3 or 4); 0 -> Manual shift (PRNDL = 1,2,5,6 or 7).

- FLG_TIP_OUT = Flag which indicates a tip-out upshift in progress: 0 -> no tip-out upshift in progress; 1 -> a tip-out upshift in progress.

PROCESS

STRATEGY MODULE: SC_VER_AUTO_SHFT_COM1

```
GR_DS <> GR_DS_LST -----------|
 (shift is desired)           |
                              |
FLG_DE_DSGR = 1 --------------|AND -| GR_DS = GR_DS_LST
 (delay in process)           |     |  (hold original desired gear until
                              |     |   new desired gear is verified)
TM_DEL_SFT > 0 ---------------|     |
 (timer not expired)                | --- ELSE ---
                                    |
GR_DS <> GR_DS_LST -----------|     |
 (shift is desired)           |     |
                              |     |
FLG_DE_DSGR = 1 --------------|AND -| FLG_DE_DSGR = 0
 (delay in process)           |     |  (new desired gear is delayed,
                              |     |   allow it to pass thru to
TM_DEL_SFT = 0 ---------------|     |   commanded gear module)
 (timer expired)                    | FLG_SF_AUTO = 1
                                    |  (this is an automatic shift,
                                    |   cleared in converted clutch)
                                    |
                                    | --- ELSE ---
GR_DS > GR_DS_LST ------------|     |
 (upshift is desired)         |AND -| FLG_DE_DSGR = 1
                              |     |  (set first pass flag)
FLG_DE_DSGR = 0 --------------|     | DO "LOAD TM_DEL_SFT FOR UPSHIFTS"
 (first pass thru)                  |  (load timer)
                                    | GR_DS = GR_DS_LST
                                    |  (hold original desired gear until
                                    |   new desired gear is delayed)
                                    | FLG_SF_AUTO = 0
                                    |  (clear auto shift flag)
                                    |
GR_DS < GR_DS_LST ------------|     | --- ELSE ---
 (downshift is desired)       |     |
                              |AND -| FLG_DE_DSGR = 1
FLG_DE_DSGR = 0 --------------|     |  (set first pass flag)
 (first pass through)               | TM_DEL_SFT = TM_DEL_DOWN
                                    |  (load timer)
                                    | DO "MANUAL DOWNSHIFT DETERMINATION"
                                    | GR_DS = GR_DS_LST
                                    |  (hold original desired gear until
                                    |   new desired gear is delayed)
                                    | FLG_SF_AUTO = 0
                                    |  (clear auto shift flag)
                                    | FLG_TIP_OUT = 0
                                    |
                                    | --- ELSE ---
                                    |
                                    | FLG_DE_DSGR = 0
                                    |  (clear first pass flag)
```

```
PDL = 3 ----------------------|
                              |AND -| FLG_SF_AUTO = 0
GEAR_CUR = 4 -----------------|     |  (clear auto shift flag for a
 (driver did manual 4-3)            |   manual 4 - 3 downshift. All
                                    |   subsequent downshifts will be
                                    |   considered automatic downshifts.
                                    |   Also clear shift flag for manual
                                    |   PRNDL positions)
```

MANUAL DOWNSHIFT DETERMINATION

```
PDL = 3 ----------------------------|
                                    |AND -| FLG_DEL_MDN = 1
GEAR_CUR = 4 -----------------------|     |
 (driver did manual 4-3)                  | --- ELSE ---
                                          |
                                          | FLG_DEL_MDN = 0
```

LOAD TM_DEL_SFT FOR UPSHIFTS LOGIC

```
TP_RATE < TO_TP_RATE ---------------------| TM_DEL_SFT = TM_DEL_TO_UP
                                          | FLG_TIP_OUT = 1
                                          |
                                          | --- ELSE ---
                                          |
                                          | TM_DEL_SFT = TM_DEL_UP
                                          | FLG_TIP_OUT = 0
```

PRNDL BASED COMMANDED GEAR DETERMINATION

OVERVIEW

The commanded transmission gear is calculated based on PRNDL position. The logic looks at the current gear and the desired gear to determine if an upshift or a downshift is required and then commands the next appropriate gear in the sequence.

In the drive or overdrive position, there are no restrictions on shifts. The commanded gear routine, therefore does not have to sequence through any particular pattern. The only restriction is that downshifts are done only after the converter clutch has unlocked.

In neutral, first gear is commanded, unless a calibratible vehicle speed is reached. At this point, the gear commanded is calibratible (GR_NEU). This feature is provided to protect the direct clutch from rotating at very high speeds in neutral. In the case of a vehicle speed sensor failure, GR_NEU will be commanded.

In the manual 2 or 1 position, the downshift sequence is as follows:

1. Unlock the converter clutch and command 3rd gear (This is because commanded 4th gear in a manual low range results in 2nd intermediate band on, based on both shift solenoids being off).

2. Apply the coast clutch in 3rd gear to absorb some inertia torque.

3. When the coast clutch has engaged and the converter clutch has unlocked command second gear, intermediate band off.

4. Delay the application of the intermediate band for a calibratable period of time.

5. Command first gear if PRNDL = 1 and below the 2 - 1 pull-in speed.

The main outputs of the commanded gear routine are:

- GR_CM, commanded gear which reflects the actual shift solenoid states.

- GR_OLD, GEAR_OLD, RT_GR_OLD, last gear the transmission was in.

- FLG_FRST_CM, global flag to indicate a shift is commanded this background pass.

DEFINITIONS

INPUTS

Registers:

- GEAR_CUR = Current transmission gear (global register).

- GR_CM = Commanded gear for shift solenoids.

- GR_CM_LST = Commanded gear in last background pass.

- GR_DS = Desired transmission gear.

- PDL = Current PRNDL position.

- RT_GR_CUR = Current transmission gear ratio.

- TP_REL = Relative throttle position, counts.

- VSBART_RT = Vehicle Speed, corrected for N/V, MPH.

Bit Flags:

- FLG_FRST_DS = First time a shift is desired flag; 0 -> no shift desired, 1 -> shift desired this background pass.

- VSFMFLG = Vehicle speed sensor FMEM flag; 1 -> sensor failed.

Calibration Constants:

- FN624(TP_REL) = Time to delay downshift to unlock converter.

- GRMSFT = Gear commanded for manual shifting.

- GR_NEU = Gear commmanded in neutral at a vehicle speed above VS_NEU.

- SW_MSF = Switch to select GR_CM manually; 1 -> manual gear selection, <> 1 -> automatic gear selection.

- VS_NEU = Vehicle speed above which an alternate gear is commanded in neutral.

OUTPUTS

Registers:

- DNUN_TM = FN624(TP_REL) = Time to delay downshift for converter to unlock.

- GEAR_OLD = Last commanded transmission gear (global register).

- GR_CM = See above.

- GR_CM_LST = See above.

- GR_OLD = Last commanded gear.

- RT_GR_OLD = Last gear transmission gear ratio.

Bit Flags:

- FLG_FRST_CM = First time a shift is commanded flag; 1 -> shift commanded this background pass, 0 -> no shift commanded this background pass.

PROCESS

STRATEGY MODULE: SC_CM_GR_DETR_COM1

```
Always ----------------------------| GR_CM_LST = GR_CM
                                   |  (Update last pass current gear)



FLG_FRST_DS = 1 -------------------| DNUN_TM = FN624(TP_REL)
 (new desired gear)                |  (time to delay downshift to unlock
                                   |   converter)



SW_MSF = 1 ------------------------| GR_CM = GRMSFT
 (Manual shift selection)          |
                                   | --- ELSE ---
                                   |
PDL = 1 ---------------------------| DO "GR_CM, PRNDL = 1" LOGIC
                                   |
                                   | --- ELSE ---
                                   |
PDL = 2 ---------------------------| DO "GR_CM, PRNDL = 2" LOGIC
                                   |
                                   | --- ELSE ---
PDL = 3 OR 4 ----------------|     |
                             |AND -| GR_CM = GR_DS
GR_DS > GR_CM ---------------|     |
                                   | --- ELSE ---
PDL = 3 OR 4 ----------------|     |
                             |AND -| DO "GR_CM, PRNDL = 3 OR 4 DOWNSHIFT"
GR_DS < GR_CM ---------------|     |     LOGIC
                                   | --- ELSE ---
PDL = 5 ---------------------|     |
 (Neutral)                   |     |
                             |AND -| GR_CM = GR_NEU
VSBART_RT > VS_NEU ----|     |     |
 (Vehicle at high      |     |     |
  speed)               |OR --|     |
                       |           |
VSFMFLG = 1 -----------|           |
 (Vehicle Speed                    |
  sensor failure)                  |
                                   | --- ELSE ---
PDL = 5 ---------------------|     |
 (Neutral)                   |     |
                             |     |
PDL = 6 ---------------------|OR --| GR_CM = 1
 (Reverse)                   |
                             |
PDL = 7 ---------------------|
 (Park)
```

```
GR_CM_LST <> GR_CM ----------------| FLG_FRST_CM = 1
 (A shift has been commanded)      |  (New commanded gear for this
                                   |   program pass only)
                                   | GR_OLD = GR_CM_LST
                                   |  (Update old gear)
                                   | GEAR_OLD = GEAR_CUR
                                   | RT_GR_OLD = RT_GR_CUR
                                   |
                                   |  --- ELSE ---
                                   |
                                   | FLG_FRST_CM = 0
                                   |  (No shift this program pass)
```

GR_CM, PRNDL = 1 LOGIC

OVERVIEW

This logic determines the commanded gear when PRNDL = 1.

DEFINITIONS

INPUTS

Registers:

- DNUN_TM = FN624(TP_REL) = Time to delay downshift for converter to unlock.
- GR_CM = Commanded gear for shift solenoids.
- PDL_LST = Manual lever position previous background pass.
- TM_UNLK_CONV = Time since converter was commanded to unlock, sec.
- VSBART_RT = Filtered vehicle speed adjusted for RT_NOVS for transmission.

Bit Flags:

- FLG_CS_ENG = Coast clutch state of engagement: 1 -> coast clutch inferred to be on; 0 -> coast clutch not engaged.
- VSFMFLG = Vehicle speed sensor failure flag: 1 -> VSS failure; 0 -> no VSS failure.

Calibration Constants:

- VS21PI = Maximum vehicle speed for 2 - 1 pull in.

OUTPUTS

Registers:

- GR_CM = See above.

PROCESS

STRATEGY MODULE: SC_CM_GR_MAN1_COM1

```
GR_CM = 4 -------------------------------------| GR_CM = 3
                                                |  (always do a 4-3,
                                                |   otherwise 2nd will
                                                |   result if no action
                                                |   is taken)
PDL_LST = 5 -----------------------|            |
                                   |            |
GR_GM = 2 -------------|           |            |
                       |OR --|     |OR --|      | --- ELSE ---
GR_CM = 3 -------------|     |     |     |      |
                             |     |     |      |
TM_UNLK_CONV >= DNUN_TM -----|AND -|     |      |
 (converter clutch unlocked) |           |      |
                             |           |      |
FLG_GC_ENG = 1 --------------|           |AND -| GR_CM = 2
 (coast clutch engaged)                  |      |
                                         |      |
VSFMFLG = 1 -----------------------|     |      |
                                   |OR --|      |
VSBART_RT > VS21PI ----------------|            |
 (above 1st gear pull-in)                       |
                                                | --- ELSE ---
TM_UNLK_CONV >= DNUN_TM -----------------|      |
 (converter clutch unlocked)             |      |
                                         |      |
FLG_CS_ENG = 1 --------------------------|AND -| GR_CM = 1
 (coast clutch engaged)                  |
                                         |
VSFMFLG = 0 -----------------------------|
                                         |
VSBART_RT <= VS21PI ---------------------|
 (below 1st gear pull-in)
```

GR_CM, PRNDL = 2 LOGIC

OVERVIEW

This module determines the commanded gear when PRNDL = 2.

DEFINITIONS

INPUTS

Registers:

- DNUN_TM = FN624(TP_REL) = Time to delay downshift for converter to unlock.
- GR_CM = Commanded gear for shift solenoid.
- PDL_LST = PRNDL position last background pass.
- TM_SFT_12MN = Manual 1-2 delay timer.
- TM_UNLK_CONV = Time since converter was commanded to unlock.

Bit Flags:

- FLG_CS_ENG = Coast clutch state of engagement; 1 -> coast clutch inferred to be on, 0 -> coast clutch not engaged.
- FLG_PWR = Power mode flag; 1 -> power on mode, 0 -> power off mode.

Calibration Constants:

- TM12MN = Time to remain in 1st gear on a manual 1-2 to allow first/reverse clutch to release.

OUTPUTS

Registers:

- GR_CM = See above.
- TM_SFT_12MN = See above.

PROCESS

STRATEGY MODULE: SC_CM_GR_MAN2_COM1

```
PDL_LST = 1 -----------------|
 (manual 1-2)                |
                             |
GR_CM = 1 -------------------|AND -| TM_SFT_12MN = TM12MN
 (current gear is first)     |     |  (load timer to retain
                             |     |   first gear until the
                             |     |   low/reverse clutch
FLG_PWR = 0 -----------------|     |   released)
 (power off)                       | GR_CM = 1
                                   |
                                   | --- ELSE ---
                                   |
TM_SFT_12MN > 0 -------------------| GR_CM = 1
 (timer not expired)               |
                                   | --- ELSE ---
                                   |
GR_CM = 4 -------------------------| GR_CM = 3
                                   |  (always do a 4-3, otherwise
                                   |   2nd will result if no
                                   |   action is taken)
                                   |
TM_UNLK_CONV >= DNUN_TM -----|     | --- ELSE ---
 (converter clutch unlocked) |     |
                             |AND -| GR_CM = 2
FLG_CS_ENG = 1 --------------|
 (coast clutch engaged)
```

GR_CM, PRNDL = 3 OR 4, DOWNSHIFT LOGIC

OVERVIEW

This module determines the commanded gear on a downshift when PRNDL = 3 or 4.

DEFINITIONS

INPUTS

Registers:

- DNUN_TM = FN624(TP_REL) = Time to delay downshift for converter to unlock.
- GR_DS = Desired transmission gear.
- TM_SFT_CCO = Time since converter clutch commanded on or off during a shift, sec.

Bit Flags:

- FLG_LK_CM = Converter clutch commanded state: 1 -> command converter clutch lockup; 0 -> command converter clutch unlock.

OUTPUTS

Registers:

- GR_CM = Commanded gear for shift solenoids.

PROCESS

STRATEGY MODULE: SC_CM_GR_AUTO_DWN_COM1

```
FLG_LK_CM = 0 -----------------------|
 (converter is unlocked already, or  |
  will be due to desired downshift)  |AND -| GR_CM = GR_DS
                                     |     |  (command downshift)
TM_SFT_CCO >= DNUN_TM ---------------|
 (converter has physically unlocked)
```

LOAD SHIFT IN PROGRESS TIMER

OVERVIEW

The shift in progress timer (TM_SFT_IN) is a down-counting timer which is used in many places in the strategy to determine that a shift is currently taking place. When the timer has a value of 0, no shift is taking place. Since the E4OD transmission has no turbine speed sensor, there is no absolute way to determine the completion of a shift. A worst case time is loaded into the timer at the start of a shift. The only exception is a power-off manual downshift. In this special case, speed ratio can be monitored to infer completion of the shift. Different default values are provided for upshifts, downshifts, power on, and power off.

DEFINITIONS

INPUTS

Registers:

- GR_CM = Current transmission gear.
- GR_OLD = Last commanded gear.
- TM_DEL_SFT = Time during which a shift is delayed.
- TM_SFT_IN = Time during which shift is in progress.
- VSBART_RT = Filtered vehicle speed adjusted for RT_NOVS for transmission.

Bit Flags:

- FLG_4X4L = Flag indicating 4X4 low mode; 1 -> in 4X4 low mode.
- FLG_FRST_CM = First time a shift is commanded flag; 1 -> shift commanded this background pass, 0 -> no shift commanded this background pass.
- FLG_PWR = Power mode flag; 1 -> power on mode, 0 -> power off mode.
- FLG_SF_AUTO = Automatic upshift/downshift flag; 1 -> automatic shift (PRNDL = 3 or 4), 0 -> manual shift (PRNDL = 2 or 1).
- FLG_SFT_IN = Shift in progress flag; 1 -> shift in progress, 0 -> no shift in progress.
- FLG_SFT_MDN = Power off manual downshift flag; 1 -> power off manual downshift in progress, 0 -> power off manual downshift not in progress.

Calibration Constants:

- TCDHMF = Time delay to infer coast clutch engagement on manual downshifts at high vehicle speed, sec.
- TCDLMF = Time delay to infer coast clutch engagement on manual downshifts at low vehicle speeds, sec.

- TCDNAF = Time to complete auto downshift, power off.
- TCDNON = Time to complete downshift, power on.
- TCUPOF = Time to complete upshift, power off.
- TCUPON = Time to complete upshift, power on.
- TCUPON4L = Time to complete upshift, power on, in 4X4L mode.
- VSDNMF = Maximum vehicle speed to use TCDLMF, MPH.

OUTPUTS

Registers:

- TM_SFT_IN = See above.

Bit Flags:

- FLG_SFT_MDN = See above.
- FLG_TIP_OUT = 0 -> no tip-out upshift in progress; 1 -> a tip-out upshift in progress.
- FLG_SFT_IN = See above.

PROCESS

STRATEGY MODULE: SC_TIMER_COM1

```
FLG_FRST_CM = 1 --------------------|
 (shift has been commanded)         |
                                    |
GR_CM > GR_OLD ---------------------|
 (upshift)                          |AND -| TM_SFT_IN = TCUPON
                                    |     |  (time to complete upshift,
FLG_PWR = 1 ------------------------|     |   power on)
 (power on)                         |     | FLG_SFT_MDN = 0
                                    |     |  (not power off manual downshift)
FLG_4X4L = 0 -----------------------|     |
 (not in 4X4 low)                         |
                                          | --- ELSE ---
FLG_FRST_CM = 1 --------------------|     |
 (shift has been commanded)         |     |
                                    |     |
GR_CM > GR_OLD ---------------------|     |
 (upshift)                          |AND -| TM_SFT_IN = TCUPON4L
                                    |     |  (time to complete upshift,
FLG_PWR = 1 ------------------------|     |   power on, in 4X4 low)
 (power on)                         |     | FLG_SFT_MDN = 0
                                    |     |  (not power off manual downshift)
FLG_4X4L = 1 -----------------------|     |
 (in 4X4 low)                             |
                                          | --- ELSE ---
FLG_FRST_CM = 1 --------------------|     |
 (shift has been commanded)         |     |
                                    |     |
GR_CM > GR_OLD ---------------------|     |
 (upshift)                          |AND -| TM_SFT_IN = TCUPOF
                                    |     |  (time to complete
FLG_PWR = 0 ------------------------|     |   upshift, power off)
 (power off)                              | FLG_SFT_MDN = 0
                                          |
FLG_FRST_CM = 1 --------------------|     |
 (shift has been commanded)         |     | --- ELSE ---
                                    |     |
GR_CM < GR_OLD ---------------------|AND -| TM_SFT_IN = TCDNON
 (downshift)                        |     |  (time to complete downshift,
                                    |     |   power on)
FLG_PWR = 1 ------------------------|     |
 (power on)                               | FLG_SFT_MDN = 0
                                          |
FLG_FRST_CM = 1 --------------------|     |
 (shift has been commanded)         |     |
                                    |     | --- ELSE ---
GR_CM < GR_OLD ---------------------|     |
 (downshift)                        |     | TM_SFT_IN = TCDNAF
                                    |AND -|  (time to complete downshift,
FLG_PWR = 0 ------------------------|     |   auto, power off)
 (power off)                        |     |
                                    |     | FLG_SFT_MDN = 0
FLG_SF_AUTO = 1 --------------------|     |
 (automatic shift)                        | --- ELSE ---
```

(continued on next page)

(continued from previous page)

```
                                                   |
FLG_FRST_CM = 1 --------------------|      |
 (shift has been commanded)         |      |
                                    |      |
GR_CM < GR_OLD ---------------------|      |
 (downshift)                        |      |
                                    |AND -| TM_SFT_IN = TCDHMF
FLG_PWR = 0 ------------------------|      |  (time to complete downshift,
 (power off)                        |      |   manual, power off, high vehicle
                                    |      |   speed)
FLG_SF_AUTO = 0 --------------------|      | FLG_SFT_MDN = 1
 (manual shift)                     |      |  (manual power off downshift
                                    |      |   in progress)
VSBART_RT > VSDNMF -----------------|      |
                                           | --- ELSE ---
FLG_FRST_CM = 1 --------------------|      |
                                    |      |
GR_CM < GR_OLD ---------------------|      |
                                    |      |
FLG_PWR = 0 ------------------------|AND -| TM_SFT_IN = TCDLMF
                                    |      |  (time to complete downshift,
FLG_SF_AUTO = 0 --------------------|      |   manual, power off, low vehicle
                                    |      |   speed)
VSBART_RT <= VSDNMF ----------------|      | FLG_SFT_MDN = 1
                                           |  (manual power off downshift
                                           |   in progress)


FLG_SFT_MDN = 1 --------------------|
 (manual power off downshift        |
  in progress)                      |AND -| TM_SFT_IN = TCDNON
                                    |      |  (reset timer to power on value
FLG_PWR = 1 ------------------------|      |   if power mode changes in the
 (suddenly becomes power on)               |   middle of a downshift)
                                           | FLG_SFT_MDN = 0
                                           |  (clear manual downshift flag)


TM_SFT_IN = 0 -----------------------------| FLG_SFT_MDN = 0
 (timer expired)                           |  (manual downshift
                                           |   is complete)
                                           | FLG_SFT_IN = 0
                                           |  (no shift in progress)
                                           |
                                           | --- ELSE ---
                                           |
                                           | FLG_SFT_IN = 1
                                           |  (shift in progress)


FLG_SFT_IN = 0 ---------------------|
 (no shift in progress)             |AND -| FLG_TIP_OUT = 0
                                    |      |  (reset tip-out flag)
TM_DEL_SFT = 0 ---------------------|
 (no delay shift in progress)
```

DETERMINE SHIFT SOLENOID STATES

OVERVIEW

The shift solenoid state logic configures the shift solenoid output states based on the commanded gear (GR_CM). Most of the time, this is very straightforward. Although the solenoids have different reaction rates (SS2 takes longer to move than SS1) shifts are normally made by moving only one solenoid at a time. The exception is when PRNDL moves from 1 or 2 while GR_CM = 2 to PRNDL 3 or 4 and 2nd gear is still commanded. This requires both solenoids to go from off to on. The timer TM_SS1_GR2 takes care of this by moving SS2 early so that they both switch at the same time.

The main outputs of the shift solenoid routine, besides the shift solenoid states are:

- GEAR_CUR, global gear indicator which reflects only the transmission gear ratio, not any transmission specific combination of engine braking bands or clutches;

- RT_GR_CUR, current transmission gear ratio.

DEFINITIONS

INPUTS

Registers:

- CYCCTR = Cold shift solenoid cycling counter.

- GR_CM = Commanded gear for shift solenoids.

- NEBART = Filtered engine RPM for transmission.

- NOBART = Filtered output shaft speed.

- PDL = Current PRNDL position.

- RT_GR_CUR = Current transmission gear ratio.

- TM_SS1_GR2 = Time delay for SS1 on manual to auto 2nd gear shift.

Bit Flags:

- FLG_FRST_CM = First time a shift is commanded flag; 1 -> shift commanded this background pass, 0 -> no shift commanded this background pass.

- FLG_FRST_TV = Start-up TV pressure flag; 0 -> do start-up TV logic, 1 -> do not do start-up TV logic.

- FLG_TVENG_CD = Flag which indicates cold temperature for engagement TV; 0 -> don't use TVEMAX in engagement TV, 1 -> use TVEMAX in engagement TV.

- FLG_TVENG_MD = Flag which indicates moderate temperature for engagement TV; 0 -> don't use TVEMOD in engagement TV, 1 -> use TVEMOD in engagement TV.

- UNDSP = Flag indicating engine mode; 1 -> cranking or underspeed, 0 -> run mode.

Calibration Constants:

- CLDCTM = Cold shift solenoid cycle period (in background loops).

- GRRAT1 = First gear ratio.

- GRRAT2 = Second gear ratio.

- GRRAT3 = Third gear ratio.

- GRRAT4 = Fourth gear ratio.

- TMS1G2 = Time to delay SS1 on manual to auto 2nd gear shift.

OUTPUTS

Registers:

- CCYCTR = See above.

- GEAR_CUR = Current transmission gear (global register).

- RT_GR_CUR = See above.

- SPD_RT_STRT = Speed ratio at start of shift.

- TM_SS1_GR2 = See above.

Bit Flags:

- FLG_SS_1 = Shift solenoid 1 output state; 1 -> SS1 energized, 0 -> SS1 de-energized.

- FLG_SS_2 = Shift solenoid 2 output state; 1 -> SS2 energized, 0 -> SS2 de-energized.

PROCESS

STRATEGY MODULE: SC_SOL_CTL_COM1

```
always ---------------------------------------| Increment CYCCTR


CYCCTR >= CLDCTM -----------------------------| CYCCTR = 0


FLG_FRST_TV = 0 -----------------------|
 (no engagement yet)                   |
                                       |
CYCCTR > CLDCTM/2 ---------------------|
 (cycle shift solenoids)               |
                                       |
FLG_TVENG_MD = 1 ----------------|     |
                                 |OR --|
FLG_TVENG_CD = 1 ----------------|     |AND -| FLG_SS_1 = 0
 (moderately cold)                     |     | FLG_SS_2 = 1
                                       |     | RT_GR_CUR = GRRAT1
UNDSP = 0 -----------------------------|     | GEAR_CUR = 1
 (RUN mode)                            |     |  (do cold shift solenoid
                                       |     |   cycling strategy)
GR_CM = 1 -----------------------------|     |
                                             | --- ELSE ---
                                             |
GR_CM = 1 -----------------------------------| FLG_SS_1 = 1
 (1st gear)                                  | FLG_SS_2 = 0
                                             | RT_GR_CUR = GRRAT1
                                             | GEAR_CUR = 1
                                             |
                                             | --- ELSE ---
                                             |
GR_CM = 2 -----------------------------|     |
                                       |AND -| FLG_SS_1 = 0
PDL <= 2 ------------------------------|     | FLG_SS_2 = 0
 (2nd gear, intermediate band on)            | TM_SS1_GR2 = TMS1G2
                                             |  (load timer in case 2nd gear
                                             |   is commanded in PDL 3 or 4)
                                             | RT_GR_CUR = GRRAT2
                                             | GEAR_CUR = 2
                                             |
                                             | --- ELSE ---
GR_CM = 2 -----------------------------|     |
                                       |AND -| FLG_SS_1 = 0
TM_SS1_GR2 > 0 ------------------------|     | FLG_SS_2 = 1
                                             |   (2nd gear has been commanded
                                             |    in PDL 3 or 4. Move SS2
                                             |    early due to its longer
                                             |    response time)
                                             | RT_GR_CUR = GRRAT2
                                             | GEAR_CUR = 2
                                             |
                                             | --- ELSE ---
```

(continued on next page)

(continued from previous page)

```
                                                   |
GR_CM = 2 -----------------------------------| FLG_SS_1 = 1
 (2nd gear, intermediate                     | FLG_SS_2 = 1
  band OFF, PRNDL position                   | RT_GR_CUR = GRRAT2
  in 3 or 4)                                 | GEAR_CUR = 2
                                             |
                                             | --- ELSE ---
                                             |
GR_CM = 3 -----------------------------------| FLG_SS_1 = 0
 (3rd gear)                                  | FLG_SS_2 = 1
                                             | RT_GR_CUR = GRRAT3
                                             | GEAR_CUR = 3
                                             |
                                             | --- ELSE ---
                                             |
GR_CM = 4 -----------------------------------| FLG_SS_1 = 0
 (4th gear if PRNDL is in                    | FLG_SS_2 = 0
  4, 2nd gear if PRNDL is                    | RT_GR_CUR = GRRAT4
  in 1 or 2)                                 | GEAR_CUR = 4


                                         NOBART
FLG_FRST_CM = 1 -----------| SPD_RT_STRT = ------ * RT_GR_CUR
                                         NEBART
```

SHIFT VALIDATION LOGIC

OVERVIEW

The Shift Validation logic verifies that a shift has taken place after it is commanded. This logic is only capable of verifying automatic upshifts which take place during a steady, off idle, throttle position.

The logic works as follows: if the transmission control has just commanded an automatic upshift, and throttle position and vehicle speed are high enough, the shift is considered verifiable and the current converter clutch state, throttle position, vehicle speed and engine rpm are recorded. The logic then waits until the shift is complete; FLG_SFT_IN = 0. At that time, the logic verifies that the throttle position and the vehicle speed have not varied significantly, and the converter clutch has not changed from a locked to unlocked state. If all conditions are met, then the engine speed should decrease if a shift actually took place.

If a shift error is detected, failure mode action will be performed for a calibratable number of warm up cycles.

DEFINITIONS

INPUTS

Registers:

- C617CNT = 1-2 shift error warm up counter.
- C617FIL = 1-2 miss shift fault filter.
- C617_KAM_BIT = 1-2 shift error detected.
- C618CNT = 2-3 shift error warm up counter.
- C618FIL = 2-3 miss shift fault filter.
- C618_KAM_BIT = 2-3 shift error detected.
- C619CNT = 3-4 shift error warm up counter.
- C619FIL = 3-4 miss shift fault filter.
- C619_KAM_BIT = 3-4 shift error detected.
- GEAR_CUR = Current commanded gear.
- GEAR_OLD = Last commanded gear.
- NEBART = Filtered engine speed.
- NEV_STRT_SFT = Engine speed at start of the shift.
- SFT_STEADY = Number of steady shifts since power-up.

- SFT_TOTAL = Total shifts commanded since power-up.
- TP_REL = Relative throttle position; TP - RATCH.
- TPV_STRT_SFT = Throttle position at start of the shift.
- VSBART_RT = Filtered vehicle speed adjusted for RT_NOVS for transmission.
- VSCTR = Count vehicle speed sensor errors.
- VSV_STRT_SFT = Vehicle speed at start of the shift.

Bit Flags:

- CCV_STRT_SFT = Converter clutch position at start of shift.
- FLG_FRST_CM = New commanded gear this pass flag.
- FLG_LK_CM = Converter clutch commanded flag.
- FLG_NOV_KAM = Flag indicating at least one update of RT_NOVS_KAM.
- FLG_SF_AUTO = Automatic shift flag.
- FLG_SFT_IN = Shift in progress flag: 1 -> shift in progress.
- FLG_SFT_VAL = Shift validity flag: 0 -> Shift cannot be verified; 1 -> Shift may be verified.
- PDL_ERROR = PRNDL sensor failure; 0 -> no PRNDL sensor failure, 1 -> PRNDL sensor failure.
- TFMFLG = Throttle position FMEM flag; 1 -> throttle position failure detected.
- VSFMFLG = Vehicle speed sensor FMEM flag; 1 -> vehicle speed sensor failure detected.

Calibration Constants:

- S_VAL_NESUB = Tolerance on NE to verify an engine speed drop (negative direction) during the shift validation.
- S_VAL_TPADD = Tolerance on TP to verify steady TP (positive direction) during the shift validation.
- S_VAL_TPSUB = Tolerance on TP to verify steady TP (negative direction) during the shift validation.
- S_VAL_VSADD = Tolerance on VS to verify steady vehicle speed (positive direction) during the shift validation.
- S_VAL_VSSUB = Tolerance on VS to verify steady vehicle speed (negative direction) during the shift validation.
- SFT_FM_LVL = Total number of warm up cycles that failure mode action will be executed after a shift error is detected.

- TP_SH_VALID = Minimum TP to validate a shift.
- VS_SH_VAL2 = Minimum vehicle speed to validate an upshift to 2nd gear.
- VS_SH_VAL3 = Minimum vehicle speed to validate an upshift to 3rd gear.
- VS_SH_VAL4 = Minimum vehicle speed to validate an upshift to 4th gear.

OUTPUTS

Registers:

- NEV_STRT_SFT = Engine speed at start of the shift.
- SFT_STEADY = See above.
- TPV_STRT_SFT = See above.
- VSV_STRT_SFT = See above.

Bit Flags:

- CCV_STRT_SFT = See above.
- FLG_SFT_VAL = See above.
- SFT_ERROR = Shift error flag: 0 -> No shift error; 1 -> Shift error.
- SFT_FM_FLG = Shift error failure mode flag; 0 -> no failure mode action, 1 -> shift error failure mode action will be executed.

PROCESS

STRATEGY MODULE: SC_VALID_COM1

```
PDL_ERROR = 0 --------------------------|
 (MLPS ok)                              |
                                        |
FLG_FRST_CM = 1 ------------------------|
 (new commanded gear this pass)         |
                                        |
FLG_SF_AUTO = 1 ------------------------|
 (automatic shift)                      |
                                        |
GEAR_CUR > GEAR_OLD --------------------|
 (upshift)                              |
                                        |
TP_REL > TP_SH_VALID -------------------|
 (off idle)                             |
                                        |
GEAR_CUR = 2 ---------------|           |AND -| CCV_STRT_SFT = FLG_LK_CM
                            |AND -|     |     |  (record CC state at
VSBART_RT > VS_SH_VAL2 -----|     |     |     |   start of shift)
                                  |     |     | TPV_STRT_SFT = TP_REL
GEAR_CUR = 3 ---------------|     |     |     |  (record TP at start of
                            |AND -|OR --|     |   shift)
VSBART_RT > VS_SH_VAL3 -----|     |     |     | NEV_STRT_SFT = NEBART
                                  |     |     |  (record filtered engine
GEAR_CUR = 4 ---------------|     |     |     |   speed at start of shift)
                            |AND -|     |     | VSV_STRT_SFT = VSBART_RT
VSBART_RT > VS_SH_VAL4 -----|           |     |  (record filtered vehicle
                                        |     |   speed at start of shift)
FLG_NOV_KAM = 1 ------------------------|     | FLG_SFT_VAL = 1
 (RT_NOVS_KAM learned)                  |     |  (shift may be checked
                                        |     |   for validity)
TFMFLG = 0 -----------------------------|     |
                                        |     |
VSCTR = 0 ------------------------------|     |
                                        |     |
VSFMFLG = 0 ----------------------------|     |
                                              | --- ELSE ---
                                              |
FLG_FRST_CM = 1 ------------------------------| FLG_SFT_VAL = 0


FLG_FRST_CM = 1 ------------------------------| SFT_TOTAL = SFT_TOTAL + 1


FLG_SFT_IN = 0 -------------------------------| DO "SHIFT VERIFICATION LOGIC"
 (shift complete, may be verified)            |
                                              | --- ELSE ---
                                              |
                                              | EXIT "SHIFT VALIDATION"
                                              |  MODULE
```

SHIFT VERIFICATION LOGIC (Part 1)

```
FLG_SFT_VAL = 1 ------------------------|
 (shift can be verified)                |
                                        |
VSBART_RT > VSV_STRT_SFT - S_VAL_VSSUB -|
 (steady vehicle speed)                 |
                                        |
VSBART_RT < VSV_STRT_SFT + S_VAL_VSADD -|AND -|  (steady conditions during
 (steady vehicle speed)                 |     |   entire shift)
                                        |     | SFT_STEADY =
TP_REL < TPV_STRT_SFT + S_VAL_TPADD ----|     |   SFT_STEADY + 1
 (steady TP)                            |     | FLG_SFT_VAL = 0
                                        |     |  (reset valid shift flag)
TP_REL > TPV_STRT_SFT - S_VAL_TPSUB ----|     | Do "Part 2" and "Shift
 (steady TP)                            |     |  Error Flag Logic"
                                        |     |
FLG_LK_CM >= CCV_STRT_SFT --------------|     |
 (CC has not moved from a locked              | --- ELSE ---
  to an unlocked state during                 |
  validation)                                 |
                                              |  (unsteady conditions)
                                              | FLG_SFT_VAL = 0
                                              |  (reset valid shift flag)
                                              | Exit "Shift Validation"
                                              |  Module
```

SHIFT VERIFICATION LOGIC (Part 2)

```
GEAR_CUR = 2 ---------------------------|
                                        |AND -| error_detected = 1
NEBART > NEV_STRT_SFT - S_VAL_NESUB ----|     |  (missed 1-2 shift)
                                              | DO: FAULT FILTER for code
                                              |  617 procedure
                                              |
                                              | --- ELSE ---
                                              |
GEAR_CUR = 2 ---------------------------------|  (1-2 shift occurred)
                                              | DO: FAULT FILTER for code
                                              |  617 procedure
                                              |
                                              | --- ELSE ---
GEAR_CUR = 3 ---------------------------|     |
                                        |AND -| error_detected = 1
NEBART > NEV_STRT_SFT - S_VAL_NESUB ----|     |  (missed 2-3 shift)
                                              | DO: FAULT FILTER for code
                                              |  618 procedure
                                              |
                                              | --- ELSE ---
                                              |
GEAR_CUR = 3 ---------------------------------|  (2-3 shift occurred)
                                              | DO: FAULT FILTER for code
                                              |  618 procedure
                                              |
                                              | --- ELSE ---
GEAR_CUR = 4 ---------------------------|     |
                                        |AND -| error_detected =1
NEBART > NEV_STRT_SFT - S_VAL_NESUB ----|     |  (missed 3-4 shift)
                                              | DO: FAULT FILTER for code
                                              |  619 procedure
                                              |
                                              | --- ELSE ---
                                              |
GEAR_CUR = 4 ---------------------------------|  (3-4 shift occurred)
                                              | DO: FAULT FILTER for code
                                              |   619 procedure
```

```
C617FIL > 0 ----------------------------|
                                        |
C618FIL > 0 ----------------------------|OR --| SFT_ERROR = 1
                                        |     |
C619FIL > 0 ----------------------------|     | --- ELSE ---
                                              |
                                              | SFT_ERROR = 0


C617_KAM_BIT = 1 -----------------|
 (1-2 shift error detected)       |
                                  |AND -|
C617CNT < SFT_FM_LVL -------------|     |
 (not enough warm up cycles have        |
  passed since failure detection)       |
                                        |
C618_KAM_BIT = 1 -----------------|     |
 (2-3 shift error detected)       |     |
                                  |AND -|OR --| SFT_FM_FLG = 1
C618CNT < SFT_FM_LVL -------------|     |     |  (a shift error has been
 (not enough warm up cycles have        |     |   detected, perform failure
  passed since failure detection)       |     |   mode action for SFT_FM_LVL
                                        |     |   warm up cycles since the
C619_KAM_BIT = 1 -----------------|     |     |   last failure detection)
 (3-4 shift error detected)       |     |     |
                                  |AND -|     | --- ELSE ---
C619CNT < SFT_FM_LVL -------------|           |
 (not enough warm up cycles have              | SFT_FM_FLG = 0
  passed since failure detection)
```

CHAPTER 17

ELECTRONIC PRESSURE CONTROL

ELECTRONIC PRESSURE CONTROL GUIDE

OVERVIEW

EPC pressure is regulated by a variable force solenoid which is under EEC-IV control. The purpose of EPC pressure is to modulate the hydraulic pressure used to apply, release, and hold the various clutches and bands in the transmission. The higher the pressure, the more torque the transmission can transmit. This input torque in a conventional transmission has been approximated by either a mechanical linkage connected to the throttle plates, or a vacuum diaphragm which sees manifold vacuum. The electronic EPC strategy looks up engine torque from a table and varies the EPC pressure to contain the static capacity requirement of the transmission.

In general, EPC pressure is calculated as follows:

- Static Capacity - This is the EPC required to hold the weakest friction element due to combustion torque (TQ_NET) and inertia torque (TQ_IALPHA) during a shift. Inertia torque is 0 when a shift is not taking place. The sum of the combustion torque and inertia torque values is multiplied by the torque converter torque ratio to determine the total torque the transmission must transmit (TQ_STAT_CAP). This in turn determines the static EPC capacity requirement (TV_STAT).

- Dynamic EPC - This is EPC required to obtain acceptable shift feel and is the powertrain developers' main calibration tool. A switch is provided (SW_DYN) to allow the developer to either freeze RPM at a the start of a shift, or to allow dynamic EPC pressure to follow RPM during the shift (TV_DYN). The combustion torque always updates, even during the shift.

- AETV - Additional EPC provided on quick tip-ins to counteract the lag in EEC-IV updates to torque and to compensate for hydraulic lag times in the VFS/EPC hydraulic system.

- Total EPC - (TV_PRES) is simply the sum of static, dynamic and AETV requirements.

- Additional features -
  - Cold Starts - additional EPC can be requested to counteract the viscous effect of cold transmission oil on engagements
  - Rock cycling and high speed engagements - additional EPC can be requested to protect the transmission capacity during severe engagement conditions
  - Tip-out from stall capacity hold - delay the release of stall EPC pressure during a quick tip-out to contain powertrain wind-up
  - Stall EPC - at low speed ratio and vehicle speed stall EPC is computed as a function of throttle position
  - Coast boost - coast boost on a manual downshift is computed as a function of output shaft speed
  - Start-up - additional EPC can be requested once per start-up to "charge" the EPC system in extremely cold ambients.
  - When sensors critical to determining the correct EPC pressure have failed, or a shift error is detected, EPC is either set to the maximum value or clipped to TVFMMN as a minimum to protect the transmission. See the logic diagrams for specifics.
  - EPC is always clipped to TVPMIN as a minimum to prevent fluid drain-back.

DEFINITIONS

INPUTS

Registers:

- CS_SFT_MULT = Cold start shift multiplier.
- GEAR_CUR = Global bear indicator, reflects only.
- GR_DS_TV = Desired gear used to compute EPC pressure.
- N = RPM.
- NEBART = Filtered engine RPM for transmission, rpm.
- NEU_RES_TMR = Neutral residency timer.
- PDL = Current MLIP position.
- PDL_LST = MLIP position last background pass.
- SPD_RATIO = Speed ratio across torque converter.
- TM_DEL_SFT = Timer to verify automatic desired shift, sec.
- TM_ENG_TV = Engagement EPC pressure ramp timer, sec.
- TOT = Transmission oil temperature, deg F.
- TP = Throttle Position, counts.
- TSFETMR = Time Since first transmission engagement (sec)
- TV_COUNTS = FN620(TV_PRES) + FN622(TOT) = Requested EPC counts based transfer function and temperature compensation, counts.
- TV_PRES = EPC pressure, psi.
- VSBART_RT = Filtered vehicle speed adjusted for RT_NOVS, mph. end list
- VSCTR = Count of mph sensor errors.

Bit Flags:

- CC_FM_FLG = Converter Clutch Failure Mode flag.
- DRV2NEU_FLG = Forward prior to neutral flag.
- ETV_TEST = Flag indicating that the ETV open/short test is in progress; 1 -> test in progress.
- FLG_DEL_MDN = Flag indicating a manual downshift is being delayed; 0 -> no manual downshift is being delayed, 1 -> a manual downshift is being delayed.
- FLG_DRV_REV = Forward gear to reverse gear engagement flag; 1 -> most recent engagement was forward to reverse.

- FLG_ENG_IN = Engagement in progress flag; 1 -> engagement in progress.

- FLG_ENG_TV = Engagement EPC pressure flag; 1 -> do engagement logic, 0 -> do not do engagement logic.

- FLG_FRST_DS = First time a shift is desired flag; 0 -> no shift desired, 1 -> shift desired this background pass.

- FLG_FWD_REV = Forward to reverse or reverse to forward engagement in progress flag: 1 -> "rock cycling" engagement in progress.

- FLG_NEU_DRV = Neutral gear to forward gear engagement flag; 1 -> most recent engagement was neutral to forward.

- FLG_NEU_REV = Neutral gear to reverse gear engagement flag; 1 -> most recent engagement was neutral to reverse.

- FLG_PWR = Power mode flag; 1 -> power on mode, 0 -> power off mode.

- FLG_REV_DRV = Reverse gear to forward gear engagement flag; 1 -> most recent engagement was reverse to forward.

- FLG_SFT_IN = Shift in progress flag; 1 -> shift in progress.

- FLG_SFT_MDN = Power off manual downshift flag; 1 -> power off manual downshift in progress, 0 -> power off manual downshift not in progress.

- FLG_TVENG_MD = Flag which indicates moderate temperature for engagement EPC; 0 -> Don't use TVEMOD in engagement EPC, 1 -> Use TVEMAX in engagement EPC.

- MFMFLG = Flag indicating MAP sensor failure; 1 -> failure.

- OTMP_EPC_FLG = Flag indicating RPM is high enough to raise EPC pressure due to transmission overtemperature.

- OTEMP_FM_FLG = Transmission overtemperature FMEM flag; 1 -> Transmission is overtemperature.

- PDL_ERROR = Flag indicating a MLIP sensor failure; 1 -> failure.

- REV2NEU_FLG = Reverse prior to neutral flag.

- SFT_FM_FLG = Shift error failure mode flag; 0 -> no failure mode action, 1 -> failure mode action will be executed.

- TFMFLG = Flag indicating a TP sensor failure; 1 -> failure.

- VSFMFLG = Flag indicating a vehicle speed sensor failure; 1 -> failure.

Calibration Constants:

- CSDYN12 = Dynamic EPC multiplier for 1-2 shifts.

- CSDYN23 = Dynamic EPC multiplier for 2-3 shifts.

- CSDYN34 = Dynamic EPC multiplier for 3-4 shifts.

- EPC_OTEMP = EPC adder for transmission overtemperature.
- FN12T(TOT) = TV_PRES multiplier versus TOT for upshift to 2nd gear.
- FN21T(TOT) = TV_PRES multiplier versus TOT for downshift to 2nd gear.
- FN23T(TOT) = TV_PRES multiplier versus TOT for upshift to 3nd gear.
- FN32T(TOT) = TV_PRES multiplier versus TOT for downshift to 3nd gear.
- FN34T(TOT) = TV_PRES multiplier versus TOT for upshift to 4nd gear.
- FN43T(TOT) = TV_PRES multiplier versus TOT for downshift to 4nd gear.
- FN620(TV_PRES) = EPC VFS transfer function.
- FN622(TOT) = EPC VFS transfer function modifier as a function transmission oil temperature, counts.
- FN622A(TOT) = TV_PRES multiplier for TOT.
- NE_OTEMP_MAX = Engine speed above which EPC pressure is raised for tranmission overtemperature.
- NE_OTEMP_MIN = Engine speed below which EPC returns to normal while tranmission is overtemperature.
- NEUTIM = Minimum time in neutral to use neutral to in-gear engagement EPC functions.
- NRUN = Minimum engine speed to exit crank mode.
- RTSTAL = Maximum SPD_RATIO to do stall epc.
- TM46BLP = Time after a forward to reverse engagement to use FN46B for EPC pressure, sec.
- TM54BLP = Time after a neutral to forward engagement to use FN54B for EPC pressure, sec.
- TM56BLP = Time after a neutral to reverse engagement to use FN56B for EPC pressure, sec.
- TM64BLP = Time after a reverse to forward engagement to use FN64B for EPC pressure, sec.
- TMDRVREV = Time to complete a forward to reverse engagement, sec.
- TMNEUDRV = Time to complete a neutral to forward engagement, sec.
- TMNEUREV = Time to complete a neutral to reverse engagement, sec.
- TMREVDRV = Time to complete a reverse to forward engagement, sec.
- TVASOF = EPC pressure for power off automatic shift, psi.
- TVFMMN = Minimum EPC clip for VS, PRNDL or RPM sensor failures, psi.

- TVPMIN = Global minimum TV_PRES clip, psi.
- TVPMN1 = Minimum EPC clip, PDL = 1, GEAR = 1, psi.
- TVPMN2 = Minimum EPC clip, PDL = 1, GEAR = 2, psi.
- TVPMN3 = Minimum EPC clip, PDL = 1, GEAR = 3, psi.
- TVPMX1 = Maximum EPC clip, PDL = 1, GEAR = 1, psi.
- TVPMX2 = Maximum EPC clip, PDL = 1, GEAR = 2, psi.
- TVPMX3 = Maximum EPC clip, PDL = 1, GEAR = 3, psi.
- VSSTAL = Maximum vehicle speed to do stall EPC.

OUTPUTS

Registers:

- NEU_RES_TMR = See above.
- TM_BLP_TV = Engagement EPC pressure blip timer, sec.
- TM_ENG_TV = See above.
- TSFETMR = See above.
- TV_COUNT_LST = EPC Pressure counts last update.
- TV_COUNTS = See above.
- TV_PRES = See above.

Bit Flags:

- DRV2NEU_FLG = See above.
- FLG_DRV_REV = See above.
- FLG_ENG_IN = See above.
- FLG_ENG_TV = See above.
- FLG_FRST_TV = Start-up EPC pressure flag; 0 -> do start-up EPC logic, 1 -> do not do start-up EPC logic.
- FLG_FWD_REV = See above.
- FLG_NEU_DRV = See above.
- FLG_NEU_REV = See above.
- FLG_REV_DRV = See above.
- OTMP_EPC_FLG = See above.
- REV2NEU_FLG = See above.

PROCESS

STRATEGY MODULE: EPC_GUIDE_COM1

```
N > NRUN -------------------|
                            |AND -|
PDL <= 4 -------------------|     |
 (in fwd)                         |OR --| FLG_FRST_TV = 1
                                  |     |  (stop charging EPC circuit
N > NRUN -------------------|     |     |   after the 1st engagement)
                            |AND -|
PDL = 6 --------------------|
 (in rev)


FLG_FRST_TV = 0 ------------------| TSFETMR = 0
                                  |
                                  | --- ELSE ---
                                  |
                                  | Increment TSFETMR (every 1 sec)
```

ENGAGEMENT FLAGS/TIMERS

```
PDL = 5 --------------------|
 (in neutral)               |OR --|
                            |     |
PDL = 7 --------------------|     |AND -| NEU_RES_TMR = NEUTIM
 (in park)                        |     |  (load neutral timer)
                                  |     | DRV2NEU_FLG = 1
PDL_LST <= 4 ---------------------|     | REV2NEU_FLG = 0
 (fwd last pass)                        |  (indicate fwd prior to neu)
                                        |
                                        | --- ELSE ---
PDL = 5 --------------------|           |
 (in neutral)               |OR --|     |
                            |     |     |
PDL = 7 --------------------|     |AND -| NEU_RES_TMR = NEUTIM
 (in park)                        |     |  (load neutral timer)
                                  |     | DRV2NEU_FLG = 0
PDL_LST = 6 ----------------------|     | REV2NEU_FLG = 1
 (rev last pass)                        |  (indicate rev prior to neu)
                                        |
                                        | --- ELSE ---
                                        |
                                        | NO ACTION
```

```
PDL = 6 --------------------------|
 (in rev)                         |
                                  |
PDL_LST = 5 ----------------|     |
 (neu last pass)            |OR --|      |  (neutral to reverse)
                            |     |AND -| TM_ENG_TV = TMNEUREV + TM56BLP
PDL_LST = 7 ----------------|     |      | TM_BLP_TV = TM56BLP
 (park last pass)                 |      | FLG_DRV_REV = 0
                                  |      | FLG_NEU_DRV = 0
NEU_RES_TMR = 0 ------------|     |      | FLG_NEU_REV = 1
 (in neu long enough)       |OR --|      | FLG_REV_DRV = 0
                            |            |
DRV2NEU_FLG = 0 ------------|            |
 (not fwd prior to neu)                  |
                                         | --- ELSE ---
                                         |
PDL = 6 --------------------------|      |  (drive to reverse)
 (in rev)                         |AND -| TM_ENG_TV = TMDRVREV + TM46BLP
                                  |      | TM_BLP_TV = TM46BLP
PDL_LST <> 6 ---------------------|      | FLG_DRV_REV = 1
 (fwd to rev or                          | FLG_NEU_DRV = 0
  fwd to neu to rev with                 | FLG_NEU_REV = 0
  short time in neu)                     | FLG_REV_DRV = 0
                                         |
                                         | --- ELSE ---
PDL <= 4 -------------------------|      |
 (in fwd)                         |      |
                                  |      |
PDL_LST = 5 ----------------|     |      |
 (neu last pass)            |OR --|      |  (neutral to drive)
                            |     |AND -| TM_ENG_TV = TMNEUDRV + TM54BLP
PDL_LST = 7 ----------------|     |      | TM_BLP_TV = TM54BLP
 (park last pass)                 |      | FLG_DRV_REV = 0
                                  |      | FLG_NEU_DRV = 1
NEU_RES_TMR = 0 ------------|     |      | FLG_NEU_REV = 0
 (in neu long enough)       |OR --|      | FLG REV_DRV = 0
                            |            |
REV2NEU_FLG = 0 ------------|            |
 (not rev prior to neu)                  |
                                         | --- ELSE ---
                                         |
PDL = <= 4 -----------------------|      |  (reverse to drive)
 (in fwd)                         |AND -| TM_ENG_TV = TMREVDRV + TM64BLP
                                  |      | TM_BLP_TV = TM64BLP
PDL_LST > 4 ----------------------|      | FLG_DRV_REV = 0
 (rev to fwd or                          | FLG_NEU_DRV = 0
  rev to neu to fwd                      | FLG_NEU_REV = 0
  with short time in neu)                | FLG_REV_DRV = 1
```

```
TM_ENG_TV > 0 --------------------------| FLG_ENG_IN = 1
                                        |  (engagement in progress)
                                        |
                                        | --- ELSE ---
                                        |
                                        | FLG_ENG_IN = 0
                                        |  (no engagement in progress)


FLG_ENG_IN = 1 -------------------|
 (engagement in progress)         |
                                  |
FLG_DRV_REV = 1 ------------|     |
 (fwd to rev)               |     |
                            |     |AND -| FLG_FWD_REV = 1
FLG_REV_DRV = 1 ------------|     |     | DRV2NEU_FLG = 0
 (rev to fwd)               |     |     | REV2NEU_FLG = 0
                            |     |     |  (set direction change flag
FLG_NEU_REV = 1 ------|     |OR --|     |   and clear "prior to neutral"
 (neu to rev)         |AND -|           |   flags)
                      |     |           |
DRV2NEU_FLG = 1 ------|     |           |
 (fwd prior to neu)         |           |
                            |           |
FLG_NEU_DRV = 1 ------|     |           |
 (neu to fwd)         |AND -|           |
                      |                 |
REV2NEU_FLG = 1 ------|                 |
 (rev prior to neu)                     |
                                        | --- ELSE ---
                                        |
FLG_ENG_IN = 0 -------------------------| FLG_FWD_REV = 0
 (no engagement in progress)            |  (clear direction change flag)
                                        |
                                        | --- ELSE ---
                                        |
                                        | NO ACTION
```

```
FLG_FWD_REV = 1 ------------------|
 (direction change)               |
                                  |
PDL >= 5  ------------------------|OR --| FLG_ENG_TV = 1
 (not in forward gear)            |     |  (perform engagement/stall TV)
                                  |     |
GR_DS_TV = GEAR_CUR --------|     |     |
 (no verify in progress)    |AND -|     |
                            |           |
FLG_ENG_IN = 1 -------------|           |
 (engagement in progress)               |
                                        | --- ELSE ---
GR_DS_TV <> GEAR_CUR -------------|     |
 (desired shift being verified)   |     |
                                  |     |
FLG_FRST_DS = 1 ------------------|OR --| FLG_ENG_TV = 0
 (shift is desired)               |     |  (Stop doing engagement/stall TV;
                                  |     |   forward engagement is over and
PDL <= 4 -------------------|     |     |   trans is warm or a shift is
 (forward gear)             |AND -|     |    pending)
                            |           |
FLG_TVENG_MD = 0 -----------|           |
 (transmission warm)                    |
                                        | --- ELSE ---
                                        |
                                        | NO ACTION
```

```
PDL = 5 OR 7 ---------------------------| Do: "START-UP EPC" LOGIC
                                        |
                                        | --- ELSE ---
FLG_ENG_TV = 1 -------------------|     |
 (no shift yet after engagement)  |     |
                                  |     |
SPD_RATIO <= RTSTAL --------|     |OR --| Do: "ENGAGEMENT/STALL EPC" LOGIC
 (Low speed ratio)          |     |     |
                            |     |     |
VSBART_RT <= VSSTAL --------|AND -|     |
 (Low vehicle speed)        |           |
                            |           |
FLG_SFT_IN = 0 -------------|           |
 (No shift in progress)                 |
                                        | --- ELSE ---
FLG_PWR = 0 ----------------------|     |
 (Power off)                      |     |
                                  |     |
FLG_SFT_MDN = 1 ------------|     |     |
 (Manual downshift)         |     |AND -| Do: "COAST BOOST" LOGIC
                            |     |     |
TM_DEL_SFT > 0 -------|     |     |     |
 (Verify/delay in     |AND -|     |     |
   progress)          |     |     |     |
                      |     |     |     |
FLG_DEL_MDN = 1 ------|     |OR --|     |
 (Manual downshift is       |           |
  being verified/delayed)   |           |
                            |           |
PDL <= 3 -------------|     |           |
 (Drive, Manual1, or  |     |           |
  Manual2)            |     |           |
                      |AND -|           |
FLG_SFT_IN = 0 -------|                 |
 (No shift in         |                 |
  progress)           |                 |
                      |                 |
TM_DEL_SFT = 0 -------|                 |
 (No delay in progress)                 |
                                        | --- ELSE ---
FLG_PWR = 0 ----------------------|     |
 (Power off)                      |     |
                                  |AND -| TV_PRES = TVASOF
FLG_SFT_IN = 1 -------------|     |     |
 (Auto shift in progress)   |     |     |
                            |OR --|     |
TM_DEL_SFT > 0 -------------|           |
 (Delay in progress)                    |
                                        | --- ELSE ---
                                        |
                                        | Do: "NORMAL EPC CALCULATION"
```

Determine Transmission Overtemperature FMEM action:

```
NEBART > NE_OTEMP_MAX ------------|S Q -| OTMP_EPC_FLG
                                  |
NEBART < NE_OTEMP_MIN ------------|C


OTEMP_FM_FLG = 1 -----------------|
 (Transmission Overtemperature)   |AND -| TV_PRES = TV_PRES + EPC_OTEMP
                                  |
OTMP_EPC_FLG = 1 -----------------|
```

Clip TV_PRES as necessary:

```
TFMFLG = 1 -----------------------|
 (TP failed)                      |
                                  |
MFMFLG = 1 -----------------------|OR --| TV_PRES = 127.5 (Maximum)
 (MAP failed)                     |     |
                                  |     |
ETV_TEST = 1 ---------------------|     |
                                        | --- ELSE ---
VSFMFLG = 1 ----------------------|     |
 (VS failed)                      |     |
                                  |     |
VSCTR > 0 ------------------------|     |
                                  |     |
SFT_FM_FLG = 1 -------------------|OR --| Clip TV_PRES to TVFMMN as a minimum
 (shift error detected)           |     |
                                  |     |
CC_FM_FLG = 1 --------------------|     |
 (conv clutch error detected)     |     |
                                  |     |
PDL_ERROR = 1 --------------------|     |
 (MLPS failed)                          |
                                        | --- ELSE ---
PDL = 1 --------------------------|     |
                                  |AND -| Clip TV_PRES to TVPMX1 as a maximum
GEAR_CUR = 1 ---------------------|     | Clip TV_PRES to TVPMN1 as a minimum
                                        |
                                        | --- ELSE ---
PDL = 1 --------------------------|     |
                                  |AND -| Clip TV_PRES to TVPMX2 as a maximum
GEAR_CUR = 2 ---------------------|     | Clip TV_PRES to TVPMN2 as a minimum
                                        |
```

(continued on next page)

(continued from previous page)

```
                                             | --- ELSE ---
PDL = 1 --------------------------|          |
                                  |AND -| Clip TV_PRES to TVPMX3 as a maximum
GEAR_CUR = 3 ---------------------|     | Clip TV_PRES to TVPMN3 as a minimum
                                        |
                                        | --- ELSE ---
FLG_SFT_IN = 0 -------------------|     |
                                  |AND -| NO ACTION
GR_DS_TV = GEAR_CUR --------------|     |  (do not adjust TV_PRES for
                                        |   transmission oil temperature
                                        |   unless shifting)
                                        |
                                        | --- ELSE ---
                                        |
GR_DS_TV = 1 ---------------------------| TV_PRES = TV_PRES * FN21T(TOT)
                                        |
                                        | --- ELSE --
GR_DS_TV = 2 ---------------------|     |
                                  |     |
FLG_SFT_UP = 1 -------------------|AND -| TV_PRES = TV_PRES * FN12T(TOT) *
                                  |     |               CSDYN12
CS_SFT_MULT <> 1 -----------------|     |
                                        | --- ELSE ---
GR_DS_TV = 2 ---------------------|     |
                                  |AND -| TV_PRES = TV_PRES * FN12T(TOT)
FLG_SFT_UP = 1 -------------------|     |
                                        | --- ELSE --
                                        |
GR_DS_TV = 2 ---------------------------| TV_PRES = TV_PRES * FN32T(TOT)
                                        |
                                        | --- ELSE --
GR_DS_TV = 3 ---------------------|     |
                                  |     |
FLG_SFT_UP = 1 -------------------|AND -| TV_PRES = TV_PRES * FN23T(TOT) *
                                  |     |               CSDYN23
CS_SFT_MULT <> 1 -----------------|     |
                                        | --- ELSE ---
GR_DS_TV = 3 ---------------------|     |
                                  |AND -| TV_PRES = TV_PRES * FN23T(TOT)
FLG_SFT_UP = 1 -------------------|     |
                                        | --- ELSE --
                                        |
GR_DS_TV = 3 ---------------------------| TV_PRES = TV_PRES * FN43T(TOT)
                                        |
```

(continued on next page)

(continued from previous page)

```
                                         | --- ELSE --
GR_DS_TV = 4 ---------------------|      |
                                  |AND -| TV_PRES = TV_PRES * FN34T(TOT) *
CS_SFT_MULT <> 1 -----------------|      |            CSDYN34
                                         |
                                         | --- ELSE ---
                                         |
GR_DS_TV = 4 ---------------------------| TV_PRES = TV_PRES * FN34T(TOT)

always ---------------------------------| TV_PRES = max (TV_PRES,TVPMIN)
                                         | TV_COUNT_LST = TV_COUNTS

ETV_TEST = 0 ---------------------------| tv_comp = TV_PRES * FN622A(TOT)
                                         | TV_COUNTS = max (FN620(tv_comp) +
                                         |             FN622(TOT), 0)
                                         |
                                         | --- ELSE ---
                                         |
                                         | TV_COUNTS = 0

always ---------------------------------| Do "TV VFS OUTPUT ROUTINE"
                                         |  (set VFS_OUT_FLG = 1 for
                                         |   repeater)
```

START-UP TV LOGIC

OVERVIEW

DEFINITIONS

INPUTS

Registers:

- RATCH = Closed throttle position, counts.
- TP = Throttle position sensor.
- TP_REL = Relative TP = TP - RATCH, counts.
- TP_REL_H = Relative TP (TP - RATCH) high byte only.

Bit Flags:

- FLG_FRST_TV = Start-up TV pressure flag; 0 -> do start-up TV logic, 1 -> do not do start-up TV logic.
- FLG_TVSTR_CD = Flag which indicates cold temperature for start-up EPC; 0 -> Don't use TVCHRG in start-up EPC, 1 -> Use TVCHRG in start-up EPC.

Calibration Constants:

- FN616(TP_REL_H) = Stall EPC pressure, psi.
- TVCHRG = EPC charge pressure for first start-up, psi.

OUTPUTS

Registers:

- TV_PRES = EPC pressure, psi.

PROCESS

STRATEGY MODULE: EPC_STARTUP_COM1

```
FLG_TVSTR_CD = 1 -|
 (cold engine)    |AND -| TV_PRES = TVCHRG
                  |     |  (charge TV circuit, first start-up only)
FLG_FRST_TV = 0 --|     |
 (first start-up)       | --- ELSE ---
                        |
                        | TV_PRES = FN616(TP_REL)
```

COAST BOOST LOGIC

OVERVIEW

Coast boost TV is supplied during manual downshifts and in manual gear positions when in power off mode only.

DEFINITIONS

INPUTS

Registers:

- GEAR_CUR = Current transmission gear.
- RT_NOVS = Ratio of actual N/V to base N/V in KAM.
- VSBART_RT = Filtered vehicle speed adjusted for RT_NOVS for transmission, mph.

Calibration Constants:

- FN1CB(VSBART_RT) = First gear coast boost TV pressure.
- FN2CB(VSBART_RT) = Second gear coast boost TV pressure.
- FN3CB(VSBART_RT) = Third gear coast boost TV pressure.

OUTPUTS

Registers:

- TV_PRES = TV pressure.

PROCESS

STRATEGY MODULE: TV_CST_BOOST_COM4

```
GEAR_CUR = 1 -----| TV_PRES = FN1CB(VSBART_RT)
                  |
                  | --- ELSE ---
                  |
GEAR_CUR = 2 -----| TV_PRES = FN2CB(VSBART_RT)
                  |
                  | --- ELSE ---
                  |
GEAR_CUR >= 3 ----| TV_PRES = FN3CB(VSBART_RT)
```

NOTE: In the case of a vehicle speed sensor failure, VSBART_RT = 0. FN1CB, FN2CB, and FN3CB should be calibrated at zero vehicle speed to provide enough coast boost to cover the worst case manual downshift. These functions should therefore be calibrated as a step function from VSBART_RT 0 to 1 mph and revert to normal coast boost TV from 1 mph and higher. This will provide proper VSS failure mode protection.

ENGAGEMENT/STALL TV LOGIC

DEFINITIONS

INPUTS

Registers:

- NEBART = Filtered engine RPM for transmission.
- PDL = Current PRNDL position.
- TOT = Transmission oil tempterature, degrees F.
- TP_REL = Relative Throttle position, counts.
- TP_REL_H = Relative TP (TP - RATCH) high byte only.
- TPBARTV = Filtered TP for TV strategy.
- TM_ENG_TV = Engagament EPC pressure ramp timer.
- VSBART_RT = Filtered vehicle speed adjusted for RT_NOVS for transmission, mph.

Bit Flags:

- FLG_DRV_REV = Forward to reverse engagement flag.
- FLG_ENG_IN = Engagement in progress flag.
- FLG_ENG_TV = Engagement TV pressure flag; 0 -> do engagement logic, 1 -> do not do engagement logic.
- FLG_FWD_REV = Rock cycling engagement in progress flag.
- FLG_NEU_DRV = Neutral to drive engagement flag.
- FLG_NEU_REV = Neutral to reverse engagement flag.
- FLG_TVENG_MD = Moderate temperature for TV engagement flag.

Calibration Constants:

- FN46B_T = Engagement EPC for first TM46BLP seconds of forward to reverse engagement, psi; function of TOT.
- FN46F_T = Engagement EPC at finish of engagement ramp for a forward to reverse engagement, psi; function of TOT.
- FN46S_T = Engagement EPC at start of engagement ramp for a forward to reverse engagement, psi; function of TOT.
- FN46_NE = Multiplier for FN46S_T and FN46F_T; function of NEBART.
- FN54B_T = Engagement EPC for first TM54BLP seconds of neutral to forward engagement, psi; function of TOT.
- FN54F_T = Engagement EPC at finish of engagement ramp for a neutral to forward engagement, psi; function of TOT.
- FN54S_T = Engagement EPC at start of engagement ramp for a neutral to forward engagement, psi; function of TOT.
- FN54_NE = Multiplier for FN54S_T and FN54F_T; function of NEBART.
- FN56B_T = Engagement EPC for first TM56BLP seconds of neutral to reverse engagement, psi; function of TOT.
- FN56F_T = Engagement EPC at finish of engagement ramp for a neutral to reverse engagement, psi; function of TOT.
- FN56S_T = Engagement EPC at start of engagement ramp for a neutral to reverse engagement, psi; function of TOT.
- FN56_NE = Multiplier for FN56S_T and FN56F_T; function of NEBART.
- FN616(TP_REL_H) = Stall TV pressure.
- FN64B_T = Engagement EPC for first TM64BLP seconds of reverse to forward engagement, psi; function of TOT.
- FN64F_T = Engagement EPC at finish of engagement ramp for a reverse to forward engagement, psi; function of TOT.
- FN64S_T = Engagement EPC at start of engagement ramp for a reverse to forward engagement, psi; function of TOT.
- FN64_NE = Multiplier for FN64S_T and FN64F_T; function of NEBART.
- NETVMN = Minimum RPM to use TVEMOD engagement TV.

- NETVMX = Minimum RPM to use TVEMAX engagement TV.

- REV_ENG_ADD = Reverse Engagement Adder.

- STALLTV_SW = Calibration switch to select FN616 during engagement; 0 -> select FN616, 1 -> do not select FN616.

- TMDRVREV = Time to complete a forward to reverse engagement, sec.

- TMNEUDRV = Time to complete a neutral to forward engagement, sec.

- TMREVDRV = Time to complete a revolution to forward engagement, sec.

- TOTVTP = Tip-out TP cahnge to hold stall Tv.

- TPTVMN = Minimum TP_REL to use TVEMOD engagement TV.

- TPTVMX = Minimum TP_REL to use TVEMAX engagement TV.

- TVEMAX = TV for worst case engagement.

- VSTVMN = Minimum vehicle speed to use TVEMOD engagement TV.

- VSTVMX = Minimum vehicle speed to use TVEMAX engagement TV.

