

AD-A107 922

NAVAL RESEARCH LAB WASHINGTON DC

F/G 9/2

FUNCTION SPECIFICATIONS FOR THE A-7E FUNCTION DRIVER MODULE. (U)

NOV 81 P C CLEMENTS

UNCLASSIFIED

NL

$$\frac{1}{2} \times \frac{1}{2} = \frac{1}{4}$$

ADA107922

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER NRL Memorandum Report 4658	2. GOVT ACCESSION NO. AD-A107	3. RECIPIENT'S CATALOG NUMBER 922
4. TITLE (and Subtitle) FUNCTION SPECIFICATIONS FOR THE A-7E FUNCTION DRIVER MODULE		5. TYPE OF REPORT & PERIOD COVERED Interim report on a continuing NRL problem.
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) P.C. Clements		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Research Laboratory Washington, DC 20375		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 62721N; XF21242101; 75-0106-01
11. CONTROLLING OFFICE NAME AND ADDRESS		12. REPORT DATE November 27, 1981
		13. NUMBER OF PAGES 176
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Specifications Real-time systems Software documentation Software engineering Avionics software		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) As part of the experimental redesign of the flight software for the Navy's A-7E aircraft, software modules were designed to encapsulate the characteristics of the behavioral requirements of the system. The purpose of these Function Driver modules is to allow the remainder of the software to remain unchanged when the required system behavior is modified without associated hardware changes. This document specifies the behavior of the system without regard to specific hardware devices.		

(Continues)

DD FORM 1473
1 JAN 73

EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

20. ABSTRACT (Continued)

This report contains an explanation of the standard organization of each functional description, a description of the review procedures, specifications for all of the functional aspects of the A-7E software, and a set of indices and cross-references to help integrate this module with the rest of the system.

As well as serving as development and maintenance documentation for the A-7E redesign, this document is intended to serve as a model for other people interested in applying our documentation and structuring techniques to other software projects.

7

Preface

The Operational Flight Program (OFF) for the Navy's A-7E aircraft is considered a successful program: it works reliably. However, it is expensive to maintain because it has problems typical of much DoD software:

- * it barely meets its time and space limitations;
- * it is not fully understood by the maintenance personnel;
- * it is poorly documented; and
- * it is difficult to change.

Although various software engineering techniques have been proposed to deal with such problems, there is a widespread reluctance to modify or abandon current techniques which, despite the problems mentioned above, are the basis for many acceptable programs such as the A-7E OFF. Two reasons for this reluctance are:

- * No one has proven that the new techniques are useful in the DoD context of complicated requirements and stringent resource limitations.
- * There are no fully worked out examples that system developers can use as models in applying the new techniques to DoD systems.

In order to demonstrate feasibility and to provide a useful model, the Naval Research Laboratory (NRL) and the Naval Weapons Center (NWC) have embarked on a joint project in which certain software engineering techniques will be used to redesign and rebuild the A-7E OFF. Among the techniques being used are the following:

- * modularity and information hiding
- * formal specifications
- * abstract interfaces
- * cooperating sequential processes
- * process synchronization routines
- * resource monitors

Accession For	
NTIS GRA&I	☒
DTIC TAB	☐
Unannounced	☐
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A	

The first product of the project was /1/. It describes the externally visible behavior required of the A-7 OFP without describing an OFP implementation. The second product of the project was /2/. It describes abstract interfaces for hardware devices connected to the aircraft computer so that changes to those devices do not result in changes to the rest of the software.

The redesigned software is divided into modules according to the information-hiding principle. For each module, we will provide two types of documentation: interface specifications, showing the externally visible characteristics of the module, and abstract programs, showing the internal implementation decisions made for the module.

This report is the third published product of the project. It includes informal specifications for the function driver modules, which are responsible for causing the externally visible behavior of the system by invoking the requirements rules that specify how each system output is a function of the aircraft state.

This report is intended to serve as an example of good module documentation. It describes module interfaces without giving away implementation details. It also demonstrates the design of a systematic procedure for reviewing module interface designs.

Preface References

- /1/ Heninger, K., et al.; Software Requirements for the A-7E Aircraft, NRL Memorandum Report 3876; November 1978.
- /2/ Parker, A., et al.; Abstract Interface Specifications for the A-7E Device Interface Module, NRL Memorandum Report 4385; November 1980.

Acknowledgements

We would like to thank all the people at the Naval Weapons Center and at the Vought Corporation who helped us, but especially the following people who answered questions patiently and reviewed the document carefully.

Jack Basden
Glenn Cooper (Vought)
Sandra Fryer
Dwight Hill (Vought)
Mark Jacobson
Mike Maxfield
Lee Thomson
Jack Williams
Janice Zenor

We would also like to thank Kathryn Heninger Britton for her patient guidance, and Alan Parker for his expert help.

Table of Contents

	<u>Page</u>
FD.S0 <u>Standard Organization</u>	FD.S0-1
FD.1 <u>ADC Functions</u>	FD-1
FD.1.1 Set ADC sea level pressure: +FD_SET_ADC_SLP_D+	FD-1
FD.1.2 Set ADC reconfiguration values: +FD_ADC_RECONFIG_D+	FD-3
FD.2 <u>Audible Signal Functions</u>	FD-4
FD.2.1 Control audible signal: +FD_AUDSIG_SW_D+	FD-4
FD.2.2 Set audible signal beep rate: +FD_AUDSIG_BEEPRATE+	FD-7
FD.3 <u>Computer Fail Signal Functions</u>	FD-8
FD.3.1 Signal tactical computer failure: +FD_COMP_FAIL_SIG_D+	FD-8
FD.4 <u>Doppler Radar Functions</u>	FD-9
FD.4.1 Start and stop Doppler radar: +FD_DRS_CNTRL_D+	FD-9
FD.5 <u>Flight Information Display Functions</u>	FD-10
FD.5.1 ADI functions	
FD.5.1.1 Set ADI azimuth display: +FD_ADI_AZ_DISPLAY_P+	FD-10
FD.5.1.2 Set ADI elevation indicator: +FD_ADI_ELEV_P+	FD-13
FD.5.2 HSI functions	
FD.5.2.1 Set HSI pointer 1 and DME: +FD_HSI1_DME_P+	FD-15
FD.5.2.2 Set HSI pointer 2: +FD_HSI2_P+	FD-20

Table of Contents

	<u>Page</u>
FD.6 <u>Forward-Looking Radar (FLR) Functions</u>	FD-21
FD.6.1 Set FLR mode: +FD_FLR_MODE_D+	FD-21
FD.6.2 Control FLR range and azimuth cursors	
FD.6.2.1 Position FLR cursors: +FD_FLR_CURSOR_POSN_P+	FD-23
FD.6.2.2 Set az. cursor display mode	
: +FD_FLR_AZ_CURSOR_MODE_D+	FD-26
FD.6.3 Set FLR direction: +FD_FLR_DIRECTN_P+	FD-27
FD.6.4 Set FLR symbol blink rate: +FD_FLR_BLINKRATE+	FD-28
 FD.7 <u>Head-Up Display Functions</u>	 FD-29
FD.7.1 HUD Location-indicator Functions	FD-29
FD.7.1.1 Control HUD Aiming Symbol (AS)	FD-30
FD.7.1.1.1 Set HUD AS mode: +FD_HUD_AS_MODE_D+	FD-30
FD.7.1.1.2 Set HUD AS posn: +FD_HUD_AS_POSN_P+	FD-32
FD.7.1.2 Control HUD Azimuth Steering Line (ASL)	FD-37
FD.7.1.2.1 Set HUD ASL mode: +FD_HUD_ASL_MODE_D+	FD-37
FD.7.1.2.2 Set HUD ASL posn: +FD_HUD_ASL_POSN_P+	FD-38
FD.7.1.3 Control HUD Flight Director (FD)	FD-42
FD.7.1.3.1 Set HUD FD mode: +FD_HUD_FLTDIR_MODE_D+	FD-42
FD.7.1.3.2 Set HUD FD posn: +FD_HUD_FLTDIR_POSN_P+	FD-43
FD.7.1.4 Control HUD Flight Path Marker (FPM)	FD-44
FD.7.1.4.1 Set HUD FPM mode: +FD_HUD_FPM_MODE_D+	FD-44
FD.7.1.4.2 Set HUD FPM posn: +FD_HUD_FPM_POSN_P+	FD-45
FD.7.1.5 HUD in-range cue mode: +FD_HUD_RNGCUE_D+	FD-47
FD.7.1.6 Control HUD Lower Solution Cue (LSC)	FD-48
FD.7.1.6.1 Set HUD LSC mode: +FD_HUD_LSC_MODE_D+	FD-48
FD.7.1.6.2 Set HUD LSC posn: +FD_HUD_LSC_POSN_P+	FD-50
FD.7.1.7 Control HUD Pullup Anticipation Cue (PUAC)	FD-52
FD.7.1.7.1 Set HUD PUAC mode: +FD_HUD_PUAC_MODE_D+	FD-52
FD.7.1.7.2 Set HUD PUAC posn: +FD_HUD_PUAC_POSN_P+	FD-54
FD.7.1.8 Set HUD pullup cue mode: +FD_HUD_PUC_MODE_D+	FD-56