PROCESS

STRATEGY MODULE: EPC_ENGMT_STALL_COM1

ENGAGEMENT/STALL TV LOGIC

```
STALLTV_SW = 1 -----------------------|
 (FN616 selection switch)             |AND -| tempreg1 = 0
                                      |     |  (do not consider FN616
FLG_ENG_IN = 1 -----------------------|     |   during engagement)
 (engagement in progress)                   |
                                            | --- ELSE ---
                                            |
                                            | tempreg1 = FN616(TP_REL_H)
                                            |  (determine stall TV)
```

```
VSBART_RT >= VSTVMX ------------|
 (high vehicle speed)           |AND -|
                                |     |
FLG_FWD_REV = 1 ----------------|     |
 (fwd/rev or rev/fwd)                 |
                                      |OR --| tempreg2 = TVEMAX
FLG_ENG_TV = 1 -----------------|     |     |  (set maximum TV engagement
 (no shift yet)                 |     |     |   pressure)
                                |     |     | Do "REVERSE ENGAGEMENT ADDER"
TP_REL >= TPTVMX ---------|     |AND -|     |  logic
 (high TP)                |     |           |
                          |     |           |
NEBART >= NETVMX ---------|OR --|           |
 (high engine speed)      |                 |
                          |                 |
FLG_TVENG_CD = 1 ---------|                 |
 (cold temperature)                         |
                                            | --- ELSE ---
VSTVMN < VSBART_RT < VSTVMX ----|           |
 (moderate vehicle speed)       |AND -|     |
                                |     |     |
FLG_FWD_REV = 1 ----------------|     |     |
 (fwd/rev or rev/fwd)                 |     |
                                      |OR --| tempreg2 = TVEMOD
FLG_ENG_TV = 1 -----------------|     |     |  (set moderate TV engagement
 (no shift yet)                 |     |     |   pressure)
                                |     |     | Do "REVERSE ENGAGEMENT ADDER"
TPTVMN < TP_REL < TPTVMX -|     |AND -|     |  logic
 (moderate TP)            |     |           |
                          |     |           |
NETVMN < NEBART < NETVMX -|OR --|           |
 (moderate engine speed)  |                 |
                          |                 |
FLG_TVENG_MD = 1 ---------|                 |
 (moderate temperature)                     |
                                            | --- ELSE ---
                                            |
FLG_ENG_IN = 1 -----------------------------| Do "DRIVE/REVERSE ENGAGEMNENT"
                                            |     logic
                                            |
                                            | --- ELSE ---
                                            |
                                            | tempreg2 = 0
```

```
tempreg2 > tempreg1 ------------------------| tempreg1 = tempreg2
                                              (choose the larger of the stall
                                               or engagement EPC pressure)


(TPVARTV - TP) >= TOTVTP -------------|
 (high tip-out rate)                  |AND -| No change to TV_PRES
                                      |     | (use last pass value to hold
TV_PRES > tempreg1 -------------------|     |  TV pressure on tip-out)
 (last pass value has enough capacity)      |
                                            | --- ELSE ---
                                            |
                                            | TV_PRES = tempreg1
                                            | (use the higher of stall TV
                                            |  or engagement TV)
```

REVERSE ENGAGEMENT ADDER LOGIC

```
PDL = 6 ------------------------------------| tempreg2 =
                                                tempreg2 + REV_ENG_ADD
```

DRIVE/REVERSE ENGAGEMENT LOGIC

```
FLG_DRV_REV = 1 -----------------|
                                 |AND -| tempreg2 = FN46B_T(TOT)
TM_BLP_TV > 0 -------------------|     |
                                       |
                                       | --- ELSE ---
                                       |
FLG_DRV_REV = 1 -----------------------| temp_ne = FN46_NE(NEBART)
                                       | tempreg2 = FN46S_T(TOT) * temp_ne +
                                       | [FN46F_T(TOT) - FN46S_T(TOT)] *
                                       | [1 - TM_ENG_TV / TMDRVREV] *
                                       | [temp_ne]
                                       |
                                       | --- ELSE ---
FLG_NEU_DRV = 1 -----------------|     |
                                 |AND -| tempreg2 = FN54B_T(TOT)
TM_BLP_TV > 0 -------------------|     |
                                       |
                                       | --- ELSE ---
                                       |
FLG_NEU_DRV = 1 -----------------------| temp_ne = FN54_NE(NEBART)
                                       | tempreg2 = FN54S_T(TOT) * temp_ne +
                                       | [FN54F_T(TOT) - FN54S_T(TOT)] *
                                       | [1 - TM_ENG_TV / TMNEUDRV] *
                                       | [temp_ne]
                                       |
                                       | --- ELSE ---
FLG_NEU_REV = 1 -----------------|     |
                                 |AND -| tempreg2 = FN56B_T(TOT)
TM_BLP_TV > 0 -------------------|     |
                                       |
                                       | --- ELSE ---
                                       |
FLG_NEU_REV = 1 -----------------------| temp_ne = FN56_NE(NEBART)
                                       | tempreg2 = FN56S_T(TOT) * temp_ne +
                                       | [FN56F_T(TOT) - FN56S_T(TOT)] *
                                       | [1 - TM_ENG_TV / TMNEUREV] *
                                       | [temp_ne]
                                       |
                                       | --- ELSE ---
FLG_REV_DRV = 1 -----------------|     |
                                 |AND -| tempreg2 = FN64B_T(TOT)
TM_BLP_TV > 0 -------------------|     |
                                       |
                                       | --- ELSE ---
                                       |
FLG_REV_DRV = 1 -----------------------| temp_ne = FN64_NE(NEBART)
                                       | tempreg2 = FN64S_T(TOT) * temp_ne +
                                       | [FN56F_T(TOT) - FN64S_T(TOT) *
                                       | [1 - TM_ENG_TV / TMREVDRV] *
                                       | [temp_ne]
```

NORMAL TV CALCULATION

OVERVIEW

Normal TV computes the TV pressure required to maintain static capacity, that is, the capacity required when no shift is in progress.

DEFINITIONS

INPUTS

Registers:

- EPC_TQ_CONV = Torque converter static capacity EPC pressure requirement.
- GR_CM = Current transmission gear.
- GR_DS_TV = Desired gear used to compute EPC pressure.
- NEBART = Filtered engine RPM for transmission.
- RATCH = Closed throttle position, counts.
- SPD_RATIO = Speed ratio across the torque converter.
- TP = Throttle Position, counts.
- TPBARTV = Filtered TP for TV strategy. This filtered throttle position is calculated using the ROLAV subroutine, but with a different time constant then the values used in the TPBART and TPBARTC calculation.

  TPBARTV is used in the TV routine to determine a tip-in condition. During a tip-in condition in the TV routine, extra TV is added.
- TP_REL = Relative Throttle Position, TP - RATCH.
- TP_STRT_SFT = TP_REL at the start of a shift, counts.
- TQ_IALPHA = I-ALPHA torque due to ratio change.
- TQ_NET = Net torque into torque converter.
- TQ_STAT_CAP = Static Capacity Torque of transmission.
- TR_STRT_SFT = Torque ratio at the start of a shift.
- TV_DYN = TV pressure required for dynamic shift control.
- TV_PRES = TV pressure.
- TV_STAT = TV pressure required for static capacity.
- TV_ST_SFT = Static TV pressure while shifting.

Bit Flags:

- FLG_4X4L = 4X4L flag; 0 -> not in 4X4 L, 1 -> in 4X4 L.
- FLG_DYN_CD = Flag which indicates that it is necessary to add dynamic TV due to cold transmission conditions.
- FLG_FRST_CM = Flag indicating a shift was commanded this background loop.
- FLG_LK_CM = Converter clutch lock-up commanded flag.
- FLG_SFT_IN = Shift in progress flag.

Calibration Constants:

- AETV = Anticipatory TV adder for heavy tip-ins.
- AETVTP = Minimum throttle change to include AETV in TV_STAT.
- EPC_TQMAX = Maximum clip on torque converter static capacity EPC pressure requirement.
- FN617(SPD_RATIO) = Torque converter torque ratio.
- MUSLP = Torque converter coefficient of friction temperature compensation; slope.
- MUINT = Torque converter coefficient of friction temperature compensation; intercept.
- SCINT1 = Static capacity TV intercept, first gear.
- SCINT2 = Static capacity TV intercept, second gear.
- SCINT3 = Static capacity TV intercept, third gear.
- SCINT4 = Static capacity TV intercept, fourth gear.
- SCSLP1 = Static capacity TV slope, first gear.
- SCSLP1SD = Static capacity TV slope, downshift to first gear.
- SCSLP2 = Static capacity TV slope, second gear.
- SCSLP2SD = Static capacity TV slope, downshift to second gear.
- SCSLP2SU = Static capacity TV slope, upshift to second gear.
- SCSLP3 = Static capacity TV slope, third gear.
- SCSLP3SD = Static capacity TV slope, downshift to third gear.
- SCSLP3SU = Static capacity TV slope, upshift to third gear.
- SCSLP4 = Static capacity TV slope, fourth gear.
- SCSLP4SU = Static capacity TV slope, upshift to fourth gear.

- TQCONVINT = Static capacity EPC intercept for the torque converter.
- TQCONVSLP = Static capacity EPC slope for the torque converter.

OUTPUTS

Registers:

- EPC_TQ_CONV = See above.
- NE_STRT_SFT = Engine RPM at start of a shift.
- TQ_STAT_CAP = See above.
- TP_STRT_SFT = See above.
- TV_DYN = See above.
- TV_PRES = See above.

PROCESS

STRATEGY MODULE: EPC_NORM_COM7

```
FLG_FRST_CM = 1 -------------------| NE_STRT_SFT = NEBART
 (start of shift)                  | TR_STRT_SFT = FN617(SPD_RATIO)
                                   | TP_STRT_SFT = TP_REL
                                   |  (capture engine speed, torque
                                   |   ratio and throttle position at
                                   |   start of shift)
                                   | DO "TQ_IALPHA CALCULATION"
```

Now calculate the minimum static capacity.

```
FLG_SFT_IN = 1 --------------------| TQ_STAT_CAP = (TQ_NET + TQ_IALPHA) *
 (shift in progress,               |               TR_STRT_SFT
  use IALPHA term)                 |  (calculate static torque capacity
                                   |   required)
                                   |
                                   | --- ELSE ---
                                   |
FLG_SFT_IN = 0 --------------------| TQ_IALPHA = 0
 (shift done, use steady torque    | TQ_STAT_CAP = TQ_NET * FN617(SPD_RATIO)
  requirement)                     |  (calculate steady state input torque
                                   |   through torque converter ratio)
```

Calculate steady state pressure for current gear.

```
FLG_SFT_IN = 0 --------------------| DO "NON-SHIFTING STATIC TV
                                   |     CALCULATION"


FLG_SFT_IN = 0 --------------|
                             |AND -| DO "TORQUE CONVERTER STATIC TV
FLG_LK_CM = 1 ---------------|     |     CALCULATION"
```

Select larger of TV_STAT and EPC_TQ_CONV.

```
FLG_SFT_IN = 0 --------------|
                             |AND -| TV_STAT = EPC_TQ_CONV
TV_STAT < EPC_TQ_CONV -------|
```

Calculate pressure for shift using shifting slopes and Dynamic TV.

```
FLG_SFT_IN = 1 ----------------|
                               |
GR_DS_TV <> GR_CM -------------|
                               |OR --| DO "SHIFTING STATIC TV CALCULATION"
FLG_DYN_CD = 1 ----------------|     | DO "DYNAMIC TV CALCULATION"
                               |
FLG_4X4L = 1 ------------------|
```

Select proper value for TV_PRES.

```
FLG_SFT_IN = 0 ----------------|
                               |
GR_DS_TV = GR_CM --|           |
                   |           |AND -| TV_PRES = TV_STAT
FLG_4X4L = 0 ------|AND -|     |     |
                   |     |     |     |
FLG_DYN_CD = 0 ----|     |OR --|     |
                         |           |
TV_STAT > TV_ST_SFT      |           |
          + TV_DYN -----|            |
                                     | --- ELSE ---
                                     |
                                     | TV_PRES = TV_ST_SFT + TV_DYN
                                     |  (Clip TV_PRES to zero minimum)
```

Now add anticipatory TV value for a tip-in.

```
FLG_SFT_IN = 0 ----------------|
                               |AND -| TV_PRES = TV_PRES + AETV
(TP - TPBARTV) >= AETVTP ------|     |  (adjust TV pressure for heavy
                                     |   tip-in to compensate for all
                                     |   system delays)
```

NON-SHIFTING STATIC TV CALCULATION

```
GR_CM = 1 ---------------------------| TV_STAT = (TQ_STAT_CAP * SCSLP1) +
                                     |              SCINT1
                                     |
                                     | --- ELSE ---
                                     |
GR_CM = 2 ---------------------------| TV_STAT = (TQ_STAT_CAP * SCSLP2) +
                                     |              SCINT2
                                     |
                                     | --- ELSE ---
                                     |
GR_CM = 3 ---------------------------| TV_STAT = (TQ_STAT_CAP * SCSLP3) +
                                     |              SCINT3
                                     |
                                     | --- ELSE ---
                                     |
                                     | TV_STAT = (TQ_STAT_CAP * SCSLP4) +
                                     |              SCINT4
```

SHIFTING STATIC TV CALCULATION

```
GR_DS_TV = 1 ------------------------| TV_ST_SFT = (TQ_STAT_CAP * SCSLP1SD) +
                                     |                SCINT1
                                     |
                                     | --- ELSE ---
GR_DS_TV = 2 ------------------|     |
                               |AND -| TV_ST_SFT = (TQ_STAT_CAP * SCSLP2SU) +
FLG_SFT_UP = 1 ----------------|     |                SCINT2
                                     |
                                     | --- ELSE ---
                                     |
GR_DS_TV = 2 ------------------------| TV_ST_SFT = (TQ_STAT_CAP * SCSLP2SD) +
                                     |                SCINT2
                                     |
                                     | --- ELSE ---
GR_DS_TV = 3 ------------------|     |
                               |AND -| TV_ST_SFT = (TQ_STAT_CAP * SCSLP3SU) +
FLG_SFT_UP = 1 ----------------|     |                SCINT3
                                     |
                                     | --- ELSE ---
                                     |
GR_DS_TV = 3 ------------------------| TV_ST_SFT = (TQ_STAT_CAP * SCSLP3SD) +
                                     |                SCINT3
                                     |
                                     | --- ELSE ---
                                     |
                                     | TV_ST_SFT = (TQ_STAT_CAP * SCSLP4SU) +
                                     |                SCINT4
```

TORQUE CONVERTER STATIC TV CALCULATION

```
FLG_SFT_IN = 0 --------------|
                             |AND -| mu_temp = (TOT * MUSLP) +
FLG_LK_CM = 1 ---------------|     |           MUINT
                                   | EPC_TQ_CONV =
                                   |     (TQ_STAT_CAP * TQCONVSLP/mu_temp) +
                                   |      TQCONVINT
                                   |
                                   | --- ELSE ---
                                   |
                                   | EPC_TQ_CONV = 0


always-----------------------------| EPC_TQ_CONV =
                                         min(EPC_TQ_CONV, EPC_TQMAX)
```

TQ_IALPHA CALCULATION

OVERVIEW

The I-ALPHA torque calculation determines the torque that results from a transmission upshift or downshift.

DEFINITIONS

INPUTS

Registers:

- GEAR_CUR = Current transmission gear.
- GEAR_OLD = Last commanded gear.
- NE_STRT_SFT = Engine RPM at start of a shift.
- RT_GR_CUR = Current transmission gear ratio.
- RT_GR_OLD = Last gear transmission gear ratio.

Calibration Constants:

- TQIA12 = I-ALPHA torque constant for 1 - 2.
- TQIA21 = I-ALPHA torque constant for 2 - 1.
- TQIA23 = I-ALPHA torque constant for 2 - 3.
- TQIA32 = I-ALPHA torque constant for 3 - 2.
- TQIA34 = I-ALPHA torque constant for 3 - 4.
- TQIA43 = I-ALPHA torque constant for 4 - 3.

OUTPUTS

Registers:

- TQ_IALPHA = I-ALPHA torque due to ratio change.

PROCESS

STRATEGY MODULE: EPC_TQ_IALPHA_COM4

```
GEAR_CUR = 2 ----------------|
                             |AND -| TQ_IALPHA = NE_STRT_SFT * TQIA12 *
GEAR_OLD = 1 ----------------|     |   [1 - (RT_GR_CUR/RT_GR_OLD)]
                                   |
                                   | --- ELSE ---
GEAR_CUR = 1 ----------------|     |
                             |AND -| TQ_IALPHA = NE_STRT_SFT * TQIA21 *
GEAR_OLD > 1 ----------------|     |   [1 - (RT_GR_CUR/RT_GR_OLD)]
                                   |
                                   | --- ELSE ---
GEAR_CUR = 3 ----------------|     |
                             |AND -| TQ_IALPHA = NE_STRT_SFT * TQIA23 *
GEAR_OLD < 3 ----------------|     |   [1 - (RT_GR_CUR/RT_GR_OLD)]
                                   |
                                   | --- ELSE ---
GEAR_CUR = 2 ----------------|     |
                             |AND -| TQ_IALPHA = NE_STRT_SFT * TQIA32 *
GEAR_OLD > 2 ----------------|     |   [1 - (RT_GR_CUR/RT_GR_OLD)]
                                   |
                                   | --- ELSE ---
GEAR_CUR = 4 ----------------|     |
                             |AND -| TQ_IALPHA = NE_STRT_SFT * TQIA34 *
GEAR_OLD < 4 ----------------|     |   [1 - (RT_GR_CUR/RT_GR_OLD)]
                                   |
                                   | --- ELSE ---
GEAR_CUR = 3 ----------------|     |
                             |AND -| TQ_IALPHA = NE_STRT_SFT * TQIA43 *
GEAR_OLD = 4 ----------------|     |   [1 - (RT_GR_CUR/RT_GR_OLD)]
```

DYNAMIC TV CALCULATION

DEFINITIONS

INPUTS

Registers:

- GEAR_CUR = Global Gear Indicator, reflects only.
- GR_DS_TV = Commanded gear for TV.
- TOT = Transmission oil temperature, deg F.
- TP_REL_H = Relative TP (TP - RATCH) high byte only.
- TP_STRT_SFT = TP at start of shift.
- TSLSFT = Time since last shift timer.

Bit Flags:

- FLG_4X4L = 4x4 low flag.
- FLG_FRST_CM = First time a shift is commanded flag; 0 -> no shift commanded.
- FLG_SFT_DN = Downshift flag.
- FLG_SFT_IN = Shift in progress flag.
- FLG_SFT_UP = Upshift flag.

Calibration Constants:

- FN12CA = Time dependent Dynamic EPC pressure adder for 1-2 shift, psi.
- FN12_DC = Dynamic TV pressure for 1-2 shift.
- FN21_DC = Dynamic TV pressure for 2-1 shift.
- FN23CA = Time dependent Dynamic EPC pressure adder for 1-2 shift, psi.
- FN23_DC = Dynamic TV pressure for 2-3 shift.
- FN32_DC = Dynamic TV pressure for 3-2 shift.
- FN34CA = Time dependent Dynamic EPC pressure adder for 3-4 shift, psi.
- FN34_DC = Dynamic TV pressure for 3-4 shift.
- FN43_DC = Dynamic TV pressure for 4-3 shift.
- SW_DYN = Software switch for dynamic TV; 0 -> allow shift dynamics to vary with rpm, 1 -> use rpm at start of shift.

- TMTVRMP_12 = TV ramp time for 1-2 shift.
- TMTVRMP_23 = TV ramp time for 2-3 shift.
- TMTVRMP_34 = TV ramp time for 3-4 shift.
- TOTTV5 = Maximum TOT to adjust TV_DYN for cold accum.
- TV_4L_12 = Dynamic TV adder for 1-2 shift in 4x4L.
- TV_4L_21 = Dynamic TV adder for 2-1 shift in 4x4L.
- TV_4L_23 = Dynamic TV adder for 2-3 shift in 4x4L.
- TV_4L_32 = Dynamic TV adder for 3-2 shift in 4x4L.
- TV_4L_34 = Dynamic TV adder for 3-4 shift in 4x4L.
- TV_4L_43 = Dynamic TV adder for 4-3 shift in 4x4L.
- TVRMP_12 = TV adder during 1-2 upshift.
- TVRMP_23 = TV adder during 2-3 upshift.
- TVRMP_34 = TV adder during 3-4 upshift.

OUTPUTS

Registers:

- TSLSFT = See above.
- TV_DYN = TV pressure required for dynamic shift control.
- TV_RAMP = TV value added during upshift.
- TV_RAMP_TMR = TV adder during upshift control timer.
- TVRMPTM = TV Ramp Timer initail value for upshifts.

PROCESS

STRATEGY MODULE: EPC_DYNAMIC_COM1

```
SW_DYN = 1 --------------|
                         |AND -| tempreg = TP_STRT_SFT
FLG_SFT_IN = 1 ----------|     |  (Do not allow shift
                               |   dynamics to vary)
                               |
                               | --- ELSE ---
                               |
                               | tempreg = TP_REL_H
                               |  (Allow shift dynamics to vary)


FLG_FRST_CM = 1 ---------------| TSLSFT = 0
                               |
                               | --- ELSE ---
                               |
                               | Increment TSLSFT


GR_DS_TV = 1 ------------------| TV_DYN = FN21_DC(tempreg)
                               |
                               | --- ELSE ---
GR_DS_TV = 2 ------------|     |
                         |AND -| TV_DYN = FN12_DC(tempreg)
FLG_SFT_UP = 1 ----------|     |
                               | --- ELSE ---
GR_DS_TV = 2 ------------|     |
                         |AND -| TV_DYN = FN32_DC(tempreg)
FLG_SFT_DN = 1 ----------|     |
                               | --- ELSE ---
GR_DS_TV = 3 ------------|     |
                         |AND -| TV_DYN = FN23_DC(tempreg)
FLG_SFT_UP = 1 ----------|     |
                               | --- ELSE ---
GR_DS_TV = 3 ------------|     |
                         |AND -| TV_DYN = FN43_DC(tempreg)
FLG_SFT_DN = 1 ----------|     |
                               | --- ELSE ---
                               |
GR_DS_TV = 4 ------------------| TV_DYN = FN34_DC(tempreg)
```

COLD ACCUMULATOR DYNAMIC TV COMPENSATION

```
TOT >= TOTTV5 -----------|
                         |OR --| Do not adjust TV_DYN for a cold accumulator
FLG_SFT_UP = 0 ----------|     |
                               | --- ELSE ---
                               |
GEAR_CUR = 1 ------------|     |
                         |OR --| TV_DYN = TV_DYN + FN12CA(TSLSFT)
GEAR_CUR = 2 ------------|     |
                               |
                               | --- ELSE ---
                               |
GEAR_CUR = 3 ------------------| TV_DYN = TV_DYN + FN23CA(TSLSFT)
                               |
                               | --- ELSE ---
                               |
GEAR_CUR = 4 ------------------| TV_DYN = TV_DYN + FN34CA(TSLSFT)
```

4X4L DYNAMIC TV COMPENSATION

```
FLG_4X4L = 0 ------------------| Do not adjust TV_DYN for 4X4L mode.
                               |
                               | --- ELSE ---
                               |
GEAR_CUR = 1 ------------------| TV_DYN = TV_DYN + TV_4L_21
                               |
                               | --- ELSE ---
GEAR_CUR = 2 ------------|     |
                         |AND -| TV_DYN = TV_DYN + TV_4L_12
FLG_SFT_UP = 1 ----------|     |
                               | --- ELSE ---
GEAR_CUR = 2 ------------|     |
                         |AND -| TV_DYN = TV_DYN + TV_4L_32
FLG_SFT_DN = 1 ----------|     |
                               | --- ELSE ---
GEAR_CUR = 3 ------------|     |
                         |AND -| TV_DYN = TV_DYN + TV_4L_23
FLG_SFT_UP = 1 ----------|     |
                               | --- ELSE ---
GEAR_CUR = 3 ------------|     |
                         |AND -| TV_DYN = TV_DYN + TV_4L_43
FLG_SFT_DN = 1 ----------|     |
                               | --- ELSE ---
                               |
GEAR_CUR = 4 ------------------| TV_DYN = TV_DYN + TV_4L_34
```

```
FLG_FRST_CM = 1 ---------|
                         |
FLG_SFT_UP = 1 ----------|AND -| TVRMPTM = TMTVRMP_12
                         |     | TV_RAMP_TMR = TMTVRMP_12
GR_DS_TV = 2 ------------|     | TV_RAMP = TVRMP_12
                               |
                               | --- ELSE ---
FLG_FRST_CM = 1 ---------|     |
                         |     |
FLG_SFT_UP = 1 ----------|AND -| TVRMPTM = TMTVRMP_23
                         |     | TV_RAMP_TMR = TMTVRMP_23
GR_DS_TV = 3 ------------|     | TV_RAMP = TVRMP_23
                               |
                               | --- ELSE ---
FLG_FRST_CM = 1 ---------|     |
                         |     |
FLG_SFT_UP = 1 ----------|AND -| TVRMPTM = TMTVRMP_34
                         |     | TV_RAMP_TMR = TMTVRMP_34
GR_DS_TV = 4 ------------|     | TV_RAMP = TVRMP_34
                               |
                               | --- ELSE ---
                               |
FLG_FRST_CM = 1 ---------|     |
                         |AND -| TVRMPTM = 1
FLG_SFT_DN = 1 ----------|     | TV_RAMP_TMR = 1
                               | TV_RAMP = 0


FLG_SFT_IN = 1 ----------------| TV_DYN = TV_DYN + TV_RAMP * [(TVRMPTM -
                               |           TV_RAMP_TMR) / TVRMPTM]
                               |
                               | --- ELSE ---
                               |
                               | No action
                               |  (shift is being verified; do not apply
                               |   the upshift TV_RAMP to TV_DYN)
```

OFMFLG LOGIC

OVERVIEW

The Electronic Pressure Control VFS is a current control device with a 0 to 1 amp control range. One amp produces the lowest line pressure (handles idle torque capacity). The ETV Overcurrent Circuit Monitor, ETVOCM, is a voltage representative of the current in the ETV solenoid. This voltage is read by the EEC-IV through the A/D.

The monitor is checked once per background loop after ETV current has had time to stabilize to verify that the solenoid is operating within specification. The expected monitor voltage is a function of battery voltage and solenoid current. Measured voltage decreases with increasing current or decreasing battery voltage.

If the measured voltage is less than the allowable minimum (ETV_OCM_MIN), the ETV solenoid failure flag, ETV_ERROR, is set and an "ETV Test" sequence is initiated. This sequence commands zero TV_COUNTS and then rechecks the voltage to differentiate an open from a short. Through the use of a "pull-up" circuit connected to the ETV output, at zero TV_COUNTS, a short will result in low voltage on ETVOCM, while an open will show a near-normal voltage. If a short is detected, the torque truncation/failure mode routine is enabled by setting the OFMFLG to 1. If an open is detected, the EPC_OPEN_FLG is set to 1. If either the OFMFLG or the EPC_OPEN_FLG and the EPC_ERR_SW is set, the TCIL will flash.

DEFINITIONS

INPUTS

Registers:

- ETVOCM = Actual ETV monitor voltage, counts.
- ETV_OCM_MIN = Minimum acceptable ETVOCM, counts.
- TM_TV_SS = ETV current settling timer, sec.
- TV_COUNTS = Commanded ETV current, counts.
- UNDSP = In Underspeed mode flag; 1 -> in Underspeed or Crank, 0 -> in run mode.
- VBAT = Battery Voltage, volts.

Bit Flags:

- ETV_TEST = Flag indicating that the ETV open/short test is in progress; 1 -> test in progress.

Calibration Constants:

- ETV_BIAS = ETVOCM voltage at 10 volts VBAT and 0 counts commanded ETV, counts.

- ETV_GAIN = ETVOCM gain per one count of commanded ETV, counts/count.

- ETV_GAIN_BAT = ETVOCM count gain per one battery volt, counts/volt.

- TV_SLT_TM = Minimum time after a significant change in TV_COUNTS before performing the current monitor test. Used to allow ETV current to stabilize before measurement, sec.

- TVCDLT = Minimum change in TV_COUNTS from previous value to reset TM_TV_SS, counts.

OUTPUTS

Registers:

- TM_TV_SS = See above.

Bit Flags:

- EPC_OPEN_FLG = Indicates EPC open circuit; 1 -> EPC open circuit detected.

- ETV_ERROR = ETV Solenoid error.

- ETV_TEST = See above.

- OFMFLG = ETV Overcurrent Monitor Failure flag.

PROCESS

STRATEGY MODULE: EPC_OFM_COM2

```
|Previous TV_COUNTS - TV_COUNTS|> TVCDLT ----| TM_TV_SS = 0
                                             |
                                             | --- ELSE ---
                                             |
                                             | Increment TM_TV_SS


VBAT >= 10.0 -------------------------|
                                      |
UNDSP = 0 ----------------------------|AND -| Do "ETV Current
                                      |     |  Monitor Test"
TM_TV_SS >= TV_SLT_TM ----------------|     |
                                            | --- ELSE ---
                                            |
                                            | Exit Routine
```

ETV CURRENT MONITOR TEST

Calculate ETV_OCM_MIN:

```
ETV_OCM_MIN = (ETV_BIAS - (ETV_GAIN * TV_COUNTS)) +
                    (Clip to zero)
              ((VBAT - 10.0) * ETV_GAIN_BAT)
                    (Clip to zero)
```

```
                                          | (Initiate Open/Short Test)
ETVOCM < ETV_OCM_MIN ------------|        |
                                 |AND -| ETV_ERROR = 1
ETV_TEST = 0 --------------------|     | OFMFLG = 0
                                       | EPC_OPEN_FLG = 0
                                       | ETV_TEST = 1
                                       | TM_TV_SS = 0
                                       |
                                       | --- ELSE ---
                                       |
                                       |  (Over Current/Short Circuit)
ETVOCM < ETV_OCM_MIN ------------|     |
                                 |AND -| ETV_ERROR = 1
ETV_TEST = 1 --------------------|     | OFMFLG = 1
                                       | EPC_OPEN_FLG = 0
                                       | ETV_TEST = 1
                                       |
                                       | --- ELSE ---
                                       |
                                       | (Under Current/Open Circuit)
ETVOCM >= ETV_OCM_MIN -----------|     |
                                 |AND -| ETV_ERROR = 1
ETV_TEST = 1 --------------------|     | OFMFLG = 0
                                       | EPC_OPEN_FLG = 1
                                       | ETV_TEST = 0
                                       | TM_TV_SS = 0
                                       |
                                       | --- ELSE ---
                                       |
                                       | (No Error)
                                       |
                                       | ETV_ERROR = 0
                                       | OFMFLG = 0
                                       | EPC_OPEN_FLG = 0
                                       | ETV_TEST = 0
```

TV VFS OUTPUT ROUTINE

OVERVIEW

TV pressure is controlled by a variable force solenoid in the transmission. Current to the VFS ranges from 0 to 1 amp_. and is controlled by a custom integrated circuit which converts an 8 bit binary value to a proportional current using the following relationship:

I ave. = requested counts/255

This 8 bit word is clocked into the IC via two low speed output lines from the EEC-IV called data and clock. The following conventions must be observed:

1) Both data and clock lines shall be held high between data words

2) Holding the data line low during a falling edge of the clock initiates an input sequence at any point in time (even in the middle of a data sequence). The LSB (Least Significant Bit) is assumed to follow next, bit 7

3) A rising edge of the clock shifts the current state of the data line into the IC buffer.

4) The eighth rising edge indicates that the MSB (Most Significant Bit), bit 0, has been shifted in. This latches the new 8 bit word in the IC and forces the output current to the commanded level.

DEFINITIONS

INPUTS

Registers:

- TV_COUNTS = FN620(TV_PRES) = Requested TV counts based on transfer function.

OUTPUTS

Bit Flags:

- FLG_TV_CLK = TV VFS clock line.

- FLG_TV_DATA = TV VFS data line.

PROCESS

STRATEGY MODULE: EPC_VFS_OUT_COM2

```
One time --------------------------| FLG_TV_DATA = 0
                                   | PAUSE
                                   | FLG_TV_CLK = 0
                                   | PAUSE
                                   |  (initiate input sequence)

Repeat eight times ----------------| FLG_TV_DATA = TV_COUNTS, bit 0
                                   | PAUSE
                                   | FLG_TV_CLK = 1
                                   | PAUSE
                                   |  (Latch data bit)
                                   | FLG_TV_DATA = 1
                                   | PAUSE
                                   | FLG_TV_CLK = 0
                                   | PAUSE
                                   |  (set up for next bit)
                                   | Shift TV_COUNTS right, 1 bit

one time --------------------------| FLG_TV_DATA = 1
                                   | FLG_TV_CLK = 1
                                   |  (return output to rest state)
```

TRANSMISSION OVERTEMPERATURE TEST

OVERVIEW

At very high temperatures, the torque capacity of the transmission is decreased, which can impact its durabilty. When a high transmission oil temperature is reached, the fault filter for the Transmission overtemperature code will count up. Once the KAM bit is set, the OTEMP_FM_FLG is set. The OTEMP_FM_FLG is used throughout the strategy for FMEM action. The OTEMP_FM_FLG will remain set until the transmission drops below the overtemperature threshold minus the hysteresis.

DEFINITIONS

INPUTS

Registers:

- TOT = Transmission oil temperature, deg F.

Calibration Constants:

- C657LVL = Threshold for Transmission overtemperature fault.
- C657UP = Transmission overtemperature fault up-count.
- TOT_OTEMP = Temperature above which the Transmission overtemperature fault filter is called.
- HYS_OTEMP = Delta Temperature below which the Transmission is considered no the be overtemperature.

OUTPUTS

Bit Flags:

- OTEMP_FM_FLG = Transmission overtemperature FMEM flag; 1 -> Transmission is overtemperature.

PROCESS

STRATEGY MODULE: EPC_OTEMP_TEST_COM2

```
TOT > TOT_OTEMP ---------------------------| error_detected = 1
(Transmission oil temperature above        | Call fault filter for
 overtemperature threshold)                | Code 657
                                           | OTEMP_FM_FLG = 1
                                           | (Execute Transmission
                                           |  overtemperature failure
                                           |  mode action)
                                           |
                                           | --- ELSE ---
                                           |
TOT <= TOT_OTEMP - HYS_OTEMP --------------| error_detected = 0
                                           | Call fault filter for
                                           | Code 657
                                           | OTEMP_FM_FLG = 0
```

CHAPTER 18

CONVERTER CLUTCH CONTROL

CONVERTER CLUTCH CONTROL

OVERVIEW

The converter clutch is an electronically controlled wet clutch which essentially bypasses the torque converter when actuated. This eliminates torque converter slippage and excess heat generation leading to fuel economy and vehicle performance benefits. The converter clutch is also used during transmission gear changes to minimize the customer perceived engine RPM change during upshifts and downshifts. Hydraulically, the converter clutch circuit is a two pass (two circuit) system. This allows only on/off control of the application of the clutch.

The converter clutch strategy is broken down into four basic parts:

1) Unconditional Unlock Logic - The converter clutch is unconditionally released under a number of conditions such as closed throttle, brake applied, high tip-in or tip-out rates, etc. A timer is then loaded to allow re-application of the converter clutch after the condition is no longer true. All times are independent such that the longest converter clutch release time for multiple release conditions controls re-application of the clutch.

   There are two versions of the E4OD control logic; one for diesel engine applications and one for gasoline engine applications. To minimize software/strategy workload, the majority of the logic is kept generic, so that it may be used for both versions.

   In the converter clutch routine, it is desired to unconditionally unlock the converter clutch during closed throttle. The means by which closed throttle is determined is different in the two versions. Therefore, a dummy variable, DD_UNC_UNL, is created. This variable is loaded with TP_REL in the gasoline version. The entire converter clutch routine would then be generic between the two strategies.

2) Shift Unlock Logic - Shift unlock logic is used to control the converter clutch during upshifts and downshifts. For power on upshifts, speed ratio is monitored after the gear change begins and is used to release the converter clutch. In this way, the drop in RPM caused by the ratio change is offset by an RPM rise due to the release of the converter clutch. After the converter clutch is released, speed ratio is monitored again to reapply the converter clutch. The end result is that the RPM change during the shift has been minimized and converter clutch control has been imperceptible to the driver. If speed ratio conditions are not met a default timer controls the converter clutch. The speed ratio check, due to the accuracy required for proper control, is done during the 1 msec. interrupt. Power off upshifts and all downshifts are controlled by the default timer due to the lower required timing accuracy. Downshifts are all performed on an open converter to aid shift quality.

3) Scheduled Lock/Unlock Logic - When there are no unconditional releases in effect and the converter clutch is not being controlled during shifts, converter clutch applies and releases are scheduled as a function of throttle position versus vehicle speed for each gear. An identical set of fox functions exists for altitude as well (BP_INTR is the interpolation factor). Vehicle speeds are modified by the learned N over V of the vehicle. Some additional features exist for scheduled converter clutch applies.

   a) Speed ratio must be greater than a minimum value. This prevents application of the converter clutch while significant torque multiplications are taking place.

   b) Throttle rate must be less than some maximum rate. This prevents application of the converter clutch when driver business would continually release and apply the converter clutch.

   c) Intermediate altitude scheduled apply delay. This delays re-application of the converter clutch when driving in mountainous terrain to prevent business due to constantly changing throttle position associated with driving up and down hills.

   d) W.O.T. Lockup Logic - WOT lockup logic is used to apply the converter clutch when at Wide Open Throttle to realize a performance and efficiency benefit. In any gear other than first, the converter clutch is automatically applied. In first gear, a minimum speed ratio criteria must be met. If the converter clutch must be released due to increasing load. The speed ratio criteria becomes increasingly more difficult so as to prevent cycling of the clutch.

4) FMEM Lock-up Logic - If a Transmission sensor critical to the Converter Clutch Control Logic fails, Lock-ups will be based on information from the available sensors.

   a) If the TP Sensor fails, locks are based on SPD_RATIO. If both the TP Sensor and the VS Sensor fail, locks are based on Engine RPM.

   b) If the Vehicle Speed Sensor fails, locks are based on a function of NEBART and TP_REL.

DEFINITIONS

INPUTS

Bit Flags:

- FLG_CRV_LK = Scheduled curve lock-up flag; 0 -> no scheduled lock-up, 1 -> scheduled lock-up.

- FLG_FMM_LK = Failure Mode Management lock-up flag; 1 -> lock converter due to FMEM action.

- FLG_SFT_UNLK = Shift control unlock flag; 0 -> no shift control unlock, 1 -> shift control unlock.

- FLG_UNC_UNLK = Converter clutch unconditional unlock flag; 0 -> no unconditional unlock, 1 -> unconditional unlock.

- FLG_WOT_LK = WOT lock-up flag; 0 -> no WOT lock-up, 1 -> WOT lock-up.

Calibration Constants:

- SW_MLK = Switch for manual converter clutch control; 0 -> automatic converter clutch control, 1 -> unconditional converter clutch lock-up, 2 -> unconditional converter clutch unlock.

OUTPUTS

Bit Flags:

- FLG_LK_CM = Converter clutch lock-up commanded flag; 0 -> de-energize solenoid, unlock converter clutch, 1 -> energize solenoid, lock converter clutch.

PROCESS

STRATEGY MODULE: CCC_COM1

```
always ------------------------------| Do "UNCONDITIONAL UNLOCK" LOGIC
                                     | Do "SHIFT UNLOCK" LOGIC
                                     | Do "SCHEDULED LOCK/UNLOCK" LOGIC
                                     | Do "WOT LOCK-UP" LOGIC
                                     | Do "FAILURE MODE MANAGEMENT LOCK-UP"
                                          LOGIC


SW_MLK = 1 --------------------------| FLG_LK_CM = 1
 (Development unconditional lock-up) |  (Lock converter clutch)
                                     |
                                     | --- ELSE ---
FLG_UNC_UNLK = 1 --------------|     |
 (Unconditional unlock)        |OR --| FLG_LK_CM = 0
                               |     |  (Unlock converter clutch)
FLG_SFT_UNLK = 1 --------------|     |
 (Shift control unlock)              |
                                     | --- ELSE ---
FLG_WOT_LK = 1 ----------------|     |
 (WOT lockup)                  |     |
                               |     |
FLG_CRV_LK = 1 ----------------|OR --| FLG_LK_CM = 1
 (Scheduled curve lock-up)     |     |  (Lock converter clutch)
                               |     |
FLG_FMM_LK = 1 ----------------|     |
 (FMEM lock-up)                      |
                                     | --- ELSE ---
                                     |
                                     | FLG_LK_CM = 0
                                     |  (Unlock converter clutch)
```

UNCONDITIONAL UNLOCK LOGIC

DEFINITIONS

INPUTS

Registers:

- BP = Barometric pressure.
- CS_SFT_MULT = Cold start shift multiplier.
- DD_UNC_UNL = The driver demand position used in the unconditional unlock routine, counts.
- GEAR_CUR = Current transmission gear.
- GEAR_OLD = Last commanded gear.
- GR_CM = Commanded gear for shift solenoids.
- GR_DS = Desired transmission gear.
- NEBART = Filtered engine speed, RPM.
- PDL = Current PRNDL position.
- TM_LK_DLY = Timer for converter clutch unconditional unlock.
- TM_UN_CT = Timer for closed throttle converter clutch relock.
- TP_RATE = Throttle position rate.
- TP_REL = Relative Throttle Position.
- VSBART_RT = Filtered vehicle speed adjusted for RT_NOVS for transmission, MPH.

Bit Flags:

- BIFLG = Brake applied flag; 0 -> brake not applied, 1 -> brake applied.
- FLG_FRST_CM = First time a shift is commanded flag; 0 -> no shift commanded
  this background pass, 1 -> shift commanded this background pass.
- FLG_LK_CM = Converter clutch lock-up commanded flag; 0 -> de-energize solenoid, unlock converter clutch, 1 -> energize solenoid, lock converter clutch.
- FLG_UN_ALT = High altitude unconditional unlock flag; 0 -> not at high altitude, 1 -> high altitude.
- FLG_UN_BRK = Brake applied unconditional unlock flag; 0 -> brake not applied, 1 -> brake applied.

- FLG_UN_CT = Closed throttle unconditional unlock flag; 0 -> not closed throttle, 1 -> closed throttle.

- FLG_UN_MDN = Manual downshift sequence unconditional unlock flag; 0 -> no manual downshift unlock, 1 -> manual downshift unlock.

- FLG_UN_NE = Engine speed unconditional unlock flag; 0 -> no low engine speed unlock, 1 -> low engine speed unlock.

- FLG_UN_PRN = PRNDL position unconditional unlock flag; 0 -> no PRNDL position unconditional unlock, 1 -> PRNDL in park, reverse, neutral, or manual one.

- FLG_UN_TEMP = Cold temperature unconditional unlock flag; 0 -> no cold unlock, 1 -> cold unlock.

- FLG_UN_TRA = Throttle rate accel unconditional unlock flag; 0 -> not high positive throttle rate, 1 -> high positive throttle rate.

- FLG_UN_TRD = Throttle rate decel unconditional unlock flag; 0 -> not high negative throttle rate, 1 -> high negative throttle rate.

- FLG_UN_ULSF = Unlocked shift unconditional unlock flag; 0 -> locked up prior to start of shift, 1 -> unlocked prior to start of shift.

- OFMFLG = ETV overcurrent monitor failure flag; 0 -> ETV O.K., 1 -> ETV failure mode.

- TFMFLG = TP FMEM flag = 0 -> no TP failure; 1 -> TP failure, operating in FMEM mode.

Calibration Constants:

- BPUNMH = Hysteresis for BPUNMN.

- BPUNMN = Minimum BP to unlock converter clutch.

- BRKDLY = Brake relock delay.

- CTDLY = Closed throttle relock delay.

- D21DLY = Unlocked 2 -> 1 shift relock delay.

- D32DLY = Unlocked 3 -> 2 shift relock delay.

- D43DLY = Unlocked 4 -> 3 shift relock delay.

- LUDLY = High altitude/cold engine relock delay.

- NELUMN = Minimum NEBART to lock-up converter clutch.

- PRNDLY = PRNDL relock delay.

- SW_MLK = Switch for manual converter clutch control; 0 -> automatic converter clutch control, 1 -> unconditional converter clutch lock-up, 2 -> unconditional converter clutch unlock.

- TMCTDY = Minimum time before relock at closed throttle.
- TPUNBRK = Maximum relative TP for brake applied unlock.
- TPUNCH = Hysteresis for TPUNCT.
- TPUNCT = Maximum relative TP for closed throttle unlock.
- TPUNTR = Relative TP breakpoint for high and low unlock throttle rate.
- TRADLY = High throttle rate, accel, relock delay.
- TRDDLY = High throttle rate, decel, relock delay.
- TRUHAC = Throttle rate unlock, high TP, accel.
- TRULAC = Throttle rate unlock, low TP, accel.
- TRUHDC = Throttle rate unlock, high TP, decel.
- TRULDC = Throttle rate unlock, low TP, decel.
- U12DLY = Unlocked 1 -> 2 shift relock delay.
- U23DLY = Unlocked 2 -> 3 shift relock delay.
- U34DLY = Unlocked 3 -> 4 shift relock delay.
- VSCTDY = Minimum vehicle speed for closed throttle relock.

OUTPUTS

Registers:

- TM_LK_DLY = See above.
- TM_UN_CT = See above.

Bit Flags:

- FLG_UN_ALT = See above.
- FLG_UN_BRK = See above.
- FLG_UN_CT = See above.
- FLG_UN_MDN = See above.
- FLG_UN_NE = See above.
- FLG_UN_PRN = See above.
- FLG_UN_TEMP = See above.
- FLG_UN_TRA = See above.
- FLG_UN_TRD = See above.
- FLG_UN_ULSF = See above.

PROCESS

STRATEGY MODULE: CCC_UNCOND_UNLCK_COM1

DD_UNC_UNL = TP_REL (Performed during engineering units conversion)

```
DD_UNC_UNL < TPUNCT --------------|S Q--| FLG_UN_CT = 1
                                  |     |  (Closed throttle unlock)
DD_UNC_UNL >= TPUNCT + TPUNCH ----|C    |
                                        | --- ELSE ---
                                        |
                                        | FLG_UN_CT = 0


TP_RATE >= TRUHAC ---------|
                           |AND -|
DD_UNC_UNL >= TPUNTR ------|     |
                                 |OR --| FLG_UN_TRA = 1
TP_RATE >= TRULAC ---------|     |     |  (High tip-in rate)
                           |AND -|     |
DD_UNC_UNL < TPUNTR -------|           | --- ELSE ---
                                       |
                                       | FLG_UN_TRA = 0


TP_RATE <= TRUHDC --------|
                          |AND -|
DD_UNC_UNL >= TPUNTR -----|     |
                                |OR --| FLG_UN_TRD = 1
TP_RATE <= TRULDC --------|     |     |  (High tip-out rate)
                          |AND -|     |
DD_UNC_UNL < TPUNTR ------|           | --- ELSE ---
                                      |
                                      | FLG_UN_TRD = 0


BIFLG = 1 ----------------------|     | FLG_UN_BRK = 1
                                |AND -|  (Brake applied)
DD_UNC_UNL <= TPUNBRK ----------|     |
                                      | --- ELSE ---
                                      |
                                      | FLG_UN_BRK = 0


PDL = 7 OR 6 or 5 or 1 ---------------| FLG_UN_PRN = 1
                                      |  (Park, reverse, neutral, or
                                      |   manual one)
                                      |
                                      | --- ELSE ---
                                      |
                                      | FLG_UN_PRN = 0
```

```
CS_SFT_MULT <> 1.0 -------------------| FLG_UN_TEMP = 1
                                      |  (Cold engine)
                                      |
                                      | --- ELSE ---
                                      |
                                      | FLG_UN_TEMP = 0


PDL = 2 OR 1 -------------------|
                                |AND -| FLG_UN_MDN = 1
GR_DS <> GR_CM -----------------|     |  (Manual downshift sequence
                                      |   not completed)
                                      |
                                      | --- ELSE ---
                                      |
                                      | FLG_UN_MDN = 0


BP <= BPUNMN -------------------|S Q--| FLG_UN_ALT = 1
                                |     |  (High altitude)
BP > BPUNMN + BPUNMH -----------|C    |
                                      | --- ELSE ---
                                      |
                                      | FLG_UN_ALT = 0


FLG_FRST_CM = 1 ----------------|
 (Shift commanded)              |
                                |AND -| FLG_UN_ULSF = 1
FLG_LK_CM = 0 ------------------|     |  (Minimum unlock time after
 (Converter already unlocked)         |   start of unlocked shift)
                                      |
                                      | --- ELSE ---
                                      |
                                      | FLG_UN_ULSF = 0


NEBART < NELUMN ----------------------| FLG_UN_NE = 1
                                      |  (Low engine speed)
                                      |
                                      | --- ELSE ---
                                      |
                                      | FLG_UN_NE = 0


FLG_UN_CT = 1 ------------------------| Allow TM_UN_CT to count up
 (Closed throttle)                    |
                                      | --- ELSE ---
                                      |
                                      | TM_UN_CT = 0
```

```
FLG_UN_CT = 1 ------------------|
 (Closed throttle)              |AND -| TM_LK_DLY = CTDLY
                                |
TM_LK_DLY < CTDLY --------------|


FLG_UN_TRA = 1 -----------------|
 (High tip-in rate)             |AND -| TM_LK_DLY = TRADLY
                                |
TM_LK_DLY < TRADLY -------------|


FLG_UN_TRD = 1 -----------------|
 (High tip-out rate)            |AND -| TM_LK_DLY = TRDDLY
                                |
TM_LK_DLY < TRDDLY -------------|


FLG_UN_BRK = 1 -----------------|
 (Brake applied)                |AND -| TM_LK_DLY = BRKDLY
                                |
TM_LK_DLY < BRKDLY -------------|


FLG_UN_PRN = 1 -----------------|
 (Park, rev, neut)              |AND -| TM_LK_DLY = PRNDLY
                                |
TM_LK_DLY < PRNDLY -------------|


OFMFLG = 1 ---------------|
 (ETV sol. shorted)       |
                          |
FLG_UN_NE = 1 ------------|
 (Low engine speed)       |
                          |
FLG_UN_TEMP = 1 ----------|
 (Cold temperature)       |
                          |
FLG_UN_MDN = 1 -----------|OR --|
 (Manual downshift)       |     |
                          |     |AND -| TM_LK_DLY = LUDLY
FLG_UN_ALT = 1 -----------|     |
 (High altitude)          |     |
                          |     |
TFMFLG = 1 ---------------|     |
 (TP failure)                   |
                                |
TM_LK_DLY < LUDLY --------------|
```

```
FLG_UN_ULSF = 1 ----------|
                          |
GEAR_CUR = 2 -------------|AND -| TM_LK_DLY = U12DLY
                          |     |
GEAR_OLD = 1 -------------|     | --- ELSE ---
                                |
FLG_UN_ULSF = 1 ----------|     |
                          |     |
GEAR_CUR = 1 -------------|AND -| TM_LK_DLY = D21DLY
                          |     |
GEAR_OLD > 1 -------------|     | --- ELSE ---
                                |
FLG_UN_ULSF = 1 ----------|     |
                          |     |
GEAR_CUR = 3 -------------|AND -| TM_LK_DLY = U23DLY
                          |     |
GEAR_OLD < 3 -------------|     | --- ELSE ---
                                |
FLG_UN_ULSF = 1 ----------|     |
                          |     |
GEAR_CUR = 2 -------------|AND -| TM_LK_DLY = D32DLY
                          |     |
GEAR_OLD > 2 -------------|     | --- ELSE ---
                                |
FLG_UN_ULSF = 1 ----------|     |
                          |     |
GEAR_CUR = 4 -------------|AND -| TM_LK_DLY = U34DLY
                          |     |
GEAR_OLD < 4 -------------|     | --- ELSE ---
                                |
FLG_UN_ULSF = 1 ----------|     |
                          |     |
GEAR_CUR = 3 -------------|AND -| TM_LK_DLY = D43DLY
                          |
GEAR_OLD = 4 -------------|
```

```
SW_MLK = 2 --------------------------|
 (Unconditional unlock)              |
                                     |
FLG_UN_CT = 1 -----------------|     |
                               |AND -|
TM_UN_CT <= TMCTDY ------------|     |
 (Closed throttle, short time)       |
                                     |
TM_UN_CT > TMCTDY -------------|     |
 (Closed throttle long enough) |     |
                               |AND -|
VSBART_RT < VSCTDY ------------|     |
 (Low speed)                         |
                                     |
FLG_UN_TRA = 1 ----------------------|
 (High tip-in)                       |
                                     |
FLG_UN_TRD = 1 ----------------------|OR --|S Q--| FLG_UNC_UNLK = 1
 (High tip-out)                      |     |     |  (Unconditionally unlock
                                     |     |     |   converter clutch)
FLG_UN_BRK = 1 ----------------------|     |     |
 (Brake applied)                     |     |     | --- ELSE ---
                                     |     |     |
FLG_UN_PRN = 1 ----------------------|     |     | FLG_UNC_UNLK = 0
 (Park, rev, neut)                   |     |     |  (Allow lock-up if
                                     |     |     |   scheduled)
FLG_UN_TEMP = 1 ---------------------|     |
 (Cold temperature)                  |     |
                                     |     |
FLG_UN_MDN = 1 ----------------------|     |
 (Manual downshift)                  |     |
                                     |     |
FLG_UN_ALT = 1 ----------------------|     |
 (High altitude)                     |     |
                                     |     |
FLG_UN_ULSF = 1 ---------------------|     |
 (Unlocked shift)                    |     |
                                     |     |
TFMFLG = 1 --------------------------|     |
 (TP failure)                        |     |
                                     |     |
FLG_UN_NE = 1 -----------------------|     |
 (Low engine speed)                  |     |
                                     |     |
OFMFLG = 1 --------------------------|     |
 (ETV solenoid shorted)                    |
                                           |
TM_LK_DLY = 0 -----------------------------|C
```

SHIFT UNLOCK LOGIC

DEFINITIONS

INPUTS

Registers:

- GR_CM = Commanded gear for shift solenoids.
- GEAR_CUR = Current transmission gear.
- GR_DS = Desired transmission gear.
- GEAR_OLD = Last commanded gear.

Bit Flags:

- FLG_DN_LK = Downshift relock control flag.
- FLG_DN_UNLK = Downshift unlock control flag.
- FLG_FRST_CM = First time a shift is commanded flag; 0 -> no shift commanded this background pass, 1 -> shift commanded this background pass.
- FLG_FRST_DS = First time a shift is desired flag; 0 -> no shift.
- FLG_UP_LK = Upshift relock control flag.
- FLG_UP_UNLK = Upshift unlock control flag.

OUTPUTS

Registers:

- TM_SFT_CCO = Time for converter clutch to unlock prior to commanded shift.

PROCESS

STRATEGY MODULE: CCC_SHFT_UNLCK_COM1

```
FLG_FRST_DS = 1 -------------|
 (Shift desired this pass)   |
                             |AND -| DO "INITIALIZE DOWNSHIFT CONVERTER
GR_DS < GR_CM ---------------|     |     CLUTCH" LOGIC
 (Downshift)                       | DO "DOWNSHIFT CONVERTER CLUTCH" LOGIC
                                   |
FLG_FRST_CM = 1 -------------|     | --- ELSE ---
 (Shift commanded this pass) |     |
                             |AND -| DO "INITIALIZE UPSHIFT CONVERTER
GEAR_CUR > GEAR_OLD ---------|     |     CLUTCH" LOGIC
 (Upshift)                         | DO "UPSHIFT CONVERTER CLUTCH" LOGIC
                                   |
FLG_DN_UNLK = 1 -------------|     | --- ELSE ---
 (Downshift unlock control)  |     |
                             |OR --| DO "DOWNSHIFT CONVERTER CLUTCH" LOGIC
FLG_DN_LK = 1 ---------------|     |
 (Downshift relock control)        | --- ELSE ---
                                   |
FLG_UP_UNLK = 1 -------------|     |
 (Upshift unlock control)    |     |
                             |OR --| DO "UPSHIFT CONVERTER CLUTCH" LOGIC
FLG_UP_LK = 1 ---------------|     |
 (Upshift relock control)          | --- ELSE ---
                                   |
                                   | TM_SFT_CCO = 0
```

INITIALIZE DOWNSHIFT

DEFINITIONS

OUTPUTS

Registers:

- TM_SFT_CCO = Time for converter clutch to unlock prior to commanded shift.

Bit Flags:

- FLG_DN_LK = Downshift relock control flag.
- FLG_DN_UNLK = Downshift unlock control flag.
- FLG_HS_LK = High speed upshift relock control flag.
- FLG_HS_UNLK = High speed upshift unlock control flag.
- FLG_SFT_UNLK = Shift control unlock flag; 0 -> no shift control unlock, 1 -> shift control unlock.
- FLG_UP_LK = Upshift relock control flag.
- FLG_UP_UNLK = Upshift unlock control flag.

PROCESS

STRATEGY MODULE: CCC_INI_DWN_COM1

```
Always ------------------------| FLG_HS_UNLK = 0
 (Downshift desired this pass) | FLG_HS_LK = 0
                               |  (Disable high speed speed ratio checks)
                               | FLG_UP_UNLK = 0
                               | FLG_UP_LK = 0
                               | FLG_DN_LK = 0
                               |  (Clear all other shift control flags)
                               | FLG_DN_UNLK = 1
                               |  (Set downshift unlock control flag)
                               | FLG_SFT_UNLK = 1
                               |  (Unlock converter clutch)
                               | TM_SFT_CCO = 0
                               |  (Reset converter clutch unlock timer)
```

DOWNSHIFT CONVERTER CLUTCH

DEFINITIONS

INPUTS

Registers:

- DNUN_TM = FN624(TP_REL) = Time to delay downshift for converter to unlock.
- TM_SFT_CCO = Time for converter clutch to unlock prior to commanded shift.

Bit Flags:

- FLG_DN_LK = Downshift relock control flag.
- FLG_DN_UNLK = Downshift unlock control flag.

Calibration Constants:

- TMDNLK = Time for converter clutch relock after commanded downshift, sec.

OUTPUTS

Registers:

- TM_SFT_CCO = See above.

Bit Flags:

- FLG_DN_LK = See above.
- FLG_DN_UNLK = See above.
- FLG_SF_AUTO = Automatic upshift/downshift flag; 0 -> manual.
- FLG_SFT_UNLK = Shift control unlock flag; 0 -> no shift control unlock, 1 -> shift control unlock.

PROCESS

STRATEGY MODULE: CCC_DWN_CNVR_CLCH_COM1

```
FLG_DN_UNLK = 1 ------------|
 (Downshift unlock control) |
                            |AND -| FLG_DN_UNLK = 0
TM_SFT_CCO >= DNUN_TM ------|     |  (Clear downshift unlock control flag)
 (Converter clutch released)      | FLG_DN_LK = 1
                                  |  (Set downshift relock control flag)
                                  | TM_SFT_CCO = 0
                                  |  (Reset converter clutch relock timer)
                                  | A downshift will be commanded at
                                  |  this time
                                  |
FLG_DN_LK = 1 --------------|     | --- ELSE ---
 (Downshift relock control) |     |
                            |AND -| FLG_DN_LK = 0
TM_SFT_CCO >= TMDNLK -------|     |  (Clear downshift relock control flag)
 (Shift is complete)              | FLG_SF_AUTO = 0
                                  |  (Automatic downshift completed)
                                  | FLG_SFT_UNLK = 0
                                  |  (Permit relock if desired)
```

INITIALIZE UPSHIFT

DEFINITIONS

INPUTS

Registers:

- GEAR_CUR = Current transmission gear.
- NOBART = Filtered output shaft speed, RPM.
- RT_GR_CUR = Current transmission gear ratio.
- RT_ULK_PWR = Power on upshift unlock speed ratio.
- SPD_RT_STRT = Speed Ratio at Start of shift.
- TP_REL = Relative Throttle Position.

Bit Flags:

- FLG_LK_CM = Conv. clutch lockup commanded flag.
- FLG_UN_UPSFT = Unlocked upshift flag.
- FLG_UP_NE = WOT engine RPM upshift flag.

Calibration Constants:

- FN2LK = Speed ratio delta to relock on upshift to 2nd gear.
- FN2ULK = Speed ratio delta to unlock on usphift to 2nd gear.
- FN3LK = Speed ratio delta to relock on upshift to 3rd gear.
- FN3ULK = Speed ratio delta to unlock on usphift to 3rd gear.
- FN4LK = Speed ratio delta to relock on upshift to 4th gear.
- FN4ULK = Speed ratio delta to unlock on usphift to 4th gear.
- PUL_PER_REV = Pulses per revolution. Number of PIPs per engine revolution for E4OD gas; 1/2 the number of fuel pump teeth for E4OD_DIESEL.
- SRLK2 = Minimum speed ratio for scheduled 2nd gear lockup.
- SRLK3 = Minimum speed ratio for scheduled 3rd gear lockup.
- SRLK4 = Minimum speed ratio for scheduled 4th gear lockup.
- TMLFN2 = Power off upshift to 2nd relock time, sec.
- TMLFN3 = Power off upshift to 3rd relock time, sec.
- TMLFN4 = Power off upshift to 4th relock time, sec.

- TMLFW2 = Power off WOT upshift to 2nd relock time, sec.
- TMLFW3 = Power off WOT upshift to 3rd relock time, sec.
- TMLFW4 = Power off WOT upshift to 4th relock time, sec.
- TMLON2 = Power on upshift to 2nd relock time, sec.
- TMLON3 = Power on upshift to 3rd relock time, sec.
- TMLON4 = Power on upshift to 4th relock time, sec.
- TMLOW2 = Power on WOT upshift to 2nd relock time, sec.
- TMLOW3 = Power on WOT upshift to 3rd relock time, sec.
- TMLOW4 = Power on WOT upshift to 4th relock time, sec.
- TMUFN2 = Power off upshift to 2nd unlock time, sec.
- TMUFN3 = Power off upshift to 3rd unlock time, sec.
- TMUFN4 = Power off upshift to 4th unlock time, sec.
- TMUFW3 = Power off WOT upshift to 3rd unlock time, sec.
- TMUFW4 = Power off WOT upshift to 4th unlock time, sec.
- TMUOW3 = Power on WOT upshift to 3rd unlock time, sec.
- TMUOW4 = Power on WOT upshift to 4th unlock time, sec.

OUTPUTS

Registers:

- RT_GR_CUR = See above.
- RT_LK_PWR = Power on upshift relock speed ratio.
- RT_ULK_PWR = See above.
- SR_PP_LIM = Power on upshift speed ratio, PIP-to-PIP limit.
- SRLK = Shift ratio to allow lockup.
- TE_LK_PWR = Power on upshift relock time.
- TE_LK_UP = Power off upshift relock time.
- TE_ULK_PWR = Power on upshift unlock time.
- TE_ULK_UP = Power off upshift unlock time.
- TM_SFT_CCO = Time for converter clutch to unlock prior to commanded shift.

Bit Flags:

- FLG_DN_LK = Downshift relock control flag.
- FLG_DN_UNLK = Downshift unlock control flag.
- FLG_HS_LK = High speed upshift relock control flag.
- FLG_HS_UNLK = High speed upshift unlock control flag.
- FLG_TIP_RATE = Upshift tip-in rate flag; 0 -> no tip-in occured during upshift, 1 -> tip-in occurred during.
- FLG_UP_LK = Upshift relock control flag.
- FLG_UP_UNLK = Upshift unlock control flag.

PROCESS

STRATEGY MODULE: CCC_INI_UP_COM1

```
always --------------| FLG_DN_UNLK = 0
                     | FLG_DN_LK = 0
                     | FLG_UP_LK = 0
                     |  (clear all other shift control flags)
                     | FLG_UP_UNLK = 1
                     |  (set upshift unlock control flag)
                     | TM_SFT_CCO = 0
                     |  (reset unlock default timer)
                     | FLG_HS_LK = 0
                     |  (disable high speed relock check)
                     | FLG_TIP_RATE = 0
                     |  (reset tip-in rate flag)


FLG_UP_NE = 0 -|
               |AND -| TE_ULK_UP = TMUFN4
GEAR_CUR = 4 --|     | TE_LK_UP = TMLFN4
                     |  (power off unlock/relock timer)
                     | TE_ULK_PWR = TMUON4
                     | TE_LK_PWR = TMLON4
                     |  (power on unlock/relock timers)
                     | RT_ULK_PWR = SPD_RT_STRT + FN4ULK(TP_REL)
                     | RT_LK_PWR = SPD_RT_STRT + FN4LK(TP_REL)
                     |  (power on speed ratio checks)
                     | SRLK = SRLK4
                     |
                     | --- ELSE ---
FLG_UP_NE = 1 -|     |
               |AND -| TE_ULK_UP = TMUFW4
GEAR_CUR = 4 --|     | TE_LK_UP = TMLFW4
                     |  (power off unlock/relock timers)
                     | TE_ULK_PWR = TMUOW4
                     | TE_LK_PWR = TMLOW4
                     |  (power on unlock/relock timers)
                     | RT_ULK_PWR = SPD_RT_STRT + FN4ULK(TP_REL)
                     | RT_LK_PWR = SPD_RT_STRT + FN4LK(TP_REL)
                     |  (power on speed ratio checks)
                     | SRLK = SRLK4
                     |
                     | --- ELSE ---
FLG_UP_NE = 0 -|     |
               |AND -| TE_ULK_UP = TMUFN3
GEAR_CUR = 3 --|     | TE_LK_UP = TMLFN3
                     |  (power off unlock/relock timers)
                     | TE_ULK_PWR = TMUON3
                     | TE_LK_PWR = TMLON3
                     |  (power on unlock/relock timers)
                     | RT_ULK_PWR = SPD_RT_STRT + FN3ULK(TP_REL)
                     | RT_LK_PWR = SPD_RT_STRT + FN3LK(TP_REL)
                     |  (power on speed ratio checks)
                     | SRLK = SRLK3
                     |
                     | --- ELSE ---
```

(continued on next page)

(continued from previous page)

```
FLG_UP_NE = 1 -|      |
               |AND -| TE_ULK_UP = TMUFW3
GEAR_CUR = 3 --|      | TE_LK_UP = TMLFW3
                      |  (power off unlock/relock timers)
                      | TE_ULK_PWR = TMUOW3
                      | TE_LK_PWR = TMLOW3
                      |  (power on unlock/relock timers)
                      | RT_ULK_PWR = SPD_RT_STRT + FN3ULK(TP_REL)
                      | RT_LK_PWR = SPD_RT_STRT + FN3LK(TP_REL)
                      |  (power on speed ratio checks)
                      | SRLK = SRLK3
                      |
                      | --- ELSE ---
FLG_UP_NE = 0 -|      |
               |AND -| TE_ULK_UP = TMUFN2
GEAR_CUR = 2 --|      | TE_LK_UP = TMLFN2
                      |  (power off unlock/relock timers)
                      | TE_ULK_PWR = TMUON2
                      | TE_LK_PWR = TMLON2
                      |  (power on unlock/relock timers)
                      | RT_ULK_PWR = SPD_RT_STRT + FN2ULK(TP_REL)
                      | RT_LK_PWR = SPD_RT_STRT + FN2LK(TP_REL)
                      |  (power on speed ratio checks)
                      | SRLK = SRLK2
                      |
                      | --- ELSE ---
FLG_UP_NE = 1 -|      |
               |AND -| TE_ULK_UP = TMUFW2
GEAR_CUR = 2 --|      | TE_LK_UP = TMLFW2
                      |  (power off unlock/relock timers)
                      | TE_ULK_PWR = TMUOW2
                      | TE_LK_PWR = TMLOW2
                      |  (power on unlock/relock timers)
                      | RT_ULK_PWR = SPD_RT_STRT + FN2ULK(TP_REL)
                      | RT_LK_PWR = SPD_RT_STRT + FN2LK(TP_REL)
                      |  (power on speed ratio checks)
                      | SRLK = SRLK2


always --------------| SR_PP_LIM = (RT_ULK_PWR * 60) / (NOBART *
                      |   RT_GR_CUR * PUL_PER_REV)
                      | FLG_HS_UNLK = 1
                      |  (enable high speed unlock check)


FLG_LK_CM = 0 -------| FLG_UN_UPSFT = 1
                      |  (upshift commanded on an unlocked converter)
                      |
                      | --- ELSE ---
                      |
                      | FLG_UN_UPSFT = 0
```

UPSHIFT CONVERTER CLUTCH

DEFINITIONS

INPUTS

Registers:

- NOBART = Filtered output shaft speed.
- RT_GR_CUR = Current transmission gear ratio.
- RT_LK_PWR = Power on upshift relock speed ratio.
- RT_ULK_PWR = Power on upshift unlock speed ratio.
- SPD_RATIO = Speed ratio across torque converter (Output/Input).
- SRLK = Shift ratio to allow lockup.
- TE_LK_PWR = Power on upshift relock time.
- TE_LK_UP = Power off upshift relock time.
- TE_ULK_PWR = Power on upshift unlock time.
- TE_ULK_UP = Power off upshift unlock time.
- TM_SFT_CCO = Time for converter clutch to unlock prior to commanded shift.
- TP_RATE = Delta TP Counts Per BG Pass = (TP - TPBART).
- TSLSFT = Time since last shift timer.

Bit Flags:

- FLG_HS_LK = High speed upshift relock control flag.
- FLG_HS_UNLK = High speed upshift unlock control flag.
- FLG_PWR = Power mode flag; 0 -> power off, 1 -> power on.
- FLG_UP_LK = Upshift relock control flag.
- FLG_TIP_RATE = Unlocked upshift tip-in rate flag; 0 -> no tip-in occurred during upshift, 1 -> tip-in occurred during.
- FLG_UP_UNLK = Upshift unlock control flag.
- FLG_UN_UPSFT = Unlocked upshift flag.