Table of Contents

	<u>Page</u>
FD.7.1.9 Control HUD Upper Solution Cue (USC)	FD-57
FD.7.1.9.1 Set HUD USC mode: +FD_HUD_USC_MODE_D+	FD-57
FD.7.1.9.2 Set HUD USC posn: +FD_HUD_USC_POSN_P+	FD-59
FD.7.1.10 Set HUD symbol blink rate: +FD_HUD_BLINKRATE+	FD-60
FD.7.2 HUD Value Indicators	FD-61
FD.7.2.1 Set HUD altitude display: +FD_HUD_BARO_ALT_P+	FD-61
FD.7.2.2 Set HUD heading display: +FD_HUD_HEADING_P+	FD-62
FD.7.2.3 Set HUD pitch/roll displays: +FD_HUD_PITCH_ROLL_P+	FD-63
FD.7.2.4 Control HUD vertical velocity/acceleration displays.	
FD.7.2.4.1 Enable vert. vel. and accel. displays: +FD_HUD_VV_MODE_D+	FD-64
FD.7.2.4.2 Set HUD vert. accel.: +FD_HUD_ACCEL_P+	FD-65
FD.7.2.4.3 Set vert. vel display: +FD_HUD_VERTVEL_P+	FD-66
 FD.8 <u>IMS Functions</u>	 FD-67
FD.8.1 Switch IMS computer control on/off: +FD_IMS_COMP_CTRL_D+	FD-67
FD.8.2 Set IMS velocity measurement scale: +FD_IMS_SCALE_D+	FD-68
FD.8.3 Adjust alignment of IMS platform x, y, and z axes	FD-69
FD.8.3.1 Small x and y adjustments: +FD_IMS_SMADJ_XY_P+	FD-70
FD.8.3.2 Large x and y adjustments: +FD_IMS_LGADJ_XY_D+	FD-72
FD.8.3.3 Adjust z axis: +FD_IMS_ADJ_Z_D+	FD-73
FD.8.4 Initialize IMS velocities: +FD_IMS_SET_VELS_D+	FD-78
FD.8.5 Set IMS reconfiguration values: +FD_IMS_RECONFIG_D+	FD-79
 FD.9 <u>Panel Functions</u>	 FD-81
FD.9.1 Set mark window display: +FD_PNL_MARK_D+	FD-81
FD.9.2 Control enter light: +FD_PNL_ENTER_D+	FD-83
FD.9.3 Display data in upper/lower window: +FD_PNL_DISPLAY_D+	FD-84

Table of Contents

	<u>Page</u>
FD.10 <u>PMDs Functions</u>	FD-101
FD.10.1 Set map indicator: +FD_MAP_IND_P+	FD-101
FD.10.2 Set map orientation angle: +FD_MAP_ORIENT_P+	FD-102
FD.10.3 Set map pointer: +FD_MAP_PTR_P+	FD-103
FD.10.4 Set map reference point: +FD_MAP_REF_PT_D+	FD-104
FD.10.5 Set map display: +FD_MAP_DISPLAY_P+	FD-105
FD.10.6 Set map scale: +FD_MAP_SCALE_D+	FD-108
FD.10.7 Set map reconfiguration values: +FD_MAP_RECONFIG_D+	FD-109
 FD.11 <u>SINS Functions</u>	 FD-111
FD.11.1 Start and stop SINS: +FD_SINS_CNTRL_D+	FD-111
 FD.12 <u>Visual Indicator Functions</u>	 FD-112
FD.12.1 "Auto-cal" indicator: +FD_AUTOCAL_IND_D+	FD-112
FD.12.2 "IMS Non-aligned" indicator: +FD_IMS_NA_IND_D+	FD-113
FD.12.3 Set visual indicator blink rates: +FD_VISIND_RATE_D+	FD-116
 FD.13 <u>Weapon Release Functions</u>	 FD-117
FD.13.1 Prepare and release a weapon: +FD_WEAP_RLS_D+	FD-117
 FD.14 <u>Ground Test Functions</u>	 FD-119
FD.14.1 Conduct the ground test.: +FD_GRTEST_D+	FD-119

Appendices

Appendix 1	Cross-Reference Between Functions of the Function Driver Module and the Requirements Document	FD.App1
Appendix 2	Timing Requirements	FD.App2
Appendix 3	List of Required Events, Ordered Alphabetically	FD.App3
Appendix 4	Dictionary of Required Input Items	FD.App4
Appendix 5	Review Procedures for the A-7 Function Driver Module	FD.App5

Function Driver Module: Standard Organization

References

- CLEM81 Clements, Interface Specifications for the A-7 Shared Services Module; draft in preparation; dated February, 1981.
- HENI78 Heninger, Kallander, Parnas, Shore, Software Requirements for the A-7E Aircraft; NRL Memorandum Report 3876; November, 1978.
- HENI81 Heninger, Weiss, Interface Specifications for the A-7 Extended Computer Module; draft in preparation; dated January, 1981.
- PARK80 Parker, Heninger, Parnas, Shore, Abstract Interface Specifications for the A-7E Device Interface Module; NRL Memorandum Report 4385; November, 1980.

This introduction defines and explains the format and conventions used for the functional specifications of the Function Driver (FD) modules. In the following, the terms "function", "Function Driver module", and "function driver" are all used interchangeably.

Each Function Driver module specification consists of the following:

1. Title line. This consists of the function number, a note indicating whether the function is demand or periodic, and a brief phrase explaining the function's purpose. The function number is of the form "FD.n.m"; 'n' is an integer from 1 to 14, and corresponds to a virtual device, as defined in Table FD.SO-a; 'm' is a sequence number to differentiate among functions that control the same virtual device.
2. Mnemonic. This is the function identifier, containing an abbreviated form of the device that the function drives. The dictionary of device abbreviations is in Table FD.SO-a. Each mnemonic is suffixed with either a '_D' (for a demand function), or '_P' (for a periodic function).

3. Output produced. Each function is responsible for generating some output value (usually just one). The value is used to direct some user-controllable feature of the Extended Computer (EC), or to drive a virtual device in the Device Interface Module (DIM). This section describes the value(s) generated, the data type(s), and the DIM or EC access program(s) to use.

4. Initiation/termination events (for periodic functions only). This section lists the events that must occur for the function to begin and end its operation, respectively. This information may be presented in the form of an event table. Sometimes, it is necessary to invoke a DIM access program at initiation or termination time. For instance, when a periodic function that updates a HUD symbol position terminates, the symbol might be removed from view. Therefore, invoking the program that removes the symbol is associated with the function's termination event. In such cases, where a program call is required concurrent with initiation or termination, the appropriate program is listed at the bottom of the event table.

5. Function definition. This section defines the output values as a function of conditions, events, and system modes. For periodic functions, this information is almost always in the form of a condition or selector table. For demand functions, this information is almost always in the form of an event table. These kinds of tables are described in Section 0.3 of HENI78; events are defined here as well. The modes in each row of the tables are listed alphabetically by class (e.g., navigation, alignment, weapon delivery, etc.), and modes in different classes are separated by a blank line.

For periodic functions, the specified conditions and input values should be re-evaluated once per execution cycle. For demand functions, which are driven by events, the output values that will be passed to the access programs should be derived or obtained only after the driving event occurs.

6. Local dictionary: If any local terms (denoted by "!!" brackets) have been used in a specification, then a local dictionary will be included to define each such term. Usually, a local term is created when a complex expression appears several times throughout a specification. When practical, the terms are ordered in the local dictionary so that no forward references are used. If a function has no local terms, there is no local dictionary.

7. Design issues: If any non-obvious design decisions were made during the creation of a specification, they are documented in this section. A typical design decision is why a certain function differs in organization or pairing from its Requirements Document counterpart. This section is omitted if no controversial decisions were made.

Overall assumptions:

The following are assumed by all Function Driver modules:

1. The virtual devices and their interfaces are accurately described in the Device Interface Module interface specifications (PARK80). The Extended Computer interfaces are accurately described in the Extended Computer interface specifications (HENI81). The services and values provided by the Shared Services module are accurately described in the Shared Services Module interface specifications (CLEM81). The values provided by the Data Banker are accurately described by the Data Banker Module interface specifications (to be drafted). The values and services provided by the Physical Models module are accurately described by the Physical Models Module specifications (to be drafted).
2. The values provided by the Data Banker are sufficiently accurate and current to allow the system requirements to be met.
3. An event is signalled soon enough after the event actually occurs to allow the system requirements to be met.
4. Every DIM or EC access program listed in the Output section of a Function Driver module specification is used by exactly one Function Driver module, and by no other module.
5. It is possible to avoid generating undesired events through incorrect use of the DIM or EC access programs. How to do so is conveyed in the DIM and EC specifications, respectively, and does not appear in the Function Driver material.
6. The event @T(!+Init complete+!) causes entry into some navigation or alignment mode.
7. The system cannot simultaneously be in two modes that appear in different rows in any condition table. The description of which modes the system may be in at the same time is described in the interface to the mode determination module in (CLEM81).

Note that the Function Driver programs will be considered correct even if any of the above assumptions are not true.

FD Standard Organization

Selector Table FD.SO-a: FD References to Virtual Devices

<u>n</u>	<u>Mnemonic Abbreviation</u>	<u>Virtual Device</u>
1	ADC	Air data computer
2	AUDSIG	Audible signal (Bomb tone)
3	COMP_FAIL	Computer fail signal
4	DRS	Doppler radar set
5	ADI, HSI, DME	Flight information displays
6	FLR	Forward-looking radar
7	HUD	Head-up display
8	IMS	Inertial measurement set
9	PNL	Computer panel
10	MAP	Projected map display set
11	SINS	Shipboard inertial navigation system
12	AUTOCAL, IMS_NA	Visual indicators
13	WEAP	Weapon release
14	GRTEST	Extended Computer self-test

Notation Conventions Used in Tables:

1. Factoring of values. " $!+Term+!$ = \$Value1\$ OR \$Value2\$" is written instead of " $!+Term+!$ = \$Value1\$ OR $!+Term+!$ = \$Value2\$".
2. Omission of boolean values. For boolean-valued terms, " $!+Term+!$ " is written instead of " $!+Term+!$ = true" and " $NOT !+Term+!$ " is written instead of " $!+Term+!$ = false".
3. Equivalent events. Let " B " be a boolean condition. The event $@F(A)$ is equivalent to the event $@T(\text{not } A)$, and may be written as such for contextual consistency.

Coordinate System

The airframe coordinate system has axes X_a , Y_a , and Z_a . The Y_a axis lies along the aircraft boresight line with the positive direction being forward (toward nose, from tail). The positive X_a axis points out in the direction of the right wing and is defined so that the X_a - Y_a plane is horizontal and the positive Z_a axis points upward when the aircraft wings are level and the aircraft is right side up.

```

*****
*
*           FD.1           ADC functions
*
*****

```

FD.1.1 DEMAND FUNCTION DESCRIPTION: Set the ADC sea level pressure.

Mnemonic: +FD_SET_ADC_SLP_D+

Output produced:

<u>Item</u>	<u>Type</u>	<u>Access program</u>
Sea level pressure	pressure	+DI_S_ADC_SLP+

Function definition:

Event Table FD.1.1-a -- Setting the ADC sea level pressure

MODES	EVENTS
	=====
	@T (!!destnot0 !!)
	OR
All align-	@T (!+Init complete+!)
ment and	WHEN (!!destnot0 !!)
navigation	OR
modes	@T (!!flyto nonzero !!)
	OR
	@T (!+New dest mslp pnl entered+!)
	WHEN (!!mark !! OR
	!!dest0 !!)
	WHEN (!+dest entry pnl+! =
	!+Fly to num+!)
	=====
Output value:	!!dest mslp !! 29.92" Hg

Local dictionary:

<u>Term</u>	<u>Definition</u>
!!dest mslp!!	!+dest mslp pnl+!, indexed by !+Fly to num+!.
!!dest0!!	!+Fly to state+! = \$Dest\$ AND !+Fly to num+! = 0
!!mark!!	!+Fly to state+! = \$Mark\$
!!destnot0!!	!+Fly to state+! = \$Dest\$ AND !+Fly to num+! noteq 0
!!flyto nonzero!!	@T(!+Fly to num changed+!) and the new !+Fly to num+! is not zero.

FD.1.2 DEMAND FUNCTION DESCRIPTION: Set the ADC Reconfiguration value.

Mnemonic: +FD_ADC_RECONFIG_D+

Output produced:

<u>Item</u>	<u>Type</u>	<u>Access program</u>
L-probe switch	boolean	+DI_S_ADC_LPROBE+

Function definition:

Event Table FD.1.2-a -- ADC reconfiguration value update

MODES

EVENTS

=====
All navigation
and alignment
modes

@T(!+New L-probe pnl entered+!)

=====
Output value:

!+L-probe pnl+!

```
*****
*
*           FD.2           Audible signal functions           *
*
*****
```

FD.2.1 DEMAND FUNCTION DESCRIPTION: Control the audible signal.

Mnemonic: +FD_AUDSIG_SW_D+

Output produced:

<u>Item</u>	<u>Type</u>	<u>Access program</u>
Audible signal mode	ind_cntrl	+DI_S_AUDIBLE_SIGNAL+

Function definition:

Event Table FD.2.1-a -- Audible signal control

MODES	EVENTS		
=====			
NBShrike	@T(!+RE pressed+!) WHEN(!+desig+!)	@F(!+RE pressed+!) OR @T(!+time to prepare+!) WHEN(!+desig+!)	X

NBnot-Shrike	@T(!+RE pressed+!) WHEN(!+desig+!)	@T(!+Rel in Progress+!) OR @F(!+RE pressed+!)	X

A/A Manrip			
CCIP	@T(!+RE pressed+!)	@F(!+RE pressed+!)	X
Manrip			

A/G Guns	@T(!+RE pressed+!) WHEN(!+Weapon Class+! = \$RK\$)	@T(!+Rel in Progress+!) OR @F(!+RE pressed+!)	X

Walleye	@T(!+RE pressed+!)	@T(!+time tone on!! gteq 1 sec)	X

LoNuke	@T(!+Rmax+!)	@T(!+time since Rmax!! gteq 1 second)	@T(!+R65+!)
	OR		
	@T(!+Rmin+!)	OR	
	OR	@T(!+Rmin+6000+!)	
	@T(!+RE pressed+!)	OR	
	WHEN(!+desig+!)	@T(!+pitch IMS+! gt 15° AND NOT !+RE pressed+!)	
		OR @T(!+Rel in Progress+!)	
	OR @T(!+time beeped!! gt 2 sec)		

HiNuke	@T(!+RE pressed+!) WHEN(!+desig+!)	@T(!+RE pressed+!)	
		OR	
		@T(!+Rel in Progress+!)	
		OR @T(!+time beeped!! gt 2 sec)	
=====			
Output value:	\$On\$	\$Off\$	\$Intermittent\$

Local dictionary:Term

!!time beeped!!

Definitionelapsed time since last call to
+DI_S_AUDIBLE_SIGNAL+(\$Intermittent\$)

!!time tone on!!

elapsed time since last call to
+DI_S_AUDIBLE_SIGNAL+(\$On\$)

!!time since Rmax!!

elapsed time since @T(!+Rmax+!) occurred.

Design issues:

1. Whether to set the audible signal's beep rate before turning the signal on intermittently. It was decided not to do so, because under the current requirements, this is the only function to cause the signal to beep; therefore, the signal is only beeped at one rate (the system default rate), which need not be changed.

FD.2.2 FUNCTION DESCRIPTION: Set the audible signal beep rate.

Mnemonic: +FD_AUDSIG_BEEPRATE+

Output produced:

<u>Item</u>	<u>Type</u>	<u>Access program</u>
New beep rate	real	+DI_S_BEEP_RATE+

Function definition:

This function is never performed, because under the current requirements, the beep rate never needs to be changed from its default value. However, should the requirements change and another beep rate be called for, this function would have the responsibility for setting the new value.

```

*****
*
*           FD.3           Computer fail signal functions
*
*
*****

```

FD.3.1 DEMAND FUNCTION DESCRIPTION: Signal tactical computer failure.

Mnemonic: +FD_COMP_FAIL_SIG_D+

Output produced:

<u>Item</u>	<u>Type</u>	<u>Access program</u>
Computer fail signal	boolean	+DI_S_COMPUTER_FAIL_SIGNAL+

Function definition:

Event Table FD.3.1-a -- Signalling computer failure

<u>MODES</u>	<u>EVENTS</u>
=====	=====
All modes	@T(!+Failed state+!) @T(!+Init complete+!)
=====	=====
Output value:	<u>true</u> <u>false</u>

```

*****
*
*           FD.4           Doppler radar functions           *
*
*****

```

FD.4.1 DEMAND FUNCTION DESCRIPTION: Start and stop the Doppler radar.

Mnemonic: +FD_DRS_CNTRL_D+

Output produced:

<u>Item</u>	<u>Type</u>	<u>Access program</u>
To start the DRS	--	+DI_START_DRS+
To stop the DRS	--	+DI_STOP_DRS+

Function definition:

Event Table FD.4.1-a -- Turning the Doppler radar on and off

<u>MODES</u>	<u>EVENTS</u>	
=====	=====	=====
All alignment and navigation modes	@T(!+In flight+!)	@F(!+In flight+!)
=====	=====	=====
Function call:	+DI_START_DRS+	+DI_STOP_DRS+

```
*****
*
*          FD.5          Flight information display functions
*
*****
```

FD.5.1.1 PERIODIC FUNCTION DESCRIPTION: Set ADI azimuth indicator display.

Mnemonic: +FD_ADI_AZ_DISPLAY_P+

Output produced:

<u>Item</u>	<u>Type</u>	<u>Access program</u>
ADI azimuth display	angle	+DI_S_ADI_AZIMUTH_INDICATOR+

Initiation/termination events:

None. Always performed.

Function definition:

Condition Table FD.5.1.1-a -- Setting the ADI azimuth display

MODES	CONDITIONS		
=====			
Airaln			
All nav modes;	NOT !+in_flight+!	!+in_flight+!	X
	OR	AND	
No mode listed below;	!+fly to num+! = 0	!+fly to num+!	
		noteq 0	
Walleye			

HUDDown1	NOT !+desig+!	NOT !+desig+!	!+desig+!
Nattack	AND	AND	
SHUDDown1	!+fly to num+! = 0	!+fly to num+!	
Snattack		noteq 0	

HUDDown2			
Noffset	X	NOT !+desig+!	!+desig+!
SHUDDown2			
Snoffset			

BOC	X	X	Always
SBOC			

BOCFlyto0	NOT !+desig+!	X	!+desig+!
SBOCFlyto0			

BOCOffset	X	NOT !+desig+!	!+desig+!
SBOCOffset			
=====			
OUTPUT VALUE:	0	!!steering error to ftpt!!	!+steering error to tgt+!

Local dictionary:

<u>Term</u>	<u>Definition</u>
!!steering error to ftpt!!	!+brg_grtk_tgt+!, if 0 lseq !+brg_grtk_tgt+! lseq 180 degrees; (!+brg_grtk_tgt+! - 360 degrees) if !+brg_grtk_tgt+! gt 180 degrees.

Design issue:

The Requirements dictate that the output value must be between -2.5 degrees and +2.5 degrees, as these are the limits of the display. This Function Driver does not perform this limiting, because the DIM's virtual ADI does that check and limiting automatically anyway. To perform the calculation here would be to invite duplicate code.

FD.5.1.2 PERIODIC FUNCTION DESCRIPTION: Set the ADI elevation indicator.