Calibration Constants:

- PUL_PER_REV = Pulses per revolution. Number of PIPs per engine revolution for E4OD gas; 1/2 the number of fuel pump teeth for E4OD_DIESEL.

- TM_ACT_SFT = Re-lock delay time after a tip-in.

- UP_TIP_RATE = Tip-in rate above which FLG_TIP_RATE is set.

OUTPUTS

Registers:

- SR_PP_LIM = Power on upshift speed ratio, PIP-to-PIP limit.

- TM_SFT_CCO = See above.

Bit Flags:

- FLG_HS_LK = See above.

- FLG_HS_UNLK = See above.

- FLG_SFT_UNLK = Shift control unlock flag; 0 -> no shift control unlock, 1 -> shift control unlock.

- FLG_SF_AUTO = Automatic shift flag.

- FLG_TIP_RATE = See above.

- FLG_UP_LK = See above.

- FLG_UP_UNLK = See above.

PROCESS

STRATEGY MODULE: CCC_UP_CNVR_CLCH_COM2

```
FLG_PWR = 0 --------------------------------| SR_PP_LIM = 0
 (power off)                                |  (zero will never meet speed
                                            |   ratio criteria, uses default
                                            |   timer)
                                            |
                                            | --- ELSE ---
FLG_PWR = 1 --------------------------|     |
 (power on)                           |     |
                                      |AND -| SR_PP_LIM = RT_ULK_PWR * 60 /
FLG_UP_UNLK = 1 ----------------------|     |   NOBART * RT_GR_CUR *
 (upshift unlock control)                   |   PUL_PER_REV
                                            |  (calculate upshift unlock
                                            |   speed ratio comparator)
                                            |
                                            | --- ELSE ---
FLG_PWR = 1 --------------------------|     |
 (power on)                           |     |
                                      |AND -| SR_PP_LIM = RT_LK_PWR * 60 /
FLG_UP_LK = 1 ------------------------|     |   NOBART * RT_GR_CUR *
 (upshift relock control)                   |   PUL_PER_REV
```

```
FLG_UN_UPSFT = 1 ---------------------|
                                      |AND -| FLG_TIP_RATE = 1
TP_RATE > UP_TIP_RATE ----------------|


FLG_UP_UNLK = 1 ----------------------|
 (upshift unlock control)             |
                                      |
FLG_HS_UNLK = 0 -----------------|    |AND -| FLG_UP_UNLK = 0
 (high speed routine unlock)     |    |     |  (done with upshift unlock
                                 |    |     |   control)
FLG_PWR = 1 ---------------|     |    |     | FLG_UP_LK = 1
 (power on)                |     |    |     |  (set upshift relock control
                           |AND -|OR -|     |   flag)
TM_SFT_CCO >= TE_ULK_PWR --|     |          | TM_SFT_CCO = 0
 (default timer expired)         |          |  (reset default relock timer)
                                 |          | FLG_SFT_UNLK = 1
FLG_PWR = 0 ---------------|     |          |  (unlock converter clutch)
 (power off)               |     |          | SR_PP_LIM = RT_LK_PWR * 60
                           |AND -|          |   NOBART*RT_GR_CUR*PUL_PER_REV
TM_SFT_CCO >= TE_ULK_UP ---|                | FLG_HS_LK = 1
 (default timer expired)                    |  (enable high speed relock
                                            |   check)
                                            | FLG_HS_UNLK = 0
                                            |  (disable high speed unlock
                                            |   check)
                                            |
                                            | --- ELSE ---
FLG_TIP_RATE = 0 ---------------------|     |
                                      |     |
FLG_UP_LK = 1 ------------------------|     |
 (upshift relock control)             |     |
                                      |AND -| FLG_UP_LK = 0
FLG_HS_LK = 0 ------------------|     |     |  (clear upshift relock flag)
 (high speed routine relock)    |     |     | FLG_HS_LK = 0
                                |     |     |  (disable high speed relock
FLG_PWR = 1 --------------|     |     |     |   check)
 (power on)               |     |     |     | FLG_SFT_UNLK = 0
                          |AND -|OR --|     |  (permit relock if permitted)
TM_SFT_CCO >= TE_LK_PWR --|     |           | FLG_SF_AUTO = 0
 (default timer expired)        |           |  (done with automatic shift)
                                |           |
FLG_PWR = 0 --------------|     |           |
 (power off)              |     |           |
                          |AND -|           |
TM_SFT_CCO >= TE_LK_UP ---|                 |
                                            | --- ELSE ---
FLG_TIP_RATE = 1 ---------------------|     |
                                      |AND -| FLG_HS_LK = 0
TSLSFT < TM_ACT_SFT ------------------|     |  (disable high speed relock
                                            |   check)
                                            | FLG_SFT_UNLK = 1
                                            |  (don't allow scheduled lock)
                                            |
                                            | --- ELSE ---
```

(continued on next page)

(continued from previous page)

```
FLG_UN_UPSFT = 1 ---------------------|      |
 (unlocked upshift)                   |      |
                                      |      |
FLG_TIP_RATE = 1 ---------------------|AND -| FLG_SFT_UNLK = 0
 (tip-in detected)                    |      |  (allow a lock if scheduled)
                                      |      | FLG_TIP_RATE = 0
SPD_RATIO >= SRLK --------------------|      |  (reset tip-in flag)
 (speed ratio is high enough)                |
                                             | --- ELSE ---
FLG_UN_UPSFT = 1 ---------------------|      |
 (unlocked upshift)                   |      |
                                      |AND -| FLG_SFT_UNLK = 1
FLG_TIP_RATE = 1 ---------------------|      |  (don't allow scheduled lock)
 (tip-in detected)
```

SCHEDULED LOCK/UNLOCK LOGIC

DEFINITIONS

INPUTS

Registers:

- BP = Barometric pressure.
- BP_INTR = Output of FN615(BP). This is used to modify shift schedules and converter lockup as a function of barometric pressure.
- GEAR_CUR = Current transmission gear.
- SPD_RATIO = Speed ratio across the torque converter.
- TM_CRV_UNLK = Timer for converter clutch scheduled relocks.
- TP = Throttle position.
- TP_REL_H = Relative TP (TP - RATCH) high byte only.
- TPBARTC = UROLAV(TP,TCTPTC)
- VSBART_RT = Filtered vehicle speed adjusted for RT_NOVS for transmission,

Bit Flags:

- FLG_CRV_DS = Scheduled lockup desired from curve flag.
- FLG_CRV_LST = Last pass value of FLG_CRV_DS.
- FLG_FRST_CM = First time a shift is commanded flag; 0 -> no shift commanded this background pass, 1 -> shift commanded this background pass.
- FLG_SCHD_DLY = Scheduled unlock flag.
- FLG_SFT_IN = Shift in progress flag.
- FLG_UNC_UNLK = Converter clutch unconditional unlock flag; 0 -> no unconditional unlock, 1 -> unconditional unlock.
- FLG_UN_BRK = Brake applied unconditional unlock flag; 0 -> brake not applied, 1 -> brake applied.
- FLG_UN_CT = Closed throttle unconditional unlock flag; 0 -> not closed throttle, 1 -> closed throttle.

Calibration Constants:

- ALTDLY = Time to delay scheduled lockups at intermediate altitudes, sec.
- BPUNMN = Minimum BP to unlock converter clutch.

- BPUNMX = Maximum BP altitude unlock delay, " Hg.
- CRVDLY = Time to delay scheduled lockups after STEADYSRLK1 throttle achieved, sec.
- FN2LA(TP_REL_H) = Delta vehicle speed for 2nd gear lockup at altitude: Input TP_REL_H, counts; Output vehicle speed, MPH.
- FN2LS(TP_REL_H) = Vehicle speed for 2nd gear lockup at sea level: Input TP_REL_H, counts; Output vehicle speed, MPH.
- FN2UA(TP_REL_H) = Delta Vehicle speed for 2nd gear unlock at altitude: Input TP_REL_H, counts; Output vehicle speed, MPH.
- FN2US(TP_REL_H) = Vehicle speed for 2nd gear unlock at sea level: Input TP_REL_H, counts; Output vehicle speed, MPH.
- FN3LA(TP_REL_H) = Delta vehicle speed for 3rd gear lockup at altitude: Input TP_REL_H, counts; Output vehicle speed, MPH.
- FN3LS(TP_REL_H) = Vehicle speed for 3rd gear lockup at sea level: Input TP_REL_H, counts; Output vehicle speed, MPH.
- FN3UA(TP_REL_H) = Delta Vehicle speed for 3rd gear unlock at altitude: Input TP_REL_H, counts; Output vehicle speed, MPH.
- FN3US(TP_REL_H) = Vehicle speed for 3rd gear unlock at sea level: Input TP_REL_H, counts; Output vehicle speed, MPH.
- FN4LA(TP_REL_H) = Delta vehicle speed for 4th gear lockup at altitude: Input TP_REL_H, counts; Output vehicle speed, MPH.
- FN4LS(TP_REL_H) = Vehicle speed for 4th gear lockup at sea level: Input TP_REL_H, counts; Output vehicle speed, MPH.
- FN4UA(TP_REL_H) = Delta Vehicle speed for 4th gear unlock at altitude: Input TP_REL_H, counts; Output vehicle speed, MPH.
- FN4US(TP_REL_H) = Vehicle speed for 4th gear unlock at sea level: Input TP_REL_H, counts; Output vehicle speed, MPH.
- SRLK2 = Minimum speed ratio for scheduled 2nd gear lockup.
- SRLK3 = Minimum speed ratio for scheduled 3rd gear lockup.
- SRLK4 = Minimum speed ratio for scheduled 4th gear lockup.
- SW_RLK = Closed throttle/brake hysteresis switch; 0 -> unlock in hysteresis zone, 1 -> remain locked in hysteresis zone.
- TPUNCT = Maximum relative TP for closed throttle unlock.
- TPUNCH = Hysteresis for TPUNCT.

OUTPUTS

Registers:

- TM_CRV_UNLK = Timer for converter clutch scheduled relocks.

Bit Flags:

- FLG_CRV_DS = See above.

- FLG_CRV_LK = Scheduled curve lock-up flag; 0 -> no scheduled lock-up, 1 -> scheduled lock-up.

- FLG_CRV_LST = See above.

- FLG_SCHD_DLY = See above.

PROCESS

STRATEGY MODULE: CCC_SCHLD_LCK_UNLCK_COM1

```
always --------------------------------------| FLG_CRV_LST = FLG_CRV_DS


FLG_UN_CT = 1 -------------------|
                                 |OR --|
FLG_UN_BRK = 1 ------------------|     |
                                       |AND -| No change to FLG_CRV_DS
SW_RLK = 1 ----------------------------|     | Retain last state while
                                             |  in hysteresis zone
                                             |
                                             | --- ELSE ---
                                             |
FLG_UNC_UNLK = 1 ----------------------------| FLG_CRV_DS = 0
 (Unconditional unlock)
```

```
GEAR_CUR = 2 --------------------------|
                                       |
SPD_RATIO >= SRLK2 --------|           |
 (Speed ratio high enough) |           |
                           |AND -|     |
FLG_SFT_IN = 0 ------------|     |     |
 (Not shifting)                  |OR --|AND -| FLG_CRV_DS = 1
                                 |     |     |
FLG_SFT_IN = 1 ------------------|     |     |
 (Shifting disregard speed ratio)      |     |
                                       |     |
VSBART_RT > FN2LS + [BP_INTR * FN2LA] -|     |
                                             | --- ELSE ---
GEAR_CUR = 3 --------------------------|     |
                                       |     |
SPD_RATIO >= SRLK3 --------|           |     |
 (Speed ratio high enough) |           |     |
                           |AND -|     |     |
FLG_SFT_IN = 0 ------------|     |     |     |
 (Not shifting)                  |OR --|AND -| FLG_CRV_DS = 1
                                 |     |     |
FLG_SFT_IN = 1 ------------------|     |     |
 (Shifting - disregard speed ratio)    |     |
                                       |     |
VSBART_RT > FN3LS + [BP_INTR * FN3LA] -|     |
                                             | --- ELSE ---
GEAR_CUR = 4 --------------------------|     |
                                       |     |
SPD_RATIO >= SRLK4 ---------|          |     |
                            |AND -|    |     |
FLG_SFT_IN = 0 -------------|     |OR -|AND -| FLG_CRV_DS = 1
                                  |    |     |
FLG_SFT_IN = 1 -------------------|    |     |
                                       |     |
VSBART_RT > FN4LS + [BP_INTR * FN4LA] -|     |
                                             | --- ELSE ---
                                             |
GEAR_CUR = 1 --------------------------------| FLG_CRV_DS = 0
                                             |
                                             | --- ELSE ---
GEAR_CUR = 2 --------------------------|     |
                                       |AND -| FLG_CRV_DS = 0
VSBART_RT < FN2US + [BP_INTR * FN2UA] -|     |
                                             | --- ELSE ---
GEAR_CUR = 3 --------------------------|     |
                                       |AND -| FLG_CRV_DS = 0
VSBART_RT < FN3US + [BP_INTR * FN3UA] -|     |
                                             | --- ELSE ---
GEAR_CUR = 4 --------------------------|     |
                                       |AND -| FLG_CRV_DS = 0
VSBART_RT < FN4US + [BP_INTR * FN4UA] -|     |
                                             | --- ELSE ---
                                             |
                                             | No change to FLG_CRV_DS
```

```
FLG_FRST_CM = 1 -----------------------------| FLG_SCHD_DLY = 0
 (Shift is commanded)                        |  (Shift about to occur, do not
                                             |   check steady state criteria)
                                             |
                                             | --- ELSE ---
FLG_CRV_DS = 0 -----------------------|      |
                                      |      |
FLG_CRV_LST = 1 ----------------------|AND -| FLG_SCHD_DLY = 1
                                      |      |  (Set flag to indicate
FLG_UNC_UNLK = 0 ---------------------|      |   scheduled unlock. Apply
 (Unlock transition due                      |   steady state throttle
  solely to schedule)                        |   criteria for relock)


FLG_SCHD_DLY = 1 ---------------------|
                                      |
FLG_CRV_DS = 1 -----------------------|AND -| TM_CRV_UNLK = CRVDLY
 (Relock after scheduled unlock)      |      |  (If throttle is not steady
                                      |      |   state load delay timer)
|TP - TPBARTC| >= TPUNSC -------------|      |
 (Throttle not steady state)


FLG_CRV_LST = 0 ----------------------|
                                      |
FLG_CRV_DS = 1 -----------------------|AND -| TM_CRV_UNLK = ALTDLY
 (Scheduled lock desired)             |      |  (Delay scheduled relock at
                                      |      |   altitude)
BPUNMN < BP < BPUNMX -----------------|
 (Intermediate altitude)


TM_CRV_UNLK = 0 -----------------------------| FLG_CRV_LK = FLG_CRV_DS
                                             |  (Allow desired state to pass
                                             |   through)
                                             | FLG_SCHD_DLY = 0
                                             |  (Reset steady state relock
                                             |   flag)
```

WOT LOCK-UP LOGIC

DEFINITIONS

INPUTS

Registers:

- GEAR_OLD = Last commanded gear.
- NEBART = Filtered engine speed, RPM.
- PDL = Current PRNDL position.
- RLKCTR = WOT converter clutch relock counter.
- SPD_RATIO = Speed ratio across the torque converter.
- TP_REL = Relative Throttle Position.

Bit Flags:

- FLG_LK_CM = Converter clutch lock-up commanded flag; 0 -> de-energize solenoid, unlock converter clutch, 1 -> energize solenoid, lock converter clutch.
- FLG_RLK_WOT = WOT relock first pass flag.

Calibration Constants:

- NELKWH = Hysteresis for NELKWOd, RPM.
- NELKWO = Minimum RPM for converter clutch WOT lockup, RPM.
- RTLKWO = Minimum speed ratio for 1st gear WOT converter clutch lockup.
- RTLKWH = Hysteresis for RTLKWO.
- TPLKWO = Minimum TP for converter clutch WOT lockup, counts.
- TPLKWH = TPLKWO hysteresis for converter clutch WOT lockup, counts.

OUTPUTS

Registers:

- RLKCTR = See above.

Bit Flags:

- FLG_RLK_WOT = See above.
- FLG_WOT_LK = WOT lock-up flag; 0 -> no WOT lock-up, 1 -> WOT lock-up.

PROCESS

STRATEGY MODULE: CCC_WOT_LCK_UP_COM1

```
TP_REL < TPLKWO - TPLKWH ------|
 (low TP)                      |
                               |
TP_REL >= TPLKWO -------|      |OR ---| RLKCTR = 1
 (high TP)              |      |      | FLG_RLK_WOT = 0
                        |      |      |  (not at WOT or at WOT
GEAR_CUR <> 1 ----------|AND --|      |   in higher gear.  Set up
 (not 1st gear)         |             |   for next time.)
                        |             |
PDL <> 2 ---------------|             |
 (not manual 2)                       | --- ELSE ---
                                      |
NEBART > NELKWO ----------------------| FLG_RLK_WOT = 1
 (above engine torque peak)           |  (set first pass relock flag)
                                      |
                                      | --- ELSE ---
NEBART > NELKWO - NELKWH ------|      |
 (in hystersis zone)           |      |
                               |      |
FLG_LK_CM = 1 -----------------|AND --| FLG_RLK_WOT = 0
 (currently locked)            |      | RLKCTR = RLKCTR + 1
                               |      |  (RPM is dropping with a
FLG_RLK_WOT = 1 ---------------|      |   locked converter. Increment
 (rpm was high)                       |   counter to relock at a
                                      |   higher speed ratio next time
                                      |   and clear first pass flag)
                                      |
                                      | --- ELSE ---
                                      |
                                      | Do Not Change
                                      | FLG_RLK_WOT or RLKCTR
```

```
TP_REL >= TPLKWO ---------------------|
 (at WOT)                             |
                                      |
GEAR_CUR <> 1 ----------|             |AND --|S  Q -| FLG_WOT_LK = 1
 (not 1st gear)         |AND --|      |      |      |  (at WOT in high enough
                        |      |      |      |      |   gear or high enough
PDL <> 2  --------------|      |      |      |      |   rpm and speed ratio;
 (not manual 2)                |OR ---|      |      |   lock up converter
                               |             |      |   regardless of lockup
NEBART > NELKWO --------|      |             |      |   curves)
 (above torque peak)    |      |             |      |
                        |AND --|             |      | --- ELSE ---
SPD_RATIO >= RTLKWO +   |                    |      |
    (RTLKWH * RLKCTR) --|                    |      | FLG_WOT_LK = 0
 (above speed ratio                          |      |  (not at WOT or rpm or
   for lockup)                               |      |   speed ratio too low;
                                             |      |   use normal lockup
TP_REL < TPLKWO - TPLKWH -------------|      |      |   curves)
                                      |      |
NEBART <= NELKWO - NELKWH ------------|OR ---|C
                                      |
SPD_RATIO < RTLKWO +                  |
             (RTLKWH * RLKCTR) -------|
```

HIGH SPEED UPSHIFT

DEFINITIONS

INPUTS

Registers:

- DT12S_AVG = Filtered PIP period for transmission, ticks.
- NOBART = Filtered output shaft speed, RPM.
- RT_GR_CUR = Current transmission gear ratio.
- SR_PP_LIM = Power on upshift speed ratio, PIP-to-PIP limit.

Bit Flags:

- FLG_HS_LK = High speed upshift relock control flag.
- FLG_HS_UNLK = High speed upshift unlock control flag.
- FLG_LK_CM = Converter clutch lock-up commanded flag; 0 -> de-energize solenoid, unlock converter clutch, 1 -> energize solenoid, lock converter clutch.

Calibration Constants:

- PUL_PER_REV = Pulses per revolution. Number of PIPs per engine revolution for E4OD gas; 1/2 the number of fuel pump teeth for E4OD_DIESEL.

OUTPUTS

Registers:

- SR_PP_LIM = See above.

Bit Flags:

- FLG_HS_LK = See above.
- FLG_HS_UNLK = See above.
- FLG_LK_CM = See above.
- FLG_SFT_UNLK = Shift control unlock flag; 0 -> no shift control unlock, 1 -> shift control unlock.

PROCESS

STRATEGY MODULE: CONVERTER_CLUTCH_REPEAT_COM3

HIGH SPEED UPSHIFT SPEED RATIO CHECK
(Executed during 1 msec repeater)

```
FLG_HS_UNLK = 1 ----------------|
 (Look for unlock speed ratio)  |
                                |AND -| FLG_HS_UNLK = 0
DT12S_AVG > SR_PP_LIM ----------|     |  (Disable high speed unlock check)
 (Speed ratio high enough)            |
                                      | FLG_SFT_UNLK = 1
                                      |  (Request conv. clutch unlock)
                                      |
                                      | FLG_LK_CM = 0
                                      |  (Unlock conv. clutch)
                                      |
                                      | --- ELSE ---
FLG_HS_LK = 1 ------------------|     |
 (Look for relock speed ratio)  |     |
                                |AND -| FLG_HS_LK = 0
DT12S_AVG > SR_PP_LIM ----------|     |  (Disable high speed relock check)
 (Speed ratio high enough)            |
                                      | FLG_SFT_UNLK = 0
                                      |  (End upshift unlock)
```

NOTE: SR_PP_LIM is computed each background pass.

$$SR\_PP\_LIM = \frac{\text{"SPEED RATIO CALIBRATION PARAMETER"} * 60}{NOBART * RT\_GR\_CUR * PUL\_PER\_REV}$$

CONVERTER CLUTCH OUTPUT CONTROL
(Executed during 1 msec repeater)

```
FLG_LK_CM = 1 ------------------| Energize conv. clutch output
                                | to apply converter clutch
                                |
                                | --- ELSE ---
                                |
FLG_LK_CM = 0 ------------------| De-energize conv. clutch output
                                | to release converter clutch
```

CONVERTER CLUTCH VALIDATION

OVERVIEW

The converter clutch validation logic verifies that the engine speed over vehicle speed computation is consistent. This in turn verifies that the converter clutch was applied.

The actual computed N/V divided by the base N/V should equal the N/V ratio stored in KAM. If the ratio remains consistent, it verifies that the converter clutch was applied during the actual N/V calculation.

If these do not equal, then either:

a) the converter clutch was not applied during the actual N/V calculation

--- OR ---

b) the converter clutch was not applied when the original N/V KAM calculation was made.

In either case, there was a fault with converter clutch, and an error code is flagged.

If the converter clutch was not applied when the N/V value was stored in KAM, yet the current calculation matches the incorrect KAM value, then no error is stored. It is, however, highly probable that the actual N/V computation will differ from the KAM value eventually during the vehicle operation (since the converter is not applied), and the error will be noted at that time.

DEFINITIONS

INPUTS

Registers:

- C628CNT = Warm up cycle counter for code 628.
- C628_KAM_BIT = Code 628 KAM bit.
- NOV_ACT = Actual N/V calculation.
- RT_NOVS_KAM = N/V value stored in KAM.
- TP_REL = Relative TP (TP - RATCH)

Bit Flags:

- FLG_FRST_NOV = Flag indicating a new value has been stored in RT_NOVS_KAM during current power-up.

- FLG_NEW_NOV = Flag, if set to 1, indicates a new NOV_ACT has been calculated.

- FLG_4X4L = 4x4l flag, 0 -> not in 4x4 low mode, 1 -> in 4x4 low mode

Calibration Constants:

- CCE_TPMN = Minimum TP required to do converter clutch validation logic.

- CC_FM_LVL = number of warm up cycles converter clutch failure mode action will be executed after a fault is detected.

- NOV_ERR_BAND = Error band allowed on the N/V calculation.

- NVBASE = Base N/V.

OUTPUTS

Bit Flags:

- CC_FM_FLG = Converter clutch failure mode flag; 0 -> normal operation, 1 -> Converter clutch failure mode

PROCESS

STRATEGY MODULE: CCC_VALID_COM2

```
FLG_4X4L = 0 --------------------------|
 (not in 4x4 low)                      |
                                       |
FLG_FRST_NOV = 1 ----------------------|
 (new NOV ratio stored in KAM)         |
                                       |
FLG_NEW_NOV = 1 -----------------------|AND -|
 (new NOV_ACT calculated)              |     |
                                       |     | error_detected := 1
|NOV_ACT               |               |     | (converter clutch
|------- - RT_NOVS_KAM| > NOV_ERR_BAND-|     |   inoperative)
| NVBASE               |               |     | DO: FAULT FILTER for code
                                       |     |  628 procedure
TP_REL > CCE_TPMN ---------------------|     |
                                             | --- ELSE ---
FLG_FRST_NOV = 1 ----------------------|     |
                                       |AND -|  (converter clutch operative)
FLG_NEW_NOV = 1 -----------------------|     | DO: FAULT FILTER for code
                                             |  628 procedure


C628_KAM_BIT = 1 --------------------|
 (converter clutch error detected)   |AND -| CC_FM_FLG = 1
                                     |     | (fault detected, execute
C628CNT < CC_FM_LVL -----------------|     |  converter clutch failure
 (not enough warm up cycles since          |  mode action)
  error detection to exit failure          |
  mode action)                             | --ELSE--
                                           |
                                           | CC_FM_FLG = 0
                                           | (no fault deteced or enough
                                           |  warm up cycles have passed
                                           |  since detection to exit
                                           |  converter clutch failure mode)
```

COAST CLUTCH CONTROL

OVERVIEW

The coast clutch is a clutch used to provide engine braking in 3rd gear when the PRNDL is in the drive position. Without the coast clutch the transmission would free-wheel in third gear while the vehicle was coasting.

* In overdrive, the coast clutch is hydraulically off. Engine braking is provided by the overdrive clutch in fourth gear.

* In drive, the software turns on the coast clutch to provide engine braking. A short delay is provided to allow the overdrive clutch to release fully.

* In manual 2 or 1 the coast clutch is applied hydraulically. Intermediate band application is delayed until the coast clutch actually engages as inferred by the shift in progress timer. This is to prevent the intermediate band from absorbing excessive driveline deceleration energy which could be better handled by the larger coast clutch.

* If the vehicle speed sensor has failed, the coast clutch is applied in all gears below fourth to provide engine braking and prevent rapid free-wheeling downshifts to first gear when the throttle is closed.

DEFINITIONS

INPUTS

Registers:

- GR_CM = Commanded gear for shift solenoids.

- GR_OLD = Last commanded gear.

- PDL = Current PRNDL position.

- PDL_LST = PRNDL position last background pass.

- TM_CS_DLY = Timer to delay coast clutch application.

- TM_CS_ENG = Timer for coast clutch to engage.

Bit Flags:

- FLG_CS_CM = Coast clutch commanded output state; 1 -> command coast clutch on, 0 -> command coast clutch off.

- FLG_CS_FRST = Coast clutch engagement first pass flag; 1 -> first time coast clutch engages, 0 -> not first time for coast clutch engagement.

- FLG_DEL_MDN = Flag which indicates a manual downshift is being delayed; 0 -> no manual downshift is being delayed, 1 -> a manual downshift is being delayed.

- FLG_FRST_CM = First time a shift is commanded flag; 1 -> shift commanded this background pass, 0 -> no shift commanded this background pass.

- FLG_PWR = Power mode flag; 1 -> power on mode, 0 -> power off mode.

- FLG_SFT_MDN = Power off manual downshift flag; 1 -> power off manual downshift in progress, 0 -> power off manual downshift not in progress.

Calibration Constants:

- TMCSE2 = Time for coast clutch to engage, PDL = 2 or 1.

- TMCSE3 = Time for coast clutch to engage, PDL = 3.

- TMCSOD = Time to delay coast clutch when PDL = 3.

OUTPUTS

Registers:

- TM_CS_DLY = See above.

- TM_CS_ENG = See above.

Bit Flags:

- FLG_CS_CM = See above.

PROCESS

STRATEGY MODULE: CCC_CST_CLTCH_CTL_COM1

TM_CS_DLY TIMER CONTROL

```
FLG_FRST_CM = 1 -------------|
 (shift commanded)           |
                             |AND -| TM_CS_DLY = TMCSOD
GR_OLD = 4 ------------------|     |  (delay coast clutch engagement
                             |     |   in third gear to prevent two
GR_CM = 3 -------------------|     |   elements being on at once)
```

TM_CS_ENG TIMER CONTROL

```
PDL = 3 ---------------------|
 (PRNDL = 3)                 |
                             |
FLG_CS_CM = 1 ---------------|AND -| TM_CS_ENG = TMCSE3
 (coast clutch commanded on) |     |  (coast clutch will be engaged when
                             |     |   timer expires)
FLG_CS_FRST = 0 -------------|     | FLG_CS_FRST = 1
 (1st pass thru)                   |  (set first pass flag)
                                   |
                                   | --- ELSE ---
PDL = 1 ---------------|           |
                       |OR --|     |
PDL = 2 ---------------|     |     |
 (M1 or M2)                  |     |
                             |AND -| TM_CS_ENG = TMCSE2
PDL_LST = 4 -----------------|     |  (coast clutch will be engaged when
 (4-2 or 4-1)                |     |   timer expires)
                             |     | FLG_CS_FRST = 1
FLG_CS_FRST = 0 -------------|     |  (set first pass flag)
 (first pass thru)
```

COAST CLUTCH OUTPUT CONTROL

```
GR_DS_TV = 4 ----------------------| FLG_CS_CM = 0
                                   |  (verifing a 3-4 upshift, de-energize
                                   |   Coast clutch)
                                   |
                                   | --- ELSE ---
CS_PDL_4 = 1 ----------|           |
                       |           |
PDL <= 3 --------------|OR --|     |
                       |     |     |
VSFMFLG = 1 -----------|     |     |
                             |AND -| FLG_CS_CM = 1
TM_CS_DLY = 0 ---------------|     |  (energize coast clutch output)
                             |     |
GR_CM <> 4 ------------------|     |
                                   | --- ELSE ---
                                   |
                                   | FLG_CS_CM = 0
                                   |  (de-energize coast clutch output.
                                   |   Coast clutch is off hydraulically
                                   |   in 4th. Coast clutch is on
                                   |   hydraulically when PDL = 2 or 1)
```

INFER COAST CLUTCH ENGAGEMENT

```
FLG_CS_CM = 0 ---------------------| FLG_CS_ENG = 0
                                   |  (coast clutch is hydraulically off)
                                   | FLG_CS_FRST = 0
                                   |
                                   | --- ELSE ---
TM_CS_ENG = 0 ---------|           |
 (timer expired)       |           |
                       |AND -|     |
TM_CS_DLY = 0 ---------|     |     |
                       |     |     |
FLG_PWR = 1 -----------|     |OR --| FLG_CS_ENG = 1
 (power on)                  |     |  (coast clutch is inferred to be
                             |     |   on. Once on, it remains on until
FLG_SFT_MDN = 0 -------|     |     |   the PRNDL goes to overdrive
 (manual downshift     |AND -|     |   position)
  complete             |
                       |
FLG_PWR = 0 -----------|
 (power off)           |
                       |
FLG_DEL_MDN = 0 -------|
 (manual downshift
  delay is complete)
```

FAILURE MODE MANAGEMENT LOCK-UP

OVERVIEW

This module determines converter clutch lock-ups during a TP Sensor or a Vehicle Speed Sensor failure. If the Vehicle Speed Sensor fails, locks are based on a function of NEBART and TP_REL. If the TP Sensor fails, locks are based on SPD_RATIO.

Once an FMEM lock-up is requested, locks continue to be based on this module, until the next power-up.

Unlocks occur as a result of Unconditional unlocks only.

DEFINITIONS

INPUTS

Registers:

- NEBART = Filter engine speed, RPM.

- SPD_RATIO = Speed Ratio across the torque converter.

Bit Flags:

- FLG_FMM_LK = Failure Mode Management lock-up flag; 1 -> lock converter due to FMEM action.

- FLG_FMM_CC = Flag used to insure once FMEM lock-up is activated, it is not de-activated until next power-up; 0 -> Failure mode lock-up is not in use this power-up, 1 -> Failure mode lock-up is in use this power-up.

- FLG_UNC_UNLK = Converter clutch unconditional unlock flag; 0 -> no unconditional unlock, 1 -> unconditional unlock.

- TFMFLG = TP FMEM flag; 0 -> no TP failure, 1 -> TP failure, operating in FMEM mode.

- VSFMFLG = Vehicle speed sensor failure flag; 1 -> VSS failure, 0 -> no VSS failure.

Calibration Constants:

- FN689L(TP_REL) = Engine Speed to lock converter when TP sensor is still available, for Failure Mode Management, RPM.
- NELK_FM = Engine Speed to lock converter when TP sensor has failed, for Failure Mode Management, RPM.
- SRLK_FM = Speed Ratio to lock converter for Failure Mode Management.

OUTPUTS

Bit Flags:

- FLG_FMM_LK = Failure Mode Management lock-up flag; 1 -> lock converter due to FMEM action.
- FLG_FMM_CC = Flag used to insure once FMEM lock-up is activated, it is not de-activated until next power-up; 0 -> Failure mode lock-up is not in use this power-up, 1 -> Failure mode lock-up is in use this power-up.

PROCESS

STRATEGY MODULE: CCC_FMEM_COM2

```
FLG_FMM_LK = 1 -----------------------------------|
(Failure Mode Lock Requested)                     |AND -| FLG_FMM_LK = 0
                                                  |     | (Reset FMEM
FLG_UNC_UNLK = 1 ---------------------------------|     |  lock-up flag)
(Unconditional Unlock Requested)



VSFMFLG = 1 --------------------------------|
(VS Sensor failure)                         |
                                            |
TFMFLG = 0 ---------------------|           |AND -|
(TP Sensor OK)                  |AND -|     |     |
                                |     |     |     |
NEBART > FN689L(TP_REL) --------|     |     |     |
(Engine Speed above failure           |OR --|     |
 mode minimum)                        |           |OR --| FLG_FMM_LK = 1
                                      |           |     | FLG_FMM_CC = 1
TFMFLG = 1 ---------------------|     |           |     | (Continue with
(TP Sensor failure)             |AND -|           |     |  FMEM Lock-up
                                |                 |     |  until next
NEBART > NELK_FM ---------------|                 |     |  power-up)
(Engine Speed above failure                       |
 mode minimum)                                    |
                                                  |
TFMFLG = 1 --------------------------|            |
(TP Sensor Failure)                  |OR --|      |
                                     |     |      |
FLG_FMM_CC = 1 ----------------------|     |AND -|
(FMEM lock-up activated this power-up)     |
                                           |
SPD_RATIO > SRLK_FM -----------------------|
(Speed Ratio above failure mode minimum)
```

CHAPTER 19

TRANSMISSION INPUT CONVERSIONS

TRANSMISSION CONTROL INDICATOR LIGHT

OVERVIEW

The Transmission Control Indicator Light (TCIL), located in the instrument panel, visually indicates the status of "Transmission Control" to the driver, or alerts the driver to certain transmission faults. If not indicating a fault, the light is on when overdrive is canceled; off when overdrive is enabled. A flashing light indicates a transmission fault. The transmission control switch and inhibition of fourth gear will continue to operate normally in the flashing mode, but there is no visual indication of cancel mode.

To disable flashing for any fault, set TCIL_TM_DLY = 31.875.

To enable flashing for specific faults, set TCIL_TM_DLY < 31.875, and set the calibration switch for that fault = 1. For example, to flash the TCIL for converter clutch errors and shift errors, set CC_ERR_SW = 1 and SFT_ERR_SW = 1, and all remaining switches = 0.

NOTE: The OFMFLG does not have a switch associated with it. When TCIL_TM_DLY < 31.875, the TCIL flash for an EPC short circuit, regardless of the state of any of the calibration switches.

DEFINITIONS

INPUTS

Registers:

- TCILTMR = Time since transmission fault occurred, sec.

- TCIL_FLASH_TMR = Time since TCIL changed states in flashing mode, sec.

Bit Flags:

- CC_FM_FLG = converter clutch failure mode flag; 0 -> normal action, 1 -> execute converter clutch failure mode action.

- CRKFLG = Flag indicating engine mode status; 0 -> not in CRANK mode, 1 -> in CRANK mode.

- EPC_OPEN_FLG = Indicates EPC open circuit; 1 -> EPC open circuit detected.

- FLG_TCS = Transmission Control flag; 0 -> overdrive enable, 1 -> overdrive lockout mode.

- OFMFLG = Flag indicating that an EPC solenoid short circuit failure has been detected; 0 -> EPC solenoid circuit not shorted, 1 -> EPC solenoid circuit shorted.

- OTEMP_FM_FLG = Transmission overtemperature FMEM flag; 1 -> Transmission is overtemperature, 0 -> Transmission temperature okay.

- MFMFLG = MAP sensor FMEM flag; 1 -> MAP sensor failure, 0 -> MAP sensor okay.

- PDL_ERROR = PRNDL error flag; 1 -> PRNDL error, 0 -> PRNDL okay.

- SFT_FM_FLG = Flag indicating whether 1-2 or 2-3 or 3-4 shifts are failing to occur properly; 0 -> shifts OK, 1 -> Shifts not OK.

- STIFLG = Flag which, if set, indicates Self Test has been requested.

- TCIL_STATE = Flag indicating state of TCIL; 0 -> TCIL off, 1 -> TCIL on.

- TFMFLG = TP FMEM flag; 0 -> no TP failure, 1 -> TP failure, operating in FMEM mode.

- VSFMFLG = Vehicle Speed FMEM flag; 1 -> Vehicle speed sensor failure, 0 -> Vehicle speed sensor okay.

Calibration Constants:

- CC_ERR_SW = Calibration selection switch to enable/disable flashing TCIL for converter clutch error; 0 -> disable, 1 -> enable.

- EPC_ERR_SW = Calibration selection switch to enable/disable flashing TCIL for EPC open circuit error; 0 -> disable, 1 -> enable.

- PDL_ERR_SW = Calibration selection switch to enable/disable flashing TCIL for PRNDL error; 0 -> disable, 1 -> enable.

- MAP_ERR_SW = Calibration selection switch to enable/disable flashing TCIL for MAP sensor failure; 0 -> disable, 1 -> enable.

- OTEMP_ERR_SW = Calibration selection switch to enable/disable flashing TCIL for Transmission overtemperature condition; 0 -> disable, 1 -> enable.

- SFT_ERR_SW = Calibration selection switch to enable/disable flashing TCIL for Shift Errors; 0 -> disable, 1 -> enable.

- TCILTM1 = Flashing TCIL "ON"/"OFF" time period, sec.

- TCIL_TM_DLY = Time after fault has occurred before the TCIL begins to flash, sec. Set to 31.875 to disable flashing.

- TP_ERR_SW = Calibration selection switch to enable/disable flashing TCIL for TP sensor failure; 0 -> disable, 1 -> enable.

- VS_ERR_SW = Calibration selection switch to enable/disable flashing TCIL for VS Sensor failure; 0 -> disable, 1 -> enable.

OUTPUTS

Registers:

- TCILTMR = See above.
- TCIL_FLASH_TMR = See above.

Bit Flags:

- TCIL_STATE = See above.

PROCESS

STRATEGY MODULE: INTRN_TCIL_STATE_COM1

```
FLG_TCS = 0 -----------------------|
 (overdrive enable)                |AND -|
                                   |     |
STIFLG = 1 ------------------|     |     |
 (in Self Test)              |OR --|     |
                             |           |OR --|  (turn TCIL off)
TCILTMR <= TCIL_TM_DLY ------|           |     | TCIL_STATE = 0
 (flashing mode entry delay)             |     | TCIL_FLASH_TMR = 0
                                         |     |
TCIL_FLASH_TMR >= TCILTM1 ---------|     |     |
 (flashing period expired)         |AND -|     |
                                   |           |
TCIL_STATE = 1 --------------------|           |
 (TCIL currently on)                           |
                                               | --- ELSE ---
FLG_TCS = 1 -----------------------|           |
 (overdrive lockout mode)          |AND -|     |
                                   |     |     |
STIFLG = 1 ------------------|     |     |     |
                             |OR --|     |OR --|  (turn TCIL on)
TCILTMR <= TCIL_TM_DLY ------|           |     | TCIL_STATE = 1
                                         |     | TCIL_FLASH_TMR = 0
TCIL_FLASH_TMR >= TCILTM1 ---------|     |     |
                                   |AND -|     |
TCIL_STATE = 0 --------------------|           |
 (TCIL currently off)                          |
                                               | --- ELSE ---
                                               |
                                               | Do not change TCIL_STATE
                                               | Increment TCIL_FLASH_TMR
```

TCILTMR LOGIC

```
STIFLG = 1 ------------------------------|
                                         |
CRKFLG = 1 ------------------------------|
                                         |
OFMFLG = 0 ------------------------|     |
                                   |     |
SFT_ERR_SW = 0 --------------|     |     |
                             |OR --|     |
SFT_FM_FLG = 0 --------------|     |     |
                                   |     |OR --| TCILTMR = 0
CC_ERR_SW = 0 ---------------|     |     |     |
                             |OR --|     |     |
CC_FM_FLG = 0 ---------------|     |     |     |
                                   |     |     |
EPC_OPEN_FLG = 0 ------------|     |     |     |
                             |OR --|     |     |
EPC_ERR_SW = 0 --------------|     |     |     |
                                   |     |     |
PDL_ERROR = 0 ---------------|     |     |     |
                             |OR --|     |     |
PDL_ERR_SW = 0 --------------|     |AND -|     |
                                   |           |
OTEMP_FM_FLG = 0 ------------|     |           |
                             |OR --|           |
OTEMP_ERR_SW = 0 ------------|     |           |
                                   |           |
TFMFLG = 0 ------------------|     |           |
                             |OR --|           |
TP_ERR_SW = 0 ---------------|     |           |
                                   |           |
MFMFLG = 0 ------------------|     |           |
                             |OR --|           |
MAP_ERR_SW = 0 --------------|     |           |
                                   |           |
VSFMFLG = 0 -----------------|     |           |
                             |OR --|           |
VS_ERR_SW = 0 ---------------|                 |
                                               | --- ELSE ---
                                               |
                                               | Increment TCILTMR
```

TCIL OUTPUT

OVERVIEW

The Transmission Control Indicator Light (TCIL), located in the instrument panel, visually indicates the status of "Overdrive Cancel" to the drive, or alerts the driver to certain transmission fault. If not indicating a fault, the light is on when overdrive is canceled; off when overdrive is enabled. A flashing light indicates a transmission fault. The transmission control switch and inhibition of fourth gear will continue to operate normally, but since the light is flashing, there is no visual indication of cancel mode selection.

DEFINITIONS

INPUTS

Bit Flags:

- TCIL_STATE = Flag indicating state of TCIL; 0 -> TCIL off, 1 -> TCIL on.

PROCESS

STRATEGY MODULE: INTRN_TCIL_REPEAT_COM1

```
TCIL_STATE = 0 -----------| De-engerize TCIL output
 (normal overdrive and    |  (turns off the transmisison control light since
  start-up mode)          |   the turned off transistor provides no ground
                          |   for the light)
                          |
                          | --- ELSE ---
                          |
TCIL_STATE = 1 -----------| Energize TCIL output
 (overdrive lockout mode)    (turns on the transmission control light since
                              the turned on transistor provides a ground for
                              the light)
```

TORQUE CALCULATION

OVERVIEW

The engine output torque, or net torque, is calculated from a table of indicated torque versus speed/load, determined at MBT spark timing, 14.6 A/F ratio and no EGR. This torque value is then adjusted by multiplying by a factor dependent on the difference between MBT spark (adjusted for actual EGR flow), actual spark advance, and if in open loop fuel control, by a factor dependent on the value of lambse. This value is then further reduced by subtracting friction torque and accessory load torque. Net torque (TQ_NET) is used in the electronic transmission control strategy in calculating the EPC required for "static capacity".

DEFINITIONS

INPUTS

Registers:

- AMT = Air Mass flow for torque calculation.
- ARCHG = Air charge inducted per intake stroke. Value is updated once per background loop at the time that AMT is computed,lb.
- EGRACT = Actual EGR rate, _%.
- ENGCYL = Number of injections per engine revolution = 2, 3, 4 for 4, 6, and 8-cylinder engines respectively.
- LAMBSE = Air/fuel equivalence ratio.
- LOAD = Normalized air charge value (ARCHG/SARCHG).
- N = Engine speed, rpm.
- N_BYTE = Byte value of engine speed, rpm.
- SAFTOT = Total spark advance, including knock and tip-in retard, deg BTDC.
- SPD_RATIO = Speed ratio across torque converter.
- SPK_DELTA = Difference between MBT spark and SAFTOT, deg BTDC.
- SPK_LAMBSE = Value of LAMBSE to be used in SPARK calculations, unitless.
- TLS_24_FLG = Torque limiting strategy - 1/2 fuel flag; 0 -> normal fuel, 1 -> 1/2 fuel.
- TLS_34_FLG = Torque limiting strategy - 3/4 fuel flag; 0 -> normal fuel, 1 -> 3/4 fuel.
- TQ_NET = Net engine torque into transmission, ft-lb.

Bit Flags:

- ACCFLG = Air conditioning clutch status flag; 1 -> A/C on.

- OLFLG = Open Loop fuel flag: 1 -> open loop fuel; 0 -> closed loop fuel.

- TLS_NV_FLG = Engine speed/Vehicle speed limiting flag; 0 -> not limiting speed, 1 -> limiting speed.

Calibration Constants:

- FN034A(LOAD) = LOAD normalizing function for torque calculation table lookups.

- FN070C(N) = Engine speed normalizing function for torque calculation table lookups.

- FN617(SPD_RATIO) = Torque converter torque ratio.

- FN618(N_BYTE) = Accessory load torque, less A/C load, ft-lb.

- FN619(N_BYTE) = Air conditioning compressor load torque, ft-lb.

- FN621(SPK_DELTA) = Indicated torque table (FN1615A) multiplier versus SPK_DELTA.

- FN623(LAMBSE) = Fuel multiplier used to calculate TQ_NET in open loop.

- FN730(SPK_LAMBSE) = Required adjustment to the base spark in order to maintain MBT as the air/fuel ratio changes.

- FN1615A = Indicated engine torque at MBT spark and no EGR, ft-lb.
  x = FN070C(N) = Normalized engine rpm.
  y = FN034A(LOAD) = Normalized LOAD.

- FN1616 = Engine friction torque, ft-lb.
  x = FN070C(N) = Normalized engine rpm.
  y = FN034A(LOAD) = Normalized LOAD.

- FN1617 = MBT spark advance with no EGR, deg BTDC.
  x = FN070C(N) = Normalized engine rpm.
  y = FN034A(LOAD) = Normalized LOAD.

- MBTEGR = Number of degrees MBT spark increases per percent EGR, deg/_%.

- SARCHG = Standard Aircharge = 4.4256E-05 * CID / of cylinders.

- TCTTA = Time constant for torque truncation aircharge filtering.

OUTPUTS

Registers:

- ARCHG = See above.

- LOAD = Normalized air charge value (ARCHG/SARCHG).
- SPK_DELTA = See above.
- SPK_LAMBSE = See above.
- TQ_NET = See above.
- TQ_OFM = Transmission input torque, ft-lb.

PROCESS

STRATEGY MODULE: INTRN_EQ_TQ_CALC_COM2

```
TLS_NV_FLG = 1 ----|
                   |
TLS_24_FLG = 1 ----|OR --| ARCHG = UROLAV(AMT/[ENGCYL*N], TCTTA)
 (1/2 fuel)        |     |  (Filter ARCHG in torque truncation)
                   |     |
TLS_34_FLG = 1 ----|     |
 (3/4 fuel)              | --- ELSE ---
                         |
                         | ARCHG = AMT/(ENGCYL * N)


Always ---------| LOAD = ARCHG / SARCHG
                | SPK_DELTA = FN1617(N,LOAD) + MBTEGR * EGRACT
                |             + FN730(SPK_LAMBSE) - SAFTOT
                           (clip SPK_DELTA to ZERO minimum)


OLFLG = 1 -----| TQ_NET = [FN1615A(N,LOAD)*FN621(SPK_DELTA)*FN623(LAMBSE)]
               |        - FN1616(N,LOAD) - FN618(N_BYTE) [- FN619(N_BYTE)]
               |           (clip TQ_NET to ZERO minimum)
               |
               | --- ELSE ---
               |
               | TQ_NET = FN1615A(N,LOAD)*FN621(SPK_DELTA) - FN1616(N,LOAD)
               |           - FN618(N_BYTE) [- FN619(N_BYTE)]
               |           (clip TQ_NET to ZERO minimum)


always --------| TQ_OFM = TQ_NET * FN617(SPD_RATIO)
```

Note: "[ ]" indicates FN619 is not always included in TQ_NET caculation. The following logic controls FN619 usage:

```
ACCFLG = 1 --------| Include FN619 in TQ_NET
                   |
                   | --- ELSE ---
                   |
                   | Do not include FN619 in TQ_NET
```

SPK_LAMBSE is one when the fuel calculation is closed loop because the air/fuel mixture will be at stoichiometry regardless of the value of LAMBSE. By assuming the SPK_LAMBSE to be one, the spark is not erroneously corrected for a mixture when closed loop fuel control has controlled to stoichiometry. Note that LAMBSE1 and LAMBSE2 are the same in the open loop calculation.

```
OLFLG = 1 -----------------| SPK_LAMBSE = LAMBSE1
 (open loop fuel control)  |
                           | --- ELSE ---
                           |
                           | SPK_LAMBSE = 1
                           |  (air/fuel mixture is close to stoichiometry)
```

E4OD TRANSMISSION CALCULATIONS

OVERVIEW

These System Equations are used in the E4OD Transmission Calculation Process.

DEFINITIONS

INPUTS

Registers:

- AM = Air Mass flow through the throttle body, lb/min.
- BP = Barometric pressure as defined in the Inferred BP Section.
- DT12S_AVG = Filtered PIP period for transmission, ticks.
- N = Engine RPM.
- NEBART = Filtered engine RPM for transmission.
- NOBART = Filtered output shaft speed.
- RATCH = Closed throttle position, counts.
- RT_GR_CUR = Current transmission gear ratio.
- RT_NOVS_KAM = Ratio of actual N/V to base N/V in KAM.
- TP = Throttle Position, counts.
- TPBART = Filtered throttle position for transmission.
- VS = Instantaneous vehicle speed.
- VSBART = Filtered vehicle speed for transmission.
- VSBART_RT = Filtered vehicle speed adjusted for RT_NOVS for transmission.

Calibration Constants:

- NVBASE = Base N/V.
- PIPFIL = Filter constant factor for DT12S_AVE, unitless.
- TCNE = Time constant for filtered RPM.
- TCTPTC = Time constant for filtered TP for converter clutch.
- TCTPTE = Time constant for filtered TP.
- TCTPTV = Time constant for filtered TP for TV pressure.

- TCVST = Time constant for vehicle speed filter.

OUTPUTS

Registers:

- BP_INTR = BP interpolation factor = FN615(BP).
- LOAD = Nondimensional, generic engine load.
- NEBART = See above.
- NOBART = See above.
- SPD_RATIO = Speed ratio across torque converter.
- TP_RATE = Throttle rate.
- TP_REL = Relative TP = TP - RATCH.
- TPBART = See above.
- VSBART = See above.
- VSBART_RT = See above.

PROCESS

STRATEGY MODULE: INTRN_E4OD_SYS_EQU_COM1

```
**********************
* FILTERED PIP PERIOD *
**********************

(Performed on PIP rising edge)

CRKFLG = 1 ----------------------| DT12S_AVG = DT12S
                                 |
                                 | --- ELSE ---
                                 |
                                 | DT12S_AVG = (1 - FK)*DT12S_AVG
                                 |             + FK*DT12S

Where:
                1
       FK = -----------
             2**PIPFIL
```

```
***************
* RELATIVE TP *
***************

TP_REL = TP - RATCH (Clip to 0 as a minimum)
```

```
***********************************************
* FILTERED ENGINE SPEED FOR TRANSMISSIONS USE *
***********************************************

NEBART = UROLAV(N,TCNE)
```

```
***********************************************
* FILTERED VEHICLE SPEED FOR TRANSMISSION USE *
***********************************************

VSBART = UROLAV(VS,TCVST)
```

```
***********************************************
* FILTERED VEHICLE SPEED ADJUSTED FOR RT_NOVS *
*            FOR TRANSMISSION USE             *
***********************************************

VSBART_RT = VSBART * RT_NOVS
```

```
**************************************************
* FILTERED THROTTLE POSITION FOR TRANSMISSION USE *
**************************************************

TPBART = UROLAV(TP,TCTPTE)


********************************************
* FILTERED TRANSMISSION OUTPUT SHAFT SPEED *
********************************************

NOBART = VSBART_RT * NVBASE


**************************
* THROTTLE POSITION RATE *
**************************

TP_RATE = TP - TPBART (TP_RATE is clipped to +/- 512 counts)


***************************
* BP INTERPOLATION FACTOR *
***************************

BP_INTR = FN615(BP)


**************************************
* SPEED RATIO ACROSS TORQUE CONVERTER *
**************************************

            NOBART * RT_GR_CUR
SPD_RATIO = ------------------
                   NEBART


*********************************************
* FILTERED THROTTLE POSITION FOR TV PRESSURE *
*********************************************

TPBARTV = UROLAV(TP,TCTPTV)


**************************************************
* FILTERED THROTTLE POSITION FOR CONVERTER CLUTCH *
**************************************************

TPBARTC = UROLAV(TP,TCTPTC)
```

VSBART_FM CALCULATION

OVERVIEW

This module defines the calculation of VSBART_FM. VSBART_FM is the vehicle speed calculated from the engine speed NEBART. It is only correct during power-on non-shifting operation. When not power-on, one-way clutches may be overrunning, resulting in VSBART_FM being lower then VSBART. During shifts, the value of VSBART_FM is frozen at the value prior to commanding the shift.

DEFINITIONS

INPUTS

Registers:

- NEBART = Filtered engine RPM for transmission.
- RT_GR_CUR = Current transmission gear ratio.
- RT_NOVS = Ratio of actual N/V to base N/V in KAM.

Bit Flags:

- FLG_SFT_IN = Shift in progress flag.

Calibration Constants:

- NVBASE = Base N/V.

OUTPUTS

Registers:

- VSBART_FM = VS calculated based on NIBART, NEBART, or NOBART.

PROCESS

STRATEGY MODULE: INTRN_CALC_VSBART_FM_COM4

```
FLG_SFT_IN = 0 -----------| VSBART_FM = NEBART / (RT_GR_CUR * RT_NOVS *
 (shift not in progress)  |   NVBASE)
                          |  (not shifting, TSS failed - used engine speed)
```

ETV OVERCURRENT MONITOR VOLTAGE

OVERVIEW

(Performed during A to D conversion)

The A to D conversion of the ETV overcurrent monitor voltage (ETVOCM) will vary depending on the actual value of VREF (+/- 5%). To increase the capability to detect a partial failure in the solenoid circuit, ETVOCM is adjusted by a calibration voltage input.

DEFINITIONS

INPUTS

Registers:

- IETVOCM = A to D conversion of the ETV overcurrent monitor voltage, counts.

- IVCAL = A to D conversion of the calibration input voltage, counts.

OUTPUTS

Registers:

- ETVOCM = Corrected ETV overcurrent monitor voltage, counts.

PROCESS

STRATEGY MODULE: INPUT_ETVOCM_COM1

ETVOCM = IETVOCM * 512/IVCAL

Where: (512/IVCAL) is clipped between 0.95 and 1.05.

TRANSMISSION CONTROL SWITCH

OVERVIEW

The momentary contact Transmission Control Switch allows the driver to 1) select an alternate shift pattern or 2) lockout overdrive (fourth gear). The function of the Transmission Control Switch is dependent on the application. On each power-up, the state of the Transmission Control Switch is 1) alternate shift pattern disabled or 2) overdrive lockout disabled.

DEFINITIONS

INPUTS

Registers:

- TM_TCS_RES = Transmission Control switch input residence timer, sec.

Bit Flags:

- FLG_FRST_TCS = Flag used to prevent multiple toggles of FLG_TCS during a single activation of TCS button; 0 -> FLG_TCS has not been toggled, 1 -> FLG_TCS has been toggled.

- ITCS = Transmission Control switch input state; 0 -> TCS depressed, 1 -> TCS not depressed.

Calibration Constants:

- TMTCS = Transmission Control switch residence time, sec.

OUTPUTS

Registers:

- TM_TCS_RES = See above.

Bit Flags:

- FLG_FRST_TCS = See above.

- FLG_TCS = Transmission control switch flag.

PROCESS

STRATEGY MODULE: INTRN_TCS_COM1

```
ITCS = 1 ---------------------------------| Allow TM_TCS_RES to count up
 (TCS button depressed, 12 volts)         |
                                          | --- ELSE ---
                                          |
ITCS = 0 ---------------------------------| TM_TCS_RES = 0
 (normal state, 0 volts)                  |  (zero residence timer)
                                          | FLG_FRST_TCS = 0
                                          |  (clear first pass flag)



TM_TCS_RES >= TMTCS ----------------|
 (button depressed long enough)     |AND -| Toggle FLG_TCS
                                    |     |  (change TCS state)
FLG_FRST_TCS = 0 -------------------|     | FLG_FRST_TCS = 1
 (1st time to toggle TCS)                 |  (set first pass flag)
```

4 x 4 LOW SWITCH

OVERVIEW

The 4 x 4 switch indicates that the driver has attempted to shift the transfer case to low range. In the "shift on the fly" package, the 4 x 4 light will flash if the transfer case has not been allowed to shift into 4 x 4 low. If the transfer case has been allowed to shift into 4 x 4 low or in the non-electronic system, the light will remain steadily on if in 4 x 4 low mode. If in this mode, the shift schedule will be adjusted by the transfer case ratio to get shifts at the correct output shift speed. This is done by modifying RT_NOVS. Since 12 volts at the module pin means normal mode and 0 volts means 4 x 4 mode, the input to the CPU is read as an inverted input.

DEFINITIONS

INPUTS

Registers:

- I4X4L = Input 4 x 4 state indicator.
- I4X4L_LST = Last pass state of I4X4L.
- TM_4X4L_RES = 4 x 4 residence timer.

Calibration Constants:

- TM4X4L = 4 X 4 low switch residence time.

OUTPUTS

Registers:

- I4X4L_LST = See above.
- TM_4X4L_RES = See above.

Bit Flags:

- FLG_4X4L = Flag indicating 4 x 4 mode; 1 -> in 4 x 4 mode.

PROCESS

STRATEGY MODULE: INTRN_E4OD_INPUT_PROCESSING_COM1

```
I4X4L_LST NOT= I4X4L -----| I4X4L_LST = I4X4L
                          |  (update state change register)
                          | TM_4X4L_RES = TM4X4L
                          |  (load residence timer)


TM_4X4L_RES = 0 ----------| FLG_4X4L = I4X4L
                          |  (pass along current state, FLG_4X4L = 1
                          |   is 4 x 4 mode)
```

RT_NOVS_KAM CALCULATION

OVERVIEW

This module calculates RT_NOVS_KAM.

DEFINITIONS

INPUTS

Registers:

- GEAR_CUR = Current transmission gear (global register).
- NEBART = Filtered engine RPM for transmission.
- NOV_ACT = Actual computed N over V.
- NOV_ACT_LST = Last pass value of NOV_ACT.
- NOVCTR = NOV calculation sampling counter.
- RT_GR_CUR = Current transmission gear ratio.
- RT_NOVS_KAM = NOV ratio in KAM.
- TM_LK_CONV = Time since converter clutch commanded on, sec.
- TM_NOV_CALC = Time since last NOV_ACT calculation.
- VSBART = Filtered vehicle speed for transmission.
- VSCTR = Count of MPH sensor errors.

Bit Flags:

- FLG_4X4L = 4X4L flag; 0 -> not in 4X4 L, 1 -> in 4X4 L.
- FLG_FRST_CM = Flag indicating a shift was commanded this background loop.
- FLG_FRST_NOV = First pass to store NOV in KAM flag; 0 -> RT_NOVS_KAM has not been loaded, 1 -> RT_NOVS_KAM has been loaded.
- FLG_LK_CM = Converter clutch commanded state; 0 -> command converter clutch unlock, 1 -> command converter clutch lock-up.
- TM_SFT_IN = Time during which shift is in progress.
- PDL_ERROR = PRNDL sensor failure; 0 -> no PRNDL sensor failure, 1 -> PRNDL sensor failure.
- ERROR_4X4L = 4x4L switch failure; 0 -> no 4x4L switch failure, 1 -> 4x4L switch failure.

- SFT_ERROR = Shift error flag; 1 -> Shift error, 0 -> No shift error.

- VSFMFLG = Vehicle speed sensor failure flag; 1 -> VSS failure, 0 -> No VSS failure.

- CC_FM_FLG = Converter clutch failure mode flag; 0 -> normal operation, 1 -> Converter clutch failure mode.

Calibration Constants:

- NOVCNT = Minimum number of good NOV samples to update KAM.
- NOVDIF = Maximum difference between NOV samples.
- NVBASE = Base N/V.
- RT4X4L = 4X4 low transfer case ratio.
- RTNVMN = Minimum valid RT_NOVS_KAM.
- RTNVMX = Maximum valid RT_NOVS_KAM.
- TMNVCAL = Time between consecutive NOV_ACT calculations.
- TMNVLK = Time delay after converter clutch commanded on before allowing NOV_ACT calculation, seconds.
- ERR_BAN_4X4L = Maximum allowed deviation between rt_novs_kam and rt_novs actual in 4x4L.

OUTPUTS

Registers:

- NOV_ACT = See above.
- NOV_ACT_LST = See above.
- NOVCTR = See above.
- RT_NOVS = Ratio of actual N/V to base N/V in RAM.
- RT_NOVS_KAM = See above.
- TM_LK_CONV = See above.
- TM_UNLK_CONV = Time since converter clutch commanded off.

Bit Flags:

- FLG_FRST_NOV = See above.
- FLG_NEW_NOV = 1 -> a new NOV_ACT has been calculated.
- FLG_NOV_KAM = Flag indicating at least one update of RT_NOVS_KAM since last KAM initialization.
- ERROR_4X4L = See above.

PROCESS

STRATEGY MODULE: INTRN_RT_NOVS_KAM_CALC_COM1

```
FLG_LK_CM = 1 ----------------------| Increment TM_LK_CONV
 (converter clutch commanded on)    | Increment TM_NOV_CALC
                                    | TM_UNLK_CONV = 0
                                    |
                                    | --- ELSE ---
                                    |
                                    | TM_LK_CONV = 0
                                    | TM_NOV_CALC = 0
                                    | Increment TM_UNLK_CONV


GEAR_CUR = 3 OR 4 ------------|
 (3rd or 4th)                 |
                              |
TM_LK_CONV >= TMNVLK ---------|
 (conv. clutch fully applied) |
                              |
TM_NOV_CALC >= TMNVCAL -------|
 (enough time since last      |
  calculation)                |
                              |AND -| NOV_ACT_LST = NOV_ACT
SFT_ERROR = 0 ----------------|     |  (update last pass NOV calc.)
                              |     | TM_NOV_CALC = 0
PDL_ERROR = 0 ----------------|     | (reset interval pacer)
                              |     | NOVCTR = NOVCTR + 1
VSFMFLG = 0 ------------------|     | (increment sample counter)
                              |     | FLG_NEW_NOV = 1
TM_SFT_IN = 0 ----------------|     | (new NOV_ACT occurred)
                              |     |
VSCTR = 0 --------------------|     |
                                    | --- ELSE ---
                                    |
                                    | FLG_NEW_NOV = 0
                                    |  (no new NOV_ACT)


FLG_NEW_NOV = 1 --------------|
                              |AND -| NOV_ACT = NEBART/(VSBART*RT_GR_CUR)
FLG_4x4L = 0 -----------------|     | (compute actual N/V, not in 4x4l)
                                    |
                                    | --- ELSE ---
FLG_NEW_NOV = 1 --------------|     |
                              |AND -| NOV_ACT = NEBART/(VSBART*RT_GR_CUR
FLG_4X4L = 1 -----------------|     |                      * RT4X4L)
                                    | (compute actual NOV, in 4x4l)
```

```
|NOV_ACT_LST - NOV_ACT| > NOVDIF ---------|
 (too much variation in NOV calculations) |OR --| NOVCTR = 0
                                          |     |  (reset sample counter)
FLG_FRST_CM = 1 --------------------------|
 (shift commanded this loop)


FLG_4X4L = 0 -----------------------------|
                                          |
NOVCTR > NOVCNT --------------------------|
 (enough consecutive matches)             |
                                          |
FLG_FRST_NOV = 0 -------------------------|AND -| RT_NOVS_KAM =
 (new value for KAM)                      |     |      (NOV_ACT/NVBASE)
                                          |     |  (store new NOV ratio in
RTNVMN <= (NOV_ACT/NVBASE) <= RTNVMX -----|     |   KAM)
                                                | FLG_FRST_NOV = 1
                                                | FLG_NOV_KAM = 1
                                                |  (indicate at least
                                                |   one update)


CC_FM_FLG = 1 ---------------------------|
                                         |
NOVCTR <= NOVCNT ------------------------|
                                         |
FLG_NEW_NOV = 0 -------------------------|OR --|
                                         |     | NO ACTION ON FAULT
FLG_4X4L = 0 ----------------------------|     | FILTER OR 4X4L ERROR
                                         |     | FLAG
FLG_NOV_KAM = 0 -------------------------|     |
                                               | --- ELSE ---
                                               |
|NOV_ACT/NVBASE - RT_NOVS_KAM| > ERR_BAN_4X4L -| error_detected = 1
                                               | CALL FAULT FILTER 691
                                               | (4x4l sw. fault filter)
                                               | ERROR_4X4L = 1
                                               |
                                               | --- ELSE ---
                                               |
                                               | error_detected = 0
                                               | CALL FAULT FILTER 691
                                               | ERROR_4X4L = 0


FLG_4X4L = 1 ---------------------|
 (in 4x4 low)                     |AND -| RT_NOVS = RT_NOVS_KAM * RT4X4L
                                  |     |
ERROR_4X4L = 0 -------------------|     |
 (4x4 is ok)                            | --- ELSE ---
                                        |
                                        | RT_NOVS = RT_NOVS_KAM
```

CHAPTER 20

SYSTEM EQUATIONS

DEFINITIONS

INPUTS

Registers:

- A3C = A/C clutch.
- AEMAP = Acceleration Enrichment Map Filter.
- AETP = Filtered TP for recognizing stable TP after AE.
- AM = Air mass flow.
- APT = Throttle mode flag.
- BG_TMR = Time to complete the previous background loop.
- BP = Barometric Pressure. (Note: Upper byte of BP_WORD.)
- BPCOR = Corrected BP = FN004(BP).
- BRAKE_INPUT = State of the brake on/off input to the module. 1 -> Brake applied, 0 -> Brake not applied.
- CLOCK_SEC = Dacable clock in seconds; 60 sec. rollover period.
- DATA_TIME = Interrupt time, clock ticks.
- DNDTI = Derivitive of RPM (unfiltered).
- EVP = EGR valve position, counts.
- EGRBAR = Rolling average EGR position.
- EOFF = Lowest filtered EGR position.
- FIRST_MPH = Flag which indicates 1st VSS edge.
- IIVPWR = Ignition voltage, A/D counts.
- INDS = Neutral / drive input.
- IVCAL = Calibration input voltage, A/D counts.
- MAP = Manifold Absolute Pressure, " Hg.
- MAPAEF = MAP sample used for aefuel calculations.
- MPHCNT = MPH sensor transition count.
- MPHTIM1 = Last MPH transition time.
- MPHTIM2 = First MPH transition time.
- N = Engine speed, RPM.

- NDDTIM =
- OI_A4LD = A4LD Solenoid or SIL Output (1 = ON).
- OLDTP = Previous A/D conversion of TP.
- OTIM = Last software TAR update time, lower 16 bits.
- PDL = PRNDL Position - from A to D conversion.
- RATCH = Lowest throttle position since start, stored in KAM.
- TAR = Throttle angle rate.
- TARTMR = S/W TAR time since OLDTP updated.
- TP = Throttle position, counts.
- TP_REL = Relative TP (TP_REL).
- TP_REL_LST = Previous value of TP_REL.
- TPBAR = Rolling average throttle angle.
- TSLMPH = Time since last rising VSS edge.
- V_MODE_SETUP = Use Throttle Mode VIP Constants in.
- VBAT = Battery voltage.
- VS = Vehicle speed.
- VSBAR = Filtered vehicle speed.
- VSCTR = Counter for unrealistic changes in vehicle speed.

Bit Flags:

- CRKFLG = Crank flag.
- FLG_LK_CM = Converter clutch lockup commanded flag.
- MUPET_FLAG = Filtered MAP update enable time: 1 -> MAP register has been updated, run AEMAP filter; 0 -> MAP has not been updated, do not run AEMAP filter.
- NDSFLG = Neutral/drive flag; 1 -> drive.
- RUNNING = RVIP enable flag.

Calibration Constants:

- AEDLMP = Minimum change in MAP to indicate manifold filling (in. Hg)
- BIHP = Calibration switch which determines if a brake on/off switch is present. 1 -> Brake on/off switch present, 0 -> Brake on/off switch not present.

- FN004 = Corrected BP as a function of actual BP.

- FN074A = Exhaust pressure as a function of AM. FN074A should be corrected to sea level when mapping the data.

- FN093 = Time constant for TAPBAR.

- FN394F = Time delay before recognition of N/D transition - Forward.

- KSF = Keypower Scaling Factor; a calibration constant which has historically been 3.731; this value can be changed on VECTOR to satisfy the requirements of different processors; a newer value for KSF is 5.5991; the user should check with the EEC Design Group to determine which value for KSF is applicable to a specific processor level.

- MAXAET = Maximum time before turning off AE.