Mnemonic: +FD_ADI_ELEV_P+

Output produced:

<u>Item</u>	<u>Type</u>	<u>Access program</u>
ADI elevation	angle	+DI_S_ADI_ELEV_INDICATOR+
indication		
To display or remove	boolean	+DI_S_ADI_ELEV_IN_VIEW+
indicator from view		

Initiation/termination events:

Event Table FD.5.1.2-a -- When ADI elevation indicator is updated

MODES	INITIATION EVENTS	TERMINATION EVENTS
=====		
IMS fail		
All alignment	X	@T(In mode)
modes except		
Airaln		
NBShrike		

HiNuke		
NBnotShrike	@F(,BS(!+LSC elevation+!)	@T(ABS(!+LSC elevation+!)
Walleye	gt 4 degrees)	gt 4 degrees)

LoNuke	@F(ABS(!+PUAC elevation+!)	@T(ABS(!+PUAC elevation+!)
	gt 4 degrees)	gt 4 degrees)
=====		
Function call:	+DI_S_ADI_ELEV_IN_VIEW+	+DI_S_ADI_ELEV_IN_VIEW+
	(true)	(false)

The function also terminates when @F(!+ADI elev avail+!) occurs. No function call is associated with this termination event.

Function definition:

Condition Table FD.5.1.2-b -- Setting the ADI elevation display

MODES		CONDITIONS	
=====			
HiNuke			
NBnotShrike	Always	X	X
Walleye			

LoNuke	X	NOT !!ac inverted!!	!!ac inverted!!
=====			
Output value:	!+LSC elevation+!	!+PUAC elevation+!	- !+PUAC elevation+!

Local dictionary

<u>Term</u>	<u>Definition</u>
!!a/c inverted!!	ABS(!+roll IMS+!) gt 90 degrees

FD.5.2.1 PERIODIC FUNCTION DESCRIPTION: Set HSI pointer 1 and DME display.

Mnemonic: +FD_HSI1_DME_P+

Output produced:

<u>Item</u>	<u>Type</u>	<u>Access program</u>
HSI 1 display	angle	+DI_S_HSI_POINTER 1+
DME display	integer	+DI_S_DME_DISPLAY+
To show/hide the DME flag	boolean	+DI_S_DME_FLAG+

Initiation/termination events:

None. Always performed.

Function definition:

Condition Table FD.5.2.1-a -- Setting HSI pointer 1 and DME displays

MODES	CONDITIONS	
=====		
All alignment and navigation modes with no modes listed below;	!+fly to num+! = 0	!+fly to num+! noteq 0
A/G Guns		
Walleye		

HUDDown1	NOT !+desig+!	!+desig+!
Nattack	AND	OR
SHUDDown1	!+fly to num+! = 0	!+fly to num+! noteq 0
Snattack		

HUDDown2		
Noffset	X	Always
SHUDDown2		
Snoffset		

BOC	X	Always
SBOC		

BOCFlyto0	NOT !+desig+!	!+desig+!
SBOCFlyto0		

BOCOffset	X	Always
SBOCOffset		
=====		
HSI pointer 1:	0 degrees	!+brg_grtk (!!refpt!!)+!
DME setting:	0	!!DME integer display!!

Event Table FD.5.2.1-b -- When to display/remove the DME flag

MODES	EVENTS	
All navigation or alignment modes when not in a mode listed below	@T(!+gr_ac_(!refpt!))+! gteq 1000 nmi)	@T(!+gr_ac_(!refpt!))+! ls 1000 nmi)
BOC *BOCFlyto0* *BOCoffset* *HUDDown1* *HUDDown2* *Nattack* *Noffset* *SBOC* *SBOCFlyto0* *SBOCoffset* *SHUDDown1* *SHUDDown2* *Snattack* *Snoffset*	@T(!+desig+! AND !+gr_ac_(!refpt!))+! gteq 10 nmi) OR @F(!+desig+!)	@T(!+desig+! AND !+gr_ac_(!refpt!))+! ls 10 nmi)
Function call:	+DI_S_DME_FLAG+(<u>true</u>)	+DI_S_DME_FLAG+(<u>false</u>)

Local dictionary:

<u>Term</u>	<u>Definition</u>
!!DME integer display!!	The value of !+gr_ac (!!refpt!!)+!, converted to an integer. Whether the integer should be the number of feet or nautical miles in the distance is determined by Table FD.5.2.1-c below.

Condition Table FD.5.2.1-c -- Units of measurement for DME display

MODES	CONDITIONS	
=====		
All navigation or alignment modes when not in a mode listed below	Always	X

BOC		
BOCFlyto0		
BOCOffset		
HUDDown1		
HUDDown2	!+desig+! AND	!+desig+! AND
Nattack	!+gr_ac (!!refpt!!)+!	!+gr_ac (!!refpt!!)+!
Noffset	gteq 10 nmi	ls 10 nmi)
SBOC		
SBOCFlyto0	OR	
SBOCOffset		
SHUDDown1	NOT !+desig+!	
SHUDDown2		
Snattack		
Snoffset		
=====		
Units:	nautical miles	thousands of feet

Local dictionary (continued):

<u>Term</u>	<u>Definition</u>
!!refpt!!	Defined by table below. The mnemonic abbreviations are also given in parentheses for each reference point. Thus, for instance, when !!refpt!! is defined to be !target!, the quantity !+brg_grtk_(!refpt!)+! actually refers to !+brg_grtk_tgt+!.

Condition Table FD.5.2.1-d -- Definition of !!refpt!! for HSI and DME

MODES	CONDITIONS		
=====			
All alignment and navigation modes, with no mode listed below;	Always	X	X
A/G Guns			
Walleye			

HUDDown1			
Nattack	NOT !+desig+!	X	!+desig+!
SHUDDown1			
Snattack			

HUDDown2	NOT !+desig+!	NOT !+desig+!	!+desig+!
Noffset	AND	AND	
SHUDDown2	!+before_	!+after_	
Snoffset	slewing+!	slewing+!	

BOC	!+before_		!+after_
SBOC	slewing+!		slewing+!
	OR	X	AND
	!+gr_ac_ftpt+!		!+gr_ac_ftpt+!
	gt 30 nmi		lseq 30 nmi

BOCFlyto0	X	X	Always
SBOCFlyto0			

BOCOffset	!+gr_ac_ftpt+!	!+gr_ac_ftpt+!	
SBOCOffset	gt 30 nmi	lseq 30 nmi	!+desig+!
	AND	AND	
	NOT !+desig+!	NOT !+desig+!	
=====			
!!refpt!!:	!Fly-to-point!	!OAP!	!target!
abbreviation:	ftpt	oap	tgt

FD.5.2.2 PERIODIC FUNCTION DESCRIPTION: Set HSI pointer 2.Mnemonic: +FD_HSI2_P+Output produced:

<u>Item</u>	<u>Type</u>	<u>Access program</u>
HSI indicator 2 setting	angle	+DI_S_HSI_POINTER_2+

Initiation/termination events:

Event Table FD.5.2.2-a -- When HSI indicator 2 is updated

MODES	Initiation events	Termination events
=====	=====	=====
All align and nav modes except *SINSaln*	@T(!+In flight+!)	@F(!+In flight+!)

SINSaln	@T(In mode)	@F(!+align_stage+! = \$CA\$)
=====		

At termination, 0 degrees should be displayed.

Function definition:

Selector Table FD.5.2.2-b -- Setting HSI indicator 2

MODES	VALUE
=====	=====
SINSaln	Oscillated between 0 and 11.3 degrees each second; displays 11.3 degrees for .6 seconds, and then 0 degrees for .4 seconds.

All alignment and nav modes except *SINSaln*	!+grtk+!
=====	

```
*****
*
*           FD.6           Forward-looking radar functions
*
*****
```

FD.6.1 DEMAND FUNCTION DESCRIPTION: Set the FLR mode.

Mnemonic: +FD_FLR_MODE_D+

Output produced:

<u>Item</u>	<u>Type</u>	<u>Access program</u>
FLR mode	flr_mode	+DI_S_FLR_MODE+

Function definition:

Event Table FD.6.1-a -- Setting the FLR mode

MODES		EVENTS	
=====			
RadarUpd	@T(In mode AND !+gr_ac_fxpt+! lseq 22 nmi)	X	X

BOC	@T(In mode AND		
BOCOffset	!+gr_ac_ftpt+!	X	X
SBOC	lseq 20 nmi)		
SBOCOffset			

BOCFlyto0	@T(In mode)	X	X
SBOCFlyto0			

!*None*!	X	X	@T(In mode)

HUDUpd			
CCIP			
HUDdown1			
HUDdown2			
Nattack	X	@T(In mode)	X
Noffset			
SHUDdown1			
SHUDdown2			
Snattack			
Snoffset			

A/G Guns		@T(!+Gun Enabled+!)	
		OR	
	X	@T(In mode AND	@F(In mode)
		!+Weapon Class+!	
		= \$RK\$)	
=====			
Output value:	\$CDCE\$	\$Ranging\$	\$Idle\$

Local dictionary:

Term	Definition
!*None*!	The state when the system is in none of the other modes listed in the Function Definition table.

FD.6.2.1 PERIODIC FUNCTION DESCRIPTION: Position the FLR azimuth and range cursors.