- MAXTTM = Maximum time delay before updating OLDTP (150 msec, not calibratable).

- MPGLSW = MPG mode converter clutch development switch; 1 -> enable TRANSW logic, 0 -> disable TRANSW logic.

- NDDELT = Time before N/D, D/N switch registers.

- RACHIV = RATCH, TPBAR AND TBART initialization value.

- SMTPDL = Deadband for stable TP - AETP (counts).

- TCAEMP = Time constant for AEMP.

- TCEGR = Time constant for EGRBAR.

- TCMBAR = Time constant for MAP.

- TCN = Time constant for N.

- TCNDT_ISC = Time constant for DNDT_ISC.

- TCNDT_SPK = Time constant for DNDT_SPK.

- TCTP = Time constant for TPBAR.

- TCTPDL = Time constant for TPDLBR.

- TCVBAT = VBAT time constant. ..Typical value - 0.1 seconds.

- TCVS = Time constant for VS.

- TPDLTA = Minimum TP change for tip-out.

- TRLOAD = Transmission Load Switch.

- TSTRAT = Transmission Strategy Switch; 0 -> no transmission control (man, AOD, ATX, etc.), 1 -> Shift Indicator.

- VCAL = The value is normally 2.5 volts; this value can be changed on VECTOR to satisfy the requirements of certain processors.

- VSCNT = Increment to VSCTR when VS changes unrealistically.
- VSDELT = Maximum realistic change in vehicle speed in one background loop.
- VSTYPE = Integrated vehicle speed/cruise control system present switch; 0 -> no MPH and no VSC, 1 -> MPH and no VSC.

OUTPUTS

Registers:

- AEMAP = Acceleration Enrichment Map Filter.
- AEMTMR = AEMAP filter timer.
- CLOCK_SEC = See above.
- DNDSUP = Drive neutral select.
- DNDT_ISC = Filtered rate of change of RPM for Idle Speed Control.
- DNDT_SPK = Filtered rate of change of RPM for OSCMOD spark.
- EGRBAR = See above.
- EOFF = See above.
- FIRST_MPH = See above.
- MAPAEF = See above.
- MAPBAR = Time-dependent rolling average filter of filtered MAP.
- MAPOPE = MAP/PEXH, unitless.
- MPHCNT = See above.
- MPHTIM1 = See above.
- MPHTIM2 = See above.
- NBAR = Filtered engine RPM.
- NDSFLG = See above.
- NOVS = N/VSBAR to infer transmission gear (rpm/mph).
- OLDTP = See above.
- OTIM = See above.
- PEXH = Absolute exhaust pressure, in. Hg, FN074A(AM) * (29.875/BPCOR) + BP.

- RATCH = See above.
- TAR = See above.
- TARTMR = See above.
- TAPBAR = Filtered TP for Spark.
- TP_REL = See above.
- TP_REL_H = Relative TP (TP - RATCH) high byte only.
- TPBAR = See above.
- TPDLBR = Filtered change of throttle position.
- TSLMPH = See above.
- VACUUM = Engine manifold vacuum (BP - MAP).
- VBAT = Rolling average of instantaneous battery voltage.
- VBAT' = Instantaneous battery voltage.
- VS = See above.
- VSBAR = See above.
- VSCTR = See above.

Bit Flags:

- ACCFLG = A/C clutch status; 0 -> disengaged, 1 -> engaged.
- ACIFLG = ISC system should prepare for A/C load.
- BIFLG = Brake applied flag. 1 -> Brake applied, 0 -> Brake not applied.
- MUPET_FLAG = See above
- TARFLG = AE Demand Flag; 1 -> Manifold filling, 0 -> Manifold NOT filling.

PROCESS

STRATEGY MODULE: EQUA_LH

S2 ACCEL ENRICHMENT TP FILTER (AETP)

It is used to sense stable TP for purposes of resetting TARFLG to enable AE. The AETP time constant TCAETP is a calibratable parameter which should be large enough to prevent TARFLG reset before the TP stops moving.

AETP = UROLAV(TP,TCAETP)

```
NDSFLG = 0 -----------------------|
 (neutral indication)             |
                                  |AND -|
NDDTIM >= NDDELT -----------|     |     |
 (transmission disengaged)  |     |     |
                            |OR --|     |
TRLOAD <= 3 ----------------|           |OR --| DNDSUP = NDSFLG
 (manual trans)                         |     |  (update delayed
                                        |     |   neutral/drive flag)
NDSFLG = 1 -----------------------|     |
 (drive indication)               |     |
                                  |AND -|
NDDTIM >= FN394F -----------|     |
 (transmission engaged)     |OR --|
                            |
TRLOAD <= 3 ----------------|
 (manual trans)
```

AUTOMATIC TRANSMISSION:

DNDSUP delays strategy recognition of a transmission shift until the transmission actually engages or disengages (regardless of the state of the gear switch (or pressure switch) inputs). The time delays, FN394R and FN394F are dependent upon the type of transmission used. Therefore, calibration of these functions should be coordinated with the appropriate transmission development activity.

MANUAL TRANSMISSION:

If TRLOAD = 0, NDSFLG is forced to 0, therefore DNDSUP is always 0. If TRLOAD is 1, 2, or 3, DNDSUP will follow the state of NDSFLG with no time delay.

BRAKE INPUT

The brake input flag, BIFLG, is set/cleared based on the status of the brake input hardware present switch, BIHP; and, the status of the brake input, BRAKE_INPUT. BIFLG is used in the converter clutch control logic.

```
BIHP = 1 -------------------------------|
                                        |AND -| BIFLG = 1
BRAKE_INPUT = 1 ------------------------|     |
                                              | --- ELSE ---
                                              |
                                              | BIFLG = 0
```

CLOCK_SEC

CLOCK_SEC is a register which contains a clock in engineering units. The clock rolls over every 60 seconds and displays seconds with millisecond accuracy. As with any other background calculation, the clock is updated every background loop.

```
always --------------------------| CLOCK_SEC = CLOCK_SEC + BG_TMR
                                 |  (add current background loop time
                                 |   to the clock)


CLOCK_SEC >= 60.000 -------------| CLOCK_SEC = CLOCK_SEC + 60.000
                                 |  (allows clock to roll over)
```

EGR POSITION FILTER (EGRBAR)

The EGRBAR calculation is a time dependent rolling average filter of instantaneous EGR valve position EVP. It is updated each background pass while in RUN or UNDERSPEED mode. The EGRBAR time constant TCEGR is calibratable, but should be set to 2.0 seconds.

EGRBAR = UROLAV(EVP,TCEGR)

EGR POSITION RATCHET (EOFF)

The lowest filtered EGR position EOFF is controlled by the following logic:

```
EGRBAR < EOFF --------------------------|
                                        |
CRKFLG = 0 -----------------------------|AND -| EOFF = EGRBAR
 (RUN or UNDERSPEED mode)               |
                                        |
APT = -1 -------------------------------|
 (closed throttle mode)
```

A/C CLUTCH STATUS (ACCFLG)

ACCFLG reflects the status of the A/C Clutch via the A3C input. The A3C input differs from the ACD input which indicates whether the driver has pressed the A/C button on the instrument panel. ACD will indicate driver demand, however A3C must be used to determine whether the A/C clutch is actually engaged.

```
A3C = 1 --------------------------------| ACCFLG = 1
 (A/C on)                               | ACIFLG = 1
                                        |  (A/C Clutch engaged)
                                        |
                                        | --- ELSE ---
                                        |
                                        | ACCFLG = 0
                                        | ACIFLG = 0
                                        |  (A/C Clutch disengaged)
```

MANIFOLD ABSOLUTE PRESSURE FILTER (MAPBAR)

The MAPBAR calculation is a time dependent rolling average filter of filtered manifold absolute pressure MAP. The MAPBAR time constant TCMBAR is a calibration parameter. MAPBAR is used in the Inferred Barometric Pressure Strategy.

MAPBAR = UROLAV(MAP,TCMBAR)

ACCEL ENRICHMENT MAP FILTER (AEMAP)

The AEMAP calculation is a time dependent rolling average filter of manifold absolute pressure MAP. It is used as a means of sensing the manifold filling effect during an acceleration, especially from Idle. The AEMAP time constant TCAEMP is a calibration parameter which should be small enough to prevent a false inference of manifold filling after the MAP has reached a stable value and AE fuel is no longer required. AEMAP will be updated only if MAP has been updated within the last background loop.

The stored MAP value, MAPAEF is used for the AEMAP filter and for TAR calculation (TAR is used to enable and disable AEFUEL). This is done to ensure a consistent MAP value for all AEFUEL calculations.

```
MUPET_FLAG = 1 --------------------| MAPAEF = MAP
                                   |  (store current MAP for AEMAP
                                   |   filter and TAR calculation)
                                   | AEMAP = UROLAV(MAPAEF,TCAEMP)
                                   | AEMTMR is the sample rate
                                   |  (filter AE MAP)
                                   | MUPET_FLAG = 0
                                   |  (wait for MAP update)
                                   | AEMTMR = 0
                                   |  (reset AEMAP timer for next
                                   |   update)
```

NOTE: MUPET_FLAG is set by the foreground MAP code after the MAP conversion is done. The above logic clears the flag.

ENGINE SPEED FILTER (NBAR)

The NBAR calculation is a time dependent rolling average filter of instantaneous engine speed N. It is updated each background pass while in RUN or UNDERSPEED mode. The NBAR time constant TCN is a calibration parameter and should be set to produce a 0.5 seconds.

NBAR = UROLAV(N,TCN)

NDS - NEUTRAL DRIVE SWITCH

This switch reflects the change in transmission states (i.e., neutral/park, drive/in gear). Automatic transmissions, except AXOD, use a Neutral/Drive switch from the transmission; Manuals use a clutch switch, gear switch, or no switch. A clutch or gear switch is recommend for manuals. Among its many uses (primarily fuel control), it is most heavily used in controlling Idle Speed. The output sets a flag (NDSFLG) equal to one if the transmission is in gear (or drive) and equal to zero if the transmission is in neutral.

INDS INPUT - NEUTRAL DRIVE SWITCH INPUT

This input reflects the applied transmission load to the engine, i.e., neutral/park, drive/in-gear.

- Manual transmissions can be configured with a clutch and gear switch, a clutch switch only, a gear switch only, or neither switch. The input therefore can be used to determine a neutral state (transmission in neutral or clutch depressed) versus an in-gear state. If neither clutch nor gear switch is used, the 5-volt module pull up provides an in-gear indication which can be overridden by proper selection of the TRLOAD software switch (set TRLOAD=0).

- Non-electronic automatic transmissions typically have a two state switch which indicates neutral or drive. All transmissions except the AXOD use a mechanical switch connected to the gearshift lever. Drive is indicated by a 5-volt signal, neutral is indicated by a 0-volt signal.

  AXOD transmissions are unique in that instead of using a Neutral/Drive switch, the AXOD uses a Neutral Pressure Switch. This is a hydraulic switch which senses hydraulic pressure in the forward clutch. The voltage indicated by the NPS is opposite to that indicated by the NDS. Drive is indicated by 0 volts and neutral is indicated by 5 volts (except in overdrive). The NPS must be used in conjunction with the two other transmission hydraulic switches (THS2/3 and THS3/4) to properly decode neutral, forward, and reverse states.

- Electronic automatic transmissions typically use a position PRNDL sensor to determine the operator selected gear. The PRNDL sensor is a ratiometric sensor with six discrete resistors in series. The sensor is decoded by looking at the differing voltages produced by each of the PRNDL positions.

The engine control strategy typically requires information on the current state of engine loading. This is provided by NDSFLG. If NDSFLG = 1, the engine is loaded (transmission in gear or in drive). If NDSFLG = 0, the engine is unloaded (transmission in neutral or clutch depressed). DNDSUP, the delayed neutral/drive flag contains exactly the same information as NDSFLG except that it is delayed (see FN393F/R, NDDTIM, etc.) in an attempt to match PRNDL movement with actual application of transmission load (manual transmissions automatically get a 0 delay time).

NDSFLG or DNDSUP are typically used in idle speed control mode select and air flow computations, fuel enrichment on auto transmission neutral/drive transitions, adaptive fuel, decel fuel shutoff and vehicle speed control (as well as VIP).

TRLOAD ASSIGNMENTS

```
TRLOAD = 0 Manual trans, no clutch or gear switch, forced
           neutral (NDSFLG = 0)
       = 1 Manual trans, no clutch or gear switch
       = 2 Manual trans, one clutch or gear switch
       = 3 Manual trans
       = 4 Auto trans, non-electronic, Neutral Drive Switch
       = 5 Auto trans, non-electronic, Neutral Pressure
           Switch (AXOD)
       = 6 Auto trans, electronic, PRNDL sensor Park,
           Reverse, Neutral, Overdrive, Manual2, Manual1
           configuration.
```

NDSFLG - INSTANTANEOUS (NON-DELAYED) TRANSMISSION STATE

```
INDS < 512 -----------------------|
                                  |AND -|
TRLOAD <= 4 ----------------------|     |
 (not NPS)                              |
                                        |
TRLOAD = 0 -----------------------------|OR --| NDSFLG = 0
 (norced neutral)                       |     |  (neutral state; zero
                                        |     |   NDDTIM timer on the
TRLOAD = 6 -----------------------|     |     |   transition)
 (PRNDL sensor)                   |     |     |
                                  |AND -|     | --- ELSE ---
PDL = 7 --------------------|     |           |
 (park)                     |OR --|           | NDSFLG = 1
                            |                 |  (drive/loaded state;
PDL = 5 --------------------|                 |   zero NDDTIM timer on
 (neutral)                                    |   the transition)
```

RATE OF CHANGE OF ENGINE RPM FILTER FOR OSCMOD SPARK AND ISC

DNDT_ISC and DNDT_SPK are time dependent rolling average filters of the rate of change of engine RPM. These are updated each time a new value of N is calculated. The time constants, (TCNDT_ISC and TCNDT_SPK) are calibration parameters. DNDT_SPK is the input to FN182, the light load RPM oscillation spark multiplier. DNDT_ISC is used to calculate the Idle Fuel Modulation multiplier, ISCMOD.

For ISC:

DNDT_ISC = ROLAV(DNDTI,TCNDT_ISC)

For OSCMOD SPARK:

DNDT_SPK = ROLAV(DNDTI,TCNDT_SPK)

Where:

- DNDTI = N - N_PREV/DT_DNDT, RPM/Sec.
- N_PREV = Previous value of N.
- DT_DNDT = Time of current PIP up-edge minus time of up-edge used to calculate N_PREV, sec.

ABSOLUTE EXHAUST PRESSURE (PEXH)

Exhaust back pressure as a function of AM and altitude, in. Hg.

PEXH = FN074A(AM) * (29.875/BPCOR) + BP

Where:

- BPCOR = BP corrected = FN004(BP).

FN074A should be calibrated at sea level since the altitude correction is made by the (29.875/BPCOR) term. Note that the altitude correction used to be (29.875/BP) however actual data obtained from the altitude chamber disagreed with the calculated correction. Therefore FN004(BP) was added to allow an empirical correction. If no correction is desired, calibrate FN004 on a diagonal, that is, (0,0), (31.875,31.875). Actual data indicates that backpressure does not increase linearly with BP, but at about half that rate, roughly (0,8), (31.875,31.875). This will generate a corrected BP to be used in calculating a more accurate PEXH and PE (PFE EGR only). Overprediction of PEXH results in a smaller MAPOPE and PE which in turn results in leaner open loop fuel values and underprediction of actual EM at altitude.

NOTE: MAPOPE = MAP/PEXH

BATTERY VOLTAGE (VBAT)

The VBAT calculation is a time dependent rolling average filter of instantaneous battery voltage. It is updated each background pass while in RUN or UNDERSPEED mode. The VBAT time constant TCVBAT is a calibration parameter and should be set to 0.1 seconds.

VBAT = UROLAV(VBAT',TCVBAT)

Instantaneous battery voltage is calculated from;

VBAT' = IIVPWR * (VCAL/IVCAL) * KSF/IVCAL

Where:

- VCAL/IVCAL is clipped between 0.00867 and 0.004; and
- VBAT' is clipped to 15.94 maximum.

THROTTLE POSITION FILTER (TPBAR)

The TPBAR calculation is a time dependent rolling average filter of instantaneous throttle position (TP). It is updated each background pass while in RUN or UNDERSPEED mode. The TPBAR time constant, TCTP, is a calibration parameter and should be set to 2.0 seconds.

TPBAR = UROLAV(TP,TCTP)

THROTTLE POSITION FILTER (TAPBAR)

The TAPBAR calculation is a time and MAP dependent rolling average filter of instantaneous throttle position (TP). It is updated each background pass while in RUN or UNDERSPEED mode. The TAPBAR time constant is FN093(MAP).

TAPBAR = UROLAV(TP,FN093(MAP))

CHANGE OF THROTTLE POSITION FILTER (TPDLBR)

TPDLBR is a time dependent rolling average filter of the change (or delta) in TP on successive background loops. It is updated each background pass while in Run or Underspeed mode. TPDLBR is used in the light load RPM oscillation spark multiplier logic. The time constant, TCTPDL, is a calibration parameter.

TPDLBR = ROLAV(TP_REL - TP_REL_LST, TCTPDL)

Where:

- TP_REL = TP - RATCH (clip to 0 as a minimum)
- TP_REL_LST = Previous value of TP_REL

THROTTLE POSITION - RELATIVE TO RATCH (TP_REL)

The parameter TP_REL is an indication of the amount of throttle movement, TP, beyond the idle setting, RATCH. TP_REL is calculated every background pass, in all engine modes.

TP_REL = TP_REL_H = TP - RATCH
(clip to zero as a minimum)

Where:

- TP_REL_H = (high byte of TP_REL)

ENGINE SPEED OVER VEHICLE SPEED (NOVS)

NOVS is the ratio of engine speed over vehicle speed. It is used to infer the transmission gear ratio selected - both in automatic and manual transmissions. NOVS is currently used to specify separate entry conditions based on transmission gear for MPG mode. NOVS is set to 255 in some special cases: vehicle speed is near 0 or converter clutch is unlocked and MPGLSW = 1. MPGLSW is a development switch to enable/disable the NOVS calculation with auto transmissions until 3rd gear converter clutch lockup is achieved.

```
VSBAR <= 5 -----------------------------| NOVS = 255
 (vehicle speed low)                    |  (vehicle speed too low)
                                        |
TSTRAT = 2 -----------------|           | --- ELSE ---
 (A4LD)                     |           |
                            |           |
OI_A4LD = 0 ----------------|AND -|     |
 (unlocked)                 |     |     |
                            |     |     |
MPGLSW = 1 -----------------|     |     |
                                  |OR --| NOVS = 255
TSTRAT = 4 -----------------|     |     |  (auto trans, clutch
 (C6E4)                     |     |     |   unlocked)
                            |     |     |
FLG_LK_CM = 0 --------------|AND -|     | --- ELSE ---
 (unlocked)                 |           |
                            |           | NOVS = N/VSBAR
MPGLSW = 1 -----------------|           |  (clip NOVS between 0 and 255)
```

TSTRAT - TRANSMISSION STRATEGY SWITCH

The TSTRAT software switch selects which transmission control strategy is to be executed.

```
TSTRAT = 0 No transmission control (Manual trans, AOD, ATX, C6, C3, etc.)
       = 1 SIL (Shift Indicator Light)
       = 2 A4LD with 3 -> 4 shift control and converter clutch control
       = 3 AXOD
       = 4 C6E4 (E4OD)
       = 5 A4LD-E
       = 6 FAX-4
       = 7 AOD-E (AOD-I)
       = 8 4EAT
       = 9 CD4E
```

THROTTLE POSITION RATCHET (RATCH)

The throttle position ratchet (RATCH) continuously seeks a lower value for both throttle angle breakpoints, CLOSED THROTTLE/PART THROTTLE AND PART THROTTLE/WOT, by seeking the lowest filtered throttle angle (TPBAR). The algorithm is not used during CRANK mode. RATCH is continuously updated during the VIP Throttle Adjust Mode.

During the VIP Throttle Adjust Mode the value of RATCH is always updated to TPBAR. RATCH is clipped to RACHIV as maximum.

During RUNNING VIP RATCH is not updated.

```
CRKFLG = 1 -----------------------------------| No change to RATCH
                                              |
                                              | --- ELSE ---
                                              |
V_MODE_SETUP = 1 -----------------------------| RATCH = TPBAR
                                              |  (clip RATCH to RACHIV
                                              |   as maximum)
                                              |
TPBAR <= RATCH -------------------------|     | --- ELSE ---
                                        |     |
N > 450 RPM ----------------------------|AND -| RATCH = TPBAR
                                        |     |
RUNNING = 0 ----------------------------|     | --- ELSE ---
                                              |
                                              | No change to RATCH
```

SOFTWARE TAR CALCULATION

Background:

The S/W TAR Logic replaces the H/W TAR Logic and circuit. The H/W TAR Logic reads and A/D channel and converts the count value into deg/sec for use in the AE Fuel Calculation (normalized input to FN1303). The S/W TAR Logic calculates a throttle rate of change directly from the TP input:

$$TAR = \frac{TP - OLDTP}{9.57 * (\text{"NTIM"} - \text{"OTIM"})} \quad \text{deg/sec}$$

Where:

- OLDTP is last TP, and ("NTIM" - "OTIM") is time between successive A/D conversions.

- Because TAR is used as an anticipatory driver demand indicator, additional logic prevents the value of TAR from becoming 0 until the Accel Enrichment requirements of the engine are met.

- TAR is calculated by the software after initiation of a tip-in, until the manifold starts to fill. In general, this TAR value remains constant until after the manifold has filled (MAP - AEMAP < AEDLMP). This "latching" of TAR causes AEFUEL to be calculated until the need for it goes away, even if TP stops moving. TAR will be reset to 0 when the engine transient has dissipated (i.e., manifold has filled) or if a decel is recognized. (Throttle moves in closed direction). During part and W.O.T. mode, the higher airflow causes TP jitter. To avoid erroneous TAR calculation, as a result of this jitter, TPDLTA must be at least 20 counts.

- The original software TAR algorithm updated TAR every background loop (approximately 13 msecs at idle). This update rate prevented recognition of accels which are less than 160 deg/sec, i_.e_., TAR = (20 counts)/(9.57 counts per degree * 13 msec) = 160 deg/sec.

- The revised S/W TAR strategy implements a pacing scheme to accommodate slow accels. The software will wait up to 150 msec (approx. 12 background loops) for the TP sensor to travel TPDLTA counts. If the TP sensor has not moved within the 12 background loops, the software will update OLDTP and "OTIM".

NOTE: The first TAR calculated after a tip-in will probably be incorrect. To enable the S/W TAR calculation, set TARHP = 0.

TARFLG LOGIC

TARFLG is the mechanism for latching TAR until engine demand conditions can be met by the normal fuel equation. TARFLG is set when the manifold is filling and reset after the TP remains stable for a period of time (or as a result of a decel).

```
MAPAEF - AEMAP >= AEDLMP ---------------| TARFLG = 1
                                        |  (manifold filling)
                                        |
OLDTP - TP >= TPDLTA -------------|     | --- ELSE ---
                                  |OR --|
TP - AETP < SMTPDL ---------------|     | TARFLG = 0
                                        |  (manifold not filling)
                                        |
                                        | --- ELSE ---
                                        |
                                        | No change to TARFLG
```

TAR CALCULATION
(Do in Engineering Units Conversion)

```
TP - OLDTP >= TPDLTA ------------|
                                 |AND -| Do TAR CONVERSION LOGIC
MAPAEF - AEMAP >= AEDLMP --|     |     | OLDTP = TP
                           |OR --|     | "OTIM" = "NTIM"
TARFLG = 0 ----------------|           | TARTMR = 0
                                       |
                                       | --- ELSE ---
0 < TP - OLDTP < TPDLTA ---------|     |
                                 |     |  (steady state mode or very
MAPAEF - AEMAP >= AEDLMP --|     |     |   slow accel)
                           |OR --|AND -|  (do not calculate TAR, wait
TARFLG = 0 ----------------|     |     |   until TP - OLDTP is larger)
                                 |     | No change to "OTIM", OLD TP
TARTMR < MAXTTM -----------------|     |  or TAR
                                       |
                                       | --- ELSE ---
OLDTP - TP >= TPDLTA ------------|     |
                                 |     |
TARFLG = 1 ----------------|     |     |
                           |AND -|OR --| TAR = 0
MAPAEF - AEMAP < AEDLMP ---|     |     | OLDTP = TP
                                 |     | "OTIM" = "NTIM"
TP - AETP < SMTPDL --------------|     | TARTMR = 0
                                 |     |  (turn off AE)
TARTMR > MAXAET -----------------|     |
                                 |     | --- ELSE ---
CRKFLG = 1 ----------------------|     |
                                       | No change to TAR
                                       | "OTIM" = "NTIM"
                                       | OLDTP = TP
                                       |  (manifold is filling due
                                       |   to previous tip-in)
```

TAR CONVERSION LOGIC

```
always -------------------------|              (TP - OLDTP)
                                | TAR' = ------------------------
                                |        9.57 * ("NTIM" - "OTIM")
                                |  (calculate TAR in temporary
                                |   register)


TAR' > TAR ---------------------| TAR = TAR'
                                |  (new value of TAR is larger than
                                |   old value, use it as TAR)
                                |
                                | --- ELSE ---
                                |
                                | TAR = UROLAV(TAR',TCTAR)
                                |  (TAR is falling off, filter down
                                |   the highest value so far)
```

VACUUM - MANIFOLD VACUUM

VACUUM is used in the idle speed logic (FN820A) and the A4LD logic (FN002A).

VACUUM = BP - MAP
(clip vacuum to 0 as a minimum)

VSS - VEHICLE SPEED SENSOR

VSS is part of the EEC system and is used also by the dashboard computer. VSS is a digital input whose frequency is proportional to vehicle speed (similar to relationship of PIP signal to RPM).

VEHICLE SPEED (VS)

The variable reluctance type Vehicle Speed Sensor produces an AC signal with frequency proportional to vehicle speed. Through appropriate gearing, the sensor generates 8000 cycles/mile, for a frequency range of 0 HZ at 0 MPH to 283.3 Hz at 127.5 MPH. Interface hardware in the EEC converts the AC signal to a digital signal for input to the CPU. The strategy updates VS once per background loop if at least one new rising edge was received (MPHCNT > 0) during the previous loop. If, after 255 milleseconds, no new signals are received (< 1.75 MPH), VS is set to 0. This ensures a zero vehicle speed if the vehicle is stopped, or if the sensor fails.

On the rising edge of the vehicle speed sensor interrupt:

```
FIRST_MPH = 0 --------------------------------| FIRST_MPH = 1
                                              | MPHTIM2 = DATA_TIME
                                              | TSLMPH = 0
                                              |
                                              | --- ELSE ---
                                              |
                                              | MPHTIM1 = DATA_TIME
                                              | MPHCNT = MPHCNT + 1
                                              | TSLMPH = 0
```

Once per background; the following logic is executed:

```
VSTYPE = 0 ------------------------|
 (no vehicle speed sensor)         |
                                   |OR --| VS = 0
TSLMPH >= 255 msec ----------------|     | MPHCNT = 0
                                         | FIRST_MPH = 0
                                         |
                                         | --- ELSE ---
                                         |
                                         |         0.45 * MPHCNT
MPHCNT > 0 ------------------------------| VS = -------------------
                                         |      (MPHTIM1 - MPHTIM2)
                                         | MPHCNT = 0
                                         | MPHTIM2 = MPHTIM1
                                         |
                                         | --- ELSE ---
                                         |
                                         | Do not update VS
                                         |  or MPHTIM2
```

NOTE:

- (MPHTIM1 - MPHTIM2) must be converted from clock ticks to seconds.

- The software will handle the units conversion from clock ticks to seconds (1 tick = 3.0*10E-6 sec., 12 MHz EEC, = 2.4*10E-6 sec., 15 MHz EEC).

VEHICLE SPEED FILTER (VSBAR)

The VSBAR calculation is a time dependent rolling average filter of instantaneous vehicle speed (VS). The time constant, TCVS, is a calibration parameter.

VSBAR = UROLAV(VS,TCVS)

VSCTR LOGIC

```
Previous VS = 0 ------------------|
                                  |AND -| Exit, no action
TSLMPH >= 255 msec ---------------|     |  (don't update counter if no
                                        |   new data)
                                        |
                                        | --- ELSE ---
                                        |
|VS - Previous VS| > VSDELT ------------| VSCTR = VSCTR + VSCNT
                                        |
                                        | --- ELSE ---
                                        |
                                        | Decrement VSCTR
```

VEHICLE SPEED FILTER (VSBAR)

The VSBAR calculation is a time dependent rolling average filter of instantaneous vehicle speed (VS). It is updated each background pass while in RUN or UNDERSPEED mode. The VSBAR time constant, TCVS, is a calibration parameter.

VSBAR = UROLAV(VS,TCVS)

VS_RATEPH CALCULATIONS

OVERVIEW

Vehicle acceleration rate VS_RATEPH is calculated by differentiating VS with time.

DEFINITIONS

INPUTS

Registers:

- CTR_VSRATE = Counts the # of background loops of VSRATE.
- TM_VSRATE = Time between VSRATE calculations.
- VSBARTL = Low byte vehicle speed for transmission, mph.
- VSBARTL_PREV = Previous value of VSBARTL.
- VS_RATEPH = Filtered vehicle acceleration rate for Powertrain Hunting prevention.

Calibration Constants:

- CNTVSRATE = Number of background loops between VSRATE calculations.
- TCVSRPH = Time constant for vehicle acceleration for Powertrain Hunting prevention.

OUTPUTS

Registers:

- CTR_VSRATE = See above.
- TM_VSRATE = See above.
- VSBARTL_PREV = See above.
- VS_RATEPH = See above.

PROCESS

STRATEGY MODULE: INPUT_VS_RATE_CALC_COM2

```
CTR_VSRATE >= CNTVSRATE ----------| vsr = (VSBARTL - VSBARTL_PREV) /
                                  |        TM_VSRATE
                                  | VS_RATEPH = ROLAV(vsr,TCVSRPH)
                                  | CTR_VSRATE = 0
                                  | TM_VSRATE = 0
                                  | VSBARTL_PREV = VSBARTL
                                  |
                                  | --- ELSE ---
                                  |
                                  | CTR_VSRATE = 1 + CTR_VSRATE
```

MANIFOLD ABSOLUTE PRESSURE (MAP_WORD and MAP)

OVERVIEW

The MAP_WORD calculation is a conversion of the SCAP (Silicon CAPacitance) sensor output. The sensor outputs a digital frequency modulated signal in the 89 to 162 Hz range. Each edge output from the sensor is the equivalent of the integration of the pressure seen at the sensor since the last edge. The conversion of SCAP edges into frequency and then into a MAP_WORD value in inches of mercury is carried out during one of two foreground routines. If the sensor is determined to be not operating properly, a background routine (CNVERT) will substitute a value for MAP_WORD. The value of MAP is a less precise value of MAP_WORD that is contained within a byte.

1. During CRANK or MAPCNT register overflow:

The calculation of MAP_WORD is performed on every other SCAP edge by dividing the number of edges (2) by the time period which starts at the time of the last calculation and ends with the second edge. This produces the value of MAP_FREQ which is then converted to MAP_WORD by a linear equation.

SCAP INPUT SIGNAL

```
 ______        ______        ______        ______
|      |______|      |______|      |______|      |_

<---period---><---period---><---period---><---period
```

2. During UNDERSPEED and RUN:

The calculation of MAP_WORD is performed in three parts. In some special cases, the second part may be extended to provide a sufficient number of edges for stability in the calculation.

a. The first part is the interpolation of the fractional SCAP edge between the time of the last PIP up-edge and the next SCAP edge following that PIP up-edge. (see diagram on next page)

b. The second part is the counting of the number of SCAP edges between PIP up-edges. The counting SCAP edges continues until there have been at least MAPEDG (minimum of 2) edges since the last calculation of MAP_WORD and a PIP up-edge is reached. If MAPCNT is not at least (MAPEDG - 1) edges by the next PIP up-edge, the third part is not computed and the time for the PIP period is accumulated in DT12SA. No registers or flags are changed and a new MAP_WORD value is not computed. This feature prevents the calculation of a new MAP_WORD using only fractional data and provides increased stability in the MAP_WORD value. In exceptional cases, it may be necessary to compute the value of MAP_WORD over an engine cycle to provide enough stability to the value. At present, the only known requirement for this is at wide open throttle and must be requested by the calibrator by setting LONG_MAP_RQD to 1. If the number of SCAP edges counted and stored in MAPCNT is greater than or equal to 28 (MAX_SCAP_EDGES), then the "MAPCNT overflow flag" (MAPOFL) is set and the calculation of MAP_FREQ and MAP_WORD is performed by the same method used in CRANK. MAPOFL is cleared on the next PIP up-edge.

c. The third part is the extrapolation of the fractional SCAP edge between the time of the last SCAP edge prior to and including the PIP up-edge. After this extrapolation, the calculation of MAP_FREQ is the summation of each of these parts divided by two (number of edges in a whole SCAP period) and the time between the PIP up-edges. The value of MAP_FREQ is checked to insure that the frequency is within an acceptable band and then it is converted to MAP_WORD by a linear equation. MAP is set equal to MAP_WORD.

```
            ________________                  __________________
PIP    ____|                |_________________|                  |_________

SCAP   __     ____      ____      ____      ____      ____      ____      _
         |____|    |____|    |____|    |____|    |____|    |____|    |____|

         -->| a|<--
       ->| b  |<--   ISF = a/b (part 1)

MAPCNT =     0  ISF  1+I  2+I  3+I  4+I  5+I  6+I  (part 2)

                                    ->| d  |<-
                                         ->|c|<-   lsf = c/d (part 3)

           |<------------- DT12SA ----------->|

MAP_FREQ = [(ISF + 6 + lsf) * 0.5] / DT12SA

MAP_WORD = SLOPE(x) * MAP_FREQ - OFSET(x)
```

3. In some exceptional cases when the amplitude of the pulsations in the manifold become large and the signal at the SCAP sensor is not symmetrical for all cylinders, it may be necessary to compute the value of MAP_WORD over either an engine revolution or over an engine cycle to provide enough stability in the computed value. At present, the only known requirement for this is at wide open throttle, and must be requested by the calibrator by setting LONG_MAP_RQD to one (1). In the event that LONG_MAP_RQD is one (1), then when the throttle mode is wide open, the flag LONG_MAP_AVG is set indicating a desire to perform the long average. MAP_WORD will continue to be computed on each PIP if enough SCAP information is available until the appropriate number of cylinders have passed since start of the routine. This allows the MAP_WORD value to follow the transient changes and then stabilize. The flag LONG_MAP_FST is used to tell the system when enough cylinders have passed and the long average can actually begin. The number of cylinders to average over is computed from the calibration parameter MAP_CYCLE, which if set to zero (0), negates the long average; if set to one (1), averages over an engine revolution; and if set to two (2), averages over an engine cycle. An example of the start of the long average is shown below for a four cylinder engine with MAP_CYCLE = 2:

```
    ____    ____    ____    ____    ____    ____    ____    ____    ____    _
PIP |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
____|   |___|   |___|   |___|   |___|   |___|   |___|   |___|   |___|   |___|


MAP calculated at arrows for period indicated
    ^       ^       ^       ^       ^       ^                               ^
    |<----->|<----->|<----->|<----->|
            |<------SAMPLE PERIOD---------->|<-------SAMPLE PERIOD--------->|

        |<-----WOT-----------------------------------------------------------
        |<--- LONG_MAP_AVG = 1 ----------------------------------------------
                                            |<--- LONG_MAP_FST = 1 ----------

MAP_PIPCNT
   .0.......0......1......2......3.......4.0.....1......2......3.......4
```

4. During SCAP sensor failure:

In a background routine (CNVRT), there are two checks to verify proper SCAP sensor operation. These checks are:

a. LAST_MAP2 = LAST_MAP - this checks to see if the sensor ever started, since once an edge is captured, the time in LAST_MAP will not be equal to LAST_MAP2. b. MAPTMR >= VMPMAX - this check uses a background timer to count the amount of time since the last SCAP edge was processed and if the timer exceeds the calibration value VMPMAX, the sensor is assumed to have stopped putting out edges.

Once the sensor has been determined to have failed, the condition of the throttle position sensor is checked, and if the TP failure flag is not set, the value of MAP and MAP_WORD are set to the output of the function FN095 (TP_REL). If the TP sensor has also failed, then the value substituted for MAP and MAP_WORD is the calibration value, MAPFMM. If the sensor has failed and the CRANK flag is set, the values substituted for MAP and MAP_WORD is a constant, 29.875.

The calculation of MAP and MAP_WORD on a PIP up-edge is based on the assumption that in engines with a vacuum tap properly placed, the signal supplied to the SCAP sensor will be of near equal height and shape for all cylinders. Therefore, obtaining the value over one or more cylinders should produce the same average and no errors for partial period averaging.

At each PIP up-edge after FIRST_PIP = 1:

```
always ----------------------------------| MAPOFL = 0
                                          |  (clear MAP overflow flag)


CRKFLG = 1 -------------------------------| Return
 (in crank)                               |
                                          | --- ELSE ---
CRKFLG = 0 -------------------------|     |
 (not in crank)                     |     | DT12SA = DT12SA + DT12S
                                    |AND -|  (PIP period accumulator)
ISF_UP_FLG = 0 ---------------------|
 (not first SCAP transition after
  PIP up-edge)


LONG_ISF_UP_FLG = 0 ----------------------| LONG_DT12SA =
 (at PIP edge and in long MAP average)    |      LONG_DT12SA + DT12S
                                          |  (PIP period accumulator)
                                          | MAP_PIPCNT = MAP_PIPCNT + 1
                                          |  (increment PIP period counter)


LONG_DT12SA > 65536 ticks ----------------| LONG_MAP_AVG = 0
 (PIP period accumulator has overflowed)  | LONG_DT12SA = 0
                                          | LONG_MAPCNT = 0
                                          | MAP_PIPCNT = 0
                                          | LONG_ISF_UP_FLG = 1
                                          |  (leave long MAP mode)
```

```
always ----------------------------|       (LAST_HI_PIP - LAST_MAP)
 (compute last sample fraction)    | lsf = ------------------------
                                   |       (LAST_MAP - LAST_MAP2)
```

NOTE: MAPEDG is the minimum number of SCAP transitions between PIP up-edges required to compute a MAP update. "One" is subtracted from MAPEDG because MAPCNT is incremented by ISF (a number less than 1) on the first SCAP transition. MAPEDG should not be calibrated larger than the minimum necessary to avoid unacceptable MAP jitter. The minimum value is 2 and the maximum recommended values are: 4 cyl = 6; 6 cyl = 4; and 8 cyl = 3. Values greater than these will result in multiple PIPs between MAP updates at low engine speeds, causing slow MAP response.

DEFINITIONS

INPUTS

Registers:

- APT = At part throttle flag.
- DT12S = Time between two PIP rising edges.
- DT12SA = An accumulation of DT12S over a SCAP averaging period.
- IMAP_WORD = Raw manifold absolute pressure.
- ISF = Initial sample fraction; the ratio of time between last PIP up-edge and latest SCAP transition, and the time between the most recent two SCAP transitions.
- LONG_DT12SA = Time period accumulator for LONG_MAP_AVG.
- LONG_ISF = Initial sample fraction for LONG_MAP_AVG.
- LONG_MAP_AVG = Flag that indicates when a long period average of MAP is in progress.
- LONG_MAPCNT = SCAP edge counter in LONG_MAP_AVG.
- LAST_HI_PIP = Time of last PIP up-edge.
- LAST_MAP = Time of most recent processed SCAP transition.
- LAST_MAP2 = Time of the second most recent processed SCAP transition.
- MAP_FREQ = Integrated value of frequency in Hertz of the output of SCAP sensor.
- MAP_PIPCNT = Cylinder counter to determine period for averaging MAP over a long period.
- MAP_WORD = Same function as MAP, but with greater precision.
- MAPTMR = Free-running timer which is cleared in background if at least one SCAP edge is recognized in the foreground. Its purpose is to provide detection of a sensor failure.
- MAPUP_NORM = Set -> MAP update is complete and ready for calculation of normalized value; Clear -> Normalized value has been calculated.
- MDELTA = Latest SCAP half period.
- NEW_MAP = Flag indicating whether SCAP edge has been received to allow clearing of MAPTMR in background.
- TP_REL = Relative TP (TP_REL).
- UNDSP = Underspeed flag; 0 -> engine is in RUN mode.

Bit Flags:

- CRKFLG = State of engine mode; 0 -> Run/Underspeed, 1 -> Crank.

- ISF_UP_FLG = Flag which indicates whether initial sample has been calculated after last PIP up-edge.

- LONG_ISF_UP_FLG = Flag to indicate new MAP average completed and to computed the new ISF for long average.

- LONG_MAP_FST = Flag to indicate first pass through a long period average of

- MAPCNT = Number of SCAP transitions occurring between PIP up-edges.

- MAPOFL = Flag that indicates that MAPCNT has reached the overflow limit and that MAP calculations will be performed as during crank.

- MFMFLG = Map failure mode flag.

- MUPET_FLAG = Filtered MAP update enable time; 1 -> MAP register has been updated, run AEMAP filter, 0 -> MAP has not been updated, do not run AEMAP filter.

- TFMFLG = TP failure mode flag.

- V_VACFLG = MAPVAC error flag.

Calibration Constants:

- FN095(TP_REL) = Provides reasonable engine load value if MAP sensor is faulty. Input: TP - RATCH, counts; Output: MAP, in Hg.

- LONG_MAP_RQD = 1 = Request long MAP avg. at WOT. MAP is in progress.

- MAPBK1 = Point of intersection of the first two line segments describing MAP function (frequency versus inches).

- MAPBK2 = Point of intersection of the second and third line segments describing the MAP function (frequency versus inches).

- MAPBK3 = Point of intersection of the third and fourth segments describing the MAP function (frequency versus inches).

- MAPBK4 = Point of intersection of fourth and fifth segments describing the MAP function (frequency versus inches).

- MAPBK5 = Point of intersection of the fifth and sixth segments describing the MAP function (frequency versus inches).

- MAPEDG = Minimum number of SCAP edges to calculate a MAP value.

- MAPFMM = Value that MAP is set equal to if both SCAP and TP sensors fail.

- MAX_SCAP_EDGES = Maximum number of SCAP edges to calculate a MAP value.

- OFSET1 = Offset for the first linear equation describing MAP as function of frequency and inches of Hg.

- OFSET2 = Offset for the second linear equation describing MAP as a function of frequency and inches of Hg.

- OFSET3 = Offset for the third linear equation describing MAP as a function of frequency and inches of Hg.

- OFSET4 = Offset for the fourth linear equation describing MAP as a function of frequency and inches of Hg.

- OFSET5 = Offset for the fifth linear equation describing MAP as a function of frequency and inches of Hg.

- OFSET6 = Offset for the sixth linear equation describing MAP as a function of frequency and inches of Hg.

- SLOPE1 = Slope for the first linear equation for MAP.

- SLOPE2 = Slope for the second linear equation for MAP.

- SLOPE3 = Slope for the third linear equation for MAP.

- SLOPE4 = Slope for the fourth linear equation for MAP.

- SLOPE5 = Slope for the fifth linear equation for MAP.

- SLOPE6 = Slope for the sixth linear equation for MAP.

- TCMAPW = Time constant to use in rolling average routine for MAP_WORD.

- VMPMAX = Maximum amount of time to wait for next SCAP edge before deciding sensor has failed.

OUTPUTS

Registers:

- DT12SA = See above.

- IMAP_WORD = See above.

- ISF = See above.

- LONG_DT12SA = See above.

- LONG_MAPCNT = See above.

- MAP = See above.

- MAP_PIPCNT = See above.

- MAP_FREQ = See above.

- MAP_WORD = See above.

- MAPCNT = See above.

- MAPUP_NORM = See above.

- MAPWBAR = Rolling average of MAP_WORD.

Bit Flags:

- ISF_UP_FLG = See above.

- LONG_ISF_UP_FLG = See above.

- LONG_MAP_AVG = See above.

- LONG_MAP_FST = See above.

- MFMFLG = See above.

- MUPET_FLAG = See above.

PROCESS

STRATEGY MODULE: SYSEQ_MAP_COM1

```
LONG_MAP_AVG = 1 -------------------|
 (need long MAP average)            |AND -| LONG_MAP_FST = 1
                                    |     | fractot = LONG_ISF + lsf +
MAP_PIPCNT >=                       |     |               LONG_MAPCNT
      (MAP_CYCLE * ENGCYL) ---------|     | MAP_FREQ =
 (enough PIPs have passed)                |  (fractot * 0.5) / LONG_DT12SA
                                          | Call Subroutine "MAP_CALC"
                                          | LONG_DT12SA = 0
                                          | LONG_MAPCNT = 0
                                          | MAP_PIPCNT = 0
                                          | LONG_ISF_UP_FLG = 1
                                          |
                                          | --- ELSE ---
LONG_MAP_AVG = 0 -------------------|     |
 (need normal MAP average)          |     |  (normal MAP average)
                                    |AND -| LONG_MAP_FST = 0
MAPCNT >= (MAPEDG - 1.0) -----------|     | fractot = MAPCNT + lsf
 (enough edges to average)          |     | MAP_FREQ =
                                    |     |      (fractot * 0.5) / DT12SA
ISF_UP_FLG = 0 ---------------------|     | Call Subroutine "MAP_CALC"
                                          | LONG_DT12SA = 0
                                          | LONG_MAPCNT = 0
                                          | MAP_PIPCNT = 0
                                          | LONG_ISF_UP_FLG = 1
                                          |
                                          | --- ELSE ---
LONG_MAP_AVG = 1 -------------------|     |
 (need long MAP avg)                |     |  (normal MAP average during
                                    |     |   first long MAP average)
LONG_MAP_FST = 0 -------------------|AND -| LONG_MAP_FST = 0
 (first time for long MAP)          |     | fractot = MAPCNT + lsf
                                    |     | MAP_FREQ =
MAPCNT >= (MAPEDG - 1.0) -----------|     |       (fractot * 0.5) / DT12SA
 (enough edges to average)                | Call Subroutine "MAP_CALC"
                                          |
                                          | --- ELSE ---
LONG_MAP_AVG = 1 -------------------|     |
                                    |     |
LONG_MAP_FST = 1 -------------------|AND -| ISF_UP_FLG = 1
                                    |     | DT12SA = 0
MAPCNT >= (MAPEDG - 1.0) -----------|     | MAPCNT = 0
                                          |  (long MAP average without
                                          |   sufficient PIP's)
                                          |
                                          | --- ELSE ---
                                          |
                                          | Return
```

As each SCAP sensor edge is processed:

Clear MAP_INT flag (new SCAP transition available), then:

```
MAPOFL = 1 ------------|
 (too many edges)      |
                       |OR --|
CRKFLG = 1 ------------|     |
 (crank mode)                |AND -| MAP_FREQ = 1/(DATA_TIME - LAST_MAP2)
                             |     | Call MAP_CALC
MAPCNT >= 1 -----------------|     |
 (average over two periods)        |
                                   | --- ELSE ---
MAPOFL = 1 ------|                 |
                 |OR --|           |
CRKFLG = 1 ------|     |           |
                       |AND -|     |
MAPCNT < 1 ------------|     |     |
                             |OR --| MAPCNT = MAPCNT + 1
CRKFLG = 0 ------------|     |     |  (increment SCAP edge counter)
                       |AND -|     |
ISF_UP_FLG = 0 --------|           |
 (not 1st SCAP after PIP)          |
                                   | --- ELSE ---
                                   |
CRKFLG = 0 ------------------|     |       (DATA_TIME - LAST_HI_PIP)
                             |AND -| ISF = -------------------------
ISF_UP_FLG = 1 --------------|     |       (DATA_TIME - LAST_MAP)
 (this is 1st SCAP after PIP)      | ISF_UP_FLG = 0
                                   | MAPCNT = ISF
                                   |  (calculate inital sample fraction)
```

After incrementing MAPCNT, check the following:

```
MAPCNT >= MAX_SCAP_EDGES -----------------| MAPOFL = 1
                                          |  (too many SCAP edges)
```

NOTE: MAX_SCAP_EDGES is set to 28 and must not be changed. This prevents the registers "MAPCNT" and "FRACTOT" from overflowing if an additional SCAP edge comes in before PIP and allows for the addition of the last sample fraction to "FRACTOT" without an overflow check.

```
CRKFLG = 0 ------------------|
                             |
LONG_MAP_AVG = 1 ------------|AND -| LONG_ISF_UP_FLG = 0
 (need long MAP avg)         |     | LONG_ISF = ISF
                             |     |
LONG_ISF_UP_FLG = 1 ---------|     |
 (this is 1st SCAP after PIP)      |
                                   | --- ELSE ---
LONG_MAP_AVG = 1 ------------|     |
                             |AND -| LONG_MAPCNT = LONG_MAPCNT + 1
LONG_ISF_UP_FLG = 0 ---------|     |  (increment SCAP edge counter)
```

Additionally:

```
         Set LAST_MAP2 = LAST_MAP      (move time of previous SCAP edge)
         Set LAST_MAP = DATA_TIME      (store time of current SCAP edge)
         Set NEW_MAP flag                     (notify self test of new edge)
         Set MDELTA = LAST_MAP - LAST_MAP2    (time between SCAP edges)
```

The subroutine "MAP CALC" consists of the conversion routine for MAP frequency into MAP_WORD. The logic below is describing the method to perform the conversion in the least number of steps. A partial example of the conversion tree is shown below:

```
                                                    MAP_FREQ
                                                        |
                                                        |
                                 yes <-------- MAP_FREQ < MAPBK3 --------> no
                                  |                                         |
                                  |                                         |
               yes <----- MAP_FREQ < MAPBK1 ------> no
                |                                    |
                |                                    |
yes <-- MAP_FREQ < FMAP1 --> no     yes <-- MAP_FREQ < MAPBK2 --> no
 |                           |       |                             |
 |                           |       |                             |
```

The logic that implements this tree is:

```
MAP_FREQ < MAPBK3 -----------|
                             |
MAP_FREQ < MAPBK1 -----------|AND -| IMAP_WORD = SLOPE1 * FMAP1 - OFSET1
                             |     |
MAP_FREQ < FMAP1 ------------|     |
                                   | --- ELSE ---
MAP_FREQ >= MAPBK3 ----------|     |
                             |     |
MAP_FREQ >= MAPBK5 ----------|AND -| IMAP_WORD = SLOPE6 * FMAP2 - OFSET6
                             |     |
MAP_FREQ >= FMAP2 -----------|     |
                                   | --- ELSE ---
MAP_FREQ < MAPBK3 -----------|     |
                             |AND -| IMAP_WORD = SLOPE1 * MAP_FREQ - OFSET1
MAP_FREQ < MAPBK1 -----------|     |
                                   | --- ELSE ---
MAP_FREQ < MAPBK3 -----------|     |
                             |AND -| IMAP_WORD = SLOPE2 * MAP_FREQ - OFSET2
MAP_FREQ < MAPBK2 -----------|     |
                                   | --- ELSE ---
                                   |
MAP_FREQ < MAPBK3 -----------------| IMAP_WORD = SLOPE3 * MAP_FREQ - OFSET3
                                   |
                                   | --- ELSE ---
MAP_FREQ < MAPBK5 -----------|     |
                             |AND -| IMAP_WORD = SLOPE4 * MAP_FREQ - OFSET4
MAP_FREQ < MAPBK4 -----------|     |
                                   | --- ELSE ---
                                   |
MAP_FREQ < MAPBK5 -----------------| IMAP_WORD = SLOPE5 * MAP_FREQ - OFSET5
                                   |
                                   | --- ELSE ---
                                   |
                                   | IMAP_WORD = SLOPE6 * MAP_FREQ - OFSET6
```

After the calculation of MAP_WORD, complete the following:

```
    Set MUPET_FLAG = 1        (New MAP value for AEMAP calculation)
    Set ISF_UP_FLG = 1        (Ready to restart MAP calculation)
    Set DT12SA = 0            (Clear normal MAP time accumulator)
    Set MAPCNT = 0            (Clear normal MAP edge counter)
    Set MAPUP_NORM = 1        (New MAP value for MAP normalizing routine)

V_VACFLG = 0 -------------------------------| MFMFLG = 0
                                            | MAP_WORD = IMAP_WORD
                                            | MAPWBAR =
                                            |   ROLAV(MAP_WORD,TCMAPW)
                                            | MAP = IMAP_WORD
                                            |
                                            | --- ELSE ---
                                            |
                                            | MFMFLG = 1

MAPUP_NORM = 1 -----------------------------| MAPWBAR =
                                            |   ROLAV(MAP_WORD,TCMAPW)
                                            |
                                            | --- ELSE ---
                                            |
                                            | Do not calculation MAPWBAR
```

In the background module CNVRT, perform the following:

```
MFMFLG = 1 ----------------------------|
 (MAP sensor failure)                  |
                                       |AND -|  (crank mode and SCAP sensor
CRKFLG = 1 ----------------------------|     |   have failed)
 (in crank)                                  | MAP = 29.875
                                             | MAP_WORD = 29.875
                                             | MFMFLG = 1
                                             | MUPET_FLAG = 1
                                             | MAPUP_NORM = 1
                                             |
                                             | --- ELSE ---
MFMFLG = 1 ----------------------|           |
                                 |           |
MAPTMR >= VMPMAX ----------------|OR --|     |
 (time since last edge too long) |     |     |
                                 |     |     |
LAST_MAP2 = LAST_MAP ------------|     |AND -|  (SCAP sensor failed to start
 (MAP sensor did not start)            |     |   or has stopped; TP okay)
                                       |     | MAP = FN095(TP_REL)
TFMFLG = 0 ----------------------------|     | MAP_WORD = FN095(TP_REL)
 (throttle position sensor okay)             | MFMFLG = 1
                                             | MUPET_FLAG = 1
                                             | MAPUP_NORM = 1
                                             |
                                             | --- ELSE ---
MFMFLG = 1 ----------------------|           |
                                 |           |
MAPTMR >= VMPMAX ----------------|OR --|     |
                                 |     |     |
LAST_MAP2 = LAST_MAP ------------|     |AND -| MAP = MAPFMM
                                       |     | MAP_WORD = MAPFMM
TFMFLG = 1 ----------------------------|     | MFMFLG = 1
 (throttle sensor failure)                   | MUPET_FLAG = 1
                                             | MAPUP_NORM = 1
                                             |  (SCAP sensor never started
                                             |   or has stopped; TP sensor
                                             |   has also failed)
                                             |
                                             | --- ELSE ---
                                             |
                                             | Calculate MAP and MAP_WORD
                                             |  in the normal manner in
                                             |  "Foreground MAP"
```

```
APT = 1 -------------------------------|
 (at wide open throttle)               |
                                       |
UNDSP = 0 -----------------------------|AND -| LONG_MAP_AVG = 1
 (in run mode)                         |     |  (need to do long MAP average)
                                       |     |
LONG_MAP_RQD = 1 ----------------------|     | --- ELSE ---
 (long average requested when above          |
  conditions are met)                        | LONG_MAP_AVG = 0
```

ENGINE SPEED CALCULATION (EQNCALC)

OVERVIEW

The rpm calculation is performed in background if the foreground has signaled that a new PIP period is available for calculation - (NEW_RPM = 1). If the time since last PIP up-edge is >= 800 msec, rpm is set to zero.

DEFINITIONS

INPUTS

Registers:

- DNDTI = Rate of change of Engine rpm.
- DT_DNDT = Time delta between PIP up-edges used to calculate current and previous values of N.
- LAST_HI_PIP = Time of last PIP up-edge.
- N = Engine rpm.
- N_PREV = Previous value of N.
- TSLPIP = A timer that indicates the time since last PIP low-to-high transition.

Bit Flags:

- FIRST_RPM = Flag indicating first PIP received.
- NEW_RPM = Flag indicating new PIP information is available for calculation of rpm.

Calibration Constants:

- ENGCYL = Number of PIPs (or injections) per revolution.
- STALLN = Stall rpm; If the first rpm calculated is greater than this value assume that there was a re-init.
- TCNDT_ISC = Time constant for DNDT_ISC.
- TCNDT_SPK = Time constant for DNDT_SPK.

OUTPUTS

Registers:

- DNDT_ISC = A derivative of rpm (filtered).
- DNDTI = See above.

- DNDT_SPK = Filtered rate of change of rpm for OSCMOD.
- FIRST_RPM = See above.
- N = See above.
- N_BYTE = Byte value of N.
- N_PREV = See above.
- NBAR = Filtered engine rpm.
- NEW_RPM = See above.
- PREV_N_PIP = Time of previous high PIP used for rpm calculation.

Bit Flags:

- FIRST_PIP = Flag indicating first PIP has been received.
- REFLG = Re-init flag; 1 -> re-init occurred, 0 -> no re-init.

PROCESS

```
TSLPIP >= 800 msec ---| FIRST_PIP = 0
 (engine is stopped)  | REFLG = 0
                      | N = 0
                      | N_BYTE = 0
                      | N_PREV = 0
                      | DNDTI = 0
                      | DNDT_ISC = 0
                      | DNDT_SPK = 0
                      | NBAR = 0
                      | Do FIRST_RPM and REFLG Logic (below)
                      | Look up normalized N
                      | EXIT RPM LOGIC
                      |
                      | --- ELSE ---
                      |
NEW_RPM = 1 ----------| NEW_RPM = 0
 (new PIP information | N_PREV = N
  is available for    | N = 60/(ENGCYL * PIP period)
  calculation of rpm) | N_BYTE = byte value of N
                      |  (resolution = 16, max value = 4080)
                      | DNDTI = (N - N_PREV)/DT_DNDT
                      | DNDT_ISC = ROLAV(DNDTI,TCNDT_ISC)
                      | DNDT_SPK = ROLAV(DNDTI,TCNDT_SPK)
                      | PREV_N_PIP = LAST_HI_PIP
                      | Calculate NBAR
                      | Do FIRST_RPM and REFLG Logic (below)
                      | Look up normalized N
                      | EXIT RPM LOGIC
                      |
                      | --- ELSE ---
                      |
                      | No action
                      | EXIT RPM LOGIC
```

WHERE: DT_DNDT = Time of current PIP up-edge (LAST_HI_PIP) minus the time of the PIP up-edge last used to calculate N (PREV_N_PIP).

NOTE: PREV_N_PIP IS INITIALIZED TO THE TIME OF THE FIRST HI PIP IN FOREGROUND.

FIRST_RPM and REFLG LOGIC

```
FIRST_RPM = 0 -|
               |AND -| FIRST_RPM = 1
N >= STALLN ---|     | REFLG = 1
                     |  (this is a re-init)
                     |
                     | --- ELSE ---
FIRST_RPM = 0 -|     |
               |AND -| FIRST_RPM = 1
N < STALLN ----|     |
                     | --- ELSE ---
                     |
                     | No action
```

SPEED DENSITY AIR MASS CALCULATION

OVERVIEW

The total air mass flow into the engine (AMPEM) is computed from the basic equation:

Mass = Pressure * Volume / (Gas Law Constant * Temperature in Rankine)

Because the pressure can not be directly computed it is inferred from engine speed and manifold absolute pressure and a table of volumetric efficiency as a function of engine speed and load. BASEMD must be calibrated to provide the engine volume for this calculation as well as the Gas Law Constant.

AMPEM and AMPEMT are calculated in the same manner but use unique air mass multipliers from functions of air charge temperature and volumetric efficiency. AMPEM and AMPEMT both get EM (EGR mass) subtracted from them in the Actual Mass Flow Equations in order to determine AM and AMT (air mass). AMT is only used for air mass in the torque calculation. AM is used everywhere else that air mass is required.

DEFINITIONS

INPUTS

Registers:

- ACT = Air Charge Temperature in degrees Fahrenheit (input to FN305).
- ACT(DEG R) = ACT in degrees Rankine (not displayed).
- AM = Air Mass (input to FN074A).
- BP = Barometric Pressure " Hg (input to PEXH equation).
- BPCOR = Barometric Pressure Corrected [output from FN004(BP)].
- ECT = Engine Coolant Temperature in degrees Fahrenheit (input to FN326).
- MAPWBAR = Rolling average of Manifold Absolute Pressure Word (" Hg).
- MAP_WORD = Manifold Absolute Pressure " Hg.
- N = Engine speed.
- NORM_MAPOPE21 = The output from the evaluation of FN021(MAPOPE), the MAPOPE normalizing function (input to FN1320).
- NORM_N070 = The output from the evaluation of FN070(N), the engine speed normalizing function (input to FN1320).
- PEXH = Absolute exhaust pressure " Hg (not displayed) = FN074(AM) * (29.75/BPCOR) + BP

Calibration Constants:

- BASEMD = 0.0234393 * Engine Displacement in Liters (lbm - deg R/in. Hg - rev).
- FN004(BP) = Empirical correction to PEXH for altitude with input a function of barometric pressure.
- FN074A(AM) = Exhaust pressure as a function of air flow. NOTE: FN074A should be corrected to sea level when mapping the data. [Exhaust Pressure (Gauge) * BP / 29.875]
- FN305(ACT) = Multiplier of air mass as a function of Air Charge Temperature in degrees Fahrenheit.
- FN326(ECT) = Multiplier of air mass as a function of Engine Coolant Temperature in degrees Fahrenheit.
- FN405(ACT) = Multiplier of air mass as a function of Air Charge Temperature, degree F.
- FN1320(NORM_N070,NORM_MAPOPE21) = TABLVF is a 10 x 10 table of volumetric efficiency multipliers for air mass as a function of NORM_N070, normalized engine speed, and NORM_MAPOPE21, normalized MAPOPE.
- FN1420(NORM_N070,NORM_MAPOPE21) = 10 x 10 table of volumetric efficiency multipliers for air mass as a function of NORM_N070, and NORM_MAPOPE21, normalized MAPOPE.
- KVEFF = AMPEM and AMPEMT multiplier.

OUTPUTS

Registers:

- AMPEM = Air mass flow plus EGR mass flow, lb/min.
- AMPEMT = Air mass flow plus EGR mass flow, lb/min.
- MAPOPE = MAP_WORD/PEXH.
- MAPWBG = MAP_WORD updated once per background pass. Used in fuel pulsewidth calculation.

PROCESS

STRATEGY MODULE: EQSDMA_LL

Store the current value of MAP_WORD for this background pass:

MAPWBG = MAP_WORD

Compute the value of AMPEM and AMPEMT:

```
AMPEM = KVEFF * BASEMD * FN305(ACT) * FN326(ECT) *
        FN1320(NORM_N070,NORM_MAPOPE21) * MAPWBG * N/ACT(deg R)

AMPEMT = KVEFF * BASEMD * FN405(ACT) * FN326(ECT) *
         FN1420(NORM_N070,NORM_MAPOPE21) * MAPWBAR * N/ACT(deg R)
```

ROLLING AVERAGE ROUTINE (ROLAV/UROLAV)

OVERVIEW

The EEC-IV filters inputs using a rolling average routine. This routine requires a time constant, a sampling rate, an old average, and a new value to compute the new average. The equation is:

NEW AVERAGE = FILTER CONSTANT * NEW VALUE + (1 - FILTER CONSTANT) * OLD AVERAGE

where FILTER CONSTANT = 1/(1 + TIME CONSTANT / SAMPLE RATE); the sampling rate is the time elapsed between new calculations. For most filters, the sampling rate will equal the background loop time. The time constant is a function of the input being filtered. When the (NEW VALUE - OLD AVERAGE) * FILTER CONSTANT is less than the bit resolution of new average, the old average is incremented or decremented by 1 bit per calculation until the new average equals the new value. The strategy will specify rolling average filters using the following structure:

```
Set new_avg      = (U)ROLAV(new_value,time_const)
where new_avg    = output of rolling average filter
      ROLAV      = signed rolling average routine
      UROLAV     = unsigned rolling average routine
      new_value  = input value to filter
      time_const = time constant
```

DEFINITIONS

INPUTS

Registers:

- FK_TMR = sampling rate (seconds).
- old average = Last output from filter routine.
- new value = Most recent value of input to be filtered.

OUTPUTS

Registers:

- new average = Latest filtered value.

PROCESS

FKxxxx = 1/[1 + (TCxxx/FK_TMR)

```
new value - old average <= 1 BIT -------------| new average = new value
                                              |
                                              | --- ELSE ---
FK * (new value - old average) < 1 BIT -|     |
                                        |AND -| new average = old average
new value < old average ----------------|     |               - 1 BIT
                                              |
                                              | --- ELSE ---
FK * (new value - old average) < 1 BIT -|     |
                                        |AND -| new average = old average
new value > old average ----------------|     |               + 1 BIT
                                              |
                                              | --- ELSE ---
                                              |
                                              | new average = old average
                                              |               + FK *
                                              |  (new value - old average)
```

INPUT LIST FOR ROLLING AVERAGE FILTER ROUTINE

new value	old average	FK_TMR	TCxxxx
AM/(ENGCYL*N)	ARCHG	BG_TMR	TCTTA
EGRACT'	EGRACT	BG_TMR	TCEACT
DELOPT'	DELOPT	BG_TMR	TCDLOP
TP	DSTPBR	BG_TMR	TCDASU
TP	DSTPBR	BG_TMR	TCDASD
DSDRPM	DESNLO	BG_TMR	TCDESN
TP	AETP	BG_TMR	TCAETP
EVP	EGRBAR	BG_TMR	TCEGR
MAP	MAPBAR	BG_TMR	TCMBAR
IECT	ECT	BG_TMR	TCECT
ACT	ECT	BG_TMR	TCECT
MAPAEF	AEMAP	AEMTMR	TCAEMP
N	NBAR	BG_TMR	TCN
N	NEBART	BG_TMR	TCNE
VBAT'	VBAT	BG_TMR	TCVBAT
TP	TPBAR	BG_TMR	TCTP
TP	TAPBAR	BG_TMR	FN093
TP	TPBARTV	BG_TMR	TCTPTV
TP	TPBARTC	BG_TMR	TCTPTC
TP	TPBART	BG_TMR	TCTPTE
TAR'	TAR	BG_TMR	TCTAR
VS	VSBAR	BG_TMR	TCVS
VS	VSBART	BG_TMR	TCVST
MAP+FN1033*BP	BPPTWT	BG_TMR	TCBP
IEGO	EGOBAR	BG_TMR	VTCEGO
TP	TBART	BG_TMR	TCTPT
TP_REL-TP_REL_LST	TPDLBR	BG_TMR	TCTPDL
DNDTI	DNDT_SPK	DT_DNDT	TCNDT_SPK
DNDTI	DNDT_ISC	DT_DNDT	TCNDT_ISC
SPK_IDLE	SPK_RAMP	BG_TMR	TCRAMP
RPMERR	RPMERR_A	BG_TMR	TCBPA
RPMERR	RPMERR_S	BG_TMR	TCFBS

CALIBRATION PHILOSOPHY

1. The values for the time constants in the base calibration were calculated using the filter constants in the base calibration an assumed background loop time of 25 msec, and the following equation:

   time constant = [(1/filter constant) - 1] * sample rate

   (Sample rate approximately equals background loop time for most filters.)

2. Several filter constants were previously non-calibratable. With EMR 8-059, the time constants for these become calibratable. The effective time constants for these have been increasing as the background loops have increased. This could develop into some problems in the calibration if the time constants were to suddenly change, so the values in the base calibration are equal to the current effective time constant (assume 25 msec loop time).

3. In previous releases the filter constant was the calibration parameter. This gave an increasing time constant as rpm (loop time) increased. Now the time constant is fixed. All filters will act differently with the implementation of EMR 8-059.

TCSTRT, ACSTRT, AND INIT_TOT ROUTINE

OVERVIEW

This routine computes TCSTRT (engine coolant temperature), ACSTRT (air charge temperature), and INIT_TOT (transmission oil temperature) at start-up. If one or more of the sensor readings are out-of-range, the other sensor's reading is substituted as indicated in the logic below.

DEFINITIONS

INPUTS

Registers:

- ACT = Air charge temperature, deg F.
- ECTCNT = Number of times that ECT sensor input was read.
- IECT = A/D conversion of ECT sensor, counts.
- PUTMR = Power-up timer, sec.

Calibration Constants:

- ECTMAX = Maximum valid A/D value for ECT sensor, counts.
- ECTMIN = Minimum valid A/D value for ECT sensor, counts.
- FN703(IECT) = Transfer function for ECT sensor.
- TKYON2 = Time at which BPKYON, TCSTRT, & ACSTRT updates begin (NOT calibratable).
- TOTMAX = Maximum allowable Transmission Oil Temperature counts.
- TOTMIN = Minimum allowable Transmission Oil Temperature counts.

OUTPUTS

Registers:

- ACSTRT = ACT at start-up; arithmetic average of first 8 readings, deg F.
- ECTCNT = See above.
- INITTOT = Transmission Oil Temperature at start-up; arithmetic average of the first 8 Transmission Oil Temperature readings.
- TCSTRT = ECT at start-up; arithmetic average of first 8 readings, deg F.