Mnemonic: +FD_FLR_CURSOR_POSN_P+

Output produced:

<u>Item</u>	<u>Type</u>	<u>Access program</u>
FLR az cursor posn	angle	+DI_S_FLR_AZ_CURSOR_POSN+
FLR rng cursor posn	distance	+DI_S_FLR_RANGE_CURSOR+

Initiation/termination events (for placing FLR range cursor):

Initiation: @T(!+FLR mode+! = \$CDCE\$)
 Termination: @F(!+FLR mode+! = \$CCDE\$)

Initiation/termination events (for placing FLR azimuth cursor):

Initiation: @T(!+FLR az cursor mode+! = \$On\$)
 Termination: @T(!+FLR az cursor mode+! = \$Off\$)

Function definition:

Condition Table FD.6.2.1-a -- FLR cursor positioning

MODES	CONDITIONS		
=====			
BOC			
BOCOffset			
SBOC	Always	!+during	X
SBOCOffset		slewing+!	
RadarUpd			

BOCflyto0	!+desig+!	!+during	NOT
SBOCflyto0		slewing+!	!+desig+!
=====			
FLR az cursor posn:	!!ltd brg grtk refpt!!	!!Az slew posn!!	0 deg.
FLR rng cursor posn:	!+sr ac (!!refpt!!)+!	!!Rng slew posn!!	8 nmi

Local dictionary:TermDefinition

!!Az slew posn!!

The new position of the azimuth cursor, computed by adding !+slew FLR delta az+! to the previous position of the azimuth cursor.

!!Rng slew posn!!

The new position of the range cursor, computed by adding !+slew FLR delta rng+! to the previous position of the range cursor.

To obtain !+slew FLR delta az+! and !+slew FLR delta rng+!, call +SS_SLEW_FLR+ (!+Slew right-left+!, !+slew FLR delta az+!, !+Slew up-down+!, !+slew FLR delta rng+!).

!!ltd brg grtk refpt!!

Under some circumstances, the FLR azimuth cursor is positioned at the left or right screen edge, as defined in the table below.

Selector Table FD.6.2.1-b -- Definition of !!ltd brg grtk refpt!!

!+brg_grtk_(!+refpt!!)+!
(in degrees)

!!ltd brg grtk refpt!!

=====

gteq 270 AND lseq 315

!+Az cursor lft max+!

gt 45 AND lseq 90

!+Az cursor rgt max+!

gt 315 OR ls 45

!+brg_grtk_(!+refpt!!)+!

Local dictionary (continued):

<u>Term</u>	<u>Definition</u>
!!refpt!!	Defined by table below. The mnemonic abbreviations are also given in parentheses for each reference point. Thus, for instance, when !!refpt!! is defined to be !target!, the quantity !+sr_ac_(!refpt!)+! actually refers to !+sr_ac_tgt+!.

Condition Table FD.6.2.1-c -- Definition of !!refpt!!

MODES		CONDITIONS			
=====					
BOC	NOT	!+desig+!	X	X	X
SBOC	!+desig+!				

BOCFlyto0	X	!+desig+!	X	X	X
SBOCFlyto0					

BOCOffset	NOT		!+desig+!		
	!+desig+!	X	OR	X	X
SBOCOffset	AND		!+after_		
	!+before_		slewing+!		
	slewing+!				

RadarUpd	X	X	X	!+before_	!+after_
				slewing+!	slewing+!
=====					
!!refpt!!:	!FLY-TO- POINT! (ftpt)	!target! (tgt)	!OAP! (oap)	!Called- up point! (cup)	!adjusted point! (ap)

FD.6.2.2 DEMAND FUNCTION DESCRIPTION: Set FLR azimuth cursor display mode.

Mnemonic: +FD_FLR_AZ_CURSOR_MODE_D+

Output produced:

<u>Item</u>	<u>Type</u>	<u>Access program</u>
FLR az cursor mode	ind_cntrl	+DI_S_FLR_AZ_CURSOR_MODE+

Function definition:

Event Table FD.6.2.2-a -- Setting the FLR azimuth cursor display mode

MODES	EVENTS
=====	
BOC	
BOCoffset	@T(!+FLR mode+! =
SBOC	@T(In mode AND \$CDCE\$ AND X
SBOCoffset	NOT !!refpt ahead!!) !!refpt ahead!!)
RadarUpd	
=====	
BOCflyto0	@T(!+desig+! AND @T(!+FLR mode+! = X
SBOCflyto0	NOT !!refpt ahead!!) \$CDCE\$ AND
	!!refpt ahead!!)
=====	
Output value:	\$Off\$ \$On\$ \$Intermittent\$

Under the current requirements, the FLR azimuth cursor mode is never \$Intermittent\$.

Local dictionary:

<u>Term</u>	<u>Definition</u>
!!refpt!!	Defined by Table 6.2.1-c.

!!refpt ahead!! (!+ !!refpt!! ahead+!), where !!refpt!! is one of ap, cup, ftpt, oap, or tgt, as defined by Table 6.2.1-c.

FD.6.3 PERIODIC FUNCTION DESCRIPTION: Set the FLR direction.Mnemonic: +FD_FLR_DIRECTN_P+Output produced:

<u>Item</u>	<u>Type</u>	<u>Access program</u>
FLR elevation	angle	+DI_S_FLR_DIRECTION+(elevation,
FLR azimuth	angle	azimuth)

Initiation/termination events:

Initiation: @T(!+FLR mode+! = \$Ranging\$)
 Termination: @F(!+FLR mode+! = \$Ranging\$)

Function definition:

Selector Table FD.6.3-- FLR direction

MODES	FLR elevation	FLR azimuth
HUUpd		
A/G Guns		
HUDown1	!+AS elevation+!	!+AS azimuth+!
HUDown2		
Nattack		
Noffset		
SHUDown1		
SHUDown2		
Snattack		
Snoffset		
CCIP	!+LSC elevation+!	!+LSC azimuth+!

FD.6.4 FUNCTION DESCRIPTION: Set the FLR symbol blink rate.

Mnemonic: +FD_FLR_BLINKRATE+

Output produced:

<u>Item</u>	<u>Type</u>	<u>Access program</u>
New beep rate	real	+DI_S_FLR_BLINK_RATE+

Function definition:

This function is never performed, because under the current requirements, the FLR symbol blink rate need never be changed from its default value. However, should the requirements change and another rate be called for, this function would have the responsibility for setting the new value.

```

*****
*
*           FD.7           Head-up display functions
*
*
*****

```

FD.7.1 -- HUD Location-indicator Functions

These functions control the display of symbols on the HUD. Most symbols give information concerning the location of the aircraft relative to some external point. One symbol represents the velocity vector of the aircraft. No function controls more than one symbol. The display mode of all symbols is controlled by the software. A symbol's display mode is either on steady, on intermittently, or off. The position of most symbols is controlled by the software. A symbol-positioning function is initiated when the display mode of its symbol becomes \$On\$ or \$Intermittent\$; it is terminated when the display mode of its symbol becomes \$Off\$.

The input items of the form !+xxx_elev+! and !+xxx_az+! (where 'xxx' represents a reference point, defined in the local dictionary of the function where it is used) define angles from the aircraft boresight to that point on the earth. The HUD is mounted so that displaying a symbol at a particular azimuth and elevation will cause that symbol to overlay a point on the ground at the same azimuth and elevation relative to the a/c Ya axis.

The functions are ordered alphabetically by the name of the symbol they control. For each symbol, there may be up to three possible concerns: the symbol's mode, the symbol's position, and movement of the symbol by slewing. For each symbol, there is a separate function for each applicable concern.

FD.7.1.1 -- Control the HUD Aiming Symbol.

The HUD Aiming Symbol is a symbol used to specify the location of a certain point outside the aircraft. Under certain conditions, the program places the symbol so that it overlays a reference point. Under other conditions, the pilot may move the symbol via the slew control, thus communicating the location of a certain point to the program.

FD.7.1.1.1 DEMAND FUNCTION DESCRIPTION: Set the HUD aiming symbol mode.

Mnemonic: +FD_HUD_AS_MODE_D+

Output produced:

<u>Item</u>	<u>Type</u>	<u>Access program</u>
AS mode	ind_cntrl	+DI_S_HUD_AS_MODE+

Function definition:

Event table FD.7.1.1.1-a -- Setting the HUD aiming symbol mode

MODES		EVENTS	
=====			
Hudaln			
A/A Guns			
A/A Manrip			
A/G Guns			
BOCFlyto0			
HUDDown1	@T(In mode)	X	X
HUDDown2			
Nattack			
Noffset			
SBOCFlyto0			
SHUDDown1			
SHUDDown2			
Snattack			
Snoffset			
BOC	@T(In mode AND !+gr_ac_HUDrefpt+! lseq 30 nmi)	@T(In mode AND !+gr_ac_HUDrefpt+! gt 30 nmi)	X
BOCOffset			
SBOC	@T(In mode AND !+gr_ac_HUDrefpt+! lseq 30 nmi)	@T(In mode AND !+gr_ac_HUDrefpt+! gt 42 nmi)	X
SBOCOffset			
RadarUpd	@T(In mode AND !+gr_ac_HUDrefpt+! lseq 20 nmi)	@T(In mode AND !+gr_ac_HUDrefpt+! gt 20 nmi)	X
HUDDUpd	@T(In mode AND !+gr_ac_HUDrefpt+! lseq 22 nmi)	@T(In mode AND !+gr_ac_HUDrefpt+! gt 22nmi)	X
Walleye	@T(In mode)	@T(!+RE pressed+!)	X
Grtest	@T(!+test_stage+! = \$SC\$)	@T(In mode)	X
No other mode listed above	X	@T(In mode)	X
=====			
Output			
value:	\$On\$	\$Off\$	\$Intermittent\$

Under the current requirements, the HUD aiming symbol mode is never \$Intermittent\$.

FD.7.1.1.2 PERIODIC FUNCTION DESCRIPTION: Set HUD aiming symbol position.

Mnemonic: +FD_HUD_AS_POSN_P+

Output produced:

<u>Item</u>	<u>Type</u>	<u>Access program</u>
AS elevation	angle	+DI_S_HUD_AS_POSITION+(AS elevation,
AS azimuth	angle	AS azimuth)

Function definition:

Condition Table FD.7.1.1.2-a

Where to position the HUD aiming symbol in alignment and test modes

=====

HUDaln	!+after slewing+!	!+before slewing+!	!+during slewing+!
----------	-------------------	--------------------	--------------------

=====

Grtest	X	Always	X
----------	---	--------	---

=====

HUD AS elev:	!+HUDrefpt_elev+!	0 deg.	!!Slewed AS elev!!
HUD AS az:	!+HUDrefpt_az+!	0 deg.	!!Slewed AS az!!