PROCESS

```
PUTMR > TKYON2 ----------------|
                               |AND -| ECTCNT = ECTCNT + 1
ECTCNT < 8 --------------------|     |
                                     | --- ELSE ---
                                     |
                                     | Exit Routine
                                     | Do NOT update TCSTRT, ACSTRT,
                                     |  INIT_TOT, or ECTCNT


Always ------------------------------| ACSTRT = ACSTRT + ACT/8


IECT <= ECTMAX ----------------|
                               |AND -| TCSTRT = TCSTRT + FN703(IECT)/8
IECT >= ECTMIN ----------------|     |
                                     | --- ELSE ---
                                     |
                                     | TCSTRT = TCSTRT + ACT/8
ITOT <= TOTMAX ----------------|
                               |AND -| INIT_TOT = INIT_TOT + FN703D(ITOT)/8
ITOT >= TOTMIN ----------------|     |
                                     | --- ELSE ---
                                     |
IECT <= ECTMAX ----------------|     |
                               |AND -| INIT_TOT = INIT_TOT + FN703(IECT)/8
IECT >= ECTMIN ----------------|     |
                                     | --- ELSE ---
                                     |
                                     | INIT_TOT = INIT_TOT + ACT/8
```

PIP NOISE FILTERING

OVERVIEW

```
     ************            ************
     *          *            *          *
     *          *            *          *
     *          *            *          *
     *          *            *          *
     *          **************          ************
     |<--------------------->|
     PIP up edge to PIP up edge
             filter
     |<-------->|
     PIP up edge to PIP down edge
             filter
```

In the EEC-IV system, there is a method of noise blanking to eliminate some of the noise that occurs on the PIP input. There is a PIP up edge to PIP up edge filter. An interval value is usually picked out of a table and is in the units of clock ticks. The equivalent millisecond value of this time corresponds to some high value of engine rpm. If a pip up edge follows a previous PIP up edge at an interval less than this time, then that pip up edge is treated as noise.

The table value (TABVAL) is divided by four and is used to filter the PIP up edge to PIP down edge interval. If the computed time from the PIP down edge to the previous PIP up edge is less than (TABVAL/4), then that PIP down edge is ignored.

A typical value for the table value would be the equivalent time interval for the PIP input at maximum engine rpm. For an 8 cylinder engine, the value would be: 2.5 milliseconds (833 clock ticks for 12 MHz).

POWER MODE (PWRMODE)

OVERVIEW

This module determines Power-On or Power-Off mode and set the flag, FLG_PWR accordingly. If the throttle position is greater than the sum of the minimum throttle position, RATCH, and some calibratable delta, DELTAT, then the engine/transmission is considered to be in the Power-On mode. Otherwise, the engine/transmission is considered to be in the Power-Off mode. Hysteresis is provided for mode stability.

actual speed ratio (SPD_RATIO). If the speed ratio is greater than some calibratable minimum, SPD_PM, then the mode is considered to be Power-Off. Again, hysteresis is provided for mode stability.

DEFINITIONS

INPUTS

Registers:

- SPD_RATIO = Speed Ratio.
- RATCH = Kicker off lowest filtered throttle position.
- TP = Throttle position, counts.
- TP_REL = Relative Throttle Position, TP - RATCH.

Bit Flags:

- TFMFLG = Flag indicating TP sensor is in/out of range.

Calibration Constants:

- DELTAT = Part throttle to closed throttle breakpoint for Power mode.
- HYSTSPD = Hysteresis for SPD_PM for FMEM Power mode.
- HYSTST = Hysteresis for DELTAT for Power Mode.
- SPD_PM = Minimum Speed Ratio for Power-Off mode.

OUTPUTS

Bit Flags:

- FLG_PWR = Power Mode flag; 1 -> Power-On mode, 0 -> Power-Off mode.

PROCESS

```
TFMFLG = 0 --------------------------|
 (Normal power mode determination)   |
                                     |AND -|
TP_REL <= DELTAT --------------|S Q--|     |
                               |           |
TP_REL > DELTAT + HYSTST ------|C          |
                                           |OR --| FLG_PWR = 0
TFMFLG = 1 --------------------------|     |     |  (Power off mode)
 (FMEM power mode determination)     |     |     |
                                     |AND -|     | --- ELSE ---
SPD_RATIO >= SPD_PM -----------|S Q--|           |
                               |                 | FLG_PWR = 1
SPD_RATIO < SPD_PM - HYSTSPD --|C                   (Power on mode)
```

COLD TEMPERATURE TV SOLENOID OPERATION

OVERVIEW

The amount of Engagement TV pressure is a function of the Transmission Oil Temperature, TOT. This logic sets the flag, FLG_TVENG_CD, if the TOT sensor indicates cold and the flag, FLGTVENG_CD if moderately cold. FLG_TVENG_CD will be cleared when the time since the first engagement exceeds a calibratable value, even if TOT does not increase.

The amount of start-up TV is a function of TOT also. This logic sets the flag, FLG_TVSTR_CD if the TOT sensor indicates cold temperature.

DEFINITIONS

INPUTS

Registers:

- TSFETMR = Time since first transmission engagement, sec.

- TOT = Transmission Oil Temperature, deg. F.

Calibration Constants:

- CD_TVENG_TM = Maximum time since the first transmission engagement to use TVEMAX engagement TV, sec.

- TOTTV1 = Maximum TOT to use TVCHRG for start-up, deg F.

- TOTTV2 = Minimum ECT to use TVEMAX engagement TV, deg F.

- TOTTV3 = Minimum ECT to use TVEMOD engagement TV, deg F.

OUTPUTS

Bit Flags:

- FLG_TVENG_CD = Flag which indicates cold temperature for engagement TV: 0 -> Don't use TVEMAX in engagement TV; 1 -> Use TVEMAX in engagement TV.

- FLG_TVENG_MD = Flag which indicates moderate temperature for engagement TV: 0 -> Don't use TVEMOD in engagement TV; 1 -> Use TVEMOD in engagement TV.

- FLG_TVSTR_CD = Flag which indicates cold temperature for start-up TV: 0 -> Don't use TVCHRG in start-up TV; 1 -> Use TVCHRG in start-up TV.

PROCESS

STRATEGY MODULE: EQCTEMP_LH

```
TOT <= TOTTV1 ----------------------| FLG_TVSTR_CD = 1
                                    |
                                    | --- ELSE ---
                                    |
                                    | FLG_TVSTR_CD = 0


TOT <= TOTTV2 ----------------|
                              |AND -| FLG_TVENG_CD = 1
TSFETMR < CD_TVENG_TM --------|     |
                                    | --- ELSE ---
                                    |
                                    | FLG_TVENG_CD = 0


TOT <= TOTTV3 ----------------------| FLG_TVENG_MD = 1
                                    |
                                    | --- ELSE ---
                                    |
                                    | FLG_TVENG_MD = 0
```

DYNAMIC TV DUE TO COLD TRANSMISSION

OVERVIEW

The accumulator pressure is the hydraulic pressure used to apply, release, and hold the various clutches and bands in the transmission during shifts. When the transmission is not shifting, the hydraulic pressure is determined by the line pressure. The accumulator pressure is less than the line pressure. Therefore, it is necessary to add additional hydraulic pressure via the TV solenoid during shifts. This additional pressure is called "dynamic TV."

The accumulator in the E4OD transmission may stick when the transmission is cold. When the accumulator sticks, the transmission always operates with accumulator pressure; during shifts and during steady-state gear conditions. Of course, the pressure is inadequate during the steady-state conditions, since the EEC-IV is not adding the additional TV which is added during the shifts. With inadequate pressure, the transmission is unable to transmit large amounts of torque.

Therefore, logic is needed to also add dynamic TV when conditions exist, such that the accumulator may stick.

DEFINITIONS

INPUTS

Registers:

- TOT = Transmission Oil Temperature.

Calibration Constants:

- TOTTV4 = The maximum temperature where it is no longer necessary to add dynamic TV due to the accumulator sticking.

OUTPUTS

Registers:

- FLG_DYN_CD = Flag which indicates that it is necessary to add dynamic TV due to cold transmission conditions.

PROCESS

STRATEGY MODULE: EQCOLDTV_LH

```
TOT < TOTTV4 ----------------------------| FLG_DYN_CD = 1
                                         |
                                         | --- ELSE ---
                                         |
                                         | FLG_DYN_CD = 0
```

COLD SHIFT MULTIPLIER (CS_SFT_MULT)

OVERVIEW

During cold weather starts/drives, the initial shifts of the transmission seem delayed. This impression results from the decrease in engine torque during the early part of the cold start. Therefore, strategy which computes shift schedules as a function of the initial Transmission Oil Temperature, TOT, (temperature at the time of the start), as well as the current TOT is required.

The strategy shown below sets CS_SFT_MULT, the cold start shift multiplier, to a value other than one, when cold transmission conditions exist. Cold transmission conditions exist when the actual TOT is not greater than the initial TOT by some calibratable value. The multiplier is then applied to the appropriate shift curve to alter the shift schedules such that the shifts occur earlier during cold weather conditions.

The cold weather strategy output CS_SFT_MULT, is also used in the unconditional converter clutch unlock strategy. When CS_SFT_MULT is not equal to one, the converter clutch is unconditionally unlocked; i.e., when cold transmission conditions exist, the converter clutch is kept unlocked.

Cold shift schedule logic will be disabled when the time since the first engagement exceeds a calibratable value, even if TOT has not increased.

CS_SFT_MULT is also used to raise the vehicle speed where shifts occur when an ETV overcurrent condition has been detected (OFMFLG = 1) to provide better drivability during torque limiting operation.

DEFINITIONS

INPUTS

Registers:

- TOT = Transmission Oil Temperature, deg F.
- TSFETMR = Time since the first transmission engagement, sec.

Bit Flags

- OFMFLG = Flag, when set, indicates an ETV solenoid overcurrent.

Calibration Constants:

- CS_MAX_TIME = Maximum time allowed in cold start shift schedules.
- CS_MUL = Cold Start Multiplier.
- FN690(INIT_TOT) = Temperature required to leave cold start shift schedules.
- OFM_MUL = Shift schedule multiplier for ETV solenoid overcurrent.

OUTPUTS

Registers:

- CS_SFT_MULT = Cold Start Shift Multiplier.

PROCESS

```
OFMFLG = 1 ----------------------------| CS_SFT_MULT = OFM_MUL
                                       |
                                       | --- ELSE ---
TOT < FN690(INIT_TOT) -----------|     |
                                 |AND -| CS_SFT_MULT = CS_MUL
TSFETMR < CS_MAX_TIME -----------|     |
                                       | --- ELSE ---
                                       |
                                       | CS_SFT_MULT = 1.0
```

MLPS CONVERSION

OVERVIEW

This strategy is a fully calibratable control strategy to allow the use of a six position Manual Lever Position Sensor (MLPS) combined with the Transmission Control Switch (TCS-if present). As defined by TAPME, all transmissions have three forward positions, neutral, reverse and park. The three forward positions are overdrive, second (if TCS present) or drive (if no TCS present) and first. The second to last PRNDL position is calibratable to second or drive using the MLPS_2 parameter discussed later.

The analog MLPS input is decoded into IPDL to give one of six positions. In this decoding, deadbands can be calibrated between each position range. The calibration constants PARKHI, PARKLO, REVHI, REVLO, NEUHI, NEULO, ODHI, ODLO, MLPS_2HI, MLPS_2LO, MAN1HI, and MAN1LO are the parameters for each corresponding PRNDL position band.

DEFINITIONS

INPUTS

Registers:

- INDS = Input from manual lever position sensor (MLPS) in counts.
- IPDL = Unverified PRNDL position.
- IPDL_LST = Last unverified PRNDL position.
- PDL = Verified PRNDL position.
- TM_PDL_RES = Input residence timer.

Bit Flags:

- FLG_TCS = Transmission control switch flag.
- PARK_ERR = High vehicle speed in park error flag; 1 -> High vehicle speed in park band error, 0 -> Vehicle speed in park range.
- PDL_ERROR = PRNDL error flag; 1 -> PRNDL error, 0 -> PRNDL in range.

Calibration Constants:

- FMMMLP = IPDL value when PDL_ERROR is present.
- MAN1HI = MLPS manual 1 position band high limit.
- MAN1LO = MLPS manual 1 position band low limit.
- MLPS_2 = IPDL value when MLPS is in the second to last position; 2 -> Manual 2 has been selected (TCS present), 3 -> Drive mode/Overdrive cancel has been selected (no TCS present).

- MLPS_2HI = MLPS second to last position band high limit.
- MLPS_2LO = MLPS second to last position band low limit.
- NEUHI = MLPS neutral position band high limit.
- NEULO = MLPS neutral position band low limit.
- ODHI = MLPS overdrive position band high limit.
- ODLO = MLPS overdrive position band low limit.
- PARKHI = MLPS park position band high limit.
- PARKLO = MLPS park position band low limit.
- PDLTIM = PRNDL load residence time.
- REVHI = MLPS reverse postion band high limit.
- REVLO = MLPS reverse position band low limit.

OUTPUTS

Registers:

- IPDL = Unverified PRNDL value.
- IPDL_LST = Last unverified PRNDL value.
- PDL = Verified PRNDL value.
- PDL_LST = Last verified PRNDL value.

Bit Flags:

- PARK_ERR = See above.
- PDL_ERROR = See above.

PROCESS

STRATEGY MODULE: INTRN_MLPS_CONV_COM1

As the PRNDL switch moves from position to position, the next contact is made before the current contact is broken. This prevents sensing an open circuit condition during PRNDL transitions.

To convert voltage to counts in the INDS register:

```
TSTRAT >= 4 ------------------------------| Exit Module
(transmission                             |
 without MLPS)                            | --- ELSE ---
                                          |
                                          | INDS = PRNDL voltage * (1023/VREF)
                                          |  (where VREF = 5 Volts +/- 5%)
```

To record the last PDL and IPDL before processing a new one.

```
always -----------------------------------| PDL_LST = PDL
                                          | IPDL_LST = IPDL
```

To verify the INDS signal is in an allowable range and convert the counts into a PRNDL position.

```
PARKLO < INDS < PARKHI ------------|
                                   |AND -| ipdl = FMMMLP
VS > VSPMIN -----------------------|     | PDL_ERROR = 1
                                         | PARK_ERR = 1
                                         |
                                         | --- ELSE ---
                                         |
PARKLO < INDS < PARKHI ------------------| (PARK position)
                                         | ipdl = 7
                                         | PDL_ERROR = 0
                                         |
                                         | --- ELSE ---
                                         |
REVLO < INDS < REVHI --------------------| (REVERSE position)
                                         | ipdl = 6
                                         | PDL_ERROR = 0
                                         |
                                         | --- ELSE ---
                                         |
NEULO < INDS < NEUHI --------------------| (NEUTRAL position)
                                         | ipdl = 5
                                         | PDL_ERROR = 0
                                         |
                                         | --- ELSE ---
                                         |
ODLO < INDS < ODHI ----------------------| (OVERDRIVE position)
                                         | ipdl = 4
                                         | PDL_ERROR = 0
                                         |
                                         | --- ELSE ---
                                         |
MLPS_2LO < INDS < MLPS_2HI --------------| (MLPS_2 postion)
                                         | ipdl = MLPS_2
                                         | PDL_ERROR = 0
                                         |
                                         | --- ELSE ---
                                         |
MAN1LO < INDS < MAN1HI ------------------| (MANUAL 1 postion)
                                         | ipdl = 1
                                         | PDL_ERROR = 0
                                         |
                                         | --- ELSE ---
                                         |
                                         | (ERROR detected)
                                         | ipdl = FMMMLP
                                         | PDL_ERROR = 1
                                         | PARK_ERR = 0
```

To perform fault filtering and set SELF TEST code:

```
PDL_ERROR = 1 ---------------------|
                                   |AND -| error_detected = 1
PARK_ERR = 1 ----------------------|     | CALL FAULT FILTER 659
                                         | (HIGH VEHICLE SPEED IN PARK
                                         |  FAULT FILTERING)
                                         |
                                         | --- ELSE ---
                                         |
PDL_ERROR = 1 ---------------------------| error_detected = 1
                                         | CALL FAULT FILTER 634
                                         | (MLPS FAULT FILTERING)
                                         |
                                         | --- ELSE ---
                                         |
                                         | CALL FAULT FILTER 634
                                         | CALL FAULT FILTER 659
```

To check if a TCS has been calibrated in and if Overdrive cancel has been selected:

```
MLPS_2 = 2 ------------------------|
 (TCS present)                     |
                                   |
FLG_TCS = 1 -----------------------|AND -| IPDL = 3
 (Overdrive canceled)              |     |  (Overdrive cancel/Drive
                                   |     |   position selected)
ipdl = 4 --------------------------|     |
 (Overdrive mode selected)               |
                                         | --- ELSE ---
                                         |
                                         | IPDL = ipdl
                                         | (let input pass through)
```

To verify that a new PRNDL position has been commanded:

```
IPDL <> IPDL_LST ------------------------| TM_PDL_RES = PDLTIM
                                         |  (load residence timer)


TM_PDL_RES = 0 --------------------------| PDL = IPDL
 (MLPS reading is stable)                |  (let input value pass through)
```

CHAPTER 21

TIMERS

TIMER SUMMARY

TIMER	DESCRIPTION
A3CTMR	Time between A3C state changes.
ADPTMR	Adaptive fuel timer (seconds)
ATMR1	Time since start (time since exiting CRANK mode) (seconds)
ATMR2	Time since engine coolant temperature became greater than TEMPFB (seconds)
ATMR3	Time since entering run mode (seconds)
AWOTMR	Time in wide open throttle (seconds)
BYPTMR	Thermactor air bypass-enable timer (seconds)
CRKPIP_CTR	PIP Counter for Cranking fuel
CTTMR	Time at closed throttle timer (0.125 seconds)
EGRTMR	EGR enabled timer (seconds)
HLTMR	High load timer (0.125 seconds)
HMUTMR	High MAPPA upstream air timer.
HTPTMR	Heat protection timer (seconds)
IDLTMR	Idle time (seconds)
ISCTMR	RPM Sample/KAM Update Delay Timer (seconds)
LMBTMR	Low MAP bypass timer (seconds)
LUTIMR	Transmission lock-up control timer (0.125 seconds)
MPGTMR	MPG mode control timer (seconds)
MULTMR	Time since incrementing LAMMUL (0.001 seconds)
NACTMR	Time not at closed throttle (seconds)
NDDTIM	Time since neutral/drive switch state change (0.125 seconds)
OSCTMR	OSCMOD delay timer (0.125 seconds)
PUTMR	Time since CPU power-up (0.001 seconds)
SETTMR	RPM Control Entry Delay Timer (0.125 seconds)
SHFTMR	Shift in progress timer (0.125 seconds)
TARTMR	Time since OLDTP was updated (0.001 seconds)

TSLEGO	Time since last EGO switch (0.001 seconds)
TSLPIP	Time since last PIP (0.001 seconds)
V_NACTMR_CUM	Accumulative time at part WOT
VOLTMR	Time of low battery voltage
WOTTMR	Time at wide open throttle

DEFINTIONS

INPUTS

Registers:

- A3C = A/C clutch.
- APT = Register indicating throttle mode.
- ATMR1_LST = Last calculated value of ATMR1.
- BP = Barometric Pressure, in. Hg.
- CRKPIP_CTR = Foreground register to count PIPs for CRANK fuel.
- CRKPIP_CTR_BG = Background register equivalent of CRKPIP_CTR.
- DASPOT = Dashpot desired mass air flow.
- ECT = Engine Coolant Temperature, deg F.
- GEAR_CUR = Register indicating current transmission gear.
- MAPPA = MAP/BP, unitless
- MAP = Manifold Absolute Pressure, in. Hg.
- N = RPM.
- NEW_PIP = New high PIP has occurred.
- NOVS = Engine speed over vehicle speed ratio, rpm/mph.
- N_BYTE = Byte form of engine speed, rpm.
- NLAST = RPM at P.T.; differentiates decel/idle.
- RPMERR_A = RPMERR(DESIRED RPM - N) filtered for ISC.
- TPDLBR = Filtered change of TP.
- VBAT = Rolling average of IKYPWR.
- VSBAR = Rolling average of Vehicle Speed, mph.
- WRMEGO = EGO sensor should be warm flag.

Bit Flags:

- CFMFLG = Flag indicating the ECT sensor is out of range.
- CRKFLG = Crank flag.
- FLG_STALL = Indicated a stall has occured.
- IDLFLG = Flag which is set to 1 when at closed throttle and below "IDLRPM" rpm. See IDLTMR logic in this chapter.
- ISCFLG = Mode indicator flag.
- LESFLG = 1 -> EGO is not switching.
- MFMFLG = Flag indicating the MAP sensor has failed.
- MPGFLG = Flag that indicates Fuel Economy Mode if set to 1.
- OLFLG = Flag indicating open loop fuel control.
- UNDSP = Underspeed flag.

Calibration Constants:

- AFECT1 = Min. ECT for starting the adaptive fuel timer
- AFECT2 = Overtemp. ECT to disable adaptive learning
- BYRPM = Maximum RPM for closed throttle bypass.
- BYRPMH = Hysteresis for BYRPM.
- CHGTM = Time delay after leaving closed throttle to permit VOLTMR to decrement, sec.
- CRKCTR_RESET = CRKPIP_CTR reset switch; 1 -> reset CRKPIP_CTR upon stall.
- CRKPIPCNT2 = CRKPIP_CTR reset value for UNDERSPEED to CRANK transitions, sec.
- CRKTM1 = Time in RUN or UNDERSPEED below which CRKPIP_CTR is reset to CRKPIPCNT2, sec.
- CRKTMR_INC = Calibration switch which determines whether CRKPIP_CTR counts when the engine state is out of CRANK.
- DASCTL = Lower daspot limit to allow RPM control (PPM).
- EGRTD7 = Time delay to ramp EGR turn on.
- FN900(VSBAR) = Time delay before enabling MPG Mode as a function of vehicle speed. NOTE: Vehicle Speed is always ZERO if a Vehicle Speed Sensor is not present.
- FN1360 = Stabilized Open Loop Fuel 8 X 10 table of lambda values as a function of N and PERLOAD.

- HIMAPF = Highest MAP that allows BYPTMR to count down.
- HMUMAP = A calibratable minimum MAPPA to increment HMUTMR for upstream air.
- HMUMPH = HMUMAP hysteresis term.
- IDLRPM = Max RPM for closed throttle mode idle, RPM.
- IDRPMH = Hysteresis for IDLRPM, RPM.
- ISCTM = Pacing to evaluate rate of change of engine speed.
- LAMRHYS = Hysteresis for LAMRICH.
- LAMRICH = Minimum lambse value for enrichment.
- LESTM = Time delay before forced open loop fuel after last ego switch (seconds).
- LMBMAP = Minimum decel MAP to increment LMBTMR.
- LOMAPF = Lowest MAP that allows BYPTMR to count up.
- LOWVOL = System voltage level, below which the battery is discharging, volts.
- MAPAHI = Maximum MAPPA value for MPG mode, unitless.
- MAPLO = Minimum MAP value for MPG mode, in. Hg.
- MAPLOH = Hysteresis for MAPLO, in. Hg.
- MAXTIM = Maximum time to wait for TAR change on SW TAR.
- MPAHIH = Hysteresis for MAPAHI, unitless.
- MPGCTH = Maximum ECT for Fuel Economy Mode, deg F.
- MPGCTL = Minimum ECT for Fuel Economy Mode, deg F.
- MPGGR = Minimum gear for Fuel Economy Mode.
- MPGNOV = Maximum NOV for Fuel Economy Mode, rpm/mph.
- MPGRPH = Hysteresis for MPGRPM, rpm.
- MPGRPM = Minimum RPM for Fuel Economy Mode, rpm.
- MPGRT = Minimum delay time to re-enter MPG mode once exited, sec.
- MPMNBP = Minimum BP for Fuel Economy Mode, in. Hg.
- MPNBPH = Hysteresis for MPMNBP, in. Hg.
- MPNOVH = Hysteresis term for MPGNOV, rpm/mph.
- NACTMR = Time at P.T. or WOT.

- NDIF = Time dependent RPM limit to differentiate decel from idle.

- O_8500_SW = Calibration switch to select vehicle speed or transmission gear for Fuel Economy Mode determination. "0" selects vehicle speed; "1" selects transmission gear. Set to "1" only for over 8500 GVW aplications with E4OD transmission.

- OLITD3 = Time to go back to closed loop fuel control (see Closed Loop/ Open Loop Logic). Closed loop from IDLTMR = 0 to OLITD1, open loop fuel from IDLTMR = OLITD1 to OLITD3, then back to closed loop fuel when IDLTMR reset to
  0 at OLITD3. Closed loop/Open loop feature can be calibrated out by setting OLITD3 to 255 sec.

- RAMPSW = Ramp EGR on switch; 0 -> ramp on every EGR turn on, 1 -> ramp first egr turn on only.

- RPMDED = RPM deadband - C/L ISC.

- SHFDLT = a calibration parameter giving TP change to infer M/T shift.

- TEMPFB = Warm Engine Temperature

- VOLHYS = Hysteresis term for LOWVOL, volts.

- VOLTCLP = Maximum clip value to freeze VOLTMR at upper threshold.

- VSMPG = Minimum vehicle speed value for MPG mode, mph.

- VSMPGH = Hysteresis for VSMPG, mph.

- VSTYPE = Must be set to 1 if Vehicle Speed Sensor is present.

OUTPUTS

Registers:

- ATMR1_LST = See above.

- CRKPIP_CTR = See above.

- CRKPIP_CTR_BG = See above.

- NEW_PIP = See above.

Bit Flags:

- IDLFLG = See above.
- LESFLG = See above.
- MPGFLG = See above.
- MPGTFG = Flag indicating a transition from Fuel Economy Mode is in progress.

PROCESS

STRATEGY MODULE: TIMER_LH

A3CTMR - A3C TRANSITION TIMER (0.001 sec)

The A3CTMR measures the time between A3C state changes.

```
always ------------------------------------------| Increment A3CTMR
```

A3CTMR is reset to 0 on any A3C transition - (LSTA3C - A3C <> 0 in successive background passes).

```
LSTA3C - A3C <> 0 -------------------------------| LSTA3C = A3C
 (A3C state changed)                             | A3CTMR = 0
```

Where, LSTA3C is the last pass A3C flag.

ADPTMR - ADAPTIVE FUEL ENABLE TIMER

```
RUN mode --------------------------------|
                                         |AND -| Increment ADPTMR
AFECT1 <= ECT <= AFECT2 ------------------|     |
                                               | --- ELSE ---
                                               |
                                               | ADPTMR = 0
```

NOTE: These coolant temperature parameters should bracket normal engine coolant temperature range.

ATMR1 - TIME SINCE ENGINE START-UP

```
CRKFLG = 0 -----------------------------------| ATMR1_LST = ATMR1
 (run OR underspeed)                          |  (save last calculated
                                              |   value of ATMR1)
                                              | Increment ATMR1
                                              |  (calculate new value)
                                              |
                                              | --- ELSE ---
                                              |
                                              | ATMR1_LST = ATMR1
                                              |  (save last calculated
                                              |   value of ATMR1)
                                              | ATMR1 = 0
```

ATMR2 - TIME SINCE ENGINE COOLANT TEMPERATURE BECAME GREATER THAN TEMPFB

```
ECT > TEMPFB----------------------------|S Q -| Increment ATMR2
                                        |     |
                                        |     | --- ELSE ---
NEVER ----------------------------------|C    |
                                              | ATMR2 = 0
```

NOTE: Except at power-up initialization; timer is used to delay closed loop fuel and EGR after a cold start.

ATMR3 - TIME SINCE ENTERING RUN MODE

```
UNDSP = 0 ------------------------------------| Increment ATMR3
 (RUN mode)
```

AWOTMR - TIME AT WIDE OPEN THROTTLE

```
APT = 1 --------------------------------------| Increment AWOTMR
 (wide open throttle mode)                    |
                                              | --- ELSE ---
                                              |
                                              | AWOTMR = 0
```

BYPTMR - THERMACTOR AIR BYPASS-ENABLE TIMER

```
LOMAPF < MAPPA < HIMAPF ----------------|
                                        |
APT >= 0 -------------------------------|AND -| Increment BYPTMR
 (part or wide open throttle)           |     | Clip at BYMAP
                                        |     |
WRMEGO = 1 -----------------------------|     | --- ELSE ---
                                              |
                                              | Decrement BYPTMR
                                              | Clip at 0
```

CRKPIP_CTR - PIP COUNTER FOR CRANKING FUEL

CRKPIP_CTR is used as a Crank Fuel Multiplier to do a lean-out or fuel shut-off during sub-zero cold operation. Some calibrators are using the logic to improve restarts following a stall during -20 deg F testing by turning off the fuel during the first seconds of crank. This allows the engine to restart on the residual fuel remaining in the manifold after the stall.

```
CRKCTR_RESET = 1 ------------------|
                                   |
FLG_STALL = 1 ---------------------|AND -| Reset counter part way
                                   |     | CRKPIP_CTR = CRKPIPCNT2
ATMR1_LST <= CRKTM1 ---------------|     | CRKPIP_CTR_BG = CRKPIPCNT2
 (in run or underspeed less              |
  than CRKTM1 seconds)                   |
                                         | --- ELSE ---
CRKFLG = 1 ------------------------|     |
 (in crank)                        |OR --| Increment CRKPIP_CTR every
                                   |     |  rising edge of PIP
CRKTMR_INC > 0 --------------------|     | CRKPIP_CTR_BG = CRKPIP_CTR
                                         |
                                         | --- ELSE ---
                                         |
                                         | Freeze CRKPIP_CTR
                                         | CRKPIP_CTR = CRKPIP_CTR_BG
```

In Foreground PIP_DATA Module:

```
always ---------------------------------------| Increment CRKPIP_CTR every
                                              |  rising edge of PIP
                                              | Clip CRKPIP_CTR at 255
                                              |  as a maximum
```

CTTMR - TIME AT CLOSED THROTTLE

```
APT = -1 -------------------------------------| Increment CTTMR
 (closed throttle mode)                       |
                                              | --- ELSE ---
                                              |
                                              | CTTMR = 0
```

EGRTMR - EGR ENABLED TIMER

```
EGR ENABLED ----------------------------------| Increment EGRTMR
                                              | Clip at EGRTD7
                                              |
                                              | --- ELSE ---
                                              |
RAMPSW = 0 -----------------------------------| EGRTMR = 0
 (ramp EGR on every time)                     |  (prepare for next ramp)
                                              |
                                              | --- ELSE ---
                                              |
RAMPSW = 1 -----------------------------------| Freeze EGRTMR
 (ramp EGR on first time only)                |  (do not ramp next time)
```

HLTMR - HIGH LOAD TIMER

The HLTMR delays Open Loop fuel control during crowds. Running Closed Loop fuel during crowds eliminates the need for Upstream Air during those conditions.

```
FN1360 <= LAMRICH ----------------------|S Q -| Increment HLTMR
                                        |     |
FN1360 > LAMRICH + LAMRHYS -------------|C    | --- ELSE ---
                                              |
                                              | HLTMR = 0
```

HMUTMR - HIGH MAPPA UPSTREAM AIR TIMER

```
MAPPA >= HMUMAP + HMUMPH ---------------|S Q -| Increment HMUTMR
                                        |     | HUMTMR_FLG = 1
MAPPA < HMUMAP -------------------------|C    |
                                              | --- ELSE ---
                                              |
                                              | HMUTMR = 0
                                              | HMUTMR_FLG = 0
```

HTPTMR - HEAT PROTECTION TIMER (BANKLINE)

```
APT = -1 -------------------------------|
 (closed throttle mode)                 |
                                        |AND -| Increment HTPTMR
N_BYTE < BYRPM --------------------|S Q -|     |  (clipped at BYSTM4)
                                   |           |
N_BYTE > BYRPM + BYRPMH -----------|C          | --- ELSE ---
                                               |
                                               | Decrement HTPTMR
                                               |  (clipped at 0)
```

IDLTMR - TIME AT IDLE

```
APT = -1 -------------------------------|
 (closed throttle mode)                 |
                                        |AND -| Increment IDLTMR
N < IDLRPM ------------------------|S Q--|     | IDLFLG = 1
                                   |           |
N > IDLRPM + IDRPMH----------------|C          | --- ELSE ---
                                               |
                                               | IDLTMR = 0
                                               | IDLFLG = 0


IDLTMR > OLITD3 -------------------------------| IDLTMR = 0
```

ISCTMR - RPM SAMPLE / KAM UPDATE DELAY TIMER (seconds)

This timer is used for two separate purposes in the normal idle speed control strategy. It is always cleared in dashpot-preposition mode.

- In RPM conrol mode, it is used to prevent ISCKAM updates while RPMERR_A is outside the deadband allowed for ISCKAM learning.

- In dashpot mode, it is used to pace the rate of change of RPM checks to determine whether or not to go to lockout mode.

```
In Mode Select Logic -------------|
                                  |AND -|
APT >= 0 ------------------------|      |
 (preposition mode)                     |
                                        |
In ISCKAM Update Logic -----------|     |
                                  |AND -|OR --| ISCTMR = 0
|RPMERR_A| > RPMDED --------------|     |     |
 (outside deadband)                     |     | --- ELSE ---
                                        |     |
In Lockout Logic -----------------|     |     | Increment ISCTMR
                                  |     |
ISCFLG < 1 -----------------------|     |
 (not RPM Control or Lockout)     |     |
                                  |AND -|
ISCTMR >= ISCTM ------------------|
 (time to check RPM)              |
                                  |
|N - NLAST| > NDIF ---------------|
 (RPM changing too fast)
```

LMBTMR - LOW MAP BYPASS TIMER

```
MAP <= LMBMAP --------------------------------| Increment LMBTMR
                                              |
                                              | --- ELSE ---
                                              |
                                              | LMBTMR = 0
```

LUTIMR - TRANSMISSION LOCK-UP CONTROL TIMER

Refer to the desired transmission section (AXOD or A4LD) for the LUTIMR logic.

MPGTMR - MPG MODE CONTROL TIMER

MPGTMR enables Fuel Economy Mode as a function of Vehicle Speed or Transmission gear, provided the Engine is operating in a steady state mode (cruise) within a limited range of Manifold Pressures and Engine Coolant Temperatures. MPGTMR is used to control the Fuel Economy Mode Flag (MPGFLG), which is used to select the Fuel Economy Mode calibrations for Fuel, Spark, EGR and Thermactor.

```
CFMFLG = 0 --------------------------------|
                                           |
MFMFLG = 0 --------------------------------|
                                           |
OLFLG = 0 ---------------------------|     |
                                     |OR --|
MPGFLG = 1 --------------------------|     |
                                           |
IDLFLG = 0 --------------------------------|
                                           |
APT < 1 -----------------------------------|
                                           |
ECT >= MPGCTL -----------------------------|
                                           |
ECT <= MPGCTH -----------------------------|AND -| Increment MPGTMR
                                           |     | Clip to FN900(VSBAR) as
GEAR_CUR >= MPGGR -------------------|     |     |  a maximum
                                     |OR --|     |
O_8500_SW = 0 -----------------------|     |     | --- ELSE ---
                                           |     |
N_BYTE >= MPGRPM + MPGRPH -----|           |     | Decrement MPGTMR
                               |           |     | Clip to FN900(VSBAR) -
MAP >= MAPLO + MAPLOH ---------|           |     |  MPGRT as a maximum
                               |           |     | Clip to ZERO as a minimum
MAPPA <= MAPAHI - MPAHIH ------|           |
                               |           |
BP >= MPMNBP + MPNBPH ---------|AND -|S Q -|
                               |     |
NOVS <= MPGNOV - MPNOVH -------|     |
                               |     |
VSBAR >= VSMPG + VSMPGH -|     |     |
                         |     |     |
VSTYPE = 0 --------------|OR --|     |
                         |           |
O_8500_SW = 1 -----------|           |
                                     |
```

(continued on next page)

(continued from previous page)

```
N_BYTE < MPGRPM ---------------|      |
                               |      |
MAP < MAPLO -------------------|      |
                               |      |
MAPPA > MAPAHI ----------------|      |
                               |OR --|C
BP < MPMNBP -------------------|
                               |
NOVS > MPGNOV -----------------|
                               |
VSBAR < VSMPG -----------------|
```

MPG MODE FLAG LOGIC (MPGFLG)

```
MPGTMR >= FN900(VSBAR) -----------------------| MPGFLG = 1
                                              |
                                              | --- ELSE ---
                                              |
                                              | MPGFLG = 0
```

MPG MODE TRANSITION FLAG SET LOGIC (MPGTFG)

```
Previous MPGFLG = 1 --------------------|
                                        |AND -| MPGTFG = 1
Current MPGFLG = 0 ---------------------|     |  (transition from MPG Mode)
```

MULTMR - TIME SINCE INCREMENTING LAMMUL

```
always ---------------------------------------| Increment MULTMR
```

NOTE: MULTMR is periodically set to 0 within the Open Loop Fuel Logic.

NACTMR - NOT AT CLOSED THROTTLE TIMER

```
APT = 0 --------------------------------|
 (part throttle mode)                   |
                                        |OR --| Increment NACTMR
APT = 1 --------------------------------|     |
 (wide open throttle mode)                    | --- ELSE ---
                                              |
                                              | NACTMR = 0
```

NDDTIM - TIME SINCE NEUTRAL/DRIVE SWITCH STATE CHANGE

```
NEUTRAL/DRIVE SWITCH STATE CHANGE ------------| NDDTIM = 0
                                              |
                                              | --- ELSE ---
                                              |
                                              | Increment NDDTIM
```

OSCTMR - OSCMOD DELAY TIMER (0.125 SEC)

This timer is used to prevent OSCMOD spark feedback during manual transmission shifts.

```
TPDLBR >= SHFDLT -----------------------------| Increment OSCTMR
                                              |
                                              | --- ELSE ---
                                              |
                                              | OSCTMR = 0
```

PUTMR - TIME SINCE CPU POWER-UP

```
CPU POWER ON ---------------------------------| Increment PUTMR
```

SETTMR - RPM CONTROL ENTRY DELAY TIMER (0.125 seconds)

This timer is used to delay entry into RPM control for a time which corresponds to the manifold stabilization time. SETTMR is cleared in Idle Speed Control MODE_SELECT.

```
In mode select -------------------------|
                                        |
APT >= 0 -------------------------|     |AND -| SETTMR = 0
 (not Closed Throttle)            |OR --|     |
                                  |           | --- ELSE ---
APT = -1 -------------------|     |           |
                            |AND -|           | Increment SETTMR
DASPOT > DASCTL ------------|
 (daspot too large)
```

SHFTMR - SHIFT IN PROGRESS TIMER

```
always ---------------------------------------| Decrement SHFTMR
 (free running)
```

NOTE: SHFTMR is loaded with SHFTTM when any 3-4 or 4-3 shift is made. Thus when SHFTMR = 0, no shift is in progress.

TARTMR - TIME SINCE OLDTP WAS UPDATED

```
always ---------------------------------------| Increment TARTMR
                                              | Clip at MAXTIM
```

NOTE: TARTMR is reset within the software TAR logic.

TSLEGO - TIME SINCE LAST EGO SWITCH

```
OPEN LOOP FUEL CONTROL -----------------|
                                        |AND -| TSLEGO = 0
LESFLG = 0 -----------------------------|     | Freeze TSLEGO
 (ego switching)                              |
                                              | --- ELSE ---
                                              |
EGO SWITCH -----------------------------------| LESFLG = 0
                                              |  (EGO switching)
                                              | Increment TSLEGO
                                              |
                                              | --- ELSE ---
                                              |
TSLEGO > LESTM -------------------------------| LESFLG = 1
                                              |  (EGO not switching)
                                              | Increment TSLEGO
                                              |
                                              | --- ELSE ---
                                              |
                                              | Increment TSLEGO
```

NOTE: TSLEGO is also set to zero within the closed loop fuel logic after the jumpback is calculated following an EGO switch.

TSLPIP - TIME SINCE LAST PIP

```
NEW_PIP = 1 ----------------------------------| TSLPIP = 0
                                              | NEW_PIP = 0
                                              |
                                              | --- ELSE ---
                                              |
                                              | Count up TSLPIP
```

NOTE: NEW_PIP is set equal to 1 upon a PIP interrupt.

V_NACTMR_CUM - SELF TEST CUMULATIVE NOT A CLOSED THROTTLE TIMER

```
APT = 0 --------------------------------|
                                        |OR --| Increment V_NACTMR_CUM
APT = 1 --------------------------------|     |
                                              | --- ELSE ---
                                              |
                                              | Freeze V_NACTMR_CUM
```

VOLTMR - LOW VOLTAGE TIMER

Low voltage timer (VOLTMR) represents the amount of time that battery voltage, as indicated by the VBAT calculation, is lower than a calibrated threshold voltage LOWVOL. VOLTMR is referenced by FN821A to determine the increase in engine speed necessary to provide battery charge compensation (see Desired RPM calculation in the Idle Speed Control Chapter).

```
CRKFLG = 0 -----------------------------|
 (run/underspeed mode)                  |
                                        |
VBAT < LOWVOL --------------------|S Q -|AND -| Count up VOLTMR
                                  |     |     | Clip VOLTMR to VOLTCLP
VBAT >= LOWVOL + VOLHYS ----------|C    |     |
                                        |     |
APT = -1 -------------------------------|     |
                                              | --- ELSE ---
NACTMR >= CHGTM ------------------------|     |
                                        |AND -| Count down VOLTMR
VBAT >= LOWVOL + VOLHYS ----------------|     |
                                              | --- ELSE ---
                                              |
                                              | Freeze VOLTMR
```

WOTTMR - TIME AT WIDE OPEN THROTTLE

```
APT = 1 --------------------------------------| Increment WOTTMR
                                              |
                                              | --- ELSE ---
                                              |
                                              | Decrement WOTTMR
```

CHAPTER 22

FAILURE MODE MANAGEMENT

FAILURE MODE STRATEGY

The Failure Mode (FMEM) strategy protects vehicle function from adverse effects of an EEC component failure. The strategy recognizes open or short circuit failure for five sensors: MAP, TP, ECT, ACT and EGR(EVP/PFE). In general, if the continuous Self Test strategy recognizes a failure the FMEM strategy will execute an alternative vehicle strategy. The alternative strategy disables logic which relies on realistic sensor values. Some sensor FMEM strategies also substitute a "safe" value for the bad sensor. A summary of the alternate strategies is tabulated below.

Alternate Strategy	Sensors				
	MAP	TP	ECT	ACT	EGR
Transmission - Do not Lock Up	X	X	X	X	
Inferred BP - No PT/WOT Update	X	X	X	X	
Adaptive Fuel - No Update	X	X	X	X	
Idle Speed - Fixed Duty Cycle	X	X	X	X	
EGR - Disabled	X	X	X	X	X
THERMACTOR - Bypass Air	X	X	X	X	
MPG MODE - Do Not Enter	X		X		
DECEL FUEL S/O - Disable	*	*			

NOTE: DFSO is disabled only if both the MAP and TP sensors have failed.

FAILURE RECOGNITION

OVERVIEW

The FMEM strategy checks the "Continuous Self Test Code" Filters to ascertain whether a sensor has failed. If the sensor failure lasts long enough to trigger a Self Test Code, the FMEM strategy will substitute an alternate value and strategy. Until the Self Test filters exceed their fault thresholds, the strategy continues to use the last known valid value. The logic diagram below APPROXIMATELY describes the Fault recognition and value substitution strategy. However, to more effectively use the Self Test Fault filters, the logic is divided into two sections; the Fault Flag logic and the Sensor input process logic. (See Specific Sensor FMEMs.)

DEFINITIONS

INPUTS

Bit Flags:

- CRKFLG = Flag indicating Crank mode.

Calibration Contants:

- FILHYS = Hysteresis term to permit normal TP update, if sensor is functioning properly.

PROCESS

STRATEGY MODULE: INPUT_FAIL_REC_COM1

```
SENSOR >= SENSORMIN -----------|      | SENSOR WITHIN
 (sensor OK)                   |      |  ACCEPTABLE RANGE
C***FIL < C***LVL - FILHYS ----|      |  UPDATE SENSOR INPUT
 (sensor high fault            |AND -|
  filter OK)                   |      | Failure Flags = 0
SENSOR <= SENSORMAX -----------|      |
 (sensor OK)                   |      |
C***FIL < C***LVL - FILHYS ----|      | --- ELSE ---
 (sensor low fault filter OK)         |
                                      | SENSOR OUTSIDE
CRANK MODE (CRKFLG = 1) -------------|  RANGE
 (Optional)                           | SENSOR = INITIAL VALUE
                                      |
                                      | --- ELSE ---
                                      |
C***FIL > C***LVL -------------|      | SENSOR OUTSIDE
 (sensor high fault)           |OR --|  RANGE - NOT DUE TO
C***FIL > C***LVL -------------|      |  LOW BATTERY VOLTAGE
 (sensor low fault)                   | Failure Flags = 1
                                      |
                                      |  ALTERNATE STRATEGY
                                      | SENSOR = SUBSTITUTED
                                      |   VALUE
                                      |
                                      | --- ELSE ---
                                      |
                                      | SENSOR DATA NOT
                                      |  RELIABLE - DO NOT
                                      |  UPDATE UNTIL CHECK
                                      |  PROVES VALUE VALID.
```

NOTE: MAP Failure Mode Recognition is described in the MAP calculation (See INPUT CONVERSIONS AND FILTERS Chapter).

Should this strategy be implemented in the Continuous Self Test Strategy, the following specifications must be met:

- FMEM Code must be executed following the A/D conversion.
- FMEM Code must be executed, once per background loop, during ALL Engine Modes (CRANK, UNDERSPEED, RUN, KEYON).
- Continuous Self Test Check provides two functions: It does Normal Fault Filtering for ALL sensors checked by the continuous Self Test; and, it also executes the FMEM_FLAG logic.

ACT SENSOR UPDATE

OVERVIEW

DEFINITIONS

INPUTS

Registers:

- C112FIL = Self Test Register which counts the number of ACT high failures.
- C113FIL = Self Test Register which counts the number of ACT low failures.
- ECT = Engine Coolant Temperature, deg F.
- IACT = A/D conversion of ACT sensor input, counts.
- IECT = A/D conversion of ECT sensor, counts.

Bit Flags:

- AFMFLG = Flag indicating that ACT sensor has failed.
- CFMFLG = Flag indicating that ECT sensor is in/out of range.
- WRMEGO = Ego sensor should be warm flag; 1 -> Ego warm, 0 -> Ego is

Calibration Constants:

- ACTFMM = FMEM default value for ACT.
- ACTMAX = Maximum ACT (ACT Open), Counts.
- ACTMIN = Minimum ACT (ACT Shorted), Counts.
- C112LVL = Threshold for ACT short fault, unitless.
- C113LVL = Threshold for ACT Open fault, unitless.
- FILHYS = Hysteresis term to prevent spurious exit of Failure Mode strategy.
- FN703 = ECT/ACT transfer function.

OUTPUTS

Registers:

- ACT = Air Charge Temperature, deg F.

Bit Flags:

- AFMFLG = Flag indicating that ACT sensor has failed.

PROCESS

STRATEGY MODULE: INPUT_ACT_COM2

This module is performed during engineering units conversion.

```
AFMFLG = 0  --------------------|
                                |
IACT <= ACTMAX -----------------|AND -| ACT = FN703(IACT)
                                |     |  (sensor OK)
IACT >= ACTMIN -----------------|     |
                                      | --- ELSE ---
WRMEGO = 0 ---------------------|     |
 (start up Open Loop)           |     |
                                |     |
CFMFLG = 0 ---------------------|     |
                                |AND -| ACT = FN703(IECT)
IECT <= ECTMAX -----------------|     |  (sensor bad, use ECT if OK)
                                |     |
IECT >= ECTMIN -----------------|     |
                                      | --- ELSE ---
                                      |
AFMFLG = 1 ---------------------------| ACT = ACTFMM
                                      |
                                      | --- ELSE ---
                                      |
                                      | Do not update ACT
```

AFMFLG LOGIC (FOR ACT SENSOR)

CONTINUOUS SELF TEST

```
C112FIL > C112LVL --------------------| afmlo = 1
                                      |
                                      | --- ELSE ---
                                      |
C112FIL < C112LVL - FILHYS -----------| afmlo = 0

C113FIL > C113LVL --------------------| afmhi = 1
                                      |
                                      | --- ELSE ---
                                      |
C113FIL < C113LVL - FILHYS -----------| afmhi = 0

afmhi --------------------------|
                                |OR --| AFMFLG = 1
afmlo --------------------------|     |
                                      | --- ELSE ---
                                      |
                                      | AFMFLG = 0
```

ADAPTIVE FUEL TABLE FMEM

OVERVIEW

This module sets and clears two adaptive fuel failure mode flags; one flag for each adaptive table in a two EGO control system.

DEFINITIONS

INPUTS

Registers:

- C179FIL = At adaptive fuel limit, system lean fault filter.
- C181FIL = At adaptive fuel limit, system rich fault filter.
- C182FIL = At adaptive fuel limit, system lean at idle fault filter.
- C183FIL = At adaptive fuel limit, system rich at idle fault filter.
- C188FIL = At adaptive fuel limit (EGO #2), system lean fault filter.
- C189FIL = At adaptive fuel limit (EGO #2), system rich fault filter.
- C191FIL = At adaptive fuel limit (EGO #2), system lean at idle fault filter.
- C192FIL = At adaptive fuel limit (EGO #2), system rich at idle fault filter.

Bit Flags:

- NUMEGO = Number of EGO sensors.

Calibration Constants:

- C179LVL = At adaptive fuel limit, system lean fault filter threshold.
- C181LVL = At adaptive fuel limit, system rich fault filter threshold.
- C182LVL = At adaptive fuel limit, system lean at idle fault filter threshold.
- C183LVL = At adaptive fuel limit, system rich at idle fault filter threshold.
- C188LVL = At adaptive fuel limit (EGO #2), system lean fault filter threshold.
- C189LVL = At adaptive fuel limit (EGO #2), system rich fault filter threshold.
- C191LVL = At adaptive fuel limit (EGO #2), system lean at idle fault filter threshold.

- C192LVL = At adaptive fuel limit (EGO #2), system rich at idle fault filter threshold.
- FILHYS = Hysteresis term to prevent spurious exit of Failure Mode strategy.

OUTPUTS

Bit Flags:

- ADT1FMFLG = Adaptive table 1 failure mode.
- ADT2FMFLG = Adaptive table 2 failure mode.

PROCESS

STRATEGY MODULE: INPUT_ADAPT_COM1

```
C179FIL > C179LVL ----------------|
 (adapt lean clip, ego 1)         |
                                  |
C181FIL > C181LVL ----------------|
 (adapt rich clip, ego 1)         |
                                  |OR --| ADT1FMFLG = 1
C182FIL > C182LVL ----------------|     |
 (adapt lean clip at idle, ego 1) |     |
                                  |     |
C183FIL > C183LVL ----------------|     |
 (adapt rich clip at idle, ego 1)       |
                                        | --- ELSE ---
C179FIL < C179LVL - FILHYS -------|     |
                                  |     |
C181FIL < C181LVL - FILHYS -------|     |
                                  |AND -| ADT1FMFLG = 0
C182FIL < C182LVL - FILHYS -------|
                                  |
C183FIL < C183LVL - FILHYS -------|
```

For Stereo EGO systems:

```
NUMEGO = 2 -----------------------|
                                  |
C188FIL > C188LVL ----------|     |
 (adapt lean clip, ego 2)   |     |
                            |     |AND -| ADT2FMFLG = 1
C189FIL > C189LVL ----------|     |     |
 (adapt rich clip, ego 2)   |     |     |
                            |OR --|     |
C191FIL > C191LVL ----------|           |
 (adapt lean clip at idle,  |           |
  ego 2)                    |           |
                            |           |
C192FIL > C192LVL ----------|           |
 (adapt rich clip at idle,              |
  ego 2)                                |
                                        | --- ELSE ---
NUMEGO = 2 -----------------------|     |
                                  |     |
C188FIL < C188LVL - FILHYS -|     |     |
                            |     |AND -| ADT2FMFLG = 0
C189FIL < C189LVL - FILHYS -|     |
                            |     |
C191FIL < C191LVL - FILHYS -|AND -|
                            |
C192FIL < C192LVL - FILHYS -|
```

ECT SENSOR UPDATE

OVERVIEW

DEFINITIONS

INPUTS

Registers:

- ACT = Air Charge Temperature, deg F.
- C117FIL = Self Test Register which counts the number of ECT high failures.
- C118FIL = Self Test Register which counts the number of ECT low failures.
- ECT = Engine Coolant Temperature, deg F.
- ECTCNT = ECT counter used in TCSTRT average.
- FMECTR = Background loop counter used to ramp failed ECT from ACT in crank mode to ECTFMM.

Bit Flags:

- CFMFLG = Flag indicating that ECT sensor is in/out of range.
- CRKFLG = Crank flag.
- IECT = A/D conversion of ECT sensor, counts.

Calibration Constants:

- C117LVL = Threshold for ECT Short Fault, unitless.
- C118LVL = Threshold for ECT Open fault, unitless.
- ECTFMM = FMEM default value for ECT, deg F.
- ECTMAX = Maximum engine ECT, Counts.
- ECTMIN = Minimum engine ECT, Counts.
- FILHYS = Hysteresis term to prevent spurious exit of Failure Mode strategy.
- FMECNT = Number of background loops between incrementing/decrementing ECT register by 2 deg.F. A good estimate of The ECT ramp is 2 deg.F/6 seconds, therefore, FMECNT = 6 sec./.030 sec. per background loop = 200.
- FN703 = ECT/ACT transfer function.

- TCECT = Time constant for ECT, sec.

OUTPUTS

Registers:

- ECT = Engine Coolant Temperature, deg F.
- FMECTR = Background loop counter used to ramp failed ECT from ACT in crank mode to ECTFMM.

Bit Flags:

- CFMFLG = Flag indicating that ECT sensor is in/out of range.

PROCESS

STRATEGY MODULE: INPUT_ECT_COM2

This module is performed during engineering units conversion.

```
CFMFLG = 0 ---------------------|
                                |
IECT <= ECTMAX -----------------|
                                |AND -| ECT = FN703(IECT)
IECT >= ECTMIN -----------------|     |  (sensor OK)
                                |     |
ECTCNT = 0 ---------------------|     |
                                      | --- ELSE ---
                                      |
ECTCNT = 0 ---------------------------| ECT = ACT
                                      |  (sensor bad, start rolling
                                      |   average at ACT)
                                      |
                                      | --- ELSE ---
CFMFLG = 0 ---------------------|     |
                                |     |
IECT <= ECTMAX -----------------|AND -| ECT = ROLAV(FN703(IECT),TCECT)
                                |     |  (sensor OK)
IECT >= ECTMIN -----------------|     |
                                      | --- ELSE ---
                                      |
CRKFLG = 1 ---------------------------| ECT = ROLAV(ACT),TCECT)
 (crank mode)                         |  (sensor bad, infer ECT from ACT)
                                      |
                                      | --- ELSE ---
                                      |
CFMFLG = 1 ---------------------------| Increment FMECTR
 (every background loop)              |  (clip at 255 as a max)
                                      | Perform ECT RAMP LOGIC
                                      |
                                      | --- ELSE ---
                                      |
                                      | Do not update ECT
```

ECT RAMP LOGIC

```
ECT > ECTFMM --------------|
                           |AND -| ECT = ECT - 2 deg. F
FMECTR >= FMECNT ----------|     | FMECTR = 0
                                 |  (ramp down to ECTFMM)
                                 |
                                 | --- ELSE ---
ECT < ECTFMM --------------|     |
                           |AND -| ECT = ECT + 2 deg. F
FMECTR >= FMECNT ----------|     | FMECTR = 0
                                 |  (ramp up to ECTFMM)
                                 |
                                 | --- ELSE ---
                                 |
ECT = ECTFMM --------------------| Do not update ECT
                                 |  (ECT is already ECTFMM)
```

CFMFLG LOGIC (FOR ECT SENSOR)

CONTINUOUS SELF TEST

```
C117FIL > C117LVL ---------------| cfmlo = 1
                                 |
                                 | --- ELSE ---
                                 |
C117FIL < C117LVL - FILHYS ------| cfmlo = 0


C118FIL > C118LVL ---------------| cfmhi = 1
                                 |
                                 | --- ELSE ---
                                 |
C118FIL < C118LVL - FILHYS ------| cfmhi = 0


cfmhi = 1 -----------------|
                           |OR --| CFMFLG = 1
cfmlo = 1 -----------------|     |
                                 | --- ELSE ---
                                 |
                                 | CFMFLG = 0
```

EVP SENSOR FMEM (PFEHP = 0)

OVERVIEW

The EVP sensor failure Mode strategy will force the EGR valve to close and will have no adverse impact on Spark or Fuel.

DEFINITIONS

INPUTS

Registers:

- C327FIL = Self Test Register which counts the number of EVP low failures.
- C337FIL = Self Test Register which counts the number of EVP high failures.
- EOFF = The EGR valve reading when the valve is fully closed in A/D counts.
- IEGR = A/D conversion of EVP or EPT sensor, counts.

Bit Flags:

- EFMFLG = Flag indicating that EVP EGR sensor has failed. (This flag performs for both Sonic and PFE EGR.)

Calibration Constants:

- C327LVL = Threshold for EVP fault, unitless.
- C337LVL = Threshold for (PFE) EVP fault, unitless.
- EVPMAX = Maximum EGR Valve position, counts.
- EVPMIN = Minimum EGR Valve position, counts.
- FILHYS = Hysteresis term to prevent spurious exit of Failure Mode strategy.

OUTPUTS

Registers:

- EVP = EGR valve position reading in A/D counts.

Bit Flags:

- EFMFLG = Flag indicating that EVP EGR sensor has failed. (This flag performs for both Sonic and PFE EGR.)

PROCESS

STRATEGY MODULE: INPUT_EVP_COM2

This module is performed during engineering units conversion.

```
EFMFLG = 0 ---------------|
                          |
IEGR <= EVPMAX -----------|AND -| EVP = IEGR
                          |     |
IEGR >= EVPMIN -----------|     | --- ELSE ---
                                |
EFMFLG = 1 ---------------------| EVP = EOFF
                                |
                                | --- ELSE ---
                                |
                                | Do NOT update EVP
```

EFMFLG LOGIC (FOR EVP SENSOR)

CONTINUOUS SELF TEST

```
C337FIL > C337LVL --------------| efmhi = 1
                                |
                                | --- ELSE ---
                                |
C337FIL < C337LVL - FILHYS -----| efmhi = 0

C327FIL > C327LVL --------------| efmlo = 1
                                |
                                | --- ELSE ---
                                |
C327FIL < C327LVL - FILHYS -----| efmlo = 0

efmhi = 1 ----------------|
                          |OR --| EFMFLG = 1
efmlo = 1 ----------------|     |
                                | --- ELSE ---
                                |
                                | EFMFLG = 0
```

TOT SENSOR FMEM

DEFINTIONS

INPUTS

Registers:

_ ECT = Engine Coolant Temperature, deg F.

_ ECTCNT = ECT counter used in TCSTRT average.

_ ITOT = Transmission Oil temperature, counts.

Bit Flags:

Calibration Constants:

_ FN703D(ITOT) = TOT transfer function; purpose: convert ITOT A/D counts into deg. F, counts.

_ TCTOT = Time constant for filtered TOT, sec.

_ TOTMAX = Maximum allowable TOT counts, counts.

_ TOTMIN = Minimum allowable TOT counts, counts.

OUTPUTS

Registers:

_ TOT = Transmission Oil Temperature, deg F.

PROCESS

STRATEGY MODULE: INPUT_TOT_COM1

This module is performed during engineering units conversion.

```
ITOT <= TOTMAX -----------|
                          |
ITOT >= TOTMIN -----------|AND -| TOT = FN703D(ITOT)
                          |     |  (sensor OK, start rolling
ECTCNT = 0 ---------------|     |   average at actual value)
                                |
                                | --- ELSE ---
ITOT <= TOTMAX -----------|     |
                          |AND -| TOT = ROLAV(FN703D(ITOT),TCTOT)
ITOT >= TOTMIN -----------|     |  (sensor OK)
                                |
                                | --- ELSE ---
                                |
                                | TOT = ECT
                                |  (sensor bad, use ECT)
```

TP SENSOR FMEM

OVERVIEW

The FMEM strategy checks the Continuous Self Test Code Filters to ascertain whether the TP sensor failed. If the sensor failure lasts long enough to trigger a Self Test Code, the FMEM strategy will infer throttle position based upon a load parameter (usually MAP). [NOTE: The load parameters has protective logic in the event of a load sensor failure. See the MAP calculation in the Systems Equation Chapter.

DEFINITIONS

INPUTS

Registers:

- C122FIL = Throttle Position(TP) sensor circuit below minimum voltage fault filter.
- C123FIL = Throttle Position(TP) sensor circuit above maximum voltage fault filter.
- MAP = Manifold absolute pressure.
- RATCH = Closed throttle position, counts.

Bit Flags:

- CRKFLG = Crank Flag.
- ITP = Throttle position value from A/D conversion, counts.
- TFMFLG = Flag indicating that TP sensor has failed.

Calibration Constants:

- C122LVL = Throttle Position(TP) sensor circuit below minimum voltage threshold.
- C123LVL = Throttle Position(TP) sensor circuit above maximum voltage threshold.
- FILHYS = Hysteresis term to prevent spurious exit of Failure Mode strategy.
- FN090 = Change in TP as a function of MAP. This function is designed to permit Closed and Part Throttle operation.
- TAPMAX = Maximum valid TP value, counts. (Calibrated by Self Test Design Section)
- TAPMIN = Minimum valid TP value, counts. (Calibrated by Self Test Design Section)

OUTPUTS

Registers:

- RATCH = Closed throttle position, counts.
- TP = Throttle position, counts.

Bit Flags:

- TFMFLG = Flag indicating that TP sensor has failed.

PROCESS

STRATEGY MODULE: INPUT_TP_COM3

This module is performed during engineering units conversion.

```
TFMFLG = 0 ------------------|
                             |
ITP >= TAPMIN ---------------|AND -| TP = ITP
                             |     |  (TP sensor within
ITP <= TAPMAX ---------------|     |   acceptable range)
                                   |
                                   | --- ELSE ---
                                   |
CRKFLG = 1 ------------------------| TP = RATCH
 (crank mode)                      |  (TP sensor out of limits)
                                   |
                                   | --- ELSE ---
                                   |
TFMFLG = 1 ------------------------| TP = RATCH + FN090(MAP)
                                   | RATCH = RACHIV
                                   |  (TP sensor out of limits
                                   |   but NOT due to Low
                                   |   battery voltage)
                                   |
                                   | --- ELSE ---
                                   |
                                   | No change to TP
                                   |  (TP sensor data unreliable
                                   |   DO NOT update until
                                   |   confident data valid)
```

TFMFLG LOGIC (FOR TP SENSOR)

CONTINUOUS SELF TEST CHECK

```
C122FIL > C122LVL -----------------| tfmlo = 1
                                   |
                                   | --- ELSE ---
                                   |
C122FIL < C122LVL - FILHYS --------| tfmlo = 0


C123FIL > C123LVL -----------------| tfmhi = 1
                                   |
                                   | --- ELSE ---
                                   |
C123FIL < C123LVL - FILHYS --------| tfmhi = 0


tfmhi = 1 -------------------|
                             |OR --| TFMFLG = 1
tfmlo = 1 -------------------|     |
                                   | --- ELSE ---
                                   |
                                   | TFMFLG = 0
```

CHAPTER 23

INFERRED BAROMETRIC PRESSURE STRATEGY

INFERRED BAROMETRIC PRESSURE STRATEGY

BACKGROUND:

Earlier EEC systems used two pressure sensors (manifold absolute and barometric absolute pressure) to provide full altitude capability. Since both sensors perform the same function (i.e. measure pressure), elimination of one sensor, via time sharing or software inference was considered a significant system cost reduction. Since MAP is a primary input for both spark and fuel control and BAP is a secondary modifier, elimination of BAP sensor resulted in cost reduction.

In the Inferred Barometric Pressure strategy, the BAP sensor is replaced by a software algorithm which uses available inputs (i.e. MAP, RPM, Throttle Position, ECT) to infer the Barometric Pressure. The EEC-IV Barometric Pressure (BP) is saved in Keep Alive Memory (KAM) to bridge the power-down to power-up sequence.

DEFINITIONS

- ATMR1 = Time since engine start-up, sec.
- BPKAM = Barometric pressure stored in KAM.
- BPKFLG = KAM flag indication state of BPKYON. If it is equal to zero, normal KEY-ON is assumed.
- BPKYON = Calculated BP while KEY ON.
- BPPTWT = Barometric pressure calculated during part throttle or WOT.
- BPUFLG = Flag which indicates that BP update is or is not permitted.
- CFMFLG = Flag indicating state of ECT sensor.
- CRKFLG = State of engine mode; 0 -> Run/Underspeed.
- ECT = Engine Coolant Temperature, deg F.
- KAM = Keep Alive Memory.
- KAMOK = Flag indicating whether KAM error exists or not.
- MAP = Manifold absolute pressure.
- MAPBAR = Calculated as rolling average filter of MAP.
- MFMFLG = Flag indicating state of MAP sensor.
- PIP = Profile ignition pickup (rpm input).
- PTPFLG = Flag indicating engine is running.

- PUTMR = Power up timer.
- RATCH = Lowest throttle position since start.
- TFMFLG = Flag indicating state of TP sensor.
- TP = Throttle position.
- TP_REL = Relative Throttle Position, TP - RATCH.

CALIBRATION CONSTANTS:

- ECTBP = Temperature at which inferred BP is enabled.
- FKBP = BPPTWT filter constant.
- FN046A = Normalizing function for AM/BP as Y-input to FN1033.
- FN047A = Normalizing function for relative TP input to FN1033.
- FN069A = The minimum relative throttle angle (TP_REL) for Inferred BP update. This function prevents BP updates when the airflow becomes sonic at a particular throttle area (constant airflow for constant throttle angle and increasing rpm).
- FN1033 = The pressure drop/BP table as a function of relative throttle position (TP_REL) and air mass flow (AM/BP). The normalizing function for AM/BP is FN046A. The normalizing function for TP_REL is FN047A.
- SSMAP = Steady state MAP.
- TKYON1 = Time at which PIP sensing is enabled. (NOT calibratable)
- TKYON2 = Time at which BPKYON update begins. (NOT calibratable)
- TKYON3 = Maximum time at which BPKYON update may begin. (NOT calibratable)
- TKYON4 = Locks out additional KEYON inferred BP updates. (NOT calibratable)

SUMMARY:

Barometric Pressure is inferred by two methods:

1. KEY ON - ENGINE OFF condition (BPKYON).
2. PART THROTTLE/WIDE OPEN THROTTLE condition (BPPTWT).

The BP calculation uses Keep Alive Memory to maintain accuracy. During Key on engine off condition, MAP is read and stored in KAM to be used as BP after start. During Part Throttle and Wide Open Throttle conditions, BP is calculated using MAP and the pressure drop across the air inlet.

The inferred Barometric Pressure Strategy is described in the following pages.

The Barometric Pressure from each method is saved separately in Keep Alive Memory (KAM). When Barometric Pressure is updated its value is also saved in the Keep Alive Memory Register called BPKAM. This register will always contain the most recent update of Barometric Pressure.

```
BPKYON updated ----------------------------------| BPKAM = BPKYON
                                                 |
                                                 | --- ELSE ---
                                                 |
BPPTWT updated ----------------------------------| BPKAM = BPPTWT
                                                 |
                                                 | --- ELSE ---
                                                 |
                                                 | BPKAM not updated
```

BPKYON - KEY ON BAROMETRIC PRESSURE UPDATE PROCEDURE:

During key-on engine-off, the Manifold Absolute Pressure is filtered and saved as Barometric Pressure in the Keep Alive Memory. The Key On BP Value BPKYON is updated as follows:

Consider the time period immediately after power up.

```
           TKYON1          TKYON2                            TKYON3
|   50msec   |    60msec     |            400msec              |
0------------+---------------+------------------+--------------+->TIME
|            |               |                  |              |
POWER       PIP            START               END            MAX
 UP       SENSING          BPKYON            BPKYON          BPKYON
          ENABLED          UPDATE            UPDATE          UPDATE
                                             IF PIP           TIME
                                             OCCURS
```

1) During the 50 millisecond period after power up:

PIP signals are ignored by the BP strategy. Power up noise transients can create false PIP signals.

The MAPBAR filter is continuously initialized at the current IMAP value.

The KAM flag BPKFLG is checked. This flag bridges the reset (loss of computer control) that can occur with starter engagement. The flag is set when the BPKYON value is actually updated. If BPKFLG is clear, a normal key-on power up is assumed. The BPKYON update will be permitted (later on). If BPKFLG is set, a starter engagement is assumed (engine cranking). The BPKYON update will not be permitted.

2) During the 50 to 110 millisecond period after power up:

MAPBAR is calculated as a rolling average filter of MAP.

PIP sensing is enabled. This time period allows the BP strategy 1) to detect engine cranking after a reset due to starter engagement or 2) to detect engine running after a spurious reset during normal operation. If the engine is turning with a PIP period less than 60 millisecond, a PIP signal should occur during this time period. (The equivalent RPM values are 500 for 4-cylinder, 333 for 6-cylinder, and 250 for 8-cylinder.)

3) During the 110 to 510 millisecond period after power up:

MAPBAR is calculated as a rolling average filter of MAP.

PIP sensing is enabled.

If a PIP has not been sensed yet and if the BPKYON update is permitted from step 1), MAPBAR is saved as barometric pressure in KAM (as both BPKAM and BPKYON). The BPKYON update continues until a PIP occurs or the time limit is reached.

4) After 10 seconds of normal run mode operation, the BPKFLG is cleared to get ready for the next power down/power up sequence.

KEY-ON BAROMETRIC PRESSURE UPDATE LOGIC

```
PUTMR > TKYON1 -------------------|
                                  |AND -| PTPFLG = 1
PIP OCCURRED AFTER TKYON1 --------|


KAMOK = 0 ------------------------|
(KAM DATA BAD)                    |OR --| Permit KEY-ON BP update
                                  |     | BPUFLG = 1
BPKFLG = 0 -----------------------|


LAST_MAP2 <> LAST_MAP ------------|
                                  |
BPUFLG = 1 -----------------------|
                                  |
PUTMR > TKYON2 -------------------|
                                  |AND --| Update KEY-ON BP
PUTMR < TKYON3 -------------------|      | BPKFLG = 1
                                  |
PTPFLG = 0 -----------------------|
(NO PIP'S YET)


BPKFLG = 1 -----------------------|
                                  |AND -| BPKAM = MAPBAR
MAPBAR > 16" Hg ------------------|     | BPKYON = MAPBAR
                                        |
                                        | --- ELSE ---
                                        |
BPKFLG = 1 -----------------------------| BPKAM = 29.875
                                        | BPKYON = 29.875


CRKFLG = 0, UNDSP = 0 ------------|
(RUN MODE)                        |AND -| Enable KEY-ON BP update
                                  |     |  for next START-UP
ATMR1 > TKYON4 -------------------|     | Clear BPKFLG = 0
```

NOTE:

1) BPKAM, BPKYON are restricted from 16" Hg to 31.875" Hg.

2) BPKAM is saved first in KAM , followed by BPKYON. Since a reset can occur between these saves, BPKAM (used by strategy) will have the highest chance to be current and correct.

BPPTWT - PART THROTTLE/WIDE OPEN THROTTLE BAROMETRIC PRESSURE UPDATE :

During Part Throttle and Wide Open Throttle conditions, Barometric Pressure is calculated as Manifold Absolute Pressure plus the pressure drop through the throttle body. This is saved as BPPTWT and BPKAM in the Keep Alive Memory.

```
MFMFLG = 0 -------------------|
 (MAP sensor OK)              |
                              |
CRKFLG = 0, UNDSP = 0 --------|
 (RUN mode)                   |
                              |
CFMFLG = 0 -------------------|
 (ECT OK)                     |
                              |
ECT > ECTBP ------------------|
                              |
TP_REL > FN069A(AM/BP) -------|AND -| Calculate BPPTWT as
                              |     |  a rolling average
MAP >= MAPBAR ----------------|     |  filter of Manifold
                              |     |  Absolute Pressure
MAP - MAPBAR <= SSMAP --------|     |  plus the Pressure Drop
                              |     |  across the throttle.
ATMR1 > TKYON4 ---------------|     | BPPTWT = UROLAV[MAP +
                              |     |   FN1033(TP_REL,AM/BP)*BP,TCBP]
TFMFLG = 0 -------------------|     | BPKAM = BPPTWT
 (TP sensor OK)               |
                              |
AFMFLG = 0 -------------------|
 (ACT sensor OK)
```

FN1033 is the pressure drop/BP table as a function of relative throttle position (TP_REL) and air mass flow (AM/BP), clipped to 0.996 as a minimum. The normalizing function for AM/BP is FN046A. The normalizing function for TP_REL is FN047A.

FN069A defines the minimum relative throttle angle [(TP_REL) clipped at 0 as a minimum] for Inferred BP update. This function prevents BP updates when the airflow becomes sonic at a particular throttle area (constant airflow for constant throttle angle and increasing rpm).

Note: BPKAM, BPPTWT are restricted from 16" Hg to 31.875" Hg. TKYON4 is also used in the key-on BP logic. It is a fixed 10 seconds.

BAROMETRIC PRESSURE PIP COUNTER CONTROL LOGIC :
==

```
PUTMR > TKYON1 ---------------------------| Increment PTPCNT
                                          |  as PIPs occur
                                          |
                                          | --- ELSE ---
                                          |
                                          | PTPCNT = 0
```

CHAPTER 24

KEEP ALIVE MEMORY

KEEP ALIVE MEMORY (KAM) QUALIFICATION TEST

OVERVIEW

Each time the vehicle is started, the data stored in KAM may not be valid. Power interruptions, noise, etc., may have altered KAM contents. Or, the computer may not be reading KAM registers correctly because of a hardware fault. When the KAM registers are initialized, a special binary pattern is written into three bytes of KAM. The KAM register names are KAMQA, KAMQB, and KAMQC. During each background loop, the KAM registers are tested. The KAM qualification test judges the validity of the KAM data by looking for the proper binary pattern. The alternate courses of action are either:

1) If the proper pattern is present, the KAM data is considered OK for use by the strategy.

2) If not present, the KAM data is suspect. The KAM is over-written to a set of initial values. The initial values are also used in place of the KAM data when the strategy references KAM.

The KAM registers KAMQA, KAMQB, and KAMQC are assigned to different areas of the KAM. This will help protect for partial KAM failures. The assignments are:

KAM Register	KAM Address
KAMQA	LOWEST ADDRESS OF KAM
KAMQB	MIDDLE ADDRESS OF KAM
KAMQC	HIGHEST ADDRESS OF KAM

The KAM qualification test is normally performed each background loop when the computer is running.

KAM QUALIFICATION TEST LOGIC
(Performed each background loop)

```
KAMQA = 10101010 BINARY----------------|
                                       |
KAMQB = 11000110 BINARY----------------|AND -| KAM_ERROR = 0
                                       |     | Assume KAM DATA is VALID
KAMQC = 01110101 BINARY----------------|     | ALL Strategy references to
                                       |     |  KAM will use KAM DATA.
RT_NOVS_KAM >= RTNVMN -----------------|     | BP = BPKAM
                                       |     |
RT_NOVS_KAM <= RTNVMX -----------------|     | --- ELSE ---
                                       |     | KAM_ERROR = 1
BPKAM >= 16 ---------------------------|     | VIP_KAM = 1
                                       |     | Assume KAM DATA is BAD
BPKYON >= 16 --------------------------|     | Initialize all KAM
                                       |     |  locations used in the
BPPTWT >= 16 --------------------------|     |  strategy
                                             | Clear all VIP CODES
                                             | Write the special
                                             |  BINARY PATTERNS to KAM:
                                             |   KAMQA = 10101010 BINARY
                                             |   KAMQB = 11000110 BINARY
                                             |   KAMQC = 01110101 BINARY
                                             | LTMTBLrc = 0.5
                                             | CHKSUM = 13568
                                             | ISCKAMn = 0.0 (n=0-5)
                                             | ISKSUM = 0.0
                                             | BPKAM = 29.875
                                             | BPKYON = 29.875
                                             | BPPTWT = 29.875
                                             | BP = 29.875
                                             | BPKFLG = 0
                                             | KWUCTR = 0
                                             | NOVCTR = 0
                                             | RT_NOV_KAM = 1.0
                                             | FLG_FRST_NOV = 0
                                             | FLG_NOV_KAM = 0
                                             | LESFLG = 0
```

ISCKAM VALIDATION PROCEDURE
(Performed during power up sequence)

```
|ISCKAM0 + ISCKAM1 +    |          | ASSUME THE ISCKAMs ARE VALID
 ISCKAM2 + ISCKAM3 +    |----------| ISKSUM = ISCKAM0 + ISCKAM1 +
 ISCKAM4 + ISCKAM5 -    |          |          ISCKAM2 + ISCKAM3
 ISKSUM| <= 1 BIT ------|          |          ISCKAM4 + ISCKAM5
                                   |
                                   | --- ELSE ---
                                   |
                                   | ASSUME ISCKAMs DATA ARE INVALID
                                   | RE-INITIALIZE THE ISCKAM
                                   | ISCKAM0 = 0
                                   | ISCKAM1 = 0
                                   | ISCKAM2 = 0
                                   | ISCKAM3 = 0
                                   | ISCKAM4 = 0
                                   | ISCKAM5 = 0
                                   | ISKSUM = 0
```

VIP THROTTLE MODE SET
(Done each background loop in running VIP)

```
V_MODE_SETUP = 1 ------------------| Re-initialize the ISCKAM
 (VIP Throttle Adjust Mode)        |  ISCKAM0 = 0
                                   |  ISCKAM1 = 0
                                   |  ISCKAM2 = 0
                                   |  ISCKAM3 = 0
                                   |  ISCKAM4 = 0
                                   |  ISCKAM5 = 0
                                   |  ISKSUM = 0
```

ADAPTIVE FUEL TABLE VALIDATION PROCEDURE

Each time the vehicle is started, the data stored in KAM may or may not be valid. Power interruptions, noise, etc., may have altered the KAM contents. Or, the computer may not be reading KAM registers correctly because of a hardware fault. The KAM qualification test judges the validity of the KAM data. Based on the result, KAM can be initialized as required. See the KAM section for more details on the KAM qualification test.

Based on the results of the KAM qualification test, validate the adaptive fuel table as follows;

```
|(SUM OF ALL KAM CELLS)------|      | ASSUME THE ADAPTIVE FUEL
- CHKSUM| <= 1 BIT           |AND -|  DATA IN KAM IS VALID.
                             |      | CHKSUM = SUM OF ADAPTIVE FUEL CELLS
CHKFLG = 0 ------------------|      |
                                    | --- ELSE ---
                                    |
                                    | ASSUME THE ADAPTIVE FUEL
                                    |  DATA IN KAM IS WRONG.
                                    | DO A TOTAL INITIALIZATION
                                    |  OF THE ADAPTIVE FUEL
                                    |  DATA IN KAM.
                                    | FOR EACH CELL:
                                    |  1) SET LTMTBLrc = 0.5
                                    |  2) SET CHKSUM = 13568
                                    | KWUCTR = 0
```

CHKSUM is a KAM memory word containing the sum of the LTMTBL contents. CHKSUM is incremented or decremented each time any LTMTBL cell is updated. A one count difference between the present sum and the stored sum is allowed to account for the case of power down after a KAM update but prior to CHKSUM update.

CHAPTER 25

EEC-IV SELF TEST

EEC-IV SELF TEST OVERVIEW

Self Test is divided into two types of testing, one which occurs only at the "request" of the service technician (the "on-demand" tests), and one which continuously surveys the system during normal operating modes (the "continuous" tests). The on-demand portion is further divided into "engine-running" and "engine-off" tests.

The engine-off portion of the test "looks" for normal engine-off sensor readings. Any out-of-limits, open, or shorted sensor input is signalled by sending a service code. If all sensors are within expected ranges, a "111" code is issued. Codes are repeated to make it easier for the technician to verify the code sequence. After the service codes, a single pulse occurs to signal the technician that the next set of codes will be from the continuous test. Continuous test codes are issued using the same format as the service codes, and are also repeated. Finally, the test enters the "output state test", which simply turns actuator outputs "on" and "off" based on "requests" from the technician (these consist of depressing the throttle and letting it return to closed position). STO is also turned "on" and "off" in this mode, so that the technician knows the state in which the other outputs should be.

The engine-running portion signals that it has begun by sending an "identification" code (=no. of cylinders/2). It then tests inputs and EEC-IV-controlled functions by forcing various conditions and "looking" for expected engine response to them. A single output pulse is sent to signal the test operator to "goose" the throttle, during which inputs are tested for dynamic response. If no RPM change is detected, a special code (code 538) will be sent to indicate that the test was incorrectly performed. When the "goose" test has completed, service codes are sent.

The "continuous" self test monitors inputs during normal operation, and stores information in keep-alive memory (KAM) when errors are detected. In general, checks are made only for open-or short-circuits. When the number of errors in a given time period exceeds a calibratable threshold, that code is stored in KAM. As a special diagnostic aid, in engine-off conditions and STI is latched or when STI=GND and the on-demand (running) test has completed, codes will be stored every time an error is detected, and STO will be turned on as long as the fault is present. This is designed to help isolate intermittent faults (eg.: the test operator can "wiggle" the harness and connectors, and STO will indicate when the intermittent fault recurs). Codes which indicate faults that have not recurred in 80 engine warm-up cycles are "erased". Codes can also be manually "erased" by opening up STI while codes are being output in the engine-off mode.

EEC-IV SELF TEST BLOCK DIAGRAMS

PROCESS

STRATEGY MODULE: VO_BLOCKDIAG_COM1

```
         ENGINE-OFF:                      ENGINE-RUNNING:


       !-------------!                   !--------------!
       ! ENGINE-OFF  !                   !  I.D. CODE   !
       !    TEST     !                   !              !
       !-------------!                   !--------------!
              |                                  |
              v                                  v
       !-------------!                   !--------------!
       !   SERVICE   !                   !    ENGINE    !
       !    CODE     !                   !   RUNNING    !
       !-------------!                   !     TEST     !
              |                          !--------------!
              v                                  v
       !-------------!                   !--------------!
       !   SINGLE    !                   !    SINGLE    !
       !   PULSE     !                   !    PULSE     !
       !-------------!                   !--------------!
              |                                  |
              v                                  v
       !-------------!                   !--------------!
       ! CONTINUOUS  !     REQUIRES      !   DYNAMIC    !
       !    TEST     !     MECHANIC----->!   RESPONSE   !
       !   CODES     !   INTERVENTION    !     TEST     !
       !-------------!         |         !--------------!
              |                |                 |
       !-------------!         |         !--------------!
       !   OUTPUT    !<--------|         !    SERVICE   !
       !   PULSING   !                   !     CODES    !
       !-------------!                   !--------------!
                                                 |
                                                 v
NOTE:  TEST CODES CAN BE                 !--------------!
       MANUALLY ERASED BY                ! 2 MIN. DELAY !
       UN-GROUNDING STI DURING           !(TIMING CHECK)!
       CODE OUTPUT (ENGINE-OFF).         !--------------!
                                                 |
                                                 v
                                         !--------------!
                           REQUIRES----->!   "WIGGLE"   !
                           MECHANIC      !     TEST     !
                         INTERVENTION    !--------------!
```

"CONTINUOUS" TESTS (STI=OPEN)

```
   ENGINE-OFF:                         ENGINE-RUNNING:


 !-------------!                     !---------------!
 !   "WIGGLE"  !                     !  CONTINUOUS   !
 !     TEST    !                     !  TEST MODE    !
 !-------------!                     !---------------!
```

CHAPTER 26

SELF TEST ENTRY/EXIT LOGIC

SELF TEST ENTRY/EXIT LOGIC

OVERVIEW

This logic checks for entry and exit conditions for the various self test modes described below:

Engine Off Test
Engine Off Wiggle Mode
Engine Running Test
Engine Running Wiggle Mode
Continuous Self Test

Engine Running Test is disabled if an automatic transmission is put into drive or if vehicle speed is above MINMPH. This is done for safety reasons.

If STI is ungrounded during the output of codes in Engine Off Test, all continuous self test codes are erased from KAM. This allows for repair verification by the mechanic.

DEFINITIONS

INPUTS

Registers:

- IEGR = EGR sensor input.
- PUTMR = Time after CPU power up.
- TSSTIL_TMR = Time since STI low timer.
- VSBAR = Filtered vehicle speed.

Bit Flags:

- CRANKING = Engine Cranking Flag.
- DISABLE_NOSTART = NST VIP disable flag.
- DISABLE_RUNNING = RVIP disable flag.
- NDSFLG = Neutral/Drive flag; 1 -> Drive.
- NO_START = NST VIP enable flag.
- RUNNING = RVIP Enable Flag.
- STIFLG = Self Test Input (demultiplexed); 1 -> STI grounded.

- STI_RESET = 1 -> Operator requested Throttle Plate.
- STO_TRIGGER = Trigger indicates STO output requested.
- STO_WORKING = Self Test Output in use.
- UNDSP = Underspeed Flag.
- VIP_ENABLE = VIP enable flag.

Calibration Constants:

- TRLOAD = Transmission load switch: See Base Strategy
- VSTYPE = Integrated vehicle speed/cruise control system present switch; 0 -> no MPH and no VSC, 1 -> MPH and no VSC.
- VTCEPT = EPT Time constant.

OUTPUTS

Registers:

- FIEPT = VIP EPT filter.

Bit Flags:

- DISABLE_NOSTART = See above.
- DISABLE_RUNNING = See above.
- NO_START = See above.
- RUNNING = See above.
- VFS_OUT_FLG = 1 -> VFS output requested.
- VIP_ENABLE = See above.
- VIP_FP_OVERRIDE = 1 -> regular strategy controls fuel pump.
- WIGFLG = Indicates Vip wiggle test.

PROCESS

STRATEGY MODULE: VO_I_EXEC_COM2

For application with EPC VFS SOLENOID, SONIC EGR

```
always ---------------------------------| VIP_FP_OVERRIDE = 0
                                        |  (make sure base strategy can
                                        |   control pump)
                                        | Call LOADPOINT


always ---------------------------------| WIGFLG = 0


PUTMR >= 4 SEC -------------------------| VIP_ENABLE = 1


NO_START = 1 ---------------------|
 (the KOEO mode)                  |
                                  |
STIFLG = 1 -----------------------|AND -|  (continue in Engine Off Test
 (STI grounded)                   |     |   continue executing NO START VIP
                                  |     |   background sequence, and
CRANKING = 1 ---------------------|     |   outputting codes)
 (engine stopped)                       | VFS_OUT_FLG = 1
                                        |  (output TV pressure)
                                        |
                                        | --- ELSE ---
NO_START = 1 ---------------------|     |
                                  |     |
STIFLG = 1 -----------|           |     |
                      |AND -|     |     |
CRANKING = 0 ---------|     |     |AND -|  (exit Engine Off Test)
                            |OR --|     | Call RAM_INIT
STIFLG = 0 -----------|     |           | NO_START = 0
                      |     |           | DISABLE_NOSTART = 1
STO_WORKING = 0 ------|AND -|           | VIP_ENABLE = 0
                      |                 |  (turn STO off)
STO_TRIGGER = 0 ------|                 |  (return to Background)
 (not outputting codes)                 |
                                        | --- ELSE ---
NO_START = 1 ---------------------|     |
                                  |     |
STIFLG = 0 -----------------------|AND -| Call VIP_CODE_ERASE
                                  |     |  (erase all continuous codes
STO_WORKING = 1 ------------|     |     |   from KAM)
                            |OR --|     |  (exit Engine Off Test)
STO_TRIGGER = 1 ------------|           | Call RAM_INIT
 (outputting codes)                     | NO_START = 0
                                        | DISABLE_NOSTART = 1
                                        | VIP_ENABLE = 0
                                        |  (turn STO off,
                                        |   return to Background)
                                        |
                                        | --- ELSE ---
```

(continued on next page)

(continued from previous page)

```
RUNNING = 1 ----------------------|     |
 (in KOER mode)                   |     |
                                  |     |
STI_RESET = 1 --------------|     |     |
                            |OR --|     |
STI_RESET = 0 --------|     |     |     |
                      |AND -|     |     |
STIFLG = 1 -----------|           |     |
                                  |     |
TRLOAD < 3 -----------------|     |     |
                            |OR --|     |
TRLOAD >= 3 ----------|     |     |AND -| FIEPT = ROLAV(IEGR,VTCEPT)
 (auto trans)         |     |     |     |  (continue in Engine Running
                      |AND -|     |     |   Test - continue executing
NDSFLG = 0 -----------|           |     |   Engine Running background
 (in neutral)                     |     |   sequence and outputting codes)
                                  |     |
VSTYPE <> 0 ----------|           |     |
 (VSS present)        |AND -|     |     |
                      |     |     |     |
VSBAR <= MINMPH ------|     |     |     |
                            |OR --|     |
VSTYPE = 0 -----------------|           |
 (no VSS)                               |
                                        | --- ELSE ---
                                        |
RUNNING = 1 ----------------------------|  (vehicle is moving, in gear,
                                        |   or exit requested)
                                        |  (exit Engine Running Test)
                                        | Call RAM_INIT
                                        | RUNNING = 0
                                        | DISABLE_RUNNING = 1
                                        | VIP_ENABLE = 0
                                        | DISABLE_NOSTART = 1
                                        |
                                        | --- ELSE ---
CRANKING = 1 ---------------------|     |
                                  |     |
STIFLG = 1 -----------------------|AND -|  (enter Engine Off Test)
                                  |     | NO_START = 1
DISABLE_NOSTART = 0 --------------|     |
                                        | --- ELSE ---
CRANKING = 1 ---------------------|     |
                                  |     |
STIFLG = 1 -----------------------|AND -|  (enter Engine Off Wiggle Mode)
                                  |     | WIGFLG = 1
DISABLE_NOSTART = 1 --------------|     |
                                        | --- ELSE ---
```

(continued on next page)

(continued from previous page)

```
CRANKING = 0 ---------------------|      |
                                  |      |
UNDSP = 0 ------------------------|      |
 (run mode)                       |      |
                                  |      |
STIFLG = 1 -----------------------|      |
 (STI grounded)                   |      |
                                  |      |
VIP_ENABLE = 1 -------------------|      |
                                  |      |
PUTMR >= 6 SEC -------------------|      |
 (time since powerup)             |      |
                                  |      |
TSSTIL_TMR >= 1 SEC --------------|AND -|  (enter Engine Running Test)
 (time since STI grounded)        |      | RUNNING = 1
                                  |      |
DISABLE_RUNNING = 0 --------------|      |
                                  |      |
TRLOAD < 3 -----------------|     |      |
                            |OR --|      |
TRLOAD => 3 ----------|     |     |      |
 (auto trans)         |AND -|     |      |
                      |           |      |
NDSFLG = 0 -----------|           |      |
 (in neutral)                     |      |
                                  |      |
VSTYPE = 0 -----------------|     |      |
 (no VSS)                   |     |      |
                            |OR --|      |
VSTYPE <> 0 ----------|     |            |
                      |AND -|            |
VSBAR <= MINMPH ------|                  |
                                         | --- ELSE ---
CRANKING = 0 ---------------------|      |
                                  |      |
UNDSP = 0 ------------------------|      |
 (run mode)                       |      |
                                  |      |
STIFLG = 1 -----------------------|      |
 (STI grounded)                   |AND -|  (Engine Running Test
                                  |      |   should not be entered
VIP_ENABLE = 1 -------------------|      |   because vehicle is
                                  |      |   moving or is in gear)
PUTMR >= 6 SEC -------------------|      | DISABLE_RUNNING = 1
 (time since powerup)             |      |
                                  |      |
TSSTIL_TMR >= 1 SEC --------------|      |
 (time since STI grounded)        |      |
                                  |      |
DISABLE_RUNNING = 0 --------------|      |
                                         | --- ELSE ---
```

(continued on next page)

(continued from previous page)

```
CRANKING = 0 ---------------------|      |
                                  |      |
UNDSP = 0 ------------------------|      |
                                  |      |
STIFLG = 1 -----------------------|      |
                                  |      |
VIP_ENABLE = 1 -------------------|AND -|  (enter Engine Running
                                  |      |   Wiggle Mode)
PUTMR >= 6 SEC -------------------|      | WIGFLG = 1
                                  |      |
TSSTIL_TMR >= 1 SEC --------------|      |
                                  |      |
DISABLE_RUNNING = 1 --------------|      |
                                         | --- ELSE ---
                                         |
VIP_ENABLE = 1 --------------------------| Enter Continuous Test
```

NOTE:

- The flag NO_START is set immediately after the two 2msec pulses are output on STO. The two 2msec pulses indicate that the module test and RAM test were successfully completed.

- A RAM_INIT sets CRKFLG=1 which causes WIGFLG to toggle during engine run-up if STI was grounded prior to cranking.

CHAPTER 27

ENGINE OFF SELF TEST

ENGINE OFF SELF TEST SEQUENCE

PROCESS

STRATEGY MODULE: VO_EOTS_COM1

```
        -----------
      (    ENTER    )
        -----------
             |
             v
       !-----------!
       !   TURN    !
       !  STO OFF  !
       !-----------!
             |
       /-----------\
      / DO MODULE   \  FAIL
     / INSTRUCTION   \--------------------|
     \     TEST      /                    |
      \             /                     |
       \-----------/                      |
             |PASS                        v
        /--------\     FAIL        !--------------!
       /   TEST   \---------------->!  TURN STO ON !
       \ KAM/RAM  /                ! CONTINUOUSLY !
        \--------/                 !   AND EXIT   !
             |PASS                 !--------------!
        /---------\    FAIL        !--------------!
       / TEST ROM  \--------------->!     SET      !
       \           /               ! SERVICE CODE !
        \---------/                !--------------!
             |PASS                        |
|----------->|                            |
|            v                            |
|    NO/----------\                       |
|-----/ PUTMR =>    \ SEE NOTE 1          |
      \ V_LANSDTM ?/                      |
       \----------/                       |
             |YES                         |
|----------->|                            |
|            v                            |
|    NO/----------\                       |
|-----/TSSTIL_TMR  \ SEE NOTE 1          |
      \V_HICKOKTM ?/                      |
       \----------/                       |
             | YES                        |
             v                            |
            (B)                          (A)
```

```
            (B)                           (A)
             |                             |
             v                             |
   !--------------------------!           |
   !SEND TWO 2MS PULSES ON STO!           |
   !2MS BETWEEN PULSES        !           |
   !--------------------------!           |
             |                             |
|----------->|<---------------------------|
|            v
|       /---------\
|  NO  /  PUTMR >  \ SEE NOTE 1
|------\   4 SEC   /
        \---------/
             |
|----------->|YES
|            v
|       /---------\
|      /FUEL PUMP  \ SEE NOTE 1
|  NO / TURNED OFF  \
|-----\ (TIMED OUT) /
       \           /
        \---------/
             |YES
             v
        /-----------\               !-----------------!
       /    TEST     \       FAIL   !       SET       !
      /   A/D'S AND   \------------>! SERVICE CODE(S) !
      \   SWITCHES    /             !-----------------!
       \             /                        |
        \-----------/PASS                     |
              |<------------------------------|
              v
             (C)
```

NOTE:

- Execute normal background until conditions are met.

- Calibration Recommendation: Since TSSTIL_TMR update may lag PUTMR by one background, V_LANSDTM should be calibrated to .275 sec. and V_HICKOKTM should be
calibrated to .180 sec to insure delay integrity.

```
                           (C)
                            |
                            v
                       /---------\    FAIL        !-----------------!
  FAIL /---------\    / TEST OCC  \-------------->!       SET       !
   |<-/FUEL PUMP  \<-|\           /               ! SERVICE CODE(S) !
   |  \CKT TEST** /  | \---------/                !-----------------!
   |   \---------/   |      |     PASS                      |
   |       |         |------|<------------------------------|
   |       |
   |       |PASS!------------------------!
   |       |--->!   . SEND SERVICE CODES !
   v         ^  !------------------------!
!--------!   |              |
!  SET   !   |              v
!SERVICE !---|  !--------------------------!
!CODE(S) !      !   . SEND SEPARATOR PULSE !
!--------!      !--------------------------!
                            |
                            v
                       /---------\    YES        !---------------!
                      /VIP_KAM =1 \------------->! SEND CODE 512*!
                      \     ?     /              !---------------!
                       \---------/                        |
                            | NO                          |
                            v                             |
                    !-------------------!                 |
                    ! . SEND CODES FROM !                 |
                    !   CONTINUOUS TEST !                 |
                    !-------------------!                 |
                            |<----------------------------|
                            v
                !----------------------------!
                !   . ENTER OUTPUT TEST MODE !
                !----------------------------!
```

*See Normal Strategy KAM qualification test logic for setting of VIP_KAM=1.
Code 512 is output during continuous code output.

** NOTE: Fuel pump check test must be performed after OCC test.

KAM/RAM TEST

PROCESS

STRATEGY MODULE: VO_KAMRAM_COM1

```
                 -----------
               (    ENTER    )
                 -----------
                      |
                      v
     !-----------------------------------!
     ! SET POINTER TO BEGINNING OF       !
     ! EXT_RAM                           !
     !-----------------------------------!
|--------------------->|
|                     v
|        !-------------------------!
|        ! SAVE REGISTER CONTENTS  !
|        !-------------------------!
|                     |
|        !-------------------------!
|        !   WRITE HEX AAAA TO     !
|        !   REGISTER              !
|        !-------------------------!
|                     |
|        !-------------------------!
|        ! READ REGISTER CONTENTS  !
|        !-------------------------!
|                     |
|              /-------------\
|             / IS CONTENTS   \  NO
|             \  = AAAA       /-------------|
|              \-------------/              |
|                     | YES                  |
|                     v                      |
|        !------------------------!         |
|        !   WRITE HEX 5555 TO    !         |
|        !     REGISTER           !         |
|        !------------------------!         |
|                     |                      |
|        !------------------------!         |
|        ! READ REGISTER CONTENTS !         |
|        !------------------------!         |
|                     |                      |
|                /---------\                |
|               /    IS     \ NO            |
|              /  CONTENTS   \------------->|
|              \   = 5555    /              |
|               \         /                v
|                \---------/           !-----------!
|                     | YES            !TURN STO ON!
|                     v                !   EXIT    !
|                    (A)               !-----------!
```

```
                          (A)
                           |
     |                     v
     |             !-------------------!
     |             ! RESTORE REGISTER  !
     |             !     CONTENTS      !
     |             !-------------------!
!------------!             |
!INCREMENT   !             |
! POINTER    !             |
!------------!             |
     ^                     |
     |       NO     /-------------\
     |<-----------/  END OF 2K     \
                  \  EXT-RAM ?     /
                   \-------------/
                           | YES
                           |
                           v
                    --------------
                 (  GO TO ROM_TEST  )
                    --------------
```

READ-ONLY MEMORY TEST

PROCESS

STRATEGY MODULE: VO_ROM_COM1

```
                 -----------
                (   ENTER    )
                 -----------
                      |
                      v
      !-----------------------------------!
      ! DO A 16-BIT ADDITION OF CONTENTS !
      ! OF ALL ROM LOCATIONS RETAINING   !
      ! THE 16 LEAST SIGNIFICANT BITS    !
      ! OF THE RESULT.                   !
      !-----------------------------------!
                      |
                      v
                 /---------\    NO    !---------!
                /  SUM=0 ?  \-------->!   SET   !
                \           /         ! CODE 511!
                 \---------/          !---------!
                      |   YES              |
                      |<-------------------|
                      v
                 -----------
                (    EXIT    )
                 -----------
```

NOTE: A specific location will contain checksum such that sum of correct ROM contents (including checksum)=0. Location is labeled "Rom_To" or "Rom_Total".

ENGINE COOLANT TEMPERATURE SENSOR

OVERVIEW

The ECT sensor test checks the ECT sensor and associated circuitry in four specific areas.

- Low voltage output such as short circuits. -Service code 117.
- High voltage output. i.e. open circuits, disconnects. -Service code 118.
- Lower ECT range fault. -Service code 116.
- Upper ECT range fault. -Service code 116.

The analog signal from the ECT sensor undergoes A/D conversion and IECT is compared to four calibration parameters; ECTMIN, ECTMAX, VIECT1, and VIECT2.

Cross checks are made first to direct service to a hard fault. -i.e. short circuits or simple disconnects. Shorts are tested first determined by ECTMIN, then open circuits determined by ECTMAX.

VIECT1 and VIECT2 define low and high range values and are subsequently compared to IECT to evaluate normal sensor readings.

Specific vehicle preparation must be performed to establish a standard against which the range is checked.

The calibration values are established by PEDD/EED and should not be changed.

DEFINITIONS

INPUTS

Registers:

- ECT
- IECT

Calibration Constants:

- ECTMIN; Minimum Engine off ECT -COUNTS
- ECTMAX; Maximum Engine off ECT -COUNTS
- VIECT1; Minimum Engine off ECT (RANGE) -COUNTS
- VIECT2; Maximum Engine off ECT (RANGE) -COUNTS

PROCESS

STRATEGY MODULE: VO_ECT_COM1

```
  -----------
(    ENTER    )
  -----------
       |
       v
  /---------\    NO      !---------!
 /  IECT >=  \---------->! CODE 117!-->|
 \  ECTMIN ? /           !---------!   |
  \---------/                          |
       |   YES                         |
       |                               |
       v                               |
  /---------\    NO      !---------!   |
 /  IECT <=  \---------->! CODE 118!-->|
 \ ECTMAX ?  /           !---------!   |
  \---------/                          |
       |   YES                         |
       v                               |
  /---------\                          |
 /   VIECT1  \   NO      !---------!   |
/  >=IECT >=  \--------->! CODE 116!-->|
\    VIECT2 ? /          !---------!   |
 \           /                         |
  \---------/                          |
       |   YES                         |
       |<------------------------------|
       v
  -----------
(     EXIT    )
  -----------
```

MANIFOLD ABSOLUTE PRESSURE SENSOR TEST

OVERVIEW

The MAP sensor test has been designed to test three conditions:

- MAP signal presence.
- Low MAP signal (BP lower range fault)
- High MAP signal (BP upper range fault)

To verify if a MAP signal is present, MAPTMR, a 1/8 second timer which counts up time since last scap edge, is compared to parameter VMPMAX. If MAPTMR exceeds VMPMAX it is ascertained that no MAP signal is present and service code 126 is output.

MAP (in this case BP) range is checked by utilizing the MAP sensors' digital frequency output. This output is translated into computer ticks and compared to parameters VMDEL1 and VMDEL2. If the signal doesn't meet or exceeds these calibratable values, service code 126 is output.

Notice that all three failure conditions will produce the same service code.

Calibration of VMPMAX is dependant on the microprocessor background loop time and the time elapsed to detect an open sensor.

VMDEL1 AND VMDEL2 must take into account running Self-Test at altitude (low BP), sea level (high BP) and tolerance stack-ups.

DEFINITIONS

INPUTS

Registers:

- MAP
- MAPTMR
- MDELTA

Calibration Constants:

- VMPMAX; Max. time since last MAP update. -MSEC
- VMDEL1; Min. MAP during engine-off Self-Test. -TICKS
- VMDEL2; MAX. MAP during engine-off Self-Test. -TICKS

PROCESS

STRATEGY MODULE: VO_MAP_COM2

```
  -----------
(    ENTER    )
  -----------
       |
       v
  /---------\    NO
 /  MAPTMR <=\-----------------|
 \  VMPMAX ? /                 |
  \---------/                  |
       |  YES                  |
       v                       |
  /---------\                  v
 /   VMDEL1  \   NO      !---------!
/<= MDELTA <= \--------->! CODE 126!
\    VMDEL2   /          !---------!
 \     ?     /                 |
  \---------/                  |
       |   YES                 |
       |<----------------------|
       v
  -----------
(     EXIT    )
  -----------
```

THROTTLE POSITION SENSOR

PROCESS

STRATEGY MODULE: VO_TP_COM1

```
      -----------
    (    ENTER    )
      -----------
           |
           v
      /---------\   NO       !---------!
     /  ITP >=   \---------->! CODE 122!-->|
     \ TAPMIN ?  /           !---------!   |
      \---------/                          |
           |   YES                         |
           |                               |
           v                               |
      /---------\     NO     !---------!   |
     /   ITP <=  \---------->! CODE 123!-->|
     \ TAPMAX ?  /           !---------!   |
      \---------/                          |
           |   YES                         |
           v                               |
      /---------\                          |
     /   VTAP1   \   NO      !---------!   |
    /  <=ITP <=   \--------->! CODE 121!-->|
    \    VTAP2 ?  /          !---------!   |
     \           /                         |
      \---------/                          |
           |  YES                          |
           |<------------------------------|
           v
      -----------
    (     EXIT    )
      -----------
```

ACT SENSOR TEST

PROCESS

STRATEGY MODULE: VO_ACT_COM1

```
     -----------
   (    ENTER    )
     -----------
          |
          v
     /---------\    NO      !---------!
    /  IACT >=  \---------->! CODE 112!---|
    \  ACTMIN ? /           !---------!   |
     \---------/                          |
          |   YES                         |
          |                               |
          v                               |
     /---------\    NO      !---------!   |
    /  IACT <=  \---------->! CODE 113!---|
    \ ACTMAX ?  /           !---------!   |
     \---------/                          |
          |   YES                         |
          v                               |
     /---------\                          |
    /   VIACT1  \   NO      !---------!   |
   /  =>IACT =>  \--------->! CODE 114!-->|
   \    VIACT2 ? /          !---------!   |
    \           /                         |
     \---------/                          |
          |   YES                         |
          |<------------------------------|
          v
     -----------
   (     EXIT    )
     -----------
```

TRANSMISSION OIL TEMPERATURE SENSOR TEST

OVERVIEW

The TOT (Transmission Oil Temp) sensor test checks the TOT sensor and associated circuitry in four specific areas.

- Low voltage output such as short circuits. -Service code 638

- High voltage output. i.e. open circuits, disconnects. -Service code 637.

- Lower TOT range fault. -Service code 636.

- Upper TOT range fault. -Service code 636.

The analog signal from the TOT sensor undergoes A/D conversion and ITOT is compared to four calibration parameters; TOTMIN, TOTMAX, VITOT1, and VITOT2.

Gross checks are made first to direct service to a hard fault. -i.e. short circuits or simple disconnects. Shorts are tested first determined by TOTMIN, then open circuits determined by TOTMAX.

VITOT1 and VITOT2 define low and high range values and are subsequently compared to ITOT to evaluate normal sensor readings.

Specific vehicle preparation must be performed to establish a standard against which the range is checked. Refer to section entitled; Initiating Self-Test.

The calibration values are established by ESD/EED and should not be changed.

DEFINITIONS

INPUTS

Registers:

- TOT
- ITOT

Calibration Constants:

- TOTMIN; Maximum Engine off TOT -COUNTS
- TOTMAX; Minimum Engine off TOT -COUNTS
- VITOT1; Minimum Engine off TOT (RANGE) -COUNTS
- VITOT2; Maximum Engine off TOT (RANGE) -COUNTS

PROCESS

STRATEGY MODULE: VO_TOT_COM2

```
       -----------
     (    ENTER    )
       -----------
            |
            v
       /---------\    NO      !---------!
      /  ITOT>=   \---------->! CODE 638!-->|
      \  TOTMIN ? /           !---------!   |
       \---------/                          |
            |   YES                         |
            |                               |
            v                               |
       /---------\    NO      !---------!   |
      /  ITOT <=  \---------->! CODE 637!-->|
      \ TOTMAX ?  /           !---------!   |
       \---------/                          |
            |   YES                         |
            v                               |
       /---------\                          |
      /   VITOT1  \   NO      !---------!   |
     /  >=ITOT >=  \--------->! CODE 636!-->|
     \    VITOT2 ? /          !---------!   |
      \           /                         |
       \---------/                          |
            |   YES                         |
            |<------------------------------|
            v
       -----------
     (     EXIT    )
       -----------
```

EXHAUST GAS RE-CIRCULATION SENSOR TEST

PROCESS

STRATEGY MODULE: VO_EGR_COM5

```
  -----------
(    ENTER    )
  -----------
       |
       v
 /---------\  NO
/  PFEHP=0  \-------------------|
\     ?     /                   |
 \---------/                    |
       | YES                    |
       v                        |
 /---------\ YES !---------!    |
/  IEGR <=  \--->! CODE 327!--->|
\  EVPMIN   /    !---------!    |
 \---------/                    |
       |NO                      |
       v                        |
 /---------\ YES !---------!    |
/  IEGR >=  \--->! CODE 337!--->|
\  EVPMAX   /    !---------!    |
 \---------/                    |
       |NO                      |
       v                        |
 /---------\ YES !---------!    |
/  IEGR <   \--->! CODE 328!--->|
\  VEVPLL ? /    !---------!    |
 \---------/                    |
       |NO                      |
       v                        |
 /---------\ YES !---------!    |
/  IEGR >   \--->! CODE 334!--->|
\  VEVPHL ? /    !---------!    |
 \---------/                    |
       |NO                      |
       |------------------------|
       v
   -----------
(     EXIT    )
   -----------
```

A/C SWITCH TEST

OVERVIEW

This test determines whether the A/C is on or the A/C input is high when the A/C switch is in the off position.

DEFINITIONS

Bit Flags:

- A3C

PROCESS

STRATEGY MODULE: VO_ACCS_COM2

```
A3C = 1 --------------------------------| Set code 539 (A/C on indicated)
```

MANUAL LEVER POSITION SENSOR INPUT TEST (E40D)

OVERVIEW

The Manual Lever Position Sensor (MLPS) Input Test is to be performed with the transmission selector lever in the PARK position (when V_SW_PARK = 1). The voltage input, INDS (in counts), is checked to verify that all 6 resistances in series (Park, Reverse, Neutral, Overdrive, Manual 2, and Manual 1) are within the range VND1 to VND2. The values for the range are determined by the tolerances of the 6 MLPS resistance to VREF and a 560 OHM resistance to VREF as the voltage divider. For vehicles without the PARK position, V_SW_PRK is set to zero and the test is run in NEUTRAL. In this case, INDS must be within the range VND3 to VND4.

The divider network consists of the input to the EEC (INDS), divided by the 560 OHM resistance to VREF and the 6 resistors in series (MLPS) to signal return.

The INDS register is in "counts". To convert voltage to counts:

INDS counts = MLPS voltage * (VREF/1023)

DEFINITIONS

Registers:

- INDS = Input Neutral/drive switch. -counts

Calibration Constants:

- VND1 = Lower limit for MLPS test. -counts
- VND2 = Upper limit for MLPS test. -counts
- VND3 = Lower limit for MLPS test in neutral. -counts
- VND4 = Upper limit for MLPS test in neutral. -counts
- V_SW_PRK = Calibration switch to select PARK (=1) or NEUTRAL (=0) test.

PROCESS

STRATEGY MODULE: VO_MLPS_INPUT_COM1

```
V_SW_PRK = 1 ---------------------|
 (test in park)                   |AND -| Set code 654
                                  |     | Not in park
INDS < VND1 ----------------|     |     |
                            |OR --|     |
INDS > VND2 ----------------|           |
                                        | --- ELSE ---
V_SW_PRK = 0 ---------------------|     |
 (test in neutral)                |AND -| Set code 655
                                  |     | Not in neutral
INDS < VND3 ----------------|     |     |
                            |OR --|     | --- ELSE ---
INDS > VND4 ----------------|           |
                                        | Exit this test
```

4X4L SWITCH INPUT TEST

PROCESS

STRATEGY MODULE: VO_4X4L_SWITCH_COM2

```
     -------
   (  ENTER  )
     -------
        |
        v
   /----------\ NO
  /  TSTRAT =4 \-------------------------|
  \            /                         |
   \----------/                          |
        | YES                            |
   /----------\                          |
  / I4X4L = 0  \  NO    !--------!       |
 /(NOT IN 4X4   \------>!CODE 633!---|   |
 \    LOW)      /       !--------!   |   |
  \            /                     |   |
   \----------/                      |   |
        |YES                         |   |
        |<---------------------------|   |
        |                                |
        |<-------------------------------|
        |
        v
     --------
   (  EXIT    )
     --------
```

POWER STEERING PRESSURE SWITCH TEST

PROCESS

STRATEGY MODULE: VO_PSPS_COM1

```
      ----------
    (   ENTER    )
      ----------
          |
          |
          v
     /---------\  NO
    / PSPSHP=1 ?\----->( GO TO NEXT TEST )
    \           /
     \---------/
          | YES
          v
    /-----------\
   /  POWSFG = 1 \ YES       !---------!
  /      ?        \-------->!CODE 519 !----|
  \ (SWITCH OPEN) /          !---------!    |
   \             /                           |
    \-----------/                            |
          | NO                               |
          |<---------------------------------|
          v
    -----------
  (   EXIT      )
    -----------
```

IVPWR INPUT TEST

PROCESS

STRATEGY MODULE: VO_IVPWR_COM1

```
  ----------
(   ENTER    )
  ----------
      |
      v
 /---------\
/ IIVPWR    \  YES     !---------!
\ < VKYPWR  /--------->!CODE 513 !
 \---------/           !---------!
      |  NO                 |
      |<--------------------|
      v
  ---------
(  EXIT     )
  ---------
```

NOTE: This test is designed to check continuity of the IVPWR circuit internal to the EEC module and can be used as a battery voltage check, if the parameter VKYPWR is calibrated to the minimum voltage to prevent occurrence of false codes.

$$\text{COUNTS} = \frac{\text{IIVPWR} * .1786 * 1023}{\text{VREF}}$$

$$\text{IIVPWR} = \frac{\text{VREF} * \text{COUNTS}}{.1786 * 1023}$$

NOTE: Recommended VKYPWR value: 400 counts (10.9 volts)

OUTPUT CIRCUIT CHECK

OVERVIEW

The OCC uses special module hardware to test certain output channels for open circuits/shorted drivers. The hardware consists of a resistor- divider network which is fed back into an A/D channel. The test begins by turning off all outputs in the network. Outputs are then turned on and off, one at a time, and the A/D channel is used to determine the change in voltage associated with each. A voltage change smaller than expected causes a fault code to be registered. The output channels, their associated fault codes, and expected voltage change calibration parameters for each appear below.

PROCESS

STRATEGY MODULE: VO_OCC_COM12

OC#	CIRCUIT FUNCT.	CAL. PARAMETER	ERROR CODE
1	AM2 1)	OCCDT1	553
2	AM1 1)	OCCDT2	552
4	EVR 2)	OCCDT4	558
5	CANP 3)	OCCDT5	565
7	FP	OCCDT7	556
10	SS-1 4)	OCCSS1	621
11	SS-2 4)	OCCSS2	622
12	CCS 4)	OCCCCS	626
13	CCC 4)	OCCCCC	629
14	TCIL 4)	OCCTCIL	631

NOTES:
1) ONLY IF THRMHP=1
2) ONLY IF PFEHP=0
3) ONLY IF CANPHP=1
4) ONLY IF TSTRAT= 4

AM1 - AIR MANAGEMENT 1 (BYPASS)
AM2 - AIR MANAGEMENT 2 (DIVERT)
EVR - ELECTRONIC VACUUM REG.
CANP - CANISTER PURGE
FP - FUEL PUMP
SS-1 - SHIFT SOLENOID #1
SS-2 - SHIFT SOLENOID #2
CCS - COAST CLUTCH SOLENOID
CCC - CONVERTER CLUTCH SOLENOID
TCIL - TRANSMISSION CONTROL INDICATOR LIGHT

OUTPUT CIRCUIT CHECK TEST STRUCTURE

PROCESS

STRATEGY MODULE: VO_OCCTS_COM4

```
                        -----------
                      (    ENTER    )
                        -----------
                             |
                !--------------------------!
                ! DE-ENERGIZE ALL OUTPUTS !
                ! . OC=1                   !
                !--------------------------!
                             |
                     !-----------------!
                     ! DELAY 50 MSEC. !
                     !-----------------!
     |----------------------->|
     |              !----------------------!
     |              ! SAVE IOCC IN OCCSAV !
     |              !----------------------!
     |                       |
     |             !------------------------!
     |             ! ENERGIZE OUTPUT #(OC)  !
     |             !------------------------!
     |                       |
     |               !-----------------!
     |               ! DELAY 50 MSEC. !
     |               !-----------------!
     |                       |
     |               /-------------\            !-------------!
     |              /  ABS[OCCSAV-  \      NO   !     SET     !
     |             /    IOCC]   >=   \--------->! APPROPRIATE !
     |             \ OCCDT(OC) COUNTS/          !  FAIL CODE  !
     |              \       ?       /           ! (SEE TABLE) !
     |               \-------------/   YES      !-------------!
     |                       |<--------------------------|
     |                       v
     |          !---------------------------!
     |          ! DE-ENERGIZE OUTPUT #(OC) !
     |          !---------------------------!
     |                       |
     |               !-----------------!
     |               ! DELAY 50 MSEC. !
!-----------!        !-----------------!
! OC= OC+1 !                 |
!-----------!           /---------\
     |           NO    /    IS     \
     |<---------------/ OC= LASTOC ?\
                      \             /
                       \           /
                        \---------/
                             |  YES
                             v
                        -----------
                      (    EXIT     )
                        -----------
```

OCC PARAMETER DEFINITIONS

NAME	DESCRIPTION	UNITS	RANGE MIN	RANGE MAX	BASE
OC	OUTPUT CIRCUIT #	-	0	9	-
OCCSAV	SAVED OCC A/D	COUNTS	0	1023	-
IOCC	OCC A/D	COUNTS	0	1023	-
OCCDTx(1-9)	MIN A/D CHANGE	COUNTS	-1023	1023	36

FUEL PUMP MONITOR TEST

OVERVIEW:

This test determines if the proper FPM input is received by the processor as the fuel pump is commanded on and off.

DEFINTIONS:

Self Test Calibration Constants:

- V_FPMDLY = Fuel Pump Monitor test fuel pump on-to-off/off-to-on stabilization delay time.

- V_FPMFLG = Fuel Pump Monitor test enable switch, 1 = enable.

Self Test Flags:

- FPM = State of the FPM input 1 = high, implying pump on.

- CODE_556 = Primary fuel pump circuit failure (from OCC test)

- CODE_543 = Fuel pump circuit open-battery to ECA

- CODE_542 = Fuel pump circuit open-ECA to motor ground

Self-Test Registers:

- VIP_CNT_EX Self test executive pointer

Base Strategy Flags:

- PUMP = Controls state of fuel pump control output 1 = commanded on.

PROCESS

STRATEGY MODULE: VO_FPM_COM2

```
VIP_CNT_EX = fpm_test_1 ------| DO: FPM TEST 1 PROCESS
                              | (If test is enabled, start
                              | pump, initialize delay
                              | timer.)
                              |
                              | ---ELSE---
                              |
VIP_CNT_EX = fpm_test_2 ------| DO: FPM TEST 2 PROCESS
                              | (Check for FPM = 1, if not,
                              | set code 543.  Stop pump,
                              | initialize delay timer.)
                              |
                              | ---ELSE---
                              |
VIP_CNT_EX = fpm_test_3 ------| DO: FPM TEST 3 PROCESS
                              | (Check for FPM = 0, if not,
                              | set code 542. End of test)
```

```
                    BEGIN: FPM TEST 1 PROCESS

VFPMFLG = 1 -------|        | (Start test)
(Test enabled)     |AND ----| VIP_TIMER_EX = 0
                   |        | (Initialize delay timer)
CODE_556 = 0 ------|        |
(No fuel pump OCC           | PUMP = 1
 failure)                   | (Command fuel pump on)
                            |
                            | VIP_CNT_EX = fpm_test_2
                            | (Set up for next process)
                            |
                            | ---ELSE---
                            |
                            | (Skip to next test)
                            | VIP_CNT_EX = fpm_test_3 + 1
                            | (Next test)

                     END: FPM TEST 1 PROCESS


                    BEGIN: FPM TEST 2 PROCESS

VIP_TIMER_EX > V_FPMDLY -|        | (pass code 543 test)
(time delay)             |AND ----| VIP_TIMER_EX = 0
                         |        | (re_init timer for next
FPM = 1 -----------------|        |  delay)
(Indicates pump on)               |
                                  | PUMP = 0
                                  | (command pump off)
                                  |
                                  | VIP_CNT_EX = fpm_test_3
                                  | (Set up for next process)
                                  |
                                  | ---ELSE---
                                  |
VIP_TIMER_EX > V_FPMDLY -|        | SET CODE_543 = 1
(Time delay)             |AND ----| (fail test)
                         |        |
FPM = 0 -----------------|        | VIP_TIMER_EX = 0
(Indicates pump off)              | (re-init timer for next
                                  |  delay)
                                  |
                                  | PUMP = 0
                                  | (command pump off)
                                  |
                                  | VIP_CNT_EX = fpm_test_3
                                  | (Set up for next process)

                     END: FPM TEST 2 PROCESS
```

BEGIN: FPM TEST 3 PROCESS

```
VIP_TIMER_EX > V_FPMDLY -|         | (pass code 542 test)
 (time delay)            |AND ----| VIP_CNT_EX = fpm_test_3+1
                         |         |  (next test)
FPM = 0 -----------------|         |
 (indicates pump off)              |
                                   | ---ELSE---
                                   |
VIP_TIMER_EX > V_FPMDLY -|         | SET CODE_542 = 1
 (time delay)            |         |  (fail test)
                         |AND ----|
FPM = 1 -----------------|         | VIP_CNT_EX = fpm_test_3+1
 (indicates pump on)               |  (next test)
```

END: FPM TEST 3 PROCESS

ELECTRONIC PRESSURE CONTROL SOLENOID TEST

PROCESS

STRATEGY MODULE: VO_EPC_SOLENOID_COM1

```
         ------
       (  ENTER  )
         ------
            |
            v
        /--------\ NO
       /TSTRAT =4 \---------------------------------------->|
       \          /                                         |
        \--------/                                          |
            |YES                                            |
        /--------\NO                                        |
       /V_ETV_TEST\---------------------------------------->|
       \  = 1 ?   /                                         |
        \--------/                                          |
            | YES                                           |
  !------------------------------!                          |
  !SET TV_COUNTS = TV1CNTS_TEST !                           |
  !------------------------------!                          |
            |                                               |
  !--------------------------!                              |
  !RUN TV VFS OUTPUT ROUTINE!                               |
  !--------------------------!                              |
            |NOTE: Refer to Transmission Strategy to        |
            |      perform this routine.                    |
            v                                               |
  !-----------------!                                       |
  !WAIT ETV_TEST_SEC!                                       |
  !-----------------!                                       |
            |                                               |
            |                                               |
  !-----------------!                                       |
  !SAVE ETVOCM ->   !                                       |
  !V_ETVOCM_SAV     !                                       |
  !-----------------!                                       |
            |                                               |
            v                                               |
  !----------------------------------------------------!    |
  !COMPUTE:                                            !    |
  !                          Clip to zero if negative !    |
  !ETV_OCM_MIN = (ETV_BIAS - (ETV_GAIN * TV_COUNTS))   !    |
  !               + ((VBAT - 10.0)*ETV_GAIN_BAT)       !    |
  !                          Clip to zero if negative !    |
  !----------------------------------------------------!    |
            |                                               |
            v                                               |
           (A)                                             (B)
```

```
        (A)                                            (B)
         |                                              |
         v                                              |
 /---------------\YES      !--------!                   |
/ ETVOCM   <      \------->!CODE 624!-------|           |
\ ETV_OCM_MIN ?   /        !--------!       |           |
 \---------------/                          |           |
         |NO                                |           |
         |                                  |           |
         v                                  |           |
  !-----------------------------!           |           |
  !SET TV_COUNTS =TV2CNTS_TEST  !           |           |
  !-----------------------------!           |           |
         |                                  |           |
         v                                  |           |
  !-----------------------------!           |           |
  !RUN TV VFS OUTPUT ROUTINE    !           |           |
  !-----------------------------!           |           |
         |                                  |           |
         v                                  |           |
  !------------------!                      |           |
  !WAIT ETV_TEST_SEC !                      |           |
  !------------------!                      |           |
         |                                  |           |
         v                                  |           |
  /--------------\                          |           |
 /ABS (V_ETVOCM_  \NO    !---------!        |           |
/SAV-ETVOCM) =>    \---->! CODE 625!        |           |
\     VETVDT ?     /     !---------!        |           |
 \                /           |             |           |
  \--------------/            |             |           |
         |  YES               v             |           |
         |<---------------------------------|           |
         v                                              |
  !-----------------!                                   |
  !SET TV_COUNTS =0 !                                   |
  !-----------------!                                   |
         |                                              |
         |<---------------------------------------------|
         v
      ------
    (  EXIT  )
      ------
```

OUTPUT TEST MODE

OVERVIEW

In this mode, outputs are turned on/off based on operator requests which consist of throttle position moving above an upper limit, then below a lower limit. A timeout is used when outputs are turned on to protect module circuity.

ONLY OUTPUTS IN THE FOLLOWING TABLE WILL BE FUNCTIONED:

1989 EFI-SD
STO
WAC
AM1
AM2
EVR
CCC
CANP
ISC
SS-2
CCS
OCIL
SS-1

PROCESS

STRATEGY MODULE: VO_OTM_LH_COM1

ON ENTRY TO THIS MODE, DE-ENERGIZE THE ABOVE OUTPUTS

```
ITP > VTAP5 ----------------------------| Set REQFLG


REQFLG SET -----------------------|
                                  |AND -| Reverse state of outputs
ITP <= VTAP6 ---------------------|     |  (OFF-->ON or ON-->OFF)
                                        | Clear REQFLG
                                        | Start OUTTMR


OUTTMR >= 10 MINUTES -------------------| Turn outputs off
```

CHAPTER 28

ENGINE RUNNING SEQUENCE

ENGINE RUNNING TEST STRUCTURE

PROCESS

STRATEGY MODULE: VR_ERTS_LH_COM1

```
       -----------
     (    ENTER    )
       -----------
            |
 !---------------------!
 !   ENGINE-RUNNING    !
 ! TEST INITIALIZATION !
 !---------------------!
            |
 !---------------------!
 !GENERIC IDLE SPEED   !
 !      CONTROL        !
 !---------------------!
            |
      /----------\
     / HIGH RPM   \
     \ ISC TEST   /
      \----------/
            |
   /----------------\
  /    A/D  AND      \
  \  SWITCH TESTS    /
   \----------------/
            |
       /---------\
      /EGO SWITCH-\
      \ ING TEST  /
       \---------/
            |
       /---------\
      / AIR TEST  \
      \           /
       \---------/
            |
       /----------\
      /  SPARK     \
      \CONTROL TEST/
       \----------/
            |
       /---------\
      /EGR SYSTEM \
      \   TEST    /
       \---------/
            |
           (A)
```

```
                 (A)
                  |
            /-----------\
           / LOW RPM ISC \
           \    TEST     /
            \-----------/
                  |
             /---------\
            /GOOSE TEST \
            \           /
             \---------/
                  |
        !--------------------!
        ! SEND SERVICE CODES !
        !--------------------!
                  |
        !-------------------!
        !   2 MINUTE DELAY  !
        ! FOR TIMING CHECK  !
        !-------------------!
                  |
        !-------------------! *
        !  THROTTLE PLATE   !
        !  ADJUST MODE      !
        !-------------------!
                  |
             -----------
           (     EXIT    )
             -----------
```

The Throttle Plate Adjust Mode Entry is dependent upon operator action as described in it's documentation unit.

EGOBAR FILTER AND STATE FLAGS

STRATEGY MODULE: VR_EGOBAR_COM1

IEGO is filtered in EGOBAR (side) -where side =left or right on stereo systems, left only on mono systems. Time constant for EGOBAR is VTCEGO (a calibratible parameter). EGOSTE (side) is the resultant ego state flag, determined as follows:

```
Non-shared ego{EGOBAR (side) > 855 counts-->EGOSTE (side)=lean(1)
              {EGOBAR (side) <=855 counts-->EGOSTE (side)=rich(0)

Shared ego/STI{EGOBAR (side) > 425 counts-->EGOSTE (side)=lean(1)
              {EGOBAR (side) <=425 counts-->EGOSTE (side)=rich(0)
```

DELAY LOGIC CLARIFICATION

PROCESS

STRATEGY MODULE: VR_DELAYLOGIC_COM1

Delay VDLY(x) means:

```
!----------!
!   SET    !
! VIPTMR=0 !
!----------!
      |
      |<-------------|
      v              |
 /---------\    YES  |
/ VIPTMR <  \--------|
\ VDLY(x) ? /
 \---------/
      |  NO
      |
      v
   CONTINUE
```

NOTE: "RESTART VIPTMR" means "SET VIPTMR = 0".

ENGINE RUNNING INITIALIZATION

OVERVIEW

The Engine Running Initialization sets specific inputs and outputs as required in preparation for performing the Engine Running On-demand Test. The inputs and outputs are determined based on the Strategy application.

Part of the initialization process is to check the FMEM failure flags for any hard fault present. These flags would have been set from the VIP EXECUTIVE when a 2 second time period is allowed to do Continuous testing prior to entry into the Engine Running Test.

If any of the failure flags are present, the test is aborted and a service code 998 and the corresponding fault code(s) will be output on STO as notification that a hard fault currently exists.

DEFINITIONS

INPUTS

Registers:

- VIP_CNT_EX = Vip State Counter.
- VIP_TIMER_EX = VIP state timer.

Bit Flags:

- AFMFLG = ACT FMEM Flag.
- CFMFLG = ECT failure mode (FMEM) flag.
- DIS_FMFLG = Dual Plug DIS FMEM flag.
- MFMFLG = MAP/MAF FMEM flag.
- TFMFLG = TP FMEM flag.

Calibration Constants:

- VIPSPK = Vip Spark advance units are Deg.
- VISCN = Extended idle CLISC desired RPM.
- VRLAM = Rich LAMBDA for EGO test units are LAMBDAS.

OUTPUTS

Registers:

- EFTR = Equil fuel transfer rate BIN 16 LBM.
- EGRDC = EGR duty cycle.
- KAMREF = Adaptive Fuel correction.
- KAMRF1 = EGO-1 Adaptive fuel correction.
- KAMRF2 = EGO-2 Adaptive fuel correction.
- LAMBSE = Closed loop desired equivalence rate.
- LAMBSE1 = LAMBDA equivalence ratio (EGO-1).
- LAMBSE2 = LAMBDA equivalence ratio (EGO-2).
- PURGDC = Purge duty cycle (FN600 output).
- RVIPRPM = RVIP "desired RPM" to CLISCP.
- SAF = Spark advance Bin 2.
- VCUTOUT = Number of injector cutout during Cylinder Bal Test.
- VIP_CNT_EX = Vip state counter.

Bit Flags:

- ACR = --
- AM1 = Air Management 1 solenoid.
- AM2 = Air Management 2 = TAD.
- BRK_NEVER_OFF = Brake always on during RVIP test.
- BRK_NEVER_ON = Brake not applied during RVIP test.
- ERROR 543 = Fuel pump fault in running VIP; Impacts EGO2.

- OCS_OPEN = Overdrive cancel switch set to open.
- OCS_SHORT = Overdrive cancel switch set to close.
- POWOFF = 1 -> power steering is OFF.
- POWON = 1 -> power steering is ON.
- RVIP_CYL_BAL = Indicates running VIP cylinder.
- RVIP_CYL_QUIT = 1 -> cyl balance test aborted.
- STI_RESET = 1 -> Operator requested Throttle Plate Adjust Mode.
- VRUN_ISCFLG = RVIP idle speed control flag.
- V_LOW_FAN_ON = 1 -> turn on low speed fan during K.O.E.R. VIP.
- V_MODE_SETUP = 1 -> Use throttle mode VIP constants.
- WAC = --

PROCESS

STRATEGY MODULE: VR_RUN_INIT_COM8

```
always ---------------------------------| Turn STO off
 (when in VR_RUN_INIT)                  | CODE COUNT = 0
                                        | STI_RESET = 0
                                        | V_MODE_SETUP = 0
```

FMEM FAULT PRESENT AND INITIALIZATION LOGIC

```
AFMFLG = 1 -----------------|
                            |
CFMFLG = 1 -----------------|OR --|
                            |     |
TFMFLG = 1 -----------------|     |
                                  |AND -|  (output Code 998 and the corres-
MFMFLG = 1 -----------------|     |     |   ponding FMEM fault codes(s)
                            |AND -|     |   except replace service code 126
V_VACFLG = 1----------------|           |   with 128)
                                        |  (exit Engine Running Self Test
                                        |   sequence)
                                        |
                                        | --- ELSE ---
MFMFLG = 1 -----------------------|     |
                                  |AND -|  (output Code 998 and Code 128)
V_VACFLG = 1 ---------------------|     |  (exit Engine Running Self Test
                                        |   sequence)
                                        |
                                        | --- ELSE ---
AFMFLG = 1 -----------------------|     |
                                  |     |
CFMFLG = 1 -----------------------|OR --|  (output Code 998 and the corres-
                                  |     |   ponding FMEM fault code(s)
TFMFLG = 1 -----------------------|     |  (exit Engine Running Self Test
                                  |     |   sequence)
MFMFLG - 1 -----------------------|     |
                                        | --- ELSE ---
                                        |
```

(continued on next page)

(continued from previous page)

```
                                 | VRUN_ISCFLG = 1
                                 | LAMBSE = VRLAM
                                 | RVIPRPM = VISCN
                                 | KNOCK_ENABLED = 0
                                 | EFTR = 0
                                 | KAMREF = 1.0
                                 | SAF = VIPSPK
                                 | EGRDC = 0
                                 | WAC = 1
                                 | AM1 = 0
                                 | AM2 = 0
                                 | PURGDC = 0
                                 | BRK_NEVER_ON = 1
                                 | BRK_NEVER_OFF = 1
                                 | POWON = 0
                                 | POWOFF = 0
                                 | OCS_OPEN
                                 | OCS_SHORT
                                 | VIP_TIMER_EX = 0
                                 | VIP_CNT_EX = VR_OUT_ENGCYL
```

The following logic outputs engine ID pulses on STO. ID code equals 1/2 number of cylinders in engine.

```
always ---------------------------------| Output ENGCYL pulses for the
 (when in VR_OUT_ENGCYL)                 |  engine I.D. code on STO
                                         |  (I.D. code =   of CYL/2)
                                         | Wait until pulsing is complete
                                         | VIP_TIMER_EX = 0
                                         | VIP_CNT_EX = VR_HICAM_ISC
```

GENERIC IDLE SPEED CONTROL

OVERVIEW

The ISC is an adaptive air bypass system designed to regulate the duty cycle to the air bypass solenoid to obtain a desired engine speed for all idle operating conditions and provide a dashpot action. Predicted airflow is adaptively corrected to minimize the impact of hardware variability.

The Self Test ISC test is designed to check control of the ISC system at an extended RPM or high cam condition, as well as a low RPM relative to, but not equal to, normal base idle. Self Test uses the base control system algorithm, but substitutes certain calibration parameters in order to maintain Self Test commonality among the many engine calibrations. The following documentation describes the Self Test portion of ISC. See the base strategy chapter on Generic ISC for detailed information.

PROCESS

STRATEGY MODULE: VR_GENISC_COM1

VIP Strategy: (VRUN_ISCFLG = 1)

DSDRPM CALCULATION

DSDRPM = RVIPRPM [+DNAC] [+DNPOWS] [+HEDFRPM]

DSDRPM is allowed to rise instantaneously, but any decreasing value is filtered to prevent a sudden drop in DSDRPM. DESNLO is the filtered value of the DSDRPM register.

```
DSDRPM < DESNLO ------------------------| Filter DSDRPM
                                        | DESNLO = UROLAV_TC(DESNLO,VTCDSN)
                                        |  (TCDESN is used when VTCDSN = 0)
```

The flag, HCAMFG, is set when in VIP. HCAMFG is used to prevent adaptive airflow updates (ISCKAM).

```
RVIPRPM <> NUBASE ----------------------| HCAMFG = 1
                                        |  (disable adaptive airflow update)
```

NOTE: A/C [DNAC], Power Steering [DNPOWS] and electrodrive fan [HEDFRPM] adders are also included in the final equation; see base strategy logic for conditions.

DESMAF CALCULATION

DESMAF = DESMAF_PRE(predicted air flow) + IPSIBR + DASPOT + ISCKAMn

DESMAF_PRE = FN875N * FN1861(ECT,ATMR3) [+ACPPM] [+PSPPM] [+EDFPPM] [+HWPPM]

FN875N is the air flow required, closed throttle, in neutral, for the desired engine speed.

[ACPPM],[PSPPM],[EDFPPM] and [HWPPM] represent the airflow needed in DESMAF_PRE for the increased load due to rpm adders.

IPSIBR adjusts DESMAF for load changes. Changes in IPSIBR results in corresponding change to bypass valve duty cycle.

IPSIBR = IPSIBR + ISCPSI
(clipped to VSIBRN as a minimum and VSIBRM as a maximum.)

Where ISCPSI is calculated as:

ISCPSI = RPMERR_A * (DESMAF_PRE/DSDRPM) * BG_TIMER/VTC_UNDER OR VTC_OVER ***

```
***
RPMERR_A >= 0 --------------------------| Use VTC_UNDER
 (RPM error for air flow correction)    |
                                        | --- ELSE ---
                                        |
                                        | Use VTC_OVER
```

RPMERR_A = ROLAV(RPMERR,TCBPA) where RPMERR = DSDRPM - N

DASPOT is the air flow to provide a preposition at open throttle positions (APT = 0 OR 1). It is bled off after a closed throttle transition allowing a smooth transition to RPM control.

DASPOT DECREMENT

```
VRUN_ISCFLG = 1 ------------------------| DASPOT = DASPOT -
                                        |   (FN879(DASPOT) * V_879_MULT)
                                        |
                                        | --- ELSE ---
                                        |
                                        | DASPOT = DASPOT - FN879(DASPOT)
```

ISCKAMn = Adaptive correction for load condition, where n is the value of ISFLAG (see base strategy for load state definition).

ISCDTY CALCULATION

The mass air flow through the ISC actuator (DEBYMA) is calculated as the mass air flow at idle (DESMAF), less the flow through the throttle plate, corrected for altitude (FN890).

ISCDTY = FN800(DEBYMA) * V820A * IDCMUL + IDCOFS

DEBYMA = (DESMAF - ITHBMA) * (29.92/BP) - FN890(BP)

ITHBMA is the air flow through the throttle plate at idle.

V820A is the ISC duty cycle multiplier used to replace base strategy FN820A, usually set to 1.

IDCMUL is a multiplier for development, usually set to 1.

IDCOFS is the ISCDTY adder for development, usually set to 0.