=====

Condition Table FD.7.1.1.2-b
Where to position the HUD aiming symbol in weapon and update modes

MODES		CONDITIONS		
=====				
Nattack	!!SK!! AND NOT !+desig+! AND (!!desig retent!! OR NOT !+during slewing+!)	NOT !!SK!! AND NOT !+desig+! AND (!!rls imminent!! OR NOT !+during slewing+!)	!+desig+! AND (!!rls imminent!! OR NOT !+during slewing+!)	(NOT !!rls imminent!! AND !+during slewing+! AND NOT !!SK!!) OR (!+during slewing+! AND !!desig retent!! AND !!SK!!)

HUDDown1	!!SK!! AND NOT !+desig+!	NOT !!SK!! AND NOT !+desig+!	!+desig+!	X

SHUDDown1	X	NOT !+desig+!	!+desig+!	X

Snattack	X	NOT !+desig+! AND (NOT !+during slewing+! OR !!rls imminent!!)	!+desig+! AND (NOT !+during slewing+! OR !!rls imminent!!)	NOT !!rls imminent!! AND !+during slewing+!

HUDDown2		NOT !+desig+!	!+desig+!	
SHUDDown2	X	AND !+before slewing+!	OR !+after slewing+!	X
=====				
HUD AS elev:	!+boresight elevation+!	!+FPM elevation+!	!+HUDrefpt _elev+!	!!Slewed AS elev!!
HUD AS az:	!+boresight azimuth+!	!+FPM azimuth+!	!+HUDrefpt az+!	!!Slewed AS az!!

Condition Table FD.7.1.1.2-b (continued)
Where to position the HUD aiming symbol in weapon and update modes

MODES		CONDITIONS		
=====				
Noffset	X	NOT !+desig+! AND	(!+desig+! AND !!rls	NOT !!rls
Snoffset		NOT !+after slewing+! AND	imminent!!) OR	imminent!! AND !+during
		(!+before slewing+! OR	(!+desig+! AND NOT!!rls	slewing+!
		(!+during slewing+! AND !!rls	imminent!! AND NOT	
		imminent!!))	!+during slewing+!)	
			OR	
			(NOT!+desig+! AND !+after slewing+!)	

HUDUpd	X	X	NOT !+during slewing+!	!+during slewing+!
RadarUpd				

A/A Guns				
A/A Manrip	X	X	Always	X
A/G Guns				

BOC			!!rls	NOT !!rls
BOCFlyto0			imminent!!	imminent!!
BOCOffset	X	X	OR NOT	AND !+during
SBOC			!+during	slewing+!
SBOCFlyto0			slewing+! OR	AND !+gr_ac_
SBOCOffset			!+gr_ac_ HUDrefpt+! gt 20 nmi	HUDrefpt+! lseq 20 nmi

Walleye	Always	X	X	X
=====				
HUD AS elev:	!+boresight elevation+!	!+FPM elevation+!	!+HUDrefpt _elev+!	!!Slewed AS elev!!
HUD AS az:	!+boresight azimuth+!	!+FPM azimuth+!	!+HUDrefpt az+!	!!Slewed AS az!!

Local dictionary:TermDefinition

!!Slewed AS elev!!

The new elevation position of the aiming symbol, computed by adding !!AS delta elev!! to the previous elevation position of the aiming symbol.

!!Slewed AS az!!

The new azimuth position of the aiming symbol, computed by adding !!AS delta az!! to the previous azimuth position of the aiming symbol.

!!AS delta elev!!
!!AS delta az!!

Defined by the table below.

Selector Table FD.7.1.1.2-c

!!AS delta elev!! & !!AS delta az!!

MODES !!AS delta az!! !!AS delta elev!!

=====

RadarUpd

BOC

BOCFlyto0 !+slew FLR

!+slew FLR

BOCoffset delta az+!

delta elev+!

SBOC

SBOCFlyto0

SBOCoffset

HUDaln

HUDUpd

!+slew HUD

!+slew HUD

Nattack delta az+!

delta elev+!

Noffset

Snattack

Snoffset

=====

To obtain !+slew FLR delta az+! and
!+slew FLR delta rng+!, call +SS_SLEW_FLR+
(!+Slew right-left+!, !+slew FLR delta az+!,
!+Slew up-down+!, !+slew FLR delta rng+!).

To obtain !+slew HUD delta az+! and
!+slew HUD delta elev+!, call +SS_SLEW_HUD+
(!+Slew right-left+!, !+slew HUD delta az+!,
!+Slew up-down+!, !+slew HUD delta elev+!).

Local dictionary (continued)TermDefinition

!!SK!!

!+Weapon Class+! = \$SK\$

!!desig retent!!

true iff !+desig+!, and the weapon mode was
 BOC, *BOCoffset*, or *BOCFlyto0* when
 @T(!+desig+!) last occurred.

!!rls imminent!!

Defined by the table below.

Selector Table FD.7.1.1.2-d

MODE	Definition of !!rls imminent!!
BOC	ABS(!+*** elevation+! -
BOCFlyto0	!+FPM elevation+!) lseq
BOCoffset	.59 deg AND !+*** elevation+!
Nattack	gt !+FPM elevation+!
Noffset	where "***" is "LSC" or "ISC".
SBOC	0 degrees lseq
SBOCFlyto0	(!+LSC elevation+! -
SBOCoffset	!+FPM elevation+!)
Snattack	lseq .59 degrees
Snoffset	

FD.7.1.2 -- Control the HUD Azimuth Steering Line (ASL)

The ASL usually shows the pilot the direction and sometimes the amount of steering error to his target or release point. Steering error is the angle between the a/c !ground track! and the line from the a/c to the target or release point. In most modes, the ASL center is placed on the line that is (a) parallel to the pitch lines and (b) passes through the flight path marker. Under the current requirements, the ASL mode is never \$Intermittent\$.

FD.7.1.2.1 DEMAND FUNCTION DESCRIPTION: Set the HUD ASL mode.

Mnemonic: +FD_HUD_ASL_MODE_D+

Output produced:

<u>Item</u>	<u>Type</u>	<u>Access program</u>
ASL mode	ind_cntrl	+DI_S_HUD_ASL_MODE+

Function definition:

Event Table FD.7.1.2.1-a -- Setting the ASL mode

MODES	EVENTS		
BOC			
BOCFlyto0			
BOCoffset			
CCIP			
HUDDown1			
HUDDown2	@T(In mode)	@F(In Mode)	X
Nattack			
Noffset			
SBOC			
SBOCFlyto0			
SBOCoffset			
SHUDDown1			
SHUDDown2			
Snattack			
Snoffset			
Output value:	\$On\$	\$Off\$	\$Intermittent\$

FD.7.1.2.2 PERIODIC FUNCTION DESCRIPTION: Set the HUD ASL position.

Mnemonic: +FD_HUD_ASL_POSN_P+

Output produced:

<u>Item</u>	<u>Type</u>	<u>Access program</u>
ASL elevation	angle	+DI_S_HUD_ASL_POSITION+(ASL elevation,
ASL azimuth	angle	ASL azimuth,
ASL rotation	angle	ASL rotation)

Function definition:

Condition Table FD.7.1.2.2-a -- ASL coordinates (1)

MODES		CONDITIONS			
=====					
BOCFlyto0					
HUDDown1					
HUDDown2					
Nattack	X	NOT	!+steering	!+OTS+!	!+GAS+!
Noffset		!+desig+!	to tgt+!		
SBOCFlyto0					
SHUDDown1					
SHUDDown2					
Snattack					
Snoffset					
=====					
BOC	!+gr_ac_		!+gr_ac_		
SBOC	HUDrefpt+!	X	HUDrefpt+!	!+OTS+!	!+GAS+!
	gt 30 nmi		lseq 30 nmi		
			AND		
			!+steering		
			to tgt+!		
=====					
BOCoffset	!+gr_ac_	NOT !+desig+!	!+gr_ac_		
SBOCoffset	HUDrefpt+!	AND	HUDrefpt+!	!+OTS+!	!+GAS+!
	gt 30 nmi	!+gr_ac_	AND		
		HUDrefpt+!	!+steering		
		lseq 30 nmi	to tgt+!		
=====					
ASL azimuth:	!!error	!!AS inter-	!!near	-1/2 x	!!closest
	weight!!	section!!	steering	!!error	edge!!
	x		display!!	weight!!	
	!+steering			x	
	error to tgt+!			!+steer-	
				ing error	
				to tgt+!	

After the ASL azimuth is computed, ASL elevation is set to
!!ASL_elev_placement!! in all cases.

Condition Table FD.7.1.2.2-b -- ASL coordinates (2)

MODES	CONDITIONS	
=====		
CCIP	!+ip_elev+! ls -20 degrees	!+ip elev+! gteq -20 degrees
=====		
ASL azimuth:	!+FPM azimuth+!	!!ASL_FPM_intersect4_az!!
ASL elevation:	!+FPM elevation+!	4 degrees below !+FPM elevation+!

Condition Table FD.7.1.2.2-c -- ASL rotation

MODES	CONDITIONS		
=====			
BOC			
BOCFlyto0			
BOCOffset			
HUDDown1			
HUDDown2			
Nattack			
Noffset	X	Always	X
SBOC			
SBOCFlyto0			
SBOCOffset			
SHUDDown1			
SHUDDown2			
Snattack			
Snoffset			

CCIP	X	!+ip elev+! ls -20 degrees	!+ip elev+! gteq -20 degrees
=====			
ASL rotation:	0	!+roll IMS+! + 90 deg	!+bomb fall line+!