HIGH RPM ISC TEST

PROCESS

STRATEGY MODULE: VR_HICAM_ISC_COM2

```
VIP_TIMER_EX < VISDL1 ------------------| RETURN
                                        |  (delay VISDL1 seconds
                                        |   for engine to stabilize
                                        |   at DSDRPM)
                                        |
                                        | --- ELSE ---
                                        |
|DSDRPM - NBAR| > ISUBND ---------------| Set CODE_412
                                        |  (RPM out of range)
                                        | DISABLE_ISC = 1
                                        |  (freeze ISC duty cycle)
                                        | VIP_TIMER_EX = 0
                                        | VIP_CNT_EX = VR_SENSOR_CHK
                                        |
                                        | --- ELSE ---
                                        |
                                        | DISABLE_ISC = 1
                                        | VIP_TIMER_EX = 0
                                        | VIP_CNT_EX = VR_SENSOR_CHK
```

ECT SENSOR TEST

PROCESS

STRATEGY MODULE: VR_ECT_COM1

```
   -----------
 (    ENTER    )
   -----------
        |
        v
   /---------\
  /   VIECT3  \   NO        !---------!
 /   >=IECT>=  \----------->! CODE 116!
 \    VIECT4 ? /            !---------!
  \           /                  |
   \---------/                   |
        |  YES                   |
        |<-----------------------|
        v
   ----------
 (     EXIT   )
   ----------
```

MAP SENSOR TEST

PROCESS

STRATEGY MODULE: VR_MAP_COM1

```
  -----------
(    ENTER    )
  -----------
       |
       v
  /---------\    NO
 /  MAPTMR <=\----------------|
 \  VMPMAX ? /                |
  \---------/                 |
       |  YES                 |
       v                      |
  /---------\                 v
 /   VMDEL3  \   NO      !---------!
/  <=MDELTA<= \--------->! CODE 126!
\    VMDEL4    /         !---------!
 \           /                |
  \---------/                 |
       |   YES                |
       |<---------------------|
       v
  -----------
(     EXIT    )
  -----------
```

THROTTLE POSITION SENSOR

PROCESS

STRATEGY MODULE: VR_TPS_COM1

```
   -----------
 (    ENTER    )
   -----------
        |
        v
   /---------\
  /   VTAP3   \   NO      !---------!
 /  <=ITP <=   \--------->! CODE 121!
 \    VTAP4 ?  /          !---------!
  \           /                |
   \---------/                 |
        |  YES                 |
        |<---------------------|
        v
   -----------
 (     EXIT    )
   -----------
```

AIR CHARGE TEMPERATURE SENSOR

PROCESS

STRATEGY MODULE: VR_ACT_COM1

```
   -----------
 (    ENTER    )
   -----------
        |
        v
   /---------\
  /   VIACT3  \    NO      !---------!
 /  >=IACT >=  \---------->! CODE 114!
 \    VIACT4 ? /           !---------!
  \           /                 |
   \---------/                  |
        |  YES                  |
        |<----------------------|
        v
   -----------
 (     EXIT    )
   -----------
```

TRANSMISSION OIL TEMPERATURE SENSOR

PROCESS

STRATEGY MODULE: VR_TOT_COM1

```
   -----------
  (    ENTER    )
   -----------
        |
        v
   /---------\
  /   VITOT3  \   NO       !---------!
 /  >=ITOT >=  \---------->! CODE 636!
 \    VITOT4 ? /           !---------!
  \           /                 |
   \---------/                  |
        |  YES                  |
        |<----------------------|
        v
   ----------
  (     EXIT   )
   ----------
```

BRAKE ON/OFF TEST

OVERVIEW

The BRAKE ON/OFF test checks the integrity of the brake switch input to the processor. This test requires the operator to depress the brake pedal any time during the ENGINE RUNNING test, from the I.D. code to the service code output, (including the GOOSE test). This will toggle the input, BIFLG, when the switch opens and closes. The BIFLG check is done every background pass and the service code setup is done just prior to the code output routine.

DEFINITIONS

Bit Flags:

- BIFLG = Brake input signal.
- BRK_NEVER_OFF = Brake always on during test; initialized to 1 in engine running initialization.
- BRK_NEVER_ON = Brake not applied during test; initialized to 1 in engine running initialization.

Calibration Constants:

- BIHP = Base strategy hardware present indicator; 1 = switch present.
- VBISW = Brake input test enable switch; 1 = enable test.

PROCESS

STRATEGY MODULE: VR_BOO_COM1

BIFLG CHECK

```
BIFLG = 1 ------------------------------| BRK_NEVER_ON = 0
 (brake on, Boo high)                   |
                                        |  --- ELSE ---
                                        |
                                        | BRK_NEVER_OFF = 0
```

SERVICE CODE SET-UP

```
VBISW = 1 ------------------|
                            |AND -|
BIHP = 1 -------------------|     |
                                  |AND -| SET CODE 536
BRK_NEVER_ON = 1 -----------|     |
                            |OR --|
BRK_NEVER_OFF = 1 ----------|
```

POWER STEERING PRESSURE SWITCH TEST

OVERVIEW

The PSPS (power steering pressure switch) test is a functional check of the pressure switch input to the processor. The power steering system must be filled with fluid, and pressurized by the operator turning the steering wheel fully in one direction to the stop and then releasing. This will toggle the input, POWSFG, as the switch opens and closes. This input is used in strategy as a power steering load adder for idle speed control. The POWSFG check is done every background pass and the service code setup is done just prior to the code output routine.

DEFINITIONS

Bit Flags:

- POWOFF = Power steering was never off; initialized to 0 in engine running initialization.

- POWON = Power steering was never on; initialized to 0 in engine running initialization.

- POWSFG = Power steering pressure switch input signal.

Calibration Constants:

- PSPSHP = Base strategy hardware present indicator; 1 -> switch present

- VPSSW = PSPS input test enable switch; 1 -> enable test.

PROCESS

STRATEGY MODULE: VR_PSPS_COM2

POWSFG CHECK

```
POWSFG = 1 -----------------------------| POWON = 1
 (power steering is on)                 |
                                        | --- ELSE ---
                                        |
                                        | POWOFF = 1
```

SERVICE CODE SET-UP

```
VPSSW = 1 ------------------------|
                                  |AND -|
PSPSHP = 1 -----------------------|     |
                                        |AND -| Set Code 521
POWON = 0 ------------------------|     |
                                  |OR --|
POWOFF = 0 -----------------------|
```

TRANSMISSION CONTROL SWITCH CIRCUIT TEST

OVERVIEW

The Transmission Control (TCS) switch circuit test checks the integrity of the circuit from the switch to the processor. This test requires the operator to manually depress and release the switch at any time during the KOER test from the I.D. code to the service code output, (including the GOOSE test). This action will cause the input, ITCS to toggle, setting flags as shown in the logic below. The ITCS input check is done every background pass and the service code set-up is done just prior to the code output routine.

DEFINITIONS

INPUTS

Bit Flags:

- ITCS = Switch input signal.
- TCS_SHORT = Transmission Control switch on state (ITCS = 1) initialized to 0 in engine running initialization.
- TCS_OPEN = Transmission Control switch off state (ITCS = 0) initialized to 0 in engine running initialization.

Calibration Constants:

- TSTRAT = Transmission Strategy Switch. The TSTRAT software switch selects which transmission control strategy is to be executed;
  - 0 -> No transmission control, (Manual trans., AOD, ATX, C6, C3, etc.),
  - 1 -> SIL (Shift Indicator Light),
  - 2 -> A4LD with 3-4 shift control and converter clutch control,
  - 3 -> AXOD,
  - 4 -> E4OD,
  - 5 -> A4LD-E,
  - 6 -> AXOD-E,
  - 7 -> AOD-I,
  - 8 -> F4E,
  - 9 -> CD4E,
  - 10 -> JATCO

OUTPUTS

Bit Flags:

- TCS_SHORT = See above.
- TCS_OPEN = See above.

PROCESS

STRATEGY MODULE: VR_TCS_COM4

```
ITCS = 1 -------------------------------| TCS_SHORT = 1
 (TCS button depressed, 12 volts)       |
                                        | --- ELSE ---
                                        |
                                        | TCS_OPEN = 1
```

SERVICE CODE SET-UP

```
TSTRAT = 4 -----------------------|
                                  |AND -| Set Code 632
TCS_OPEN = 0 ---------------|     |
                            |OR --|
TCS_SHORT = 0 --------------|
```

EGO SWITCHING TEST

OVERVIEW

The Ego switching test has been designed to check the switching capability of the Ego sensor, Ego circuit continuity and connections. Hego heater supply voltage is not addressed.

Test preconditioning is done during the initialization sequence where engine rpm is adjusted to an upper level and lambda is set to a slightly rich condition (VRLAM). Spark advance is fixed to VIPSPK and selected items such as canister purge, thermactor air, and egr are de-activated to eleminate interaction with engine A/F ratio.

An Ego warm-up delay (VISDL1) is used to help warm-up the Ego sensor before entering the test. This delay is also used to allow the ISC circuit to settle.

The test begins by comparing NBAR (filtered rpm) to VNMIN. If engine rpm is below VNMIN, the test is aborted and jumps to the goose test section. A service code 416 will be output to indicate a low rpm condition at the end of the Engine-Running test.

If the engine rpm is above VNMIN, the test continues by ramping fuel lean from VRLAM to the clip LEQV at the rate of VIPLR1. Ego state is monitored during this ramping sequence and a time constraint (VIPTM3) is used to determine the maximum time allowed. If VIPTM3 is calibrated larger than the time needed to reach LEQV, a dwell time at the clip will result.

If an Ego switch has not occured within this time period, a service code 173 (fuel always rich) is set and the test exits under a specific set of conditions. Lambse jumps back to VRLAM and AM1 and AM2 is turned off. A delay (VISDL8) is used just prior to entering the egr test for stabilization.

If a lean Ego switch has occured, fuel is ramped rich from the lean switch point to the rich clip REQV at the rich rate VIPRR1. Again, Ego state is monitored during this process and a dwell timer VIPTM4 is used at the rich clip.

If a rich ego switch has not occured within the VIPTM4 dwell, a service code 172 (fuel always lean) is set and the test exits under the same conditions as in the code 173 failure.

If a rich Ego switch has occured within the allowable time, the test exits with the next path being either the thermactor air test and/or the egr test.

It should be noted that in a correctly operating system the maximum timers VIPTM3/VIPTM4 may never be realized if the Ego responds quickly. The test therefor takes only as long as necessary to verify ego switches.

Calibration of the timers and clips have to take into account how accurate the fuel strategy volumetric efficiency tables are calibrated atthe speed/load point the Ego switching test is performed. Wider fuel ramp excursions may be necessary if stoichiometry is observed at a lambda of 1.25 for example.

The following is a list of Ego switching test calibration parameters and recommended values.

PARAMETER	RECOMMENDED VALUE
VNMIN	1000 rpm
VIPLR1	.05 lambda/sec.
LEQV	1.3 lambda
VIPTM3	13 sec.
VIPRR1	.075 lambda/sec.
REQV	.75 lambda
VIPTM4	5 sec.

PROCESS

STRATEGY MODULE: VR_EGO_COM1

```
  -----------       /-----------\ YES   ----
(    ENTER    )--->/ V_EGOHP = 0 \---->( EXIT )
  -----------      \             /      ----
       |            \-----------/
       |                 |NO
       |<----------------|
       |
       v
 /---------\  NO
/ NBAR >=   \------>(EXIT TO LOWER RPM TEST)
\  VNMIN ?  /
 \---------/
       | YES
       v
!---------------!     !----------!
!.RAMP FUEL     !     ! CONTINUE !
! LEAN AT VIPLR1!<----! RAMPING  !
!.CLIP AT LEAN  !     !----------!
! LIMIT LEQV    !          ^
!---------------!          |
       |                   |
       v                   | NO          !---------!
  /---------\  NO     /---------\  YES   !   SET   !
 / EGOSTE=1 ?\------>/VIP_TIMER_EX------>! CODE 173!----->( AA )
 \  (LEAN)   /       \>= VIPTM3 ?/       !---------!
  \---------/         \---------/
       |  YES
       v                        |<------------------------------|
!---------------!               v                               |
!.RAMP LAMBSE   !          /---------\  YES       ----          |
! RICH AT VIPRR1!--------->/ EGOSTE=0  \------>  ( EXIT )       |
! CLIP AT RICH  !         \  (RICH) ?  /          ----          |
! LIMIT REQV    !          \---------/                          |
!---------------!               | NO                            |
       ^                        |                               |
       |                        v                               |
!-------------!       NO   /---------\  YES   /---------\  NO   |
!.CONTINUE    !<----------/ IS LAMBSE \------>/ VIPTMR >= \-----|
! RAMPING     !           \ <= REQV ? /       \  VIPTM4 ? /
!.ZERO        !            \---------/         \---------/
! VIP_TIMER_EX!                                     | YES
!-------------!                                     v
                                               !---------!
                                               !  SET    !
                                               ! CODE 172!
                                               !---------!
                                                    |
                                                    v
                                               -----------
                                              ( EXIT TO AA )
                                               -----------
```

THERMACTOR AIR TEST

PROCESS

STRATEGY MODULE: VR_THERMAIR_COM2

```
       -------
      ( ENTER )
       -----
         |
         v
    /---------\  NO
   / THRMHP=1? \------>( GO TO AA )
   \           /
    \---------/
         |  YES
         v
    /---------\  NO
   / VAIRFL=1 ?\------>( AA )
   \           /
    \---------/
         |  YES
         v
!----------------!
! RAMP FUEL      !
! RICH AT VIPRR1!
! FOR VDLY1 SEC !
!----------------!
         |
         v
!----------------------!
! .AIR UPSTREAM        !
!----------------------!
         |
         |<-------------------|
         v                    | NO
    /---------\  NO     /---------\  YES   !-------------!
   / EGOSTE=1 ?\------>/VIP_TIMER_EX------>! SET CODE 311!
   \  (LEAN)   /       \>= VIPTM2 ?/       !-------------!
    \---------/         \---------/              |
         |  YES                                  v
         v                                     ( AA )
!------------------------!
! .AIR DOWNSTREAM        !
!------------------------!
         |
         |<-------------------|
         v                    | NO
    /---------\  NO     /---------\  YES   !-------------!
   / EGOSTE=0 ?\------>/VIP_TIMER_EX------>! SET CODE 312!------|
   \  (RICH)   /       \>= VATMR2 ?/       !-------------!      |
    \---------/         \---------/                             |
         |  YES                                                 |
         |<-----------------------------------------------------|
         v
       ( F2 )
```

```
      ( F2 )
        |
        v
   /---------\  NO
  / VTABFL=1 ?\------>( AA )
  \          /
   \---------/
        |  YES
        v
 !--------------!
 ! AIR UPSTREAM !
 ! AND BYPASSED !
 !--------------!
        |
        v
 !--------------!
 ! DELAY VDLY2  !
 !     SEC      !
 !--------------!
        |
        v
   /---------\  NO    !-------------!
  / EGOSTE=0 ?\------>! SET CODE 313!
  \  (RICH)   /       !-------------!
   \---------/               |
        |  YES               |
        |<-------------------|
        v
      ( AA )
        |
        v
!--------------!
! .SET LAMBSE= !
!    VRLAM     !
! .AM1 AND AM2 !
!     OFF      !
!--------------!
        |
        v
   -------------------
 (GO TO SPARK CNTRL TEST )
   ------------------
```

SPARK CONTROL TEST

OVERVIEW

The Spark Control Test provides verification that the EEC spark output is actually controlling by means of using the engine RPM as feedback while other parameters are held constant. The ISC duty cycle is held constant to prevent engine speed correction and LAMBSE is fixed to VRLAM as the spark output is ramped from an initial starting point to a final spark and the engine speed (NBAR) input is compared to a calibrated delta for the expected change. The delta distance to be ramped divided by the spark ramp rate plus the delay times result in total test time required when there is a failure. However, when the delta is reached, indicating the system is controlling, the test time is less.

DEFINITIONS

Registers:

- NSAV = Temporary register to store current NBAR.

- NBAR = Filtered engine speed input used to compare to NSAV.

- SAF = Final spark advance output to spark controller.

- VIP_TIMER_EX = VIP execution time; 1/8 second later, also used as medium for ramping spark advance.

Calibration Constants:

- V_SPK_ENABL = Calibration switch to enable/disable test, flag bit 0 = bypass test; 1 = do test.

- V_SPK_INIT = Initial spark starting point to begin ramping from.

- VDLY8 = Delay time used before beginning spark ramp or used as delay before entering the EGR TEST when the SPARK TEST is bypassed.

- V_SPK_RATE = Spark ramp rate required during test.

- V_SPK_FINAL = Final spark for spark ramp to end.

- VIPSPK = Normal VIP spark (30 deg's BTC).

- VSPTDL = Delay time before exiting test after setting SAF = VIPSPK.

- CODE 213 = Service code which indicated spark control fault present.

PROCESS

STRATEGY MODULE: VR_SPARK_COM1

```
                -----
              ( ENTER )
                -----
                  |
                  v
             /------------\NO   !-------!      -----
            / V_SPK_ENABL \--->! DELAY !-->(  EXIT )
            \      = 1     /    ! VDLY8 !      -----
             \------------/     !-------!
                  |YES
                  v
            !----------------!
            ! SAF=V_SPK_INIT!
            !----------------!
                  |
             !-------------!
             ! DELAY VDLY8!
             !-------------!
                  |
          !------------------!
          !    . NSAV = NBAR  !
          !    . VIPTMR = 0   !
          !------------------!
                  |
|---------------->|
|         !------------------!
|         !SAV=V_SPK_INIT - !
|         !V_SPK_RATE*VIPTMR!
|         !------------------!
|                 |
|          /---------------\YES
|         /ABS(NSAV-NBAR)   \------------------------|
|         \    => VSPRPM    /                        |
|          \---------------/                         |
|                 |NO                                 |
|    YES   /-------------\NO    !----------!         |
|--------/     SAF >     \---->!  SET     !         |
          \ V_SPK_FINAL  /      !CODE 213 !         |
           \-------------/      !----------!         |
                                    |                |
                                    v                |
                  |-----------------------------------|
                  v
             !-----------!
             !SAF=VIPSPK!
             !-----------!
                  |
            !---------------!         -----
            ! DELAY VSPTDL !-----> ( EXIT  )
            !---------------!         -----
```

EXHAUST GAS RE-CIRCULATION SYSTEM TEST (SONIC)

PROCESS

STRATEGY MODULE: VR_EGR_COM6

```
  ---------
(  ENTER    )
  ---------
      |
      v
 /---------\  NO
/  PFEHP=0  \--------------------|
\    ?      /                    |
 \---------/                     |
      |  YES                     |
      v                          |
 /---------\ YES  !---------!    |
/  IEGR <=  \---->! CODE 327!--->|
\  EVPMIN ? /     !---------!    |
 \---------/                     |
      | NO                       |
      v                          |
 /---------\ YES  !---------!    |
/  IEGR >=  \---->! CODE 337!--->|
\  EVPMAX ? /     !---------!    |
 \---------/                     |
      | NO                       |
      v                          |
 /---------\ YES  !---------!    |
/  IEGR <   \---->! CODE 328!--->|
\  VEVPLL ? /     !---------!    |
 \---------/                     |
      | NO                       |
      v                          |
 /---------\ YES  !---------!    |
/  IEGR >   \---->! CODE 334!--->|
\  VEVPHL ? /     !---------!    |
 \---------/                     |
      | NO                       |
      v              ----        |
    ( B )          ( EXIT )<-----|
                     ----
```

```
             ( B )
               |
               v
    !---------------------!
    !.SET VIP_TIMER_EX = 0!
    !.SAVE IEGR -> TPSAV  !
    !---------------------!
               |
 |------------>|
 |             v
 |   !---------------------!
 |   !.RAMP EGRDC AT VEGRAT !
 |   ! START FROM VDCMIN TO !
 |   ! VDCMAX.              !
 |   !.EGRDC=VDCMIN +       !
 |   ! VEGRAT*VIP_TIMER_EX  !
 |   !---------------------!
 |             |
 |             |
 |             |
 |             v
 |        /-----------\   YES
 |       / IEGR-IEVPSV \----|
 |       \ > VEVPDL ?  /    |
 |        \-----------/     |
 |             | NO         |
 |             |            |
 |             v            |
 |    NO  /---------\       |
 |-------/ EGRDC >=  \      |
        \  VDCMAX   /       |
         \---------/        |
               |            |
               |            |
               | YES        |
               v            |
          !----------!      |
          ! CODE 332 !      |
          !----------!      |
               |            |
               |            |
               |<-----------|
               v
      !---------------------!
      !.SET EGRDC=0          !
      !.SET VIP_TIMER_EX= 0 !
      !---------------------!
               |
               |
               v
           ---------
         (   EXIT    )
           ---------
```

LOW RPM ISC TEST

PROCESS

STRATEGY MODULE: VR_LOW_ISC_COM4

```
VIP_CNT_EX = VR_LOW_ISC ----------------| DISABLE_ISC = 0
                                        | RVIPRPM = NGOOSE
                                        | VIP_TIMER_EX = 0
                                        |  (continue with ISC test)


VIP_TIMER_EX < VISDL3 ------------------| RETURN
                                        |  (delay VISDL3 seconds
                                        |   for engine to stabilize
                                        |   at DSDRPM)
                                        |
                                        | --- ELSE ---
                                        |
|DSDRPM - NBAR| > ISLBND ---------------| Set CODE_411
                                        |  (RPM out of range)
                                        | VIP_TIMER_EX = 0
                                        | VIP_CNT_EX = VR_GOOSE
                                        |
                                        | --- ELSE ---
                                        |
                                        | VIP_TIMER_EX = 0
                                        | VIP_CNT_EX = VR_GOOSE
```

GOOSE TEST
SPEED DENSITY, MONO EGO

OVERVIEW

Test operator is directed to Goose the throttle as soon as he sees the single pulse (readout of 10 on Star unit) so that dynamic response can be tested. The test will end when one of the following conditions are met.

- V_GOOS_DELAY seconds elapse after the RPM response occurs
- Time in test exceeds VGOOSEC seconds.

In the second case, a code 538 is sent to indicate that the test was incorrectly performed.

DEFINITIONS

Registers:

- NSAV = Temporary register to store current NBAR.
- TPSAV = Temporary register to store current TP value.
- MAPSAV = Temporary register to store current MAP value.

Bit Flags:

- VF1 = Flag indicating VIPTMR has been loaded with a new value, (VGOOS_DELAY), based on an RPM response.
- CODE_225 = Flag bit for error code 225, to be cleared during the test if knock is present.
- CODE_129 = Flag bit for error code 129, to be cleared during the test when the MAP indicates movement beyond a calibrated target.
- CODE_167 = Flag bit for error code 167, to be cleared during the test when the TP indicates movement beyond a calibrated target.
- CODE_538 = Flag bit for error code 538, to be cleared during the test when the RPM has increased above a calibrated target.

Calibration Constants:

- V_GOOSW = Flag to enter GOOSE test (1=enter;0-bypass).
- V_GOOSPK = Spark advance for knock sensing (50 deg's BTC) NOTE: V_GOOSPK is limited by normal strategy parameter SPUCLP.
- V_GOOSN = RPM change to determine GOOSE test has been performed (400 RPM).
- V_GOOSTP = TP change required in GOOSE test (200 cnts).

- V_GOOSMP = MAP change required in GOOSE test (10in.Hg).
- V_GOOSEC = Time in GOOSE test loop (15 sec's).
- V_GOOS_DELAY = Time elapse after RPM change to exit test (min. 1.5 sec's).
- VIPSPK = Normal VIP spark (30 deg's BTC).
- V_KNK_DLY = Delay time for GOOSE initialization (1 sec).
- V_KNK_FUEL = New LAMBSE setting for engine knock enhancement (0.8 LAMBDAS).
- V_KTS = Knock threshold pulse time (25 ticks).
- VRLAM = Normal LAMBSE setting from EGO switching test (0.9 LAMBDAS).

PROCESS

STRATEGY MODULE: VR_GOOSE_SD_COM1

```
      -------
    (  ENTER  )
      -------
         |
         v
    /---------\  YES        ----
   /V_GOOSW=0 ?\-------->( EXIT )
   \           /          ----
    \---------/
         |  NO
         |               !---------------------------!
         v               ! .SET SAF TO V_GOOSPK      !
    /---------\   YES    ! .SET LAMBSE = V_KNK_FUEL !
   / KIHP=1 ?  \-------->! WINDOW_DELTA = V_KTS      !
   \           /         ! (OVERIDE NORMAL STRATEGY !
    \---------/          !  KTS PULSE WIDTH)         !
         |  NO           !---------------------------!
         |                              |
         |                     !----------------!
         |                     !DELAY V_KNK_DLY!
         |                     !----------------!
         |<-----------------------------|
         v
 !-----------------------------!
 ! .SET CODE_225, CODE_129,    !
 !      CODE_167, CODE_538     !
 ! .SAVE CURRENT NBAR-->NSAV  !
 ! .SAVE CURRENT TP-->TPSAV   !
 ! .SAVE CURRENT MAP-->MAPSAV !
 ! VIP_KNOCK = 0               !
 ! (begin sensing for knock)  !
 !-----------------------------!
         |
         v
 !-----------------------------!
 ! SEND SINGLE 0.5 SEC PULSE  !
 ! ON STO                      !
 !-----------------------------!
         |
         v
    !---------------!
    !VIP_TIMER_EX=0!
    !---------------!
         |
         v
      ( G1 )
```

```
          ( G1 )
|--------->|
|          v              /----------------\
|     /---------\  YES   /VIP_KNOCK = 1     \YES  !---------------!
|    / KIHP=1 ?  \-----> \(KNOCK DETECTED?) /---->! CLEAR CODE_225!
|    \           /        \----------------/      !---------------!
|     \---------/               |                        |
|          |  NO                |  NO                    |
|          |<-------------------------------------------|
|          v
|     /---------\
|    /           \  YES   !---------------!
|   / (N-NSAV) >= \------>! CLEAR CODE_538!
|   \  V_GOOSN ?  /       !---------------!
|    \           /                |
|     \---------/                 |
|          |  NO                  |
|          |<--------------------|
|          v
|     /---------\
|    /    ABS    \  YES   !----------------!
|   /  (TP-TPSAV) \------>! CLEAR CODE_167 !
|   \      >=     /       !----------------!
|    \ V_GOOSTP ?/                  |
|     \---------/                   |
|          |  NO                    |
|          |<----------------------|
|          v
|     /---------\
|    /    ABS    \  YES   !----------------!
|   /(MAP-MAPSAV) \------>! CLEAR CODE_129 !
|   \ >= V_GOOSMP?/       !----------------!
|    \           /                  |
|     \---------/                   |
|          |  NO                    |
|          |<----------------------|
|          v
|     /---------\
|    /    IS     \  YES    /---------\  YES   !--------------!
|   / CODE_538    \------>/    VF1    \------>! .SET VF1     !
|   \   CLEAR ?   /       \  CLEAR ?  /       ! VIP_TIMER_EX=!
|    \           /         \---------/        ! (V_GOOSEC-   !
|     \---------/              |  NO          ! V_GOOS_DELAY)!
|          |  NO               v              !--------------!
|          |<----------------------------------------|
|          v
| NO  /-----------\YES   /--------\YES  !------------------!
|----/VIP_TIMER_EX \--->/ KIHP=1?  \--->!.SET LAMBSE =VRLAM!---|
     \ > V_GOOSEC ?/    \          /    !------------------!   |
      \-----------/      \--------/                            |
                             |NO                               v
                             |------------------------------->(G2)
```

```
                      ( G2 )
                         |
                         v
                 YES/--------\
         |-----------/ CODE_ 538\
         |           \    SET ? /
         v            \--------/
!----------------!       |NO
!CALL SETUP_CODE !       |
!                !    /--------\ YES  !-------------------!
!ENTER ERROR INTO!   / CODE_129 \---->!CALL SETUP_CODE    !
!SERVICE CODE    !   \  SET ?   /     !ENTER ERROR INTO   !
!  TABLE         !    \--------/      !SERVICE CODE TABLE!
!----------------!       |NO          !-------------------!
         |               |                      |
         |               |<-------------------|
         |               v
         |            /--------\ YES !-------------------!
         |           / CODE_167 \--->!CALL SETUP_CODE    !
         |           \  SET ?   /    !ENTER ERROR INTO   !
         |            \--------/     !SERVICE CODE TABLE!
         |               |NO         !-------------------!
         |               |                      |
         |               |<-------------------|
         |               v
         |          NO /--------\
         |<-----------/IS KIHP = \
         |            \  1 ?     /
         |             \--------/
         |               |YES
         |               |
         |            /--------\ YES  !-------------------!
         |           / CODE_225 \---->!CALL SETUP_CODE    !
         |           \  SET ?   /     !ENTER ERROR INTO   !
         |            \--------/      !SERVICE CODE TABLE!
         |               |NO          !-------------------!
         |               |                      |
         |---------------->|<--------------------|
                         v
                   !------------!
                   !SAF = VIPSPK!
                   !------------!
                         |
                         v
                 !-------------------!
                 ! GO TO SERVICE     !
                 ! OUTPUT ROUTINE    !
                 !                   !
                 ! SEND ALL SERVICE  !
                 ! CODES NOT CLEARED!
                 !-------------------!
```

THROTTLE PLATE ADJUSTMENT MODE

OVERVIEW

The THROTTLE ADJUST MODE allows checking and if necessary adjustment of the throttle hard set at the desired rpm V_RPM_SET.

NOTE: The ignition timing should be checked and adjusted to specification prior to any adjustment of the throttle.

The operator can enter the throttle adjust mode any time during the 2 minute timing check by ungrounding and again grounding STI within a 4 second lapsed time (STI_RESET).

STI_RESET is cleared on entry to Engine Running when VRUN_ISCFLG is set to 1. Exit from Engine Running Test will only be allowed when STI_RESET is 0. STI_RESET can only be set to 1 during the 2 minute timing check. See exit logic page for other conditions which will cause exit from Engine Running Test.

Once the mode is entered the preset engine conditions are allowed to stabilize for a calibrated period of time. To signal that this time has elapsed and throttle adjustment may proceed, a seperator pulse is output on Self Test Output (STO). The MODE SET UP LOGIC can be exited by ungrounding the STI.

NOTE: Applications which have electrodrive fan require low speed fan operation during this mode. (Flag V_LOW_FAN_ON = 1)

VIP flags, RUNNING and VRUN_ISCFLG, remain set (=1) during this mode.

The STO is also used as feedback to the operator during the adjust mode. If the idle speed is within the range the STO will be "on" constantly, otherwise it will "flash" at a rate of 1 Hz when below the range or at a rate of 4 Hz when above the range. If at anytime during this mode the TP sensor goes out of range the STO will flash at a rate of 8 Hz.

The Adjustment Mode ends when a calibrated time period (V_MODE_END), or a maximum of 10 minutes (600 sec's) is reached. As long as the time period is less than (V_MODE_END) the mode can be re-entered by ungrounding STI and again grounding STI within 4 sec's. The re-entry point is at the MODE SET UP LOGIC where all the parameters are set up and the STO signals that the mode is entered.

DEFINITIONS

INPUTS

Registers:

- VIP_TIMER_EX = VIP execution time; timer (eighths of sec).

- OUTTMR = VIP output test timer (sec's); revised from flag driven to free running timer.

Bit Flags:

- DISABLE_ISC = Flag used for idle fuel modulation determination 1 = disable, 0 = enable.

- OLFLG = Base Strategy flag which indicates type of fuel control, 0 = closed loop control, 1 = open loop control.

- V_MODE_SETUP = Flag indicates entry in the Throttle Adjust Mode, 1 = enabled.

- STIFLG = Self Test input which indicates VIP testing requested, 1 = tester input is grounded.

- STI_RESET = STI status flag during 2 minute timing check. See STI_RESET LOGIC and TIMER LOGIC within the process.

Calibration Constants:

- VDLY_ENTER = Delay time to stabilize engine before sending pulse on STO which indicates adjust mode entered, base value = 4 sec's.

- V_ISCMOD_MAX = Lean limit clip on idle fuel modulation, base value = 1.1.

- V_ISCMOD_MIN = Rich limit clip on idle fuel modulation, base value = 0.9.

- V_KDNDT = VIP gain for idle fuel modulation, base value = 0.0005.

- V_MODE_END = Time to allow in service mode, base value = 300 sec's.

- V_MODE_OPT = Enable flag for throttle plate adjust mode, 1 = enable.

- V_NHIGH = Maximum range of rpm set, base value = 50 rpm.

- V_NLOW = Minimum range of rpm set, base value = 30 rpm.

- V_RPM_SET = Desired rpm to be used in DSDRPM calculation, base value = 1000 rpm. Note: Must be above NUBASE rpm.

- V_STO_DELAY = Time delay to allow the tester to clear after sending the 1/2 second pulse on STO, base value = 4 sec's. Note: clipped to a minimum of 4 seconds.

OUTPUTS

Registers:

- VIP_TIMER_EX = See above.

- OUTTMR = See above.

Bit Flags:

- STI_RESET = See above.

- V_MODE_SETUP = See above.

- OLFLG = See above.

- DISABLE_ISC = See above.

PROCESS

STRATEGY MODULE: VR_TPADJ_MODE_COM1

This logic is performed once STI_RESET = 1 from the 2 min. delay logic. The purpose is to allow a 4 second time period to enter the TP ADJUST MODE by again grounding STI. If the 4 seconds elapses without action on STI the result is to exit RUNNING VIP. When TP ADJUST MODE is requested, DSDRPM is calculated based on the calibration parameter V_RPM_SET and fuel control is forced closed loop.

```
VIP_CNT_EX = STI_RESET TIMER LOGIC -----| PROCESS: STI_RESET TIMER
                                        |  LOGIC
```

STI_RESET TIMER LOGIC

```
                                        | Allow 1/4 second for switch
VIP_TIMER_EX < .25 seconds -------------| debounce to prevent clearing
                                        | ISCKAM when not entering the
                                        | THROTTLE ADJUST MODE
                                        |
                                        | --- ELSE ---
                                        |
VIP_TIMER_EX < 4 seconds ---------|     | Allow 4 seconds to determine
                                  |AND -| if THROTTLE ADJUST MODE
STIFLG = 0 -----------------------|     | entry is selected.
 (STI not grounded)                     |
                                        | --- ELSE ---
                                        |
VIP_TIMER_EX < 4 seconds ---------|     | V_MODE_SETUP = 1
                                  |AND -| RVIPRPM = V_RPM_SET
STIFLG = 1 -----------------------|     | OLFLG = 0 (closed loop fuel)
 (STI is grounded)                      | VIP_TIMER_EX = 0
                                        | VIP_CNT_EX = MODE SETUP
                                        |  LOGIC
                                        |
                                        | --- ELSE ---
                                        |
VIP_TIMER_EX >OR= 4 seconds ------|     | STI_RESET = 0
                                  |AND -| VIP_CNT_EX = VIP_REINIT
STIFLG = 0 -----------------------      |  (TP ADJUST MODE not
 (STI not grounded)                     |    selected)
                                        |
                                        | ---ELSE---
                                        |
VIP_TIMER_EX >OR= 4 seconds ------|     | STI_RESET = 0
                                  |AND -| VIP_CNT_EX = VIP_REINIT
STIFLG = 1 -----------------------|     |  (TP ADJUST MODE has been
 ( STI is grounded)                     |   bypassed and WIGGLE TEST
                                        |   is requested)
```

This logic sets up the conditions for the test and allows the engine to stabilize. Feedback spark is locked at the mean idle spark value and Kam is disabled from learning by clearing each background loop. The total airflow becomes the predicted airflow by ignoring airflow correction factor IPSIBR AND DASPOT.

Idle Fuel Modulation is used for idle stability of speed density systems. Most MAF systems do not need Idle Fuel Modulation and can be disabled by setting the gain term V_KDNDT = 0. Certain MAF configurations may require compensation which is opposite of what is required for speed density. In these cases the gain term V_KDNDT can be made negative.

Please note that Engine Running VIP also has Idle Fuel Modulation capability and if it was necessary to use it there, then it is needed here also.

```
VIP_CNT_EX = MODE SET UP LOGIC ---------| OUTTMR = 0
                                        | PROCESS: MODE SET UP LOGIC
```

MODE SET UP LOGIC

```
STIFLG = 0 -----------------------------| VIP_CNT_EX = STI_RESET TIMER
                                        |  LOGIC
                                        |
                                        | --- ELSE ---
                                        |
                                        | RETURN TO BACKGROUND
                                        | Wait for engine to stabilize
                                        | Base strategy actions:
                                        |  DESMAF = DESMAF_PRE
VIP_TIMER_EX < VDLY_ENTER --------------|   (ISCKAM is cleared each
                                        |    background loop. IPSIBR
                                        |    is ignored)
                                        |  SPK_FBS = SPK_IDLE
                                        |  (KSPARK = 0; mean idle spark)
                                        |
                                        | --- ELSE ---
                                        |
                                        | Send single 0.5 sec. pulse on STO
                                        | VIP_TIMER_EX = 0
```

TP ADJUST MODE FEEDBACK LOGIC

```
VIP_TIMER_EX < V_STO_DELAY -------------| Delay to clear tester after
 (Clipped to 4 sec's min.)              | 0.5 sec. pulse on STO
                                        |
                                        | --- ELSE ---
                                        |
                                        | Test has timed out
OUTTMR => V_MODE_END -------------------| STO = 0
 (Vector clipped at 600 sec's max.)     | VIP_TIMER_EX = 0
                                        | VIP_CNT_EX = (VIP_REINIT)
                                        |
                                        | --- ELSE ---
                                        |
STI_FLG = 0 ----------------------------| VIP_CNT_EX = STI_RESET TIMER
                                        |  LOGIC
                                        |
                                        | --- ELSE ---
                                        |
ITP < VTAP3 ----------------------|     |
                                  |OR --| Pulse STO at 8 Hz
ITP > VTAP4 ----------------------|     |
                                        | --- ELSE ---
                                        |
NBAR <= (DSDRPM - V_NLOW) --------------| Pulse STO at 1 Hz
                                        |
                                        | --- ELSE ---
NBAR > (DSDRPM - V_NLOW) ---------|     |
                                  |AND -| Turn STO on continuous
NBAR < (DSDRPM + V_NHIGH) --------|     |
                                        | --- ELSE ---
                                        |
NBAR >= (DSDRPM + V_NHIGH) -------------| Pulse STO at 4 Hz
```

NOTE: VIP flags V_MODE_SETUP and STI_RESET are cleared on exit
  from the TP ADJUST MODE during the re-init procedure.

CHAPTER 29

CONTINUOUS TEST STRUCTURE

FILTERING LOGIC

OVERVIEW

Each fault to be detected and stored requires an event counter-timer which will be incremented by an "Up-count" value (calibratable) each time a fault is detected, and decremented by 1 each time the fault is not detected. Fault detection and up/down counting are done once per background loop. When the counter-timer for a particular fault exceeds a "threshold" value (calibratable) for that fault, the corresponding KAM fault code will be stored.

The "Wiggle Test" is a special case. Whenever the wiggle test mode is active and any one of the fault filters active during the wiggle mode exceeds its threshold (WIGLVL), STO is turned "on" (otherwise it will be "off").

PROCESS

STRATEGY MODULE: VC_FILTER_COM2

```
                          -----------
                        (    ENTER    )
                          -----------
                               |
                               v
                 NO       /---------\     YES
          |-------------/ FAULT (N) \-------------|
          |             \ DETECTED ?/             |
          v              \---------/              v
   !--------------!                        !--------------!
   ! CXXXFIL   =  !                        ! CXXXFIL  =   !
   ! CXXXFIL   -1 !                        ! CXXXFIL  +   !
   !--------------!                        ! CXXXUP       !
          |                                !--------------!
          |                                       |
          |------------->|<-----------------------|
                         |
                         v
                   /---------\
                  /   CXXXFIL \    YES
                 /     >       \----------------|
                 \    CXXXLVL  /                |
                  \     ?     /                 |
                   \---------/                  |
                         |  NO                  v
                         v              !---------------------!
                   -----------          ! STORE KAM CODE (N)  !
                 (     EXIT    )<-----! ZERO WARM-UP CYCLE  !
                   -----------          ! COUNTER (N)         !
                                        !---------------------!
```

FAULT THRESHOLD/UPCOUNT VALUE SELECTION "WIGGLE" VERSUS NORMAL CONTINUOUS TEST

OVERVIEW

Continuous Self-Test can operate in one of two modes, normal continuous and Wiggle. In normal continuous, the calibrated values for fault filter upcounts and thresholds (CxxxUP and CxxxLVL) are used to control the setting of service codes. During Wiggle mode, a value of 255 is used for all non-zero upcounts (except as noted) and the value of WIGLVL is used for all thresholds (except as noted).Also, STO will be activated at any time that a continuous fault is present. This will cause a STAR tester to output a tone whenever a failure is caused by a service technician manipulating the EEC harness and/or connectors. This is done to assist the diagnosis of intermittent harness problems.

Wiggle mode is entered when Self-Test In (STI) is grounded after initiating and exiting, or aborting, on demand self test (KOEO or KOER). This can be done by grounding, ungrounding, then re-grounding STI with the engine off or running. Or, wiggle will be entered after an on demand Self-Test (KOEO or KOER) is completed. If the vehicle is in gear when engine running (KOER) Self-Test is initiated, or the vehicle is placed in gear during engine running Self-Test; wiggle mode will be entered. See the Self-Test entry/exit logic for a complete description.

PROCESS

STRATEGY MODULE: VC_FAULT_COM1

```
WIGFLG = 1 -----------------------------| Wiggle test:
 (set in Self-Test entry logic)         | Set all non-zero upcounts to
                                        | 255 except as noted in test
                                        | descriptions
                                        | Set all continuous thresholds
                                        | to WIGLVL except as noted in
                                        | test descriptions
                                        |
                                        | --- ELSE ---
                                        |
                                        | Not in Wiggle
                                        | Use calibrated upcounts and
                                        | thresholds
                                        |
WIGFLG = 1 -----------------------|     |
 (in wiggle test)                 |AND -| Turn STO on
                                  |     |
CxxxFIL > CxxxLVL-----------------|     |
 (any fault filter greater              |
   than threshold)                      |
                                        |
                                        | --- ELSE ---
                                        |
WIGFLG = 1 -----------------------------| Turn STO off
 (in wiggle test)                       |
                                        |
 (all fault filters less                |
  than or equal to the                  |
   threshold)                           |
```

CONTINUOUS CODE PARAMETER NAMING CONVENTION

FAULT	FILTER	UPCOUNT	THRESHOLD	WARM-UP COUNTER	VECTOR RAM FLAG
FAULT DESCRIPTION	CXXXFIL	CXXXUP	CXXXLVL	CXXXCNT	CXXX_KAM_BIT

KAM CODE WARM_UP COUNTER/ERASE LOGIC

OVERVIEW

Each KAM code has a counter for "Number of Engine warm-ups" since the fault was last stored. The warm up counters are incremented once per each power up, only if a true warm up has occurred as described below.

Each individual code is erased when its counter is >= 80. Codes can also be manually cleared by ungrounding STI during Engine-Off code output mode.

DEFINITIONS

Calibration Constants:

- VECT3 = coolant temp. limit to trigger warm up counters (150 deg. F).
- VECT5 = starting coolant temp. for warm up counters (120 deg. F).

PROCESS

STRATEGY MODULE: VC_WARMUP_KAM_ERASE_COM1

WARM_UP COUNTER LOGIC

```
WARM_UP = 0 ----------------------|
                                  |
TCSTRT < VECT5 -------------------|
                                  |AND -|SET  WARM_UP = 1
ECT > VECT3 ----------------------|     | (the flag WARM_UP is
                                  |     |  initialized to 0 on
RUN MODE -------------------------|     |  power up and is never
 (CRKFLG=0, UNSP=0)                     |  cleared once a true
                                        |  warmup has occurred


STIFLG = 0 -----------------------|
                                  |     |INCREMENT ALL CODE
POWER_UP = 1 ---------------------|AND -| WARM-UP COUNTERS
 (set in RAM INIT.)               |     |
                                  |     |SET  POWER_UP  = 0
WARM_UP = 1 ----------------------|     |
```

KAM CODE ERASE LOGIC

```
WARM-UP COUNTER(N) >= 80 ---------------|ERASE KAM CODE(N)
                                        |
                                        | --- ELSE ---
                                        |
STO_WORKING = 1 ------------|           |
 (engine off code output    |           |
  mode*)                    |OR --|     |
                            |     |     |
STO_TRIGGER = 1 ------------|     |     |
                                  |AND -|ERASE ALL KAM CODES
NO_START = 1 ---------------------|     |
 (engine off test in process)     |     |
                                  |     |
STIFLG = 0 -----------------------|     |
 (self test not requested)
```

* Note: Includes the output of any codes in Engine Off Self Test; service codes, separator pulse, and continuous codes.

COOLING SYSTEM TEST

OVERVIEW

This module is designed to verify that the cooling system is controlling the engine temperature by monitoring the ECT input to determine whether the thermostat has opened. It is based upon the observation of a predicted temperature drop within a specific control range.

Certain entry conditions must be met to enable the testing process. A stabilization time is required to initialize the test. There is a predetermined starting point at which processing will begin. A time limit is allowed for the system to control engine temperature during which the determination is made that either: 1) there is a system fault; or 2) the system is controlling as expected. Once a decision is made the test will be bypassed for the rest of that power up.

The system faults are indicated by two service codes as listed below.

Condition:

1) code 338; cooling system is not heating (i.e. thermostat stuck open, low coolant level, very cold ambient temperature, etc.)

2) code 339; cooling system is not cooling (i.e. thermostat stuck closed, flow restriction within the system, coolant level/ condition or system operating pressure, etc.)

DEFINITIONS

INPUTS

Registers:

- C338CNT = Continuous code 338 warmup counter.
- C339CNT = Continuous code 339 warm up counter.
- C338FIL = Cont. code 338 register update, counts. Initialized to 0.
- C339FIL = Cont. code 339 register update, counts. Initialized to 0.

- V_ATMR2 = Time since ECT became greater than V_ECTCTMIN, seconds.

- V_ECTCTL = Register used in determining the delta temperature drop for setting the thermostat open flag, V_THMOPN, deg F. initialized to 254 deg. F.

- V_ECT_CTR = Counter which increments when the ECT is within the calibrated control range. Initialized to 0.

- V_ECTHI = Register used to load current ECT input as long as the value is increasing, deg. F. Initialized to 0.

Continued on Next Page

Continued from Previous Page

- OUTTMR = Counter used throughout VIP to limit test time, seconds. Initialized to 0.

Bit Flags:

- C338_KAM_BIT = Code 338 KAM bit used in service code output routine.

- C339_KAM_BIT = Code 339 KAM bit used in service code output routine.

- VATMR2_FLG = Flag used to count up V_ATMR2.

- V_CST_PASS = Flag used to bypass testing once a decision has been made that either there is a fault or the system is controlling within the temperature range, 1 = bypass. Initialized to 0.

- V_STABLFLG = Flag with is set upon stabilization time V_STABLTIM. Initialized to 0.

- V_THMOPN = Flag which indicates the thermostat has opened, 1 = opened. Initialized to 0.

- V_WARM_FLG = Flag which indicates the engine has warmed up, 1 = warm. Initialized to 0.

Calibration Constants:

- C338LVL = Continuous code 338 filter level. Base value = 250 cnts.

- C339LVL = Continuous code 339 filter level. Base value = 250 cnts.

- C338UP = Continuous code 338 filter upcount. Base value = 255 cnts.

- C339UP = Continuous code 339 filter upcount. Base value = 255 cnts.

- V_CSTE_SW = Cooling System Test enable. 1 = enable, 0 = disable.

- V_ECT_DEL = Delta temperature drop required which indicates that the thermostat has opened. Base value = 10 deg. F.

- V_ECT_LIM = Number of valid ECT readings within the control range before the system is considered warmed up. Base value = 20.

- V_ECTCTMAX = ECT high limit within the control range. Base value = 220 deg. F.

- V_ECTCTMIN = ECT low limit within the control range. Base value = 170 deg. F.

Continued on Next Page

Continued from Previous Page

- V_ECT_TIME = Time allowed for engine warmup from V_ECTCTMIN. Base value = 240 secs.

- V_STABLTIM = Time allowed for cooling system stabilization after run mode is entered. Base value = 30 secs.

- V_TIME_LIM = Total time allowed for engine warmup from run mode. Base value = 1200 secs.

OUTPUTS

Registers:

- V_ECTCTL = See above

- V_ECT_CTR = See above

_ V_ECTHI = See above

Bit Flags:

- V_CST_PASS = See above

- V_STABLFLG = See above

- V_THMOPN = See above

- V_WARM_FLG = See above

PROCESS

STRATEGY MODULE: VC_COOLING_SYS_COM1

TEST ENTRY CONDITIONS:

```
V_CSTE_SW = 0 ----------|
 (Test not enabled)     |
C118FIL > 0 ------------|       | V_CST_PASS = 1
 (ECT sensor fault)     |       | (bullet proof to prevent
C117FIL > 0 ------------|OR ----|  storing a false code).
 (ECT sensor fault)     |       | EXIT: VC_COOLING_SYS_TEST
V_CST_PASS = 1 ---------|       |
 (Test done this pwrup) |       | --- ELSE ---
                                |
                                | DO: STABILIZATION CHECK
                                |
```

BEGIN: STABILIZATION CHECK

```
V_STABLFLG = 1 -----------------| DO: STARTING POINT CHECK
                                |
                                | ---ELSE---
                                |
                                | OUTTMR = 0
                                | V_STABLFLG = 1
ATMR1 >OR= V_STABLTIM ----------| V_ECTHI = ECT
 (Stabilization time)           | EXIT: VC_COOLING_SYS_TEST
                                |
                                | --- ELSE ---
                                |
                                | EXIT: VC_COOLING_SYS_TEST
```

END: STABILIZATION CHECK

BEGIN: STARTING POINT CHECK

```
ECT >OR= V_ECTCTMIN ------------| DO: ECT MONITORING LOGIC
 (ECT above control min.)       | DO: COOLING SYSTEM FUNCTION LOGIC
                                |
                                | ---ELSE---
                                |
                                | DO: COOLING SYSTEM FUNCTION LOGIC
```

END: STARTING POINT CHECK

BEGIN: ECT MONITORING LOGIC

```
ECT >OR= V_ECTHI ----------------| V_ECTHI = ECT
 (Coolant Temp. rising)          |
                                 | --- ELSE ---
ECT < V_ECTHI -------------|     |
 (Coolant temp. falling)   |     |
ECT < V_ECTCTL ------------|     |
 (Sys. controlling temp.)  |AND--| V_ECTCTL = ECT
ECT > V_ECTCTMIN ----------|     | (Note: on power up V_ECTCTL
 (ECT above control min.)  |     |  is initialized to 254 deg. F)


V_WARM_FLG = 1 ------------------|
 (Engine has warmed up)          |      |
V_THMOPN = 0 --------------------|AND---| V_THMOPN = 1
 (Thermostat not open yet)       |      |  (Thermostat has opened)
V_ECTHI - V_ECTCTL > V_ECT_DEL---|      |
 (ECT has dropped required delta)|


V_ECT_CTR >OR= V_ECT_LIM---|            |
 (Valid ECT readings limit)|AND--|      |
V_ATMR2 >OR= V_ECT_TIME----|     |OR----| V_WARM_FLG = 1
 (Time allowed for warmup)       |      |  (Warmup occurred at some
                                 |      |   previous time)
ECT >OR= V_ECTCTMAX -------------|      |
 (ECT above control max.)


ECT > V_ECTCTMIN ----------------|
 (ECT above control min.)        |      |
                                 |AND---| V_ECT_CTR = V_ECT_CTR + 1
ECT < V_ECTCTMAX ----------------|      |  (ECT within control range;
 (ECT below control max.)               |    increment counter,clip
                                             at 255)
```

END: ECT MONITORING LOGIC

NOTE: V_ATMR2 starts running based on the calibration constant
      V_ECTCTMIN.

BEGIN: COOLING SYSTEM FUNCTION LOGIC

```
V_WARM_FLG = 0 --------------|      | V_CST_PASS = 1
 (Engine not warm yet)       |      | Set:  ERROR_DETECTED
                             |AND--| Cooling system is not heating
                             |      | Do:  FAULT FILTERING C338
OUTTMR >OR= V_TIME_LIM-------|      | (Note: calibrate fault filter
 (Total time allowed to warmup)     |  to store code in one upcount).
                                    |
                                    | ---ELSE ---
V_WARM_FLG = 1----------|           |
 (warmup completed)     |           | V_CST_PASS = 1
                        |           | Cooling system is not cooling
V_THMOPN = 0------------|AND-------| Set:  ERROR_DETECTED
 (Thermostat not open)  |           | Do:  FAULT FILTERING C339
ECT > V_ECTCTMAX--------|           | (Note: calibrate fault filter
 (ECT above control limit)          |  to store code in one upcount).
                                    |
                                    | ---ELSE---
                                    |
V_WARM_FLG = 1 ---------|           | V_CST_PASS = 1
 (Warmup completed)     |AND-------| (Cooling sys. is controlling)
V_THMOPN = 1------------|           | EXIT: VC_COOLING_SYS_TEST
 (Thermostat open)                  | Do: FAULT FILTERING C338 and C339
                                    |
                                    | ---ELSE---
                                    |
                                    | EXIT: VC_COOLING_SYS_TEST
```

END: COOLING SYSTEM FUNCTION LOGIC

TIMER CONTROL LOGIC
(Included within this test module)

V_ATMR2 - TIME SINCE ECT BECAME GREATER THAN V_ECTCTMIN

```
ECT > V_ECTCTMIN -----|             | VATMR2_FLG = 1
                      |AND --------| V_ATMR2 = 0
VATMR2_FLG = 0 -------|             |
                                    | ---ELSE---
                                    |
VATMR2_FLG = 1 --------------------| COUNT UP V_ATMR2
                                    |
                                    | ---ELSE---
                                    |
                                    | V_ATMR2 = 0
```

ECT OPEN/SHORT TESTS

PROCESS

STRATEGY MODULE: VC_ECT_COM1

```
       -----------
     (    ENTER    )
       -----------
            |
            v
       /---------\   NO      !----------!
      /  IECT >=  \--------->! FAULT 117!
      \  ECTMIN ? /          !----------!
       \---------/                 |
            |  YES                 |
            |<---------------------|
            v
    !--------------!
    !  DO FAULT    !
    ! FILTERING 117!
    !--------------!
            |
            v
       /---------\   NO      !----------!
      /  IECT <=  \--------->! FAULT 118!
      \  ECTMAX ? /          !----------!
       \---------/                 |
            |  YES                 |
            |<---------------------|
            v
    !--------------!
    !  DO FAULT    !
    ! FILTERING 118!
    !--------------!
            |
            v
       -----------
     (     EXIT    )
       -----------
```

ACT SENSOR TEST

PROCESS

STRATEGY MODULE: VC_ACT_COM1

```
                -----------
              (    ENTER    )
                -----------
                     |
                     v
                /---------\   NO      !----------!
               /  IACT >=  \--------->! FAULT 112!
               \  ACTMIN ? /          !----------!
                \---------/                |
                     |  YES                |
                     |<--------------------|
                     v
             !--------------!
             !   DO FAULT   !
             ! FILTERING 112!
             !--------------!
                     |
                     v
                /---------\   NO      !----------!
               /  IACT <=  \--------->! FAULT 113!
               \  ACTMAX ? /          !----------!
                \---------/                |
                     |  YES                |
                     |<--------------------|
                     v
             !--------------!
             !   DO FAULT   !
             ! FILTERING 113!
             !--------------!
                     |
                     v
                -----------
              (     EXIT    )
                -----------
```

TOT OPEN/SHORT TESTS

PROCESS

STRATEGY MODULE: VC_TOT_COM2

```
    -----------
  (    ENTER    )
    -----------
         |
         v
    /---------\   NO      !----------!
   /  ITOT =>  \--------->! FAULT 638!
   \  TOTMIN ? /          !----------!
    \---------/                 |
         |  YES                 |
         |<---------------------|
         v
 !--------------!
 !  DO FAULT    !
 ! FILTERING    !
 !--------------!
         |
         v
    /---------\   NO      !----------!
   /  ITOT <=  \--------->! FAULT 637!
   \  TOTMAX ? /          !----------!
    \---------/                 |
         |  YES                 |
         |<---------------------|
         v
 !--------------!
 !  DO FAULT    !
 ! FILTERING    !
 !--------------!
         |
         v
    -----------
  (     EXIT    )
    -----------
```

MAP SENSOR TEST

PROCESS

STRATEGY MODULE: VC_MAP_COM1

```
     -----------
   (    ENTER    )
     -----------
          |
          v
     /---------\   NO
    / MAPTMR <= \----------------|
    \ VMPMAX ?  /                |
     \---------/                 |
          |  YES                 |
          v                      v
     /---------\   NO      !----------!
    /   VMDEL5  \--------->! FAULT 126!
   / <= MDELTA<= \         !----------!
   \   VMDEL6 ?  /               |
    \           /                |
     \---------/                 |
          |  YES                 |
          |<---------------------|
          v
  !--------------!
  !   DO FAULT   !
  ! FILTERING 126!
  !--------------!
          |
          v
     -----------
   (     EXIT    )
     -----------
```

MAP SENSOR VACUUM CIRCUIT TEST

OVERVIEW

The purpose of this test is to check the vacuum circuit integrity of the map sensor in speed density systems.

The test philosophy is such that a required difference between BP and MAP should be observed. This difference is evaluated each background loop and if present, normal down fault is executed.

Operating modes or situations that prevent this change would be:

1) Cranking/Stall
2) WOT or,
3) MAP vacuum hose disconnect. (MAP = BP)

In order to isolate the disconnect, testing is conducted in Continuous Self Test and is restricted to closed throttle and run mode where the BP-MAP change is greatest.

TP and MAP electrical integrity must be assured and checked because they are essential inputs to the testing conditions. If either input has failed, the test is immediately exited.

The test is further restricted so that updated BP is greater than about 20" (V_BPMIN). This is to prevent executing the test due to low updated BP at altitude which may cause the MAP difference to be calculated low. In addition, a time since last PIP limit (V_PIPMAP_LMT) parameter is utilized to abort testing if a stall is imminent. This is to preclude false failure recognition.

If all testing requirements are met and the BP-MAP change is less than V_MAPDIF i.e. 2", then a fault is indicated and up filtering is executed. When the fault filter exceeds the calibrated threshold, an error code (128*/81**) is stored in KAM, and a fault flag (V_VACFLG) is set and passed to base strategy to be used for activating MAP MFMFLG.

Continuous checks are made on the fault filter to ascertain whether or not a fault is present in order to set or clear the failure flag.

The test may be calibrated out if desired by setting V_MAPFLG = 0.

* 3-digit codes
** 2-digit codes

DEFINITIONS

Self Test Registers:

- V_IMAP = Stored MAP after MAP_WORD calculation.
- C126*/22**FIL = Error C126/22 fault filter.
- C128*/81**CNT = Warm-up counter for Fault 128/81.
- C128*/81**FIL = Fault filter for error code 128/81.

Self Test Calibration Constants:

- V_MAPDIF = Required minimum MAP change.
- V_PIPMAP_LMT = Max TSLPIP to do MAPVAC test.
- V_MAPFLG = MAP vacuum circuit test enable flag 1 = enable.
- V_BPMIN = Minimum BP required to do test.
- C128*/81**LVL = Threshold level for fault 128/81.
- C128*/81**UP = Upcount for fault 128/81.

Self Test Flags:

- WIGFLG = Flag to indicate wiggle mode. 1 = enable.
- V_VACFLG = Error flag passed to base strategy to indicate vacuum circuit failure. 1 = failure.
- ERROR_DETECTED = Flag passed to fault filter routine indicating self test detected a failure.

* 3-digit codes
** 2-digit codes

Base Strategy Registers:

- BP = Inferred BP used by control strategy.

- APT = Throttle mode.

Base Strategy Flags:

- UNDSP = Underspeed flag.

- TFMFLG = TP FMEM flag.

Base Strategy Calibration Constants:

- FILHYS = FMEM filter count hysteresis.

NOTE: V_MAPFLG is set to 0 on Power-up.

PROCESS

STRATEGY MODULE: VC_MAPVAC_COM1

```
V_MAPFLG = 0 ---------------------|
 (test cal'd out)                 |OR --| DO: Fault Filter C128*/81**
                                  |     | DO: Failure Mode Logic
BP-IMAP_WORD => V_MAPDIF ---------|     |
 (MAP difference indicated)             |
                                        |
                                        | --- ELSE ---
                                        |
WIGFLG = 0 -----------------------|     |
 (not in Wiggle Mode)             |     |
                                  |     |
UNDSP = 0 ------------------------|     |
 (Run mode)                       |     |
                                  |     |
TFMFLG = 0 -----------------------|     |
 (TP ok)                          |     |
                                  |     |
C126*FIL = 0 ---------------------|AND -| SET: ERROR_DETECTED
 22**                             |     | DO: FAULT FILTER C128*/81**
 (MAP elect ok)                   |     | DO: FAILURE MODE LOGIC
                                  |     |
TSLPIP <= V_PIPMAP_LMT -----------|     |
 (RPM high enough)                |     |
                                  |     |
APT = -1 -------------------------|     |
 (closed throttle)                |     |
                                  |     |
BP > V_BPMIN ---------------------|     |
 (min BP req'd)                   |     |
                                  |     |
BP-IMAP_WORD < V_MAPDIF ----------|     |
 (req'd MAP diff)                       |
                                        |
                                        |  --- ELSE ---
                                        |
V_VACFLG = 0 ---------------------------| SET: C128*FIL = 0
 (error flag set)                       |       81**
                                        |
                                        | EXIT: MODULE
                                        |
                                        |  --- ELSE ---
                                        |
                                        | EXIT: MODULE
```

* 3-Digit code
** 2-digit code

FAILURE MODE LOGIC:

```
C128*FIL > C128*LVL --------------------| SET: V_VACFLG = 1
  81**       81**                       |
 (failure indicated)                    |
                                        | --- ELSE ---
                                        |
C128*FIL <= C128*LVL - FILHYS ----------| SET: V_VACFLG = 0
  81**        81**                      |
 (failure removed)                      |
                                        | --- ELSE ---
                                        |
                                        | EXIT
```

* 3-digit code
** 2-digit code

SONIC EGR SYSTEM TEST

OVERVIEW

In the documentation, this module is enabled via the lower case parameters egr_system and sonic_egr from VC_xxxx_SEL_COMn. In the software, a similar parameter may be used or the test may be enabled based on whatever combinations of PFEHP, PFEHP_FG, EGRHP_FG, etc. are available.

When the test is entered, the IEGR input is checked to insure that it is within the max/min limits. If this test fails, the appropriate fault filter is upcounted and no further testing is required. If the voltage is within limits, it is then checked to insure that it is not below the lowest voltage expected from a closed valve. Note that these checks are bypassed until EGR is enabled for the first time.

Additional checks are made when the EGR is commanded off to insure that the valve is returning to the closed position. When the EGR is on (high duty cycle), a check is made (if the vacuum is high enough) to see if the valve has moved sufficiently from the closed position.

Lowercase parameters such as sonic_egr_test_ena, sonic_egr_off_test, etc. are used to control flow through the ladder diagrams. The use of these parameters in the documentation does not imply that similar parameters are required in the software.

Note that the flags V_EGR_STK_ON and, V_EGR_ON_CR are used to reset any fault filters that have partially counted up and control timers when a particular test mode is first entered.

DEFINITIONS

INPUTS

Registers:

- BP = Barometric pressure. (note: Upper byte of BP_WORD).
- C332FIL = EGR valve opening not detected (Sonic, PFE) fault filter
- C334FIL = EVP voltage above closed limit (sonic) fault filter.
- EGRDC = EGR duty cycle.
- IEGR = EGR sensor input.
- MAP = Manifold absolute pressure BIN 3.
- V_EGR_RDY = Continuous EGR test ready flag (Latched)
- VIP_TIMER_EX = VIP state timer.

Bit Flags:

- V_EGR_STK_ON = Indicates that the EGR off continuous test is in progress.
- WIGFLG = Indicates VIP wiggle test.

Calibration Constants:

- C332LVL = EGR valve opening not detected (sonic/PFE) threshold.
- C334LVL = EVP voltage above closed limit (sonic) threshold.
- EVPMAX = Maximum EVP reading (open) units are counts.
- EVPMIN = Minimum EVP reading (short) units are counts.
- VCRTDC = GR cruise test duty cycle limit, percent.
- VEGVAC = Minimum manifold VAC for cruise test units are inches hg.
- V_EGR_CTMR = EGR cruise test timer limit.
- VEITMR = EGR idle test timer limit units are seconds.
- V_EGR_ON_CR = Indicates that the EGR on cruise test (test for flow) was in progress last background loop.
- VEVPCL = EVP cruise test limit units are counts.
- VEVPHL = EVP high limit (valve closed) units are counts end of essential ordering of constants.
- VEVPLL = EVP low limit (valve closed) units are counts

PROCESS

STRATEGY MODULE: VC_EGR_SON_SD_COM10

Step one (if sonic EGR is present) is to test for an input voltage that is out of range. If an open or short circuit is detected, the test is complete. Otherwise, the test continues to check for flow or a stuck open valve.

!!
NOTE:
EFMLO is passed to the fault filter routine every time it is called for code 327 and returns value of 1 if the code is set and returns a value of 0 if the code is clear. Similarly, EFMHI is passed with code 337. EFMHI and EFMLO are used to set and clear EFMFLG. This logic is described in the EGR chapter.

!!

```
egr_system <> sonic_egr ----------------| sonic_egr_test_ena := 0
 (parameters egr_system                 |  (no sonic EGR hardware, no
  and sonic_egr from the                |   testing)
  VC_xxxx_SEL_COMn module)              |
                                        | --- ELSE ---
IEGR <= EVPMIN ------------------|      |
 (voltage below minimum)         |AND -| error_detected := 1
                                 |      | Fault Filter Code 327
V_EGR_RDY = 1 --------------|    |      |  (upcount fault)
 (EGR has been              |    |      |  (see note above)
  enabled)                  |OR --|     |
                            |           | Fault Filter Code 337
WIGFLG = 1 -----------------|           |  (see note above)
 (Wiggle mode - allow testing)          | Fault Filter Code 328
                                        |  (downcount faults not present)
                                        |
                                        | V_EGR_ON_CR := 0
                                        | V_EGR_STK_ON := 0
                                        |  (clear flags - modes not to be
                                        |   active)
                                        |
                                        | sonic_egr_test_ena := 0
                                        |  (done testing)
                                        |
                                        | --- ELSE ---
```

(continued on next page)

(continued from previous page)

```
                                             |
IEGR >= EVPMAX --------------------|      |
 (voltage above maximum)           |      |
                                   |AND -| error_detected := 1
V_EGR_RDY = 1 --------------|      |      | Fault Filter Code 337
 (EGR has been              |      |      |  (upcount fault)
  enabled)                  |OR --|      |  (see note above)
                            |             |
WIGFLG = 1 -----------------|             | Fault Filter Code 327
 (Wiggle mode - allow testing)            |  (see note above)
                                          | Fault Filter Code 328
                                          |  (downcount faults not present)
                                          |
                                          | V_EGR_ON_CR := 0
                                          | V_EGR_STK_ON := 0
                                          |  (clear flags - modes not to be
                                          |   active)
                                          |
                                          | sonic_egr_test_ena := 0
                                          |  (done testing)
                                          |
                                          | --- ELSE ---
IEGR < VEVPLL ---------------------|      |
 (voltage below closed             |      |
  valve voltage)                   |AND -| error_detected := 1
                                   |      | Fault Filter Code 328
WIGFLG = 0 ------------------------|      |  (upcount fault)
                                          |
                                          | Fault Filter Code 327
                                          | Fault Filter Code 337
                                          |  (downcount faults not present)
                                          |
                                          | sonic_egr_test_ena := 1
                                          |  (allow further testing)
                                          |
                                          | --- ELSE ---
                                          |
                                          | Fault Filter Code 327
                                          | Fault Filter Code 337
                                          | Fault Filter Code 328
                                          |  (no faults, downcount)
                                          |
                                          | sonic_egr_test_ena := 1
                                          |  (allow further testing)
```

To get here, sonic_egr_test_ena must be set by the previous diagram (sonic hardware present, no open or short fault). This diagram determines which EGR mode (on or off) is to be tested. For whichever mode is enabled, the flags that control action in the other mode are cleared.

```
sonic_egr_test_ena := 1 ----------|
 (testing allowed                 |
  by previous diagram)            |AND -| V_EGR_ON_CR := 0
                                  |     | sonic_egr_on_test := 0
EGRDC = 0 ------------------------|     |  (EGR on test not active)
 (EGR not on)                           |
                                        | sonic_egr_off_test := 1
                                        |  (perform test for valve stuck
                                        |   open)
                                        |
                                        | --- ELSE ---
                                        |
sonic_egr_test_ena := 1 ----------------| V_EGR_STK_ON := 0
 (testing allowed                       | sonic_egr_off_test := 0
  by previous diagram; EGR is on)       |  (EGR off test not active)
                                        |
                                        | sonic_egr_on_test := 1
                                        |  (perform test for valve opening)
                                        |
                                        | --- ELSE ---
                                        |
                                        | sonic_egr_off_test := 0
                                        | sonic_egr_on_test := 0
                                        |  (no more testing)
```

Test to see if this is the first pass through the EGR off test. If so, reset the timer VIP_TIMER_EX and clear out the fault filter for code 334. If it is not the first pass, continue on. Of course, if the EGR off test is not enabled, no action is required.

```
V_EGR_STK_ON = 0 -----------------|
 (last pass was not in            |
  EGR off test)                   |AND -| V_EGR_STK_ON := 1
                                  |     |  (this is the first background
sonic_egr_off_test = 1 -----------|     |   loop in this test mode)
 (mode enabled)                         |
                                        | VIP_TIMER_EX := 0
                                        |  (reset for time delay before
                                        |   testing)
                                        |
                                        | Do: EGR Off Fault Filter Reset
                                        | Do: EGR Off Test
                                        |
                                        | --- ELSE ---
                                        |
sonic_egr_off_test = 1 -----------------| Do: EGR Off Test
 (mode enabled)
```

BEGIN: EGR Off Test

This diagram, which follows the previous, tests to determine if the input voltage is too high for a closed valve (valve not closing). If it is, code 334 may be set after the time delay.

```
IEGR > VEVPHL --------------------|
 (voltage above closed value)     |
                                  |
VIP_TIMER_EX >= VEITMR -----------|AND -| error_detected := 1
 (mode enabled)                   |     | Fault Filter Code 334
                                  |     |  (upcount fault filter)
WIGFLG = 0 -----------------------|     |
 (not in wiggle mode)                   |
                                        | --- ELSE ---
                                        |
IEGR <= VEVPHL -------------------------| Fault Filter Code 334
 (voltage OK)                           |  (downcount fault filter)
                                        |
                                        | --- ELSE ---
                                        |
                                        | no action
                                        |  (mode not active or fault present
                                        |   but not timed out yet)
```

END: EGR Off Test

This test (enabled by sonic_egr_on_test = 1) checks to see if the conditions are right to test the EGR valve for opening. This requires that there is adequate vacuum and the EGR duty cycle be high to indicate that the EEC is trying to open the valve. Alternately, if code 332 is already set, the test will be enabled to allow the system to clear the code (and the MIL light) if the failure disappears.

```
sonic_egr_on_test <> 1 -----------------| no action
 (do not perform test                   |
   for valve opening)                   |
                                        | --- ELSE ---
C332FIL > C332LVL ----------------|      |
 (Code set, bypass                |      |
  entry conditions)               |OR --| Do: Test for EGR Flow
                                  |      |  (perform check to see
BP - MAP > VEGVAC ----------|     |      |   if valve opening can be
 (enough muscle vacuum)     |AND -|      |   detected)
                            |            |
EGRDC > VCRTDC -------------|            |
 (trying to open EGR valve)              |
                                         | --- ELSE ---
                                         |
                                         | V_EGR_ON_CR = 0
                                         |  (can't run test yet because
                                         |   vacuum / duty cycle entry
                                         |   conditions have not been met)
```

BEGIN: Test for EGR FLow

At this point, the vacuum and EGR duty cycle (or code 332) requirements have been met to determine if valve opening can be detected. A check is made to see if this is the first pass through the test - if it is, C332FIL may be reset. Otherwise, the test continues on.

```
V_EGR_ON_CR = 0 -----------------------| V_EGR_ON_CR := 1
 (not in this mode last                |  (set up to skip this step next
  background pass)                     |   pass)
                                       |
                                       | VIP_TIMER_EX := 0
                                       |  (start up timer)
                                       |
                                       | Do: EGR On Fault Filter Reset
                                       |
                                       | --- ELSE ---
                                       |
                                       | no action
                                       |  (continue to next part of test)
```

This is the test to determine if the EGR valve has opened. At this point all of the conditions should be right to insure that the valve should be opened a significant amount. If an adequate opening is not detected for a period of time, a fault is assumed.

```
IEGR <= VEVPCL -------------------|
 (open valve not observed)        |
                                  |
VIP_TIMER_EX >= V_EGR_CTMR -------|AND -| error_detected := 1
 (enough time in test)            |     | Fault Filter Code 332
                                  |     |  (upcount fault filter)
WIGFLG = 0 -----------------------|     |
 (not in wiggle mode)                   |
                                        | --- ELSE ---
                                        |
IEGR > VEVPCL --------------------------| Fault Filter Code 332
 (voltage OK.)                          |  (downcount fault filter)
                                        |
                                        | --- ELSE ---
                                        |
                                        | no action
                                        |  (mode not active or fault present
                                        |   but not timed out yet)
```

END: Test for EGR FLow

BEGIN: EGR Off Fault Filter Reset

This is called on the first entry into the EGR off test to allow resetting the fault filter if it has yet set it's code to avoid "pumping up" the filter and setting a fault if the mode is repeatedly exited and entered.

```
C334FIL <= C334LVL ---------------------| C334FIL := 0
 (code 334 not set)                     |  (Get fresh start on fault filter)
                                        |
                                        | --- ELSE ---
                                        |
                                        |  (no action)
```

END: EGR Off Fault Filter Reset

BEGIN: EGR On Fault Filter Reset

This is called on the first entry into the EGR on test to allow resetting the fault filter if it has yet set it's code to avoid "pumping up" the filter and setting a fault if the mode is repeatedly exited and entered.

```
C332FIL <= C332LVL ---------------------| C332FIL := 0
 (code 332 not set)                     |  (Get fresh start on fault filter)
                                        |
                                        | --- ELSE ---
                                        |
                                        |  (no action)
```

END: EGR On Fault Filter Reset

EGR SYSTEM SELECTION (SONIC)

OVERVIEW

This module determines if EGR is or is not present based on PFEHP. If there is no EGR, the FMEM flags and Code 332 are set to zero for bullet proofing. It also sets the value of egr_system to enable/disable the Sonic test if PFEHP is not equal to two.