Local dictionary:

Term	Definition
!!closest edge!!	Edge of HUD to the closest return. If !!GAS left!! then -(CHUD_symbol_az_maxç); if NOT !!GAS left!! then +(CHUD_symbol_az_maxç).
!!GAS left!!	!+GAS+! AND !+brg_grtk_tgt+! gt 180 degrees

Local Dictionary (continued):

<u>Term</u>	<u>Definition</u>
-------------	-------------------

!!near steering display!!	
---------------------------	--

Condition Table FD.7.1.2.2-d
Definition of !!near steering display!!

MODES	CONDITIONS	
=====		
BOC		
BOCFlyto0		
BOCOffset	!+Weapon Class+!	!+Weapon Class+!
HUDDown1	noteq \$SK\$	= \$SK\$
HUDDown2		
Nattack		
Noffset		

SBOC		
SBOCFlyto0		
SBOCOffset		
SHUDDown1	Always	X
SHUDDown2		
Snattack		
Snoffset		
=====		
!!alternate steering display!!:	(1/2 x !!error weight!! x !!alternate steering error!!)	!+AS azimuth+! + !+drift angle+!

!!error weight!!

When a/c pitch gets large, the steering error is multiplied by the weight factor !!error weight!! defined below, to prevent the steering error from becoming too large to display.

!+pitch IMS+! (in degrees)	Weighting Factor
=====	
lseq 0	1
gt 0 and lseq 60	$1 - 1.5 \times \frac{!+pitch IMS+!}{360 \text{ degrees}}$
gt 60 and lseq 80	$3 - 13.5 \times \frac{!+pitch IMS+!}{360 \text{ degrees}}$
gt 80 and lseq 90	0
=====	

!!alternate steering error!!

!+steering error to tgt+!, if !+gr_ac_tgt+! gt 48,000 feet; !+steering error to rls+! otherwise.

Local dictionary (continued):Term

!!AS intersection!!

Definition

This is the azimuth angle that places the HUD ASL center on the imaginary line that runs through the FPM parallel to the pitch lines such that (given the current ASL rotation) the ASL will intersect the HUD aiming symbol.

!!ASL_FPM_intersect4_az!!

The azimuth position of the ASL such that, given the current rotation angle of the ASL, (1) the ASL center is placed four degrees in elevation lower than the elevation of the Flight Path Marker; and (2) the ASL intersects the Flight Path Marker.

!!ASL elev placement!!

The ASL elevation on the HUD such that, given the current ASL rotation and ASL azimuth, the ASL center is placed on the imaginary line that is both parallel to the pitch lines, and intersects the FPM.

FD.7.1.3 -- Control the HUD Flight Director (FD)

The HUD Flight Director can be positioned in azimuth only. Its displacement from HUD center shows the azimuth steering error; that is, the angle between the projections into the horizontal plane of the a/c heading and the line from the a/c to the !Fly-to-point!. Under the current requirements, the flight director mode is never \$Intermittent\$.

FD.7.1.3.1 DEMAND FUNCTION DESCRIPTION: Set the HUD flight director mode.

Mnemonic: +FD_HUD_FLTDIR_MODE_D+

Output produced:

<u>Item</u>	<u>Type</u>	<u>Access program</u>
FLTDIR mode	ind_cntrl	+DI_S_HUD_FLTDIR_MODE+

Function definition:

Event Table FD.7.1.3.1-a -- Setting the HUD flight director mode

<u>MODES</u>	<u>EVENTS</u>
=====	=====
All alignment and navigation modes	@T(!+Init complete+!) OR @T(!+Weapon Mode+!) @F(!+Weapon Mode+!) = \$None\$
	X
=====	=====
Output value:	\$On\$ \$Off\$ \$Intermittent\$

FD.7.1.3.2 PERIODIC FUNCTION DESCRIPTION: Set HUD flight director position.

Mnemonic: +FD_HUD_FLTDIR_POSN_P+

Output produced:

<u>Item</u>	<u>Type</u>	<u>Access program</u>
Flight director azimuth	angle	+DI_S_HUD_FLTDIR_POSITION+

Function definition:

Condition table FD.7.1.3.2-a -- Positioning the HUD flight director

<u>MODES</u>	<u>CONDITIONS</u>	
=====	=====	=====
All alignment and navigation modes	!+Fly to num+! noteq 0	!+Fly to num+! = 0
=====	=====	=====
Output value:	!!ltd brg_ac_ftpt!!	0 degrees

Local dictionary:

<u>Term</u>	<u>Definition</u>
!!ltd brg_ac_ftpt!!	+SS_LIMIT_FN+(!!steering error to ftpt!!, 1, 5)
!!steering error to ftpt!!	!+brg_ac_ftpt+!, if 0 degrees lseq !+brg_ac_ftpt+! lseq 180 degrees; (!+brg_ac_ftpt+! - 360 degrees) otherwise. This translates between the full-circle measurement of !+brg_ac_ftpt+!, and the plus/minus measurement needed for a steering error.

FD.7.1.4 -- Control the HUD Flight Path Marker (FPM)

The HUD Flight Path Marker shows the direction of the aircraft velocity vector. If the aircraft is in straight and level flight, the FPM is optically centered on the HUD. The azimuth displacement from HUD center shows the lateral velocity component and the elevation displacement from HUD center shows the vertical velocity component.

FD.7.1.4.1 DEMAND FUNCTION DESCRIPTION: Set the HUD flight path marker mode.

Mnemonic: +FD_HUD_FPM_MODE_D+

Output produced:

<u>Item</u>	<u>Type</u>	<u>Access program</u>
FPM mode	ind_cntrl	+DI_S_HUD_FPM_MODE+

Function definition:

Event Table FD.7.1.4.1-a -- Setting the HUD FPM mode

MODES	EVENTS		
=====			
All alignment and navigation modes with no mode listed below	@T(!+VV mode+! = \$On\$)	@F(!+VV mode+! = \$On\$)	X

A/A Manrip			
BOC			
BOCFlyto0			
BOCOffset			
CCIP	@T(!+VV mode+!= \$On\$	@F(!+VV mode+!	@T(!+VV mode+! =
Manrip	AND	= \$On\$)	\$On\$ AND
Nattack	!!time FPM blinked!!		!+stik empty+!)
Noffset	gteq 2.5 seconds)		
SBOC			
SBOCFlyto0			
SBOCOffset			
Snattack			
Snoffset			
Walleye			
=====			
Output value:	\$On\$	\$Off\$	\$Intermittent\$

Local dictionary:

<u>Term</u>	<u>Definition</u>
!!time FPM blinked!!	elapsed time since last call to +DI_S_HUD_FPM_MODE+(\$Intermittent\$)

FD.7.1.4.2 PERIODIC FUNCTION DESCRIPTION: Set the HUD FPM position.Mnemonic: +FD_HUD_FPM_POSN_P+Output produced:

<u>Item</u>	<u>Type</u>	<u>Access program</u>
FPM elevation	angle	+DI_S_HUD_FPM_POSITION+(FPM elevation,
FPM azimuth	angle	FPM azimuth)

Function definition:

Condition Table FD.7.1.4.2-a -- Setting the FPM position

MODES		CONDITIONS	
=====			
Airaln	X	!+FM stage complete+! !+adc alt up+!	NOT !+FM stage complete+! AND NOT !+adc alt up+!

All alignment modes except *Airaln*	Always	X	X

DI			
DIG	X	Always	X
PolarDI			
UDI			

I	NOT !+in_flight+!	!+in_flight+!	X
PolarI			

Grid	NOT !+in_flight+!	!+adc tas up+!	NOT !+adc tas up+!
OLB		AND	AND
Mag sl		!+in_flight+!	!+in_flight+!

IMS fail	NOT !+in_flight+!	X	!+in_flight+!
=====			
FPM elevation:	0 degrees	!!ltd vert vels!!	!+AOA+!
FPM azimuth:	0 degrees	!!ltd lat vels!!	0 degrees

Local dictionary:Term

!!ltd vert vels!!

Definition

MIN(!!FPM elev from vels!!, 4.3 degrees)
if !!FPM elev from vels!! gt 0 degrees;

MAX(!!FPM elev from vels!!, -11.7 degrees)
if !!FPM elev from vels!! lseq 0 degrees.

!!ltd lat vels!!

SIGN(!!FPM az from vels!!) x
ABS(MIN(!!FPM az from vels!!, 6 degrees))

!!FPM az from vels!!

The azimuth angle at which the FPM should be placed, assuming it is to depict the direction of the aircraft's velocity vector, derived from !System velocities!.

!!FPM az from vels!! =

System velocity lateral component
System velocity forward component

where the result is interpreted as an angle in radians. The lateral and forward components are derived (via the Physical Models module) from !+Velocity north system+! and !+Velocity east system+!.

!!FPM elev from vels!!

The elevation angle at which the FPM should be placed, assuming it is to depict the direction of the aircraft's velocity vector, derived from !System velocities!.

!!FPM elev from vels!! =

!+Velocity vertical system+!
System velocity forward component

where the result is interpreted as angle in radians. The forward component is derived (via the Physical Model) from !+Velocity north system+! and !+Velocity east system+!.

FD.7.1.5 DEMAND FUNCTION DESCRIPTION: Set the HUD in-range cue mode.

Mnemonic: +FD_HUD_RNGCUE_D+

Output produced:

<u>Item</u>	<u>Type</u>	<u>Access program</u>
RNGCUE mode	ind_cntrl	+DI_S_HUD_RNGCUE_MODE+

Function definition:

Event Table FD.7.1.5-a -- Setting the HUD in-range cue mode

<u>MODES</u>	<u>EVENTS</u>
=====	=====
A/G Guns @T(!+target in range+!) OR @T(!+Gun Enable+! AND !+sr reasonable+!) WHEN(!+target in range+!)	@F(!+target in range+!) OR @F(In mode) OR @F(!+Gun Enable+! AND !+sr reason- able+!) WHEN (NOT !+target in range+!)
Walleye @T(!+tgt ahead+! AND !+target in range+! AND !+desig+!)	@F(!+target in range+!) OR @F(!+desig+!) OR @F(In mode) OR @F(!+tgt ahead+!)
=====	=====
Output value: \$On\$	\$Off\$ \$Intermittent\$

FD.7.1.6 -- Control the HUD Lower Solution Cue (LSC)

The HUD Lower Solution Cue serves as a warning that a release point is approaching, or as an indicator of the impact point.

FD.7.1.6.1 DEMAND FUNCTION DESCRIPTION: Set the HUD lower solution cue mode.

Mnemonic: +FD_HUD_LSC_MODE_D+

Output produced:

<u>Item</u>	<u>Type</u>	<u>Access program</u>
LSC mode	ind_cntrl	+DI_S_HUD_LSC_MODE+

Function definition:

Event table FD.7.1.6.1-a -- Setting the lower solution cue mode (1)

MODES	EVENTS	
=====	=====	=====
NBnotShrike	@T(!+target in range+! AND !+desig+! AND NOT !+GAS+! AND NOT !+during slewing+!) OR @T(1 second before !+target in range+!) WHEN(!+pitch IMS+! = 42 deg)	@F(!+target in range+!) OR @F(!+desig+!) OR @T(!+GAS+!) OR @T(!+during slewing+!)
NBShrike	@T(!+target in range+! AND !+desig+!)	@F(!+target in range+!) OR @F(!+desig+!)
CCIP	@T(!!impact angle proper!!)	@F(!!impact angle proper!!)
SBOC	@T((!+Special in range+! AND !+desig! AND	@T((NOT !+Special in range+! OR NOT In mode
SBOCflyto0	NOT !+GAS+! AND	OR !+GAS+!
SBOCoffset	NOT !+during slewing+!)	OR !+during slewing+!)
SHUDDown1	AND	AND
SHUDDown2	(NOT !+low drag release+!	(NOT !+low drag release+! OR
Snattack	OR !+tgt ahead+!)	!+OTS+! OR !+Rmax+! OR
Snoffset		NOT !+tgt ahead+!)
! *None*!	X	@T(In mode)
=====	=====	=====
Output value:	\$On\$	\$Off\$

Event Table FD.7.1.6.1-b -- Setting the lower solution cue mode (2)

MODES	EVENTS	
=====		
HUDDown1		
HUDDown2		
Nattack	@T(!!FLR sampled!!	@T(!!FLR sampled!!
Noffset	AND	AND
SHUDDown1	NOT !+sr reasonable+!)	!+sr reasonable+!)
SHUDDown2		
Snattack		
Snoffset		

None of the eight modes above and not in *Grtest*	X	@T(In mode)
=====		
Output value:	\$Intermittent\$	!!stale LSC mode!!

Local dictionary:

<u>Term</u>	<u>Definition</u>
!!FLR sampled!!	@T(!!+desig+!) OR @F(!!+Slew displacement non-zero+!) WHEN(!!+desig+!)
!!impact angle proper!!	ABS(!!+ip elev+!) lseq 16 degrees AND ABS(!!+ip az+!) lseq 12 degrees)
!*None*!	The system is in mode !*None*! when it is <u>not</u> in any of the following modes: **NBShrike**, **NBnotShrike**, *CCIP*, *Snattack*, *Snoffset*, *SBOC*, *SBOCFlyto0*, *SBOCoffset*, *SHUDDown1*, OR *SHUDDown2*.
!!stale LSC mode!!	The value of !+LSC mode+! before the call to +DI_S_HUD_LSC_MODE+(\$Intermittent\$)

FD.7.1.6.2 PERIODIC FUNCTION DESCRIPTION: Set the HUD LSC position.

Mnemonic: +FD_HUD_LSC_POSN_P+

Output produced:

<u>Item</u>	<u>Type</u>	<u>Access program</u>
LSC elevation	angle	+DI_S_HUD_LSC_POSITION+(elevation,
LSC azimuth	angle	azimuth)

Function definition:

Condition Table FD.7.1.6.2-a -- Placement of lower solution cue (1)

MODES

CONDITIONS

BOC

BOCflyto0

BOCoffset

!+OTS+!

!+low drag release+! NOT !+low drag
AND NOT !+OTS+! release+! AND
NOT !+OTS+!

HUDDown1

HUDDown2

Nattack

Noffset

=====

LSC	!+ASL elevation+!	!+ASL elevation+! -	!+ASL elevation+! -
elevation:	- 4 degrees	!!ltd dive pullup!!	!!ltd sr ac rls!!

LSC

azimuth:	!!LSC_az_on_AS!!	!!LSC_az_on_AS!!	!!LSC_az_on_AS!!
----------	------------------	------------------	------------------

Condition Table FD.7.1.6.2-b -- Placement of lower solution cue (2)

MODES	CONDITIONS			
=====				
HiNuke	!+TOS+!	NOT !+TOS+! AND !+sr_ac_rls+! gt 0 feet	NOT !+TOS+! AND !+sr_ac_rls+! lseq 0 feet	
=====				
LSC				
elevation:	!+FPM elevation+! + 4 degrees	!+FPM elevation+! + !!ltd sr ac rls!!	!+FPM elevation+! + !!wtd sr ac rls!!	
LSC				
azimuth:	!!LSC az on ASL!!	!!LSC az on ASL!!	!!LSC az on ASL!!	

Selector Table FD.7.1.6.2-c -- Placement of lower solution cue (3)

MODE	LSC elevation	LSC azimuth
=====		
CCIP	!+ip_elev+!	!!LSC_az_on_ASL!!

LoNuke	!+FPM elevation+! + !!wtd gracmax!!	!!LSC_az_on_ASL!!
=====		

Local dictionary:

<u>Term</u>	<u>Definition</u>
!!LSC_az_on_ASL!!	The value returned by +SS_SYMBOL_AS_ON_ASL+ when called with the newly-computed !+LSC elevation+!; this is the azimuth angle at which to place the LSC so that it intersects the ASL.
!!ltd dive pullup!!	+SS_LIMIT_FN+(!+dive_pullup+!, 1/8, 4)
!!ltd sr ac rls!!	+SS_LIMIT_FN+(!+sr_ac_rls+!, 1/1000, 4)
!!wtd gracmax!!	+SS_WEIGHT_FN+(!+gr_ac_rmax+!, 1/1000, 4)
!!wtd sr ac rls!!	-1 x +SS_WEIGHT_FN+(!+sr_ac_rls+!, 1/1000, 3.5)

FD.7.1.7 -- Control the HUD Pullup Anticipation Cue (PUAC)

In most modes, the Pullup Anticipation Cue shows the pilot how far he is from the "pullup point"; that is, the point where the pilot must execute a 4g pullup to avoid either the ground or the blast radius of a released weapon. It is flashed when NOT !+Master Arm+! to remind the pilot that a release is not possible. For the PUAC elevation requirements in *Snattack*, *Snoffset*, *SBOC*, *SBOCFlyto0*, and *SBOCoffset* modes, see the classified Addendum.

FD.7.1.7.1 DEMAND FUNCTION DESCRIPTION: Set the HUD PUAC mode.

Mnemonic: +FD_HUD_PUAC_D+

Output produced:

<u>Item</u>	<u>Type</u>	<u>Access program</u>
PUAC mode	ind_cntrl	+DI_S_HUD_PUAC_MODE+

Function definition:

Event Table FD.7.1.7.1-a -- Setting HUD PUAC mode

MODES		EVENTS	
=====			
BOCFlyto0			
CCIP	@T(In mode AND	X	@T(In mode AND
Nattack	!+Master Arm+!)		NOT !+Master Arm+!)
Noffset			

A/G Guns	@T(In mode AND	X	@T(In mode AND
	(!+Master Arm+!		NOT !+Master Arm+!)
	OR		WHEN(!+Weapon
	!+Weapon Class+!		Class+! = \$GN\$
	= \$GN\$ OR \$RK\$)		OR \$RK\$)

BOC	@T(In mode AND		@T(In mode AND
BOCoffset	!+Master Arm+! AND	@T(!+gr_ac_ftpt+!	NOT !+Master Arm+!)
	!+gr_ac_ftpt+!	gt 30 nmi AND	
	lseq 30 nmi)	!+Master Arm+!)	

*SBOC	@T((!+Master Arm+!	@T(!+Master Arm+!	@T(In mode AND
SBOCFlyto0	OR !+high drag	AND	NOT !+Master Arm+!)
SBOCoffset	release+!	!!Off special!!)	
Snattack	OR !+Rmax+6000+!)	OR	WHEN(!+low drag
Snoffset		@T(!+high drag	release+!)
		release+!)	

Walleye	@T(In mode AND	X	@T(In mode AND
	!+Master Arm+!)		NOT !+Master Arm+!)

!*None*!	X	@T(!+Weapon Mode+!	X
		= \$None\$)	
=====			
Output value:	\$On\$	\$Off\$	\$Intermittent\$

Local dictionary:

Term	Definition
!!Off special!!	(!+Rmin+6000+! AND !+stik created+!)
	OR
	(!+gr_ac_ftpt+! gteq 10 nmi AND !+stik empty)
	OR
	(!+pitch IMS+!=-30 degrees AND !+stik empty)

!*None*! The state when the system is in none of the other modes listed in the Function