By putting this logic in a separate module outside the actual EGR tests, the total number of unique strategy modules required to document the tests for the various combinations of EGR types along with the variations on PFEHP, PFEHP_FG, no PFEHP, etc. are reduced. Note that the use of lower case parameters (egr_system, pfe_egr) does not imply that a parameter be created in the software - PFEHP can still be used in the assembly code if the same functionality is maintained.

DEFINITIONS

Registers:

- C332FIL = EGR valve opening not detected (Sonic, PFE) fault filter.

Bit Flags:

- EFMHI = EGRHI FMEM flag.

- EFMLO = EGRLO FMEM flag.

Calibration Constants:

- PFEHP = Switch to select EGR strategy; 2 -> do not use any EGR, 1 -> use PFE strategy, 0 -> use sonic strategy.

PROCESS

STRATEGY MODULE: VC_SON_SEL_COM2

```
unconditionally ------------------------| sonic_egr := 0
                                        | no_egr := 2
                                        |  (supply numeric value because
                                        |   symbolic names are not part of
                                        |   this language. Note that there is
                                        |   no real reason to actually look
                                        |   at this diagram since the numeric
                                        |   values are not actually used
                                        |   anywhere)


PFEHP <> 0 -----------------------------| EFMLO := 0
                                        | EFMHI := 0
                                        | C332FIL := 0
                                        |  (No EGR system, therefore no EGR
                                        |    failure)
                                        |
                                        | egr_system := no_egr
                                        |  (Done testing EGR system, exit)
                                        |
                                        | --- ELSE ---
                                        |
                                        | egr_system := sonic_egr
                                        |  (continue to EGR tests; this
                                        |   parameter is used to enable
                                        |   EGR test)
```

IGNITION DIAGNOSTIC MONITOR (IDM)

OVERVIEW:

The continuous PIP/IDM routine basically checks if time since last PIP (TSLPIP) and time since last IDM (TSLIDM) have exceeded a calibrated timeout period.

Decisions are made in software to assure the engine is running and stabilized before the test is executed. RPM is compared against VLORPM and if greater, entry into the PIP/IDM test is permitted.

The PIP/IDM test utilizes free-running timers for processing PIP and IDM and high speed digital inputs are used to re-start the timers.

Each transition of pip starts a new time-out function. When the time since last pip > VPIPTM, a pip fault is present.

If a pip fault has been detected, software bypasses the IDM test for a calibrated time period VIDMST. This is to insure sufficient time for the TFI module to calculate a TACH output once PIPS have been restored.

The IDM test is similar to the PIP test in that each transition of IDM also starts a time-out function. When the time since last IDM is > VIDMTM, an IDM fault is present.

Both PIP and IDM tests use traditional fault filtering.

EOS_IDM MODULE

The following logic describes the input processing which occurs when an IDM transition (IDM_INT = 1) occurs:

NOTE: IDM_HIGH reflects the state of the High Speed Input (HSI) pin and not the IDM voltage. Because of an inversion, when IDM voltage = 0, IDM_HIGH = 1, and when IDM voltage is greater than 3.5 volts, IDM_HIGH = 0.

```
ALWAYS -----------------------| Clear IDM_INT

IDM_HIGH = 0 -----------------| Set NEW_IDM
```

PIP/IDM LOGIC (BACKGROUND)(FOR TACH BUFFER) PROCESS

STRATEGY MODULE: VC_PIP_IDM_COM2

```
          -----------       /---------\ YES   !---------!
        (    ENTER    )--->/  IS N  <  \----->!   SET   !
          -----------      \  VLORPM ? /      !TSLIDM =0!
                            \---------/       !---------!
                 |<-----------------|NO            |
                 |                                 v
                 |                                ----
                 |                              ( EXIT )
                 |                                ----
                 |
                 | NO
       !----------------!
       ! CLEAR ERROR    !--------------------|
       !DETECTED FLAGS  !                    |
       !----------------!                    |
                                             |
 !----------------------------!              |
 !.SET ERROR DETECTED FLAG=1  !              v
 !.SET IDM_BYPASS FLAG=1      !   YES  /---------\
 !.RESET C212FIL = 0          !<------/  TSLPIP   \
 !.RESET TSLIDM =0            !       \ > VPIPTM ?/
 !----------------------------!        \---------/
                 |                           | NO
                 |-------------------------->|
                                             v
                                      !------------!
                                      ! CALL FAULT !
                                      ! FILTER C211!
                                      !------------!
                                              |
                                              v
!-----------!  YES /---------\    NO /-------------\ YES
!.SET ERROR !<----/  TSLIDM   \<----/ IS IDM_BYPASS \----|
! DETECTED  !     \ > VIDMTM ?/     \   FLAG =1?    /    |
! FLAG =1   !      \---------/       \-------------/     |
!-----------!           | NO                             |
      |                 |                     YES /------------\
      |---------------->|                    |---/ IS TSLIDM >  \
                        |                    |   \   VIDMST ?   /
                        |                    |    \------------/
                        |                    v           | NO
                        v           !-------------------!  |
             !-------------------!  !.SET TSLIDM=0      !  |
             !.CALL FAULT FILTER!   !.CLEAR IDM_BYPASS  !  |
             !IDM_FAULT_CNT C212!   ! FLAG              !  |
             !-------------------!  !-------------------!  |
                        |                    |             |
                   -----------               v             |
                 (    EXIT    )<-------------------------|
                   -----------
```

NOTE: Pip fault filtering uses C211LVL.
IDM fault filtering uses C212LVL.

CONVERTER CLUTCH VALIDITY TEST
PROCESS
STRATEGY MODULE: VC_CONVERTER_CLUTCH_COM1

```
   ------         /---------\YES    ----
 ( ENTER  )---->/WIGFLG = 1\--->( EXIT )
   ------       \           /       ----
                 \---------/
      |---------------|NO
      v
 /--------\ NO
/TSTRAT=4  \----------------------|
\          /                      |
 \--------/                       |
      | YES                       |
      v                           |
 /--------\ NO      !---------!   |
/CC_ERROR  \------->!FAULT 628!   |
\   = 0    /        !---------!   |
 \--------/              |        |
      |YES               v        |
      |<--------------------------|
!----------!
!DO FAULT  !
!FILTERING !
!   628    !
!----------!
      |
      |
      v
  -------
(  EXIT   )
  -------
```

ELECTRONIC PRESSURE CONTROL SOLENOID TEST
PROCESS

STRATEGY MODULE: VC_EPC_SOLENOID_COM2

```
          ------
        ( ENTER  )
          ------
            |
            v
      /----------\NO
     / TSTRAT = 4 \---------------------|
     \     ?      /                     |
      \----------/                      |
            |YES                        |
            |                           |
            v                           |
      /----------\YES      !----------!  |
     / ETV_ERROR  \------>!FAULT 624 !  |
     \   = 1 ?    /       !----------!  |
      \----------/             |        |
            |NO                v        |
            |<--------------------------|
            |
            v
    !----------------!
    !DO FAULT FILTER-!
    !   ING 624      !
    !----------------!
            |
            v
          ------
        (  EXIT  )
          ------
```

TP SENSOR TEST

PROCESS

STRATEGY MODULE: VC_TPS_COM1

```
           -----------
         (    ENTER    )
           -----------
                |
                v
           /---------\   NO      !----------!
          / ITP >=    \--------->! FAULT 122!
          \ TAPMIN ?  /          !----------!
           \---------/                 |
                |  YES                 |
                |<---------------------|
                v
        !--------------!
        !   DO FAULT   !
        ! FILTERING 122!
        !--------------!
                |
                v
           /---------\   NO      !----------!
          / ITP <=    \--------->! FAULT 123!
          \ TAPMAX ?  /          !----------!
           \---------/                 |
                |  YES                 |
                |<---------------------|
                v
        !--------------!
        !   DO FAULT   !
        ! FILTERING 123!
        !--------------!
                |
                |
                v
           -----------
         (     EXIT    )
           -----------
```

VEHICLE SPEED SENSOR TEST

OVERVIEW

The Vehicle Speed Sensor (VSS) Test monitors the input for VSBAR = or > VSSMN1. When there is not sensor input and the parameters that infer the vehicle is moving are true, [FN689V (speed, load (TP)), and indicators of transmission in gear with brake not applied], a timer, (VSSTMR), is incremented. After enough settling time has elapsed, (VSSTMR = or > VSSTIM), it is assumed the input is not working. The fault flag (VSFMFLG), which is used by control strategy for shift schedule determination, is set or cleared based on three independent checks, 1) prior to continuous VIP entry if a fault code is stored in KAM the flag is set, otherwise it is cleared 2) during continuous VIP if the fault filter exceeds the level the flag is set, otherwise 3) any time the input VSBAR is equal to or greater than the minimum, VSSMN1, the flag is cleared.

DEFINITIONS

Registers:

- C452FIL = Insufficient input from Vehicle Speed Sensor (VSS) fault filter.

- NEBART = Filtered Engine RPM For Transmission.

- VSBAR = Filtered vehicle speed.

- VSSTMR = Vehicle speed sensor test timer.

Bit Flags:

- BIFLG = Brake on flag.

- NDSFLG = Neutral/drive flag; 1 -> drive.

- TFMFLG = TP FMEM flag.

- VIP_ENABLE = VIP enable flag.

- VSFMFLG = Vehicle speed sensor FMEM flag.

Calibration Constants:

- BIHP = BRAKE INPUT "Hardware Present" indicator; 0 -> NO, 1 -> YES.

- C452LVL = Insufficient input from Vehicle Speed Sensor(VSS) threshold.

- FN689V = Minimum engine speed at a given TP to test VSS input.

- VSSMN1 = Maximum vehicle speed to enter VSS test, MPH.

- VSSSW = CVIP VSS test enable switch, unitless; 1 -> do VSS test.

- VSSTIM = Minimum stablized time before performing CVIP VSS test, sec.

- VSTYPE = Integrated vehicle speed/cruise control system present switch; 0 -> no MPH and no VSC, 1 -> MPH and no VSC.

- V_VSS_MULT = Multiplier for FN689V when in 4x4 mode.

PROCESS

STRATEGY MODULE: VC_VSS_COM6

```
TFMFLG = 0 -----------------------|
 (TP Sensor OK)                   |
                                  |
NDSFLG = 1 -----------------------|
 (In Drive)                       |
                                  |
I4X4L = 0 ------------|           |
 (Not in 4x4 mode)    |AND -|     |
                      |     |     |
NEBART => FN689V -----|     |     |
 (min. rpm)                 |OR --|
                            |     |
NEBART => FN689V *          |     |AND -| INCREMENT VSSTMR
           V_VSS_MULT ------|     |     |
 (min. rpm in 4x4 mode)           |     |
                                  |     |
BIHP = 0 -------------------|     |     |
                            |OR --|     |
BIHP = 1 -------------|     |     |     |
                      |AND -|     |     |
BIFLG = 0 ------------|           |     |
 (Brake off)                      |     |
                                  |     |
VSBAR < VSSMN1 -------------------|     |
 (low VSS reading)                      |
                                        | --- ELSE ---
                                        |
                                        | VSSTMR = 0
```

```
VSSSW = 1-------------------------|
                                  |
VSTYPE <> 0 ----------------------|AND -| SET ERROR DETECTED
                                  |     | Do: FAULT FILTERING
VSSTMR => VSSTIM -----------------|     |
                                        | --- ELSE ---
                                        |
                                        | Do: FAULT FILTERING


VIP_ENABLE = 1 -------------------|
 (VIP in progress)                |AND -| VSFMFLG = 1
                                  |     |
C452FIL > C452LVL ----------------|     |
                                        | --- ELSE ---
                                        |
                                        | NO ACTION ON VSFMFLG


VSBAR => VSSMN1 ------------------------| VSFMFLG = 0
                                        |
                                        | --- ELSE ---
                                        |
                                        | NO ACTION ON VSFMFLG
```

BRAKE ON/OFF CIRCUIT TEST

OVERVIEW

If BIFLG (BOO input) does not change state after V_ZTOSPD_CNT transitions from a vehicle speed of zero to a vehicle speed of V_BOO_SPD sustained for V_BOOSPD_TM, then the BOO input is assumed to be faulty.

DEFINITIONS

Self Test Registers:

- C452FIL = Fault filter register for VSS test.
- V_ZTOSPD_CTR = Number of vehicle speed transitions from a speed of zero to a speed of V_BOO_SPD without a BOO transition.

Self Test Flags:

- V_SPDTOZ_FLG = When set this flag indicates a transition from a vehicle speed of V_BOO_SPD to a speed of zero has occurred.
- V_BOO_OLD = BIFLG from previous background loop.
- WIGFLG = When 1 indicates wiggle mode.

Self Test Calibration Constants:

- C536UP = Continuous BOO test fault filter increment.
- C536LVL = Continuous BOO test fault filter threshold.
- V_BOOSPD_TM = VSBAR threshold to reset V_BOOSPD_TMR.
- V_BOO_SPD = Vehicle speed required to increment V_ZTOSPD_CTR.
- V_CBOO_ENA = Continuous BOO test enable flag, 1 = test enabled.
- V_ZTOSPD_CNT = Number of zero MPH to vehicle speed = V_BOO_SPD transitions without a brake input considered a BOO failure.

Self Test Timers:

- V_BOOSPD_TMR = Non cumulative time VSBAR greater than V_BOO_SPD.

Base Strategy Registers:

- VSBAR

Base Strategy Flags:

- BIFLG
- CRKFLG
- NO_START
- RUNNING

Base Strategy Calibration Constants:

- BIHP

Base Strategy Timers:

- PUTMR

PROCESS

STRATEGY MODULE: VC_BOO_COM1

Test Entry Conditions:

```
V_CBOO_ENA = 0 -------------------|
 (test cal'd out)                 |
                                  |
BIHP = 0 -------------------------|
 (no brake input hardware)        |
                                  |
RUNNING = 1 ----------------------|
 (KOER)                           |
                                  |OR --| exit BRAKE ON/OFF CIRCUIT
NO_START = 1 ---------------------|     |  TEST
 (KOEO)                           |     |
                                  |     |
CRKFLG = 1 -----------------|     |     |
 (in crank mode)            |AND -|     |
                            |     |     |
WIGFLG = 0 -----------------|     |     |
                                  |     |
PUTMR < 4 ------------------------|     |
 (less than 4 sec since:                |
  powerup, exiting KOEO or              |
  KOER VIP, or a reset)                 |
                                        |
                                        | --- ELSE ---
                                        |
                                        | DO: BOO_SPD_TMR
                                        | DO: BOO_TEST_MAIN
                                        |  (test is performed)
```

```
BEGIN:  BOO_TEST_MAIN


V_BOO_OLD <> BIFLG ---------------------| V_BOO_OLD := BIFLG
 (boo input changed)                    | V_ZTOSPD_CTR := 0
                                        | call fault filter for code
                                        |  536
                                        |
                                        | --- ELSE ---
                                        |
V_BOOSPD_TMR > V_BOOSPD_TM -------|     |
                                  |AND -| V_ZTOSPD_CTR :=
V_SPDTOZ_FLG = 1 -----------------|     |  V_ZTOSPD_CTR + 1
                                        | V_SPDTOZ_FLG := 0
                                        | DO: BOO_FAILURE_PROCESS
                                        | call fault filter for code
                                        |  536
                                        |
                                        | --- ELSE ---
                                        |
VSBAR = 0 ------------------------------| V_SPDTOZ_FLG := 1
                                        | call fault filter for code
                                        |  536
                                        |
                                        | --- ELSE ---
                                        |
                                        | call fault filter for code 536


END: BOO_TEST_MAIN



BEGIN: BOO_FAILURE_PROCESS


C452FIL > 0 ----------------------------| C536FIL := 0
 (VSS failure indication)               | V_ZTOSPD_CTR := 0
                                        | return
                                        |
                                        | --- ELSE ---
                                        |
                                        |
V_ZTOSPD_CTR > V_ZTOSPD_CNT ------------| ERROR_DETECTED := 1
 (failure)                              | V_ZTOSPD_CTR := 0
                                        | return
                                        |
                                        | --- ELSE ---
                                        |
                                        | return


END: BOO_FAILURE_PROCESS
```

```
BEGIN: BOO_SPD_TMR

VSBAR < V_BOO_SPD -----------------------| V_BOOSPD_TMR := 0

END: BOO_SPD_TMR
```

ADAPTIVE TABLE CLIP TEST - SPEED DENSITY

OVERVIEW:

When the fuel system is using a KAMREF that is at its rich or lean limit a service code is stored in KAM. The test distinguishes between the lean limit and the rich limit and also whether at idle or not at idle.

DEFINITIONS

Self Test Registers:

- C179FIL = At adaptive fuel limit, system lean fault filter.
- C181FIL = At adaptive fuel limit, system rich fault filter.
- C182FIL = At adaptive fuel limit, system lean at idle fault filter.
- C183FIL = At adaptive fuel limit, system rich at idle fault filter.

Self Test Calibration Constants:

- C179LVL = At adaptive fuel limit, system lean fault filter threshold.
- C179UP = At adaptive fuel limit, system lean fault filter increment.
- C181LVL = At adaptive fuel limit, system rich fault filter threshold.
- C181UP = At adaptive fuel limit, system rich fault filter increment.
- C182LVL = At adaptive fuel limit, system lean at idle fault filter threshold.
- C182UP = At adaptive fuel limit, system lean at idle fault filter increment.
- C183LVL = At adaptive fuel limit, system rich at idle fault filter threshold.
- C183UP = At adaptive fuel limit, system rich at idle fault filter increment.

Continued on Next Page

Continued from Previous Page

- V_ADAPTV_ENA = Adaptive fuel clip test enable switch, 1 = enable.
- V_ADPMP_MAX = Maximum MAPPA to perform adaptive clip test.
- V_ADPMP_MIN = Minimum MAPPA to perform adaptive clip test.
- V_ADPN_MAX = Maximum N to perform adaptive clip test.
- V_ADPN_MIN = Minimum N to perform adaptive clip test.
- V_LAMAV_MAX = LAMAVEn above which a clipped adaptive cell may be failed.
- V_LAMAV_MIN = LAMAVEn below which a clipped adaptive cell may be failed.

Self Test Flags:

- ERROR_DETECTED = Flag indicates a failure is indicated, 1 = failure.

Base Strategy Registers:

- KAMREF = Total learned fuel system correction.
- LAMAVE = Average LAMBSE
- MAPPA = MAP/BP.
- N = Engine speed, RPM.

Base Strategy Constants:

- MAXADP = Maximum adaptive correction.
- MINADP = Minimum adaptive correction.

Base Strategy Flags:

- REFFLG = Indication of Idle Air Flow; 1 = Idle Air Flow.

Continued on Next Page

Continued from Previous Page

PROCESS:

STRATEGY MODULE: VC_SDADP_COM1

Fault Filters not currently illuminating the MIL (CxxFIL <= CxxLVL get a fresh start each time the test conditions become true.

```
C179FIL <= C179LVL -------------------------|
 (failure no longer indicated)              |
MAPPA <= V_ADPMP_MIN ----------------|      |AND --| C179FIL :=0
MAPPA >= V_ADPMP_MAX ----------------|      |      |
N >= V_ADPN_MAX ---------------------|OR ---|
N <= V_ADPN_MIN ---------------------|
 (outside test condition)            |
REFFLG = 1 --------------------------|
 (idle air flow)


C181FIL <= C181LVL -------------------------|
 (failure no longer indicated)              |
MAPPA <= V_ADPMP_MIN-----------------|      |AND --| C181FIL :=0
MAPPA >= V_ADPMP_MAX ----------------|      |      |
N >= V_ADPN_MAX ---------------------|OR ---|
N <= V_ADPN_MIN ---------------------|
 (outside test condition)            |
REFFLG = 1---------------------------|
 (idle air flow)


C182FIL <= C182LVL ------------------|
 (failure no longer indicated)       |AND --| C182FIL := 0
REFFLG = 0 --------------------------|
 (not idle air flow)


C183FIL <= C183LVL ------------------|
 (failure no longer indicated)       |AND --| C183FIL := 0
REFFLG = 0 --------------------------|
 (not idle air flow)
```

Continued on Next Page

Continued from Previous Page

Test Logic.

```
REFFLG = 0 -----------------------|       | ERROR_DETECTED := 1
 (not idle air flow)              |       | call fault filter routine
V_ADPMP_MAX > MAPPA > V_ADPMP_MIN-|       |  (code 179)
 (load window)                    |       | call fault filter routine
V_ADPN_MAX > N > V_ADPN_MIN ------|AND --|  (code 181)
 (engine speed window)            |       | call fault filter routine
KAMREF = MINADP + 0.5-------------|       |  (code 182)
 (failure criteria)               |       | call fault filter routine
V_ADAPTV_ENA = 1 -----------------|       |  (code 183
 (test cal'd in)                  |       |
LAMAVE >= V_LAMAV_MAX-------------|       |
                                          |
                                          | ---ELSE---
                                          |
REFFLG = 0 -----------------------|       | ERROR_DETECTED := 1
 (not idle air flow)              |       | call fault filter routine
V_ADPMP_MAX > MAPPA > V_ADPMP_MIN-|       |  (code 181)
 (load window)                    |AND --| call fault filter routine
V_ADPN_MAX > N > V_ADPN_MIN ------|       |  (code 179)
 (engine speed window)            |       | call fault filter routine
KAMREF = MAXADP + 0.5-------------|       |  (code 182)
 (failure criteria)               |       | call fault filter routine
V_ADAPTV_ENA = 1 -----------------|       |  (code 183)
 (test cal'd in)                  |       |
LAMAVE <= V_LAMAV_MIN ------------|       |
                                          |
                                          | ---ELSE---
                                          |
REFFLG = 1  ----------------------|       | ERROR_DETECTED := 1
 (idle air flow)                  |AND --| call fault filter routine
KAMREF = MINADP + 0.5-------------|       |  (code 182)
 (failure criteria)               |       | call fault filter routine
V_ADAPTV_ENA = 1 -----------------|       |  (code 179)
 (test cal'd in)                  |       | call fault filter routine
LAMAVE >= V_LAMAV_MAX ------------|       |  (code 181)
                                          | call fault filter routine
                                          |  (code 183)
                                          |
                                          | ---ELSE---
                                          |
```

Continued on Next Page

Continued from Previous Page

```
REFFLG = 1 -----------------------|       | ERROR_DETECTED := 1
 (idle air flow)                  |AND --| call fault filter routine
KAMREF = MAXADP + 0.5-------------|       |  (code 183)
 (failure criteria)               |       | call fault filter routine
V_ADAPTV_ENA = 1 -----------------|       |  (code 179)
 (test cal'd in)                  |       | call fault filter routine
LAMAVE <= V_LAMAV_MIN ------------|       |  (code 181)
                                          | call fault filter routine
                                          |  (code 182)
                                          |
                                          | ---ELSE---
                                          |
V_ADPMP_MAX >MAPPA > |                    |
 V_ADPMP_MIN --------|                    |
  (load window)      |AND----|            |
V_ADPN MAX > N >     |       |OR --------|
 V_ADPN_MIN ---------|       |            | call fault filter routine
  (engine speed window)      |            |  (code 179)
REFFLG = 1 ------------------|            | call fault filter routine
 (idle air flow)             |            |  (code 181)
V_ADAPTV_ENA = 0 ------------|            | call fault filter routine
 (test cal'd out)                         |  (code 182)
                                          | call fault filter routine
                                          | (code 183)
```

EGO SWITCHING TEST - SD W/SINGLE EGO STRATEGIES

OVERVIEW

This test determines that the EGO sensor is switching properly when conditions are present that will allow a functional fuel system to cause the EGO sensor to switch.

DEFINITIONS

Registers:

- AEFUEL = Acceleration enrichment fuel flow, lb/hr.
- ATMR1 = Time since exiting crank mode.
- C171FIL = Lack of EGO switch, adaptive fuel at limit fault filter.
- C172FIL = Lack of EGO switch, EGO indicates lean fault filter.
- C173FIL = Lack of EGO switch, EGO indicates rich fault filter.
- EGOSSS = Number of EGO switches since start.
- EGOTSTCUMTMR = Accumulated time that the EGO test test conditions are true.
- KAMREF = Total learned fuel system correction.
- MAPPA = MAP/BP.
- PPCTR = PIP counter; updated at PIP rising edge before injector pulsewidth is calculated and output.
- PRGTMR = Total accumulative purge on time.
- PURGDC = Purge duty cycle.
- PUTMR = Time since powerup.
- TP_REL = Relative Throttle Position, counts. TP - RATCH.
- V_EGOTST_TMR = Time since self test closed loop conditions have been met.
- V_EGR_DLYTMR = Test delay timer for EGR transitions.
- V_LEGOTMR = EGO sensor test time since last EGO state change while testing is active.
- V_LESTMR = Self test lack of EGO switchng timer.
- V_PRG_DLYTMR = Test delay timer for purge transitions.
- V_VACPRGTMR = Indicates when non-EEC controlled purge canister is presumed empty.

Bit Flags:

- AFMFLG = Flag indicating that ACT sensor is in/out of range.
- CFMFLG = Flag indicating that ECT sensor is in/out of range.
- CRKFLG = Flag indicating crank mode.
- EGOFL = EGO sensor state flag; 1 -> rich.
- error_detected = Flag indicates a failure is indicated; 1 -> failure.
- FLG_OPEN_LOOP = Open loop fuel flag; 1 -> Open loop fuel
- ISCFLG = ISC mode indicator flag; -1 -> dashpot mode, 0 -> dashpot preposition mode, 1 -> closed loop RPM control mode, 2 -> closed loop RPM control lockout.
- LEGONOTPURG = In EGO test, when equal to 1, testing while purging is appropriate.
- LESFLG = Lack of EGO switching flag; 0 -> switching.
- MFMFLG = MAP failure flag; 1 -> MAP sensor fails.
- MPGFLG = Flag that indicates whether in Fuel Economy mode; 1 -> in Fuel Economy mode.
- MPGTFG = MPG transition mode flag; 1 -> MPG mode exit into Closed Loop fuel.
- NO_START = Engine off VIP enable flag.
- RUNNING = Flag which indicates that idle speed is being controlled by Engine Running VIP.
- SWTFL = EGO switch flag; 1 -> EGO switched this background loop.
- TFMFLG = Flag indicating TP sensor is in/out of range.
- V_EGO_BYPS = Prevents additional EGO service code.
- V_EGOL_BYPS = Prevents additional EGO service code.
- V_LAMJMP = 1 -> base strategy caused a LAMBSE jump since last EGO switch.
- WRMEGO = 1 -> EGO sensor is warm, 0 -> sensor has cooled off.

Calibration Constants:

- C171LVL = Lack of EGO switch, adaptive fuel at limit fault filter threshold.
- C171UP = Lack of EGO switch, adaptive fuel at limit fault filter increment.
- C172LVL = Lack of EGO switch, EGO indicates lean fault fault filter threshold.

- C172UP = Lack of EGO switch, EGO indicates lean fault filter increment.
- C173LVL = Lack of EGO switch, EGO indicates rich fault filter threshold.
- C173UP = Lack of EGO switch, EGO indicates rich fault filter increment.
- ETST_SWCUMTM = Accumulated time EGO test active before failure is indicated because of number of switches failure.
- MAXADP = Maximum adaptive correction.
- MINADP = Minimum adaptive correction.
- PIPNUM = Number of PIPs for DFSO exit fuel ramp.
- PRG_DEC = Purge DC decrement amount when purge overwhelms fuel control.
- V_EEC_PRG = 1 -> EEC controlled purge, 0 -> mechanical purge.
- V_EGOAEMAX = Maximum AEFUEL to perform EGO test.
- V_EGOIDL_ENA = Switch to enable EGO sensor test at idle air flows; 1 -> enable.
- V_EGOMAP_MAX = Maximum MAPPA to perform EGO sensor test when not at idle.
- V_EGOMAP_MIN = Minimum MAPPA to perform EGO sensor test when not at idle.
- V_EGORNTM = Time since exiting crank mode (ATMR1) to wait before enabling ego test.
- V_EGO_EGR_SW = When equal to zero allows EGO test to ignore egr transitions.
- V_EGOSWNUM = Value of EGOSSS .LT. indicating an ego
- V_EGOTP_MIN = Minimum TP_REL to perform EGO sensor test when not at idle.
- V_EGOTST_TM = Time since closed loop conditions minus EGO input have been met.
- V_EGO_ENA = Continuous EGO test enable switch; 1 -> enable.
- V_LEGO_MAX = Maximum time since last EGO switch before failure indication.
- V_LEGO_MAX2 = Maximum time since last EGOn switch before failure incication.
- V_LESTM = Time since last EGO switch limit to fail continuous EGO switching test.
- V_PRGTOT = Total accumulative purge on time to perform continuous EGO test.

PROCESS

STRATEGY MODULE: VC_EGO_TEST_COM13

Test Entry Conditions:

```
V_EGO_ENA = 0 ----------|
 (test cal'd out)       |
                        |
RUNNING = 1 ------------|OR --| EXIT EGO SWITCHING TEST
 (KOER)                 |     |
                        |     |
NO_START = 1 -----------|     |
 (KOEO)                       |
                              | --- ELSE ---
CRKFLG = 1 -------------|     |
 (in crank mode)        |OR --| EGOTSTCUMTMR := 0
                        |     | Do: EGO_TEST_TMR_CLEAR
PUTMR < 4 --------------|     |  (test is bypassed, clear timers
 (less than 4 sec since:      |   until test is executed)
  powerup, exiting KOEO       | Do: CANISTER_FILLING_TIMER_CONTROL
  or KOER VIP, or a reset)    |  (used when purge not controlled by EEC)
                              |
                              | --- ELSE ---
                              |
                              | Do: EGR_TRANSITION_DELAY_FLAG_CONTROL
                              | Do: EGO_PURG_CHK
                              | Do: CANISTER_FILLING_TIMER_CONTROL
                              |  (used when purge not controlled by EEC)
                              | Do: PURG_NOT_CAUSE_FOR_FAILURE
                              | Do: EGO_SWITCHING_FAILURE_INDICATION_CONTROL
                              | Do: ACCUM_TIMER_CONTROL
                              | Do: EGOTST_TMR_CONTROL
                              | Do: EGOTST_TMR_CHK
```

BEGIN: EGOTST_TMR_CONTROL

The following test conditions logic must be true for V_EGOTST_TM seconds before the fuel system can be tested.

```
WRMEGO = 1 -----------------------------|
                                        |
MPGFLG = 0 -----------------------------|
                                        |
MPGTFG = 0 -----------------------------|
                                        |
FLG_OPEN_LOOP = 0 ----------------------|
                                        |
PPCTR >= PIPNUM ------------------------|
 (past decel fuel shutoff)              |
                                        |
AFMFLG = 0 -----------------------------|
 (act ok)                               |
                                        |
CFMFLG = 0 -----------------------------|
 (ect ok)                               |
                                        |
TFMFLG = 0 -----------------------------|
 (tp ok)                                |AND -| (test conditions true)
                                        |     | allow V_EGOTST_TMR to run
MFMFLG = 0 -----------------------------|     |
                                        |     | EGOTSTCUMFLG := 1
ATMR1 >= V_EGORNTM ---------------------|     |
                                        |     |
V_PRG_DLYTMR = 0 -----------------------|     |
 (purge transition delay past)          |     |
                                        |     |
V_EGR_DLYTMR = 0 -----------------------|     |
 (EGR transition delay past)            |     |
                                        |     |
AEFUEL <= V_EGOAEMAX -------------------|     |
 (max AEFUEL to perform test)           |     |
                                        |     |
ISCFLG = 0 -----------------|           |     |
                            |           |     |
TP_REL >= V_EGOTP_MIN ------|           |     |
                            |AND -|     |     |
MAPPA > V_EGOMAP_MIN -------|     |     |     |
 (load indicator)           |     |     |     |
                            |     |OR --|     |
MAPPA < V_EGOMAP_MAX -------|     |           |
                                  |           |
ISCFLG = 1 -----------------|     |           |
                            |AND -|           |
V_EGOIDL_ENA ---------------|                 |
                                              | --- ELSE ---
                                              |
                                              | Do: EGO_TEST_TMR_CLEAR
                                              | Do: VEGOFIL_ZERO
                                              | EGOTSTCUMFLG := 0
```

END: EGOTST_TMR_CONTROL

BEGIN: EGOTST_TMR_CHK

Control of how long the test conditions must be present before testing is to take place is accomplished with the calibration parameter V_EGOTST_TM.

```
V_EGOTST_TMR > V_EGOTST_TM -------------| Do: EGO_TESTS
 (test conditions present
  sufficiently long to test)
```

END: EGOTST_TMR_CHK

BEGIN: ACCUM_TIMER_CONTROL

The timer EGOTSTCUMTMR is a cumulative timer that documents total accumulated time that the EGO test conditions are true while there are no purging restrictions since the last time crank mode was exitted. If crank mode is re-entered, this timer is cleared (see "Test Entry Conditions").

```
EGOTSTCUMFLG = 1 -----------------------| allow EGOTSTCUMTMR to run
 (test conditions true)                 |
                                        |
                                        | --- ELSE ---
                                        |
                                        | freeze EGOTSTCUMTMR
```

END: ACCUM_TIMER_CONTROL

BEGIN: EGO_TEST_TMR_CLEAR

Conditions other than fuel system failures which may prevent EGO switches are present. The following timers which are indicators of fuel system failure are
cleared until conditions for testing become true.

```
V_LESTMR := 0
V_EGOTST_TMR := 0
V_LEGOTMR := 0
```

END: EGO_TEST_TMR_CLEAR

BEGIN: EGO_PURG_CHK

```
PRG_DEC = 0 --------------|
                          |OR --|
PURGDC = 0 ---------------|     |
 (not purging)                  |AND -|
                                |     |
V_EEC_PRG = 1 ------------------|     |
 (EEC controlled purge)               |
                                      |OR --| ego_purg_byps := 0
V_VACPRGTMR >= V_PRGTOT --|           |     |  (no purging restrictions
 (canister sufficiently   |           |     |   on ego test)
  empty)                  |OR --|     |     |
                          |     |     |     |
APT = -1 -----------------|     |AND -|     |
 (not purging)                  |           |
                                |           |
V_EEC_PRG = 0 ------------------|           |
 (mechanical purge)                         |
                                            | --- ELSE ---
                                            |
                                            | ego_purg_byps := 1
                                            |
                                            |  (purging restrictions on
                                            |   on ego test in effect)
```

END: EGO_PURG_CHK

BEGIN: PURG_NOT_CAUSE_OF_FAILURE

If the control system has gone open loop (LESFLG = 1) while the ego test was being bypassed then testing for a rich failure may take place if no purging is allowed during open loop (PURGSW = 0).

```
LESFLG = 1 -----------------------|
 (byples timeout)                 |AND -| LEGONOTPURG := 1
                                  |     |  (conditions imply that purge
PURGSW = 0 -----------------------|     |   was not the cause of EGO
 (no purge while open loop)             |   stuck rich)
                                        |
                                        | --- ELSE ---
                                        |
                                        | LEGONOTPURG := 0
```

END: PURG_NOT_CAUSE_OF_FAILURE

BEGIN: VEGOFIL_ZERO

Fault filters not currently illuminating the MIL (CxxFIL > CxxLVL) get a fresh start each time the test conditions become true.

```
1.  C171FIL <= C171LVL -----------------| C171FIL := 0

2.  C172FIL <= C172LVL -----------------| C172FIL := 0

3.  C173FIL <= C173LVL -----------------| C173FIL := 0
```

END: VEGOFIL_ZERO

BEGIN: EGO_TESTS

Base strategy will assign LAMBSE a new value of after TSLEGO > LESTM. Failure codes 171, 172 or 173 as appropriate will be stored upon failure, with the MIL being illuminated. But as a result of the Base Strategy action, the failure condition is overwritten. The fault filter will down count. The service code will remain stored in KAM, but the MIL will be turned off.

The following logic segment within the below logic will maintain the fault filter value after failure (CxxxFIL > (CxxxLVL) until the sensor starts switching, causing the MIL to remain illuminated until the actual failure condition is not present.

```
C17xFIL > C17xLVL ------------|
 (failure has been indicated) |AND -|
                              |     |
LESFLG = 1 -------------------|     |
 (ego hasn't switched since         |
  failure)
```

```
C171FIL > C171LVL -----------------------|
 (failure has been indicated)            |AND -| error_detected := 1
                                         |     | call fault filter routine
LESFLG = 1 ------------------------------|     |  (code 171)
 (ego hasn't switched since                    | call fault filter routine
  failure)                                     |  (code 172)
                                               | call fault filter routine
                                               |  (code 173)
                                               |
                                               | --- ELSE ---
C172FIL > C172LVL -----------------------|     |
 (failure has been indicated)            |AND -| error_detected := 1
                                         |     | call fault filter routine
LESFLG = 1 ------------------------------|     |  (code 172)
 (ego hasn't switched since                    | call fault filter routine
  failure)                                     |  (code 171)
                                               | call fault filter routine
                                               |  (code 173)
                                               |
                                               | --- ELSE ---
C173FIL > C173LVL -----------------------|     |
 (failure has been indicated)            |AND -| error_detected := 1
                                         |     | call fault filter routine
LESFLG = 1 ------------------------------|     |  (code 173)
 (ego hasn't switched since                    | call fault filter routine
  failure)                                     |  (code 171)
                                               | call fault filter routine
                                               |  (code 172)
                                               |
                                               | --- ELSE ---
```

(continued on next page)

(continued from previous page)

```
V_LESTMR > V_LESTM ----------------|           |
 (failure indicated)               |           |
                                   |           |
EGOSSS < V_EGOSWNUM ---------|     |           |
                             |     |           |
ego_purg_byps = 0 -----|     |AND -|           |
                       |OR --|     |OR --|     |
EGOFL = 0 -------------|     |     |     |     |
                             |     |     |     |
EGOTSTCUMTMR > ETST_SWCUMTM -|     |     |     |
 (sufficient cumulated test time)  |     |AND -| error_detected := 1
                                   |     |     |  (code 171)
V_LEGOTMR > V_LEGO_MAX ------------|     |     | call fault filter routine
                                         |     |  (code 172)
KAMREF = 0.5 + MINADP -------------|     |     | call fault filter routine
 (adaptive clip)                   |OR --|     |  (code 173)
                                   |           |  (fuel system failure)
KAMREF = 0.5 + MAXADP -------------|           |
                                               |
                                               |
                                               | --- ELSE ---
V_LESTMR > V_LESTM ----------------|           |
 (failure indicated)               |           |
                                   |           |
EGOSSS < V_EGOSWNUM ---------|     |           |
                             |     |OR --|     |
V_EGOL_BYPS = 0 -------------|AND -|     |     |
                             |     |     |AND -| error_detected := 1
EGOTSTCUMTMR > ETST_SWCUMTM -|     |     |     | call fault filter routine
 (sufficient cumulated test time)  |     |     |  (code 172)
                                   |     |     | call fault filter routine
V_LEGOTMR > V_LEGO_MAX ------------|     |     |  (code 171)
                                         |     | call fault filter routine
EGOFL = 0 -------------------------------|     |  (code 173)
 (lean)                                        | V_EGO_BYPS := 1
                                               |
                                               | --- ELSE ---
```

(continued on next page)

(continued from previous page)

```
LEGONOTPURG = 1 -------------|                 |
 (purge not implicated as    |                 |
  cause of failure)          |AND -|           |
                             |     |           |
V_LEGOTMR > V_LEGO_MAX2 -----|     |           |
                                   |           |
V_LESTMR > V_LESTM ----------------|           |
 (lambse at clip)                  |           |
                                   |           |
EGOSSS < V_EGOSWNUM ---------|     |           |
 (fail from start)           |     |OR --|     |
                             |     |     |     |
ego_purg_byps = 0 -----------|AND -|     |     |
                             |     |     |     |
V_EGO_BYPS = 0 --------------|     |     |     |
                             |     |     |AND -| error_detected := 1
EGOTSTCUMTMR > ETST_SWCUMTM -|     |     |     | Call fault filter routine
                                   |     |     |  (code 173)
V_LEGOTMR > V_LEGO_MAX ------------|     |     | Call fault filter routine
 (gross lack of ego switch)              |     |  (code 171)
                                         |     |  call fault filter routine
EGOFL = 1 -------------------------------|     |  (code 172)
 (rich)                                        | V_EGOL_BYPS := 1
                                               |
                                               | --- ELSE ---
                                               |
                                               | call fault filter routine
                                               |  (code 171)
                                               | call fault filter routine
                                               |  (code 172)
                                               | call fault filter routine
                                               |  (code 173)
                                               |  (no failures present)
```

BEGIN: EGR_TRANSITION_DELAY_FLAG_CONTROL

The EGR transition flag V_EGR_DLYFG when set indicates that an EGR transition has taken place. To allow the EGO test to disregard EGR transitions, calibrate V_EGO_EGR_SW = 0. Otherwise any EGR transition (on/off, off/on) will reset the EGO test timer and associated EGO fuel system failure indicators.

```
EGRDC = 0 ------------------------------| temp := 1
 (egr off)                              |
                                        | --- ELSE ---
                                        |
                                        | temp := 0


V_EGR_OLD <> temp ----------------|
 (egr transition)                 |AND -| V_EGR_DLYFG := 1
                                  |     |  (ego test reset due to
V_EGO_EGR_SW = 1 -----------------|     |   egr transition)
 (egr considered to                     | V_EGR_OLD := temp
  affect ego)                           |
                                        | --- ELSE ---
                                        |
                                        | V_EGR_DLYFG := 0
                                           (egr not affecting ego test)
```

END: EGR_TRANSITION_DELAY_FLAG_CONTROL

BEGIN: EGO_SWITCHING_FAILURE_INDICATION_CONTROL

Timer V_LESTMR indicate the time LAMBSE has been at its clip.

VIP Lack of EGO Switching Timer:

```
LAMBSE <= LAMMIN -----------------|
 (rich clip, lean system)         |
                                  |OR --| (V_LESTMR runs indicating
LAMBSE >= LAMMAX -----------|     |     |  potential lambse/ego based
 (lean clip, rich system)   |AND -|     |  failure)
                            |           |
ego_purg_byps = 0 ----------|           |
 (no purging restrictions)              |
                                        | --- ELSE ---
                                        |
                                        | V_LESTMR := 0
                                        | (no lambse/ego based
                                        |  errors indicated)
```

Time Since Last EGO Switch Logic:

```
SWTFL = 1 ------------------------------| V_LEGOTMR := 0
 (ego switched this BG)                 |
                                        | --- ELSE ---
V_LAMJMP = 1 ---------------------|     |
 (LAMBSE jump occurred)           |     |
                                  |OR --| V_LEGOTMR := 0
ego_purg_byps = 1 ----------|     |     | V_LAMJMP = 0
 (purg is affecting         |     |     | (reinitialize due to LAMBSE
  ego performance)          |     |     |   jump)
                            |AND -|
EGOFL = 1 ------------------|
 (rich)                     |
                            |
LEGONOTPURG = 0 ------------|
 (purge implicated as
  a cause of failure)
```

END: EGO_SWITCHING_FAILURE_INDICATION_CONTROL

```
BEGIN: CANISTER_FILLING_TIMER_CONTROL


V_EEC_PRG = 0 --------------------------| Do: CANISTER_CONTENT_MODEL
 (purge is not controlled by EEC)

END: CANISTER_FILLING_TIMER_CONTROL


BEGIN: CANISTER_CONTENT_MODEL

TP_REL <= V_TPREL_PRG ------------------| decrement V_VACPRGTMR
                                        |  (canister is filling)
                                        |
                                        | --- ELSE ---
                                        |
V_VACPRGTMR > V_PRGTOT -----------------| freeze V_VACPRGTMR
                                        |  (max required purge time
                                        |    reached)
                                        |
                                        | --- ELSE ---
                                        |
                                        | increment V_VACPRGTMR
                                        |  (canister emptying)

END: CANISTER_CONTENT_MODEL
```

FUEL PUMP CIRCUIT TEST

OVERVIEW

The state of the Fuel Pump Monitor (FPM) is compared to the expected state based on the fuel pump on/off command. Also, after a stall, the Output Circuit Check (OCC) is performed on the fuel pump relay.

When the fuel pump is commanded off, FPM should be low. If it is not (after waiting V_FPMTM seconds for the relay to settle) code 542 is set. This indicates that either the fuel pump relay is stuck "on" or there is an open circuit between the ECA and the fuel pump ground (circuit ooo in the diagram) allowing the ECA pull-up circuit to hold FPM high.

If the fuel pump is commanded on, FPM should be high. If it is not (after waiting V_FPMTM seconds) code 543 is set. This indicates a break in the line between the battery and the FPM input to the ECA (circuit xxx in diagram) or no contact inside the fuel pump relay.

If the engine stalls, the fuel pump is commanded off and then exercised to perform the output circuit check (OCC). An OCC failure indicates a break in the line between the battery and the driver inside the ECA (circuit *** in the diagram).

FUEL PUMP / FUEL PUMP MONITOR (FPM) CIRCUIT

```
B+ xxxxxxxxxxxxxxxx                oooooooooooo
                  x              ..o..        o    g
b+ *************  x              ..o..        o    r
        S      *  x            ..     ..    -----  o
        O .....*..x....       .   PUMP  .    ---   u
        L !    >  x   !        ..     ..      -    n
        E !  <     \  !          ..o..             d
        N !    >  x \ !          ..o..
        O !  <    x   !            o
        I !....*..x...!            o
        D      *  xxxxxxxxxxxooooooo
               *             x
         ......*.............o...............
         !     *             x              !
         !     *** OCC       o        ECA   !
         !     *             x     (EEC-IV  !
         !   Driver         FPM     MODULE) !
         !..................................!
```

DEFINITIONS

Self test Registers:

- IOCC = OCC A/D input level.

- OCCSAV = Saved value of OCC A/D input.

Self Test Calibration Constants:

- OCCDT7 = Fuel pump primary OCC calibration level.

- V_FPMDLY = Fuel pump monitor test fuel pump on-to-off off-to-on stabilization delay time.

- V_FPMFLG = Fuel pump monitor test enable switch, 1 = enable.

- V_FPMTM = Fuel pump transition delay time.

- VPUMP_LAST = State of fuel pump during last background loop.

Self Test Flags:

- ERROR_DETECTED = Flag passed to fault filter routine indicating self test detected a failure.

- FPM = State of the FPM input. 1 = high (Pump on).

Self Test Timers:

- VIP_FPMTMR = Fuel pump on-to-off transition delay timer.

Base Strategy Flags:

- PUMP = Controls state of fuel pump control output. 1 = commanded on.

PROCESS

STRATEGY MODULE: VC_FUEL_PUMP_COM2

Every Background Loop (in continuous or not):

```
PUMP <> VPUMP_LAST ---------------------| VIP_FPMTMR = 0
 (Fuel Pump changed on/off state)       |  (restart timer)
                                        |
                                        | VPUMP_LAST =  PUMP
PUTMR < 4 -----------------------|      |
                                 |      |
NO_START = 1 --------------------|OR --| Exit this test
                                 |      |  (not in "continuous")
RUNNING = 1 ---------------------|      |
                                        | --- ELSE ---
                                        |
V_FPMFLG = 0 --------------------------| call fault filter for code 542
 (test cal'd out)                       | call fault filter for code 543
                                        | call fault filter for code 556
                                        | exit this test
                                        |
                                        | --- ELSE ---
                                        |
PUMP = 1 -------------------------------| DO: FP COMMANDED ON PROCESS
                                        |
                                        | --- ELSE ---
                                        |
RMSPRU = 1 ----------------------------| De-energize all OCC outputs
 (Run mode since powerup = true         | delay 50 msec
  and PUMP = 0, engine has stalled)     | OCCSAV = IOCC
                                        | PUMP = 1
                                        | (command fuel pump on)
                                        | delay 50 msec
                                        | DO:  FP OCC PROCESS
                                        |
                                        | --- ELSE ---
                                        |
VIP_FPMTMR >= V_FPMTM ------------|      | (FPM high when pump
 (Pump has been off for           |      |  commanded off, code 542)
  V_FPMTM seconds)                |AND -| ERROR_DETECTED = 1
                                  |      |
FPM = 1 ---------------------------|      | call fault filter for code 542
 (indicates pump on)                    | call fault filter for code 556
                                        |
                                        | --- ELSE ---
                                        |
                                        | (No errors detected)
                                        | call fault filter for code 542
                                        | call fault filter for code 556
```

```
                    BEGIN: FP COMMANDED ON PROCESS


UNDSP = 0 ------------------------------| RMSPRU = 1
 (Run mode)                             |  (run mode since powerup =
                                        |   true)


                                        |  (FPM low with pump commanded
                                        |   on  - fault 543 detected)
FPM = 0 -------------------------|      |
 (FPM input low)                 |AND -| ERROR_DETECTED = 1
                                 |      | call fault filter for code 543
VIP_FPMTMR >= V_FPMTM ------------|      | call fault filter for code 542
 (Pump has been on for                  | call fault filter for code 556
  at least v_FPMTM sec.)                |
                                        |
                                        | --- ELSE ---
                                        |
                                        |  (No error detected)
                                        | call fault filter for code 543
                                        | call fault filter for code 542
                                        | call fault filter for code 556

                    END: FP COMMANDED ON PROCESS


                         BEGIN: FP OCC PROCESS

                                        |  (FP relay primary circuit
|OCCSAV-IOCC| < OCCDT7 -----------------|   failure; code 556)
 (insufficient change in OCC)           | PUMP = 0
                                        | RMSPRU = 0
                                        | ERROR_DETECTED = 1
                                        | call fault filter for code 556
                                        |
                                        | --- ELSE ---
                                        |
                                        |  (No failure detected)
                                        | PUMP = 0
                                        | RMSPRU = 0
                                        | call fault filter for code 556

                          END: FP OCC PROCESS
```

MALFUNCTION INDICATOR LIGHT - SPEED DENSITY - SINGLE EGO

OVERVIEW

The purpose of the Malfunction Indicator Light (MIL) is to alert the driver that the computer has detected a fault with the EEC-IV system. If the MIL was not present, the driver may not be aware that a problem exists. The Failure Mode Effects Management (FMEM) strategy is capable of maintaining good drive characteristics with a fault present. However, the vehicle will not be operating at an optimum point with regard to emissions, economy and performance. When the MIL is on, the driver of the vehicle should seek service at his earliest convenience. It is not necessary to immediately shut the vehicle down and have it towed in for service.

The malfunction indicator light (MIL) warning system was implemented to comply with California regulations for the 1988 model year. A pilot program was carried out in 1987 on the 2.3L TC T'Bird/Cougar, 2.3L OHC Mustang and the 3.0L Taurus/Sable. By the 1989 model year both California and 49 States EEC-IV equipped passenger car and light and medium duty trucks were equipped with MIL. The only exception was the 2.3L TC Merkur XR4Ti.

The light, which is labeled "Check Engine" or "Service Engine Soon" is located on the dashboard such that the driver can see it. Power is supplied to the light whenever the ignition switch is in the run or crank position. The ground circuit for the light is provided through the EEC module self-test output (STO). Whenever the EEC-IV strategy determines that the light should be on, the STO output driver is turned on (STO voltage will be low). Since the light is controlled by STO, the self-test error codes can be determined by counting the check engine light pulses during self-test.

The light will be turned on by the EEC module (after a delay of FMDTM seconds) whenever a fault is detected for any of the monitored signals. When MILTMR exceeds the calibrateable delay time (FMDTM), the light will turn on for at least a calibrateable time of V-MILONTM seconds. If the fault is no longer present, then the light will turn off as soon as the light has been on for at least V-MILONTM seconds.

Continued on Next Page

Continued from Previous Page

If the light is on for a period less than V_MILONTM seconds, it indicates that the light was not activated by the check engine light strategy. This could be caused by an intermittent short to ground of the STO wire, intermittent operation in HLOS, or fault detection while in the wiggle self-test mode (STI grounded). With the exception of the grounded STO wire, operation in HLOS, or an intermittent fault with VSS in the wiggle test mode, a continuous error code will always be present, indicating the reason for the fault.

The check engine light is turned on in the crank mode until a PIP signal is detected as a bulb check. The bulb check can be disabled by setting MILLIM to zero. If the light does not turn off while the engine is cranking, it indicates that the EEC module is not receiving PIP signals.

The calibration parameter MIL_SW can take values of 0, 1 or 2. If MIL_SW is set to zero, the MIL light is not activated by the MIL logic. Note that the MIL light will still turn on whenever STO is grounded (HLOS, self-test). Production calibrations must have MIL_SW set to 1 to meet California regulations. When MIL-SW is set to 2, the MIL light will activate whenever any continuous fault filter indicates a fault is present for at least FMDTM seconds.

DEFINITIONS

INPUTS

Registers:

- C332FIL = Fault Filter indicating EGR flow problem.

- CXXXFIL = Fault Filter for any continuous fault.

- MILTMR = Timer used to record the time that a continuous fault has been present, sec.

Bit Flags:

- ADT1FMFLG = Flag indicating that one or more of the adaptive table one cells have reached the max or min clip values (MAXADP OR MINADP).
  1 -> TABLE IS AT CLIP.
  0 -> TABLE NOT AT CLIP.

Continued on Next Page

Continued from Previous Page

- EGO1FMFLG = Flag indicating there is an EGO-1 failure.

- AFMFLG = Flag indicating the ACT sensor has failed.

- CFMFLG = Flag indicating the ECT sensor has failed.

- CRKFLG = Flag indicating status of CRANK MODE (1 -> in CRANK MODE 0 -> not in CRANK MODE).

- DISABLE_NOSTART = Flag set to 1 when KOEO VIP test is entered Disables bulb check when KoEO test is exited.

- EFMFLG = Flag indicating the EVP/EPT sensor has failed.

- FIRST_PIP = Flag set to 1 when the first PIP is detected. Reset to zero on power-up or stall.

- MFMFLG = Flag indicating the MAF sensor has failed.

- RUNNING = Flag indicating that engine-running VIP is active.
  1 -> IN ENGINE-RUNNING VIP.
  0 -> NOT IN ENGINE-RUNNING VIP

- STIFLG = Flag indicating the state of STI (1 -> low, Self-Test requested; 0 -> high, Self-Test not requested).

- TFMFLG = Flag indicating the TP sensor has failed.

Continued on Next Page

Continued from Previous Page

Calibration Constants:

- C332LVL = Fault threshold for EGR flow problem.

- C332UP = Upcount for fault filter 332.

- CXXXLVL = Fault threshold for any continuous fault.

- CXXXUP = Upcount for any continuous fault.

- FMDTM = Time delay after fault is detected to turn on MIL, sec.

- MILLIM = Software switch to enable/disable bulb check, unitless (1 -> enable; 0 -> disable).

- MIL_SW = MIL enable switch, unitless,
  0 = Do not turn on MIL
  1 = Do MIL logic to meet California regulation, must be 1 for production
  2 = Do MIL logic and turn MIL light on for any continuous fault, development calibration only.
- V_MILONTM = Minimum MIL on time, sec.

OUTPUTS

Registers:

- MILTMR = See above.

Calibration Information

FMDTM should be set at 4.5 seconds. This will allow the check engine light to turn on within 5 seconds of the fault occurrence. The .5 seconds of time is used to allow the fault filter to reach the fault threshold.

V_MILONTM should be large enough to easily distinguish between an intermittent harness short and a strategy controlled activation of the check engine light. A 10 second on time is suggested. If set to 8191.875, then the MIL light will remain on once it is on, until the engine is turned off.

MIL_SW must be set to 1 for production calibration.

PROCESS:

STRATEGY MODULE: VC_MIL_SD_COM1

MILTMR IS A FREE-RUNNING INCREMENTING TIMER

```
NO_START = 1 ------------|
(KOEO VIP)               |
                         |OR ------------| EXIT
RUNNING = 1 -------------|               |
 (ENGINE RUNNING VIP)                    | ---ELSE ---
                                         |
                                         | ZERO MILTMR
STIFLG = 1 ------------------------------|
 (STI GROUNDED, NOW IN WIGGLE MODE)      |
                                         |
                                         | ---ELSE ---
CRKFLG = 1 --------------|               |
 (IN CRANK)              |               |
                         |AND------------| DO BULB CHECK
DISABLE_NOSTART = 0 -----|               | (TURN STO ON)
 (KOEO VIP NOT REQUESTED)|               | ZERO MILTMR
                         |               |
FIRST_PIP = 0 -----------|               |
 (PIP NOT YET RECEIVED)  |               |
                         |               |
MILLIM = 1 --------------|               | ---ELSE ---
 (BULB CHECK REQUIRED)                   |
                                         |
MIL_SW = 0 --------------|               | TURN MIL OFF
                         |               |  (STO OFF)
CRKFLG = 1 --------------|OR ------------| ZERO MILTMR
                         |               |
CHECK MIL FAULT NOT      |               |
 PRESENT LOGIC ----------|               | ---ELSE ---
 (MIL FAULT IS NOT PRESENT)              |
                                         |
                                         | TURN MIL OFF
MILTMR <= FMDTM -------------------------| (STO OFF)
                                         |
                                         |
                                         | ---ELSE ---
                                         |
                                         | TURN MIL ON
                                         |  (STO ON)
                                         |
```

MIL FAULT NOT PRESENT LOGIC

```
          AFMFLG = 0 --------------|
           (ACT)                   |
          CFMFLG = 0 --------------|
           (ECT)                   |         | MIL FAULT IS NOT
          EFMFLG = 0---------------|         |  PRESENT
           (EGR)                   |         |
          MFMFLG = 0 --------------|AND ----|
           (MASS AIR FLOW OR MAP)  |         | ---ELSE---
          TFMFLG = 0 --------------|         |
           (TP)                    |         | MIL FAULT IS
          EGO1FMFLG = 0 -----------|         |  PRESENT
          ADT1FMFLG = 0 -----------|
           (ADAPTIVE FUEL)         |
          C332FIL <= C332LVL ------|
           (EGR FLOW)              |
                                   |
MIL_SW <> 2 -----------|           |
                       |OR---------|
CXXXFIL<= CXXXLVL -----|           |
 (NO OTHER CONTINUOUS FAULTS)      |
                                   |
                                   |
MILTMR <= FMDTM--------|           |
                       |OR --------|
MILTMR >= FMDTM +      |           |
 V_MILONTM ------------|           |
  (MIL ON MINIMUM TIME)
```

CHAPTER 30

ERROR CODE DESCRIPTION

ERROR CODE DESCRIPTION

ERROR CODE	DESCRIPTION	SELF TEST SECTION		
		K.O.E.O	E.R.	CONT.
111	PASS	X	X	X
112	254 deg. ind. ACT-ckt. grounded.	X	XX	X
113	-40 deg. ind. ACT-sensor ckt. open.	X	XX	X
114	ACT out of S-T range.	X	X	
116	ECT out of S-T range.	X	X	
117	254 deg. ind. ECT-ckt. grounded.	X	XX	X
118	-40 deg. ind. ECT-sensor ckt. open.	X	XX	X
121	TP out of S-T range.	X	X	
122	TPS ckt. below minimum voltage.	X	XX	X
123	TPS ckt. above max. voltage.	X	XX	X
126	MAP/BP out of S-T range.	X	XX	X
128	MAP vacuum circuit failure			X
129	Insuff. MAP change-dyn resp. test.		X	
167	Insuff. TP change-dyn resp. test.		X	
172	EGO sensor ckt. ind. system lean.		X	X
173	EGO sensor ckt. ind. system rich.		X	X
179	Adaptive Fuel Limit Lean			X
181	Adaptive Fuel Limit Rich			X
182	Adaptive Fuel Limit Lean @idle			X
183	Adaptive Fuel Limit Rich @idle			X
194	Hego switch rate too fast			X
211	PIP ckt. fault.			X
212	Loss of tach input to processor.			X
213	Spark control fault present		X	
225	Knock not sensed-dyn response test.		X	
311	Themactor air system inop.		X	
312	Thermactor air upstream during S-T.		X	
313	Therm. air not bypassed during S-T.		X	
327	EPT/EVP below min. voltage.	X	X	X
328	EVP volt below closed lim (SONIC)	X	X	X
332	EGR valve not opening (SONIC).		X	X
334	EVP volt. above closed limit.	X	X	X
337	EVP ckt. above max volt.	X	X	X
338	Cooling System not Heating			X
339	Cooling System not Cooling			X
411	RPM not within S-T lower band limit		X	
412	RPM not within S-T upper band limit		X	
452	Insufficient input from VSS.			X

ERROR CODE	DESCRIPTION	SELF TEST SECTION K.O.E.O.	E.R.	CONT.
511	ROM test failed	X		
512	KAM Test Failed			X
513	Failure in EEC ref. voltage.	X		
519	PSPS ckt. open.	X		
521	PSPS did not change states.		X	
536	BOO Sw. Ckt failed open/closed -ECA input open or brake not actuated during test		X	
538	Operator error-dyn response test.		X	
539	A/C Swith error	X		
542	FP Ckt Open -ECA to Motor Ground	X		X
543	FP Ckt Open Bat. to Relay	X		X
552	Air Management 1 (AM1) ckt. failure	X		
553	Air Management 2 (AM2) ckt. failure	X		
556	Fuel pump ckt. failure.	X		X
558	Elect. vac. reg. (EVR) ckt. failure	X		
565	Canister Purge (CANP) ckt. failure.	X		
617	1-2 Shift Error			X *
618	2-3 Shift Error			X *
619	3-4 Shift Error			X *
621	SS1 Sol Ckt Failure	X		
622	SS2 Sol. Ckt. Failure	X		
624	EPC Solenoid Circuit Failure/Shor- Output Driver	X		X
626	CCS Sol Ckt Failure	X		
628	Converter Clutch Failure			X
629	CCC Sol. Ckt. Failure	X		
631	OCIL Ckt. Failure	X		
632	OCS Not Changing State		X	
633	4X4 Switch closed	X		
634	MLPS Out of Range			X
636	TOT Out of S-T range	X	X	
637	-40 deg. ind. TOT sensor ckt open	X		X
638	315 Deg ind. TOT sensor ckt grounded	X		X
654	MLPS not in park	X		
655	MLPS not in neutral	X		
998	FMEM failure/Failed (open)EPC Output Driver	X	XX	

XX - Service Code 998 and corresponding code(s) are output which constitute FMEM mode failure.

* SEE BASE STRATEGY FOR TEST DOCUMENTATION

CHAPTER 31

ROM IDENTIFICATION CODE

ROM IDENTIFICATION CODE

ROM identification codes are used by both IC and module suppliers. The IC suppliers require a means of identifying ROM chip contents quickly since they produce different calibration bit patterns on the same wafer. The module suppliers utilize these codes to insure that the ROM/module combination is correct.

In the past, the ROM identification codes (CALID and VERID) were generated by hand. They were then distributed to Engine Systems to put in their calibrations for Cert. If any change to the Cert calibration was made or a different strategy used, new values had to be generated and calibrated in.

For 1988 and beyond, the procedure has been changed to make this process easier. The new process removes CALID and replaces it with ROM_TO. In addition, VERID has been deleted and a new parameter "FIXSUM" has been added. FIXSUM should always be set to 0. Specifically:

1. The non-modifiable Vector parameter "ROM_TO" replaces the old CALID parameter as the ROM chip identifier. The ROM_TO value is generated by Vector during a calibration release and is located at 200A HEX. This value is the complement of the ROM pattern CHECKSUM and is also used to perform the EEC-IV diagnostic "CHECKSUM Memory Test".

2. The new parameter "FIXSUM" is a Vector calibration parameter located at 2004 HEX and should always be set to 0. This parameter will be used to assure the ROM_TO values are unique and will only be changed by the SWDV engineer if a duplicate ROM_TO value is found.

ROM IDENTIFICATION CODE PROCEDURES

1. PEDD SW will set the value of the calibration parameter FIXSUM to 0 in the base release.

2. When Engine Systems releases a CERT calibration, the ROM chip ID code, ROM_TO, will be automatically generated by VECTOR, and the value is to be recorded on the calibration release sheet submitted to SWDV.

3. PEDD SWDV will verify that the ROM_TO value is not the same as any other previous ROM_TO prior to sending the binary file to EED. If the ROM_TO value matches another, PEDD SWDV will change the value of FIXSUM and generate a new ROM_TO value, which will be checked again for a match. This process is repeated until a unique ROM_TO value is generated.

4. For production calibrations only, EED systems will receive and record the value of ROM_TO for final ROM verification.

5. EED will then transmit the binary file to the vendors, verify the ROM chip against the binary file, verify the checksum, verify the RAM read/write test, and verify the ROM_TO value and location.

INDEX

Please feel free to use this form to record faults with the book, and/or general comments and complaints about Strategy Group documentation. As we make improvements, we would also like to know about the things you would like to see in the strategy book.

This form, if used, should be directed to the responsible person listed on the cover of this document.

****

Do you have any general comments on this book, or its availability?

****

Specific errors found in this book. Please list the page on which the error was found, the error itself, and the correction if known.

****

Please indicate your area, and/or primary use made of this book.

If you would like to be notified when a change is made, or if you would like to discuss something, please write your name and extension below.

Name______________________ Ext_____________ Date___________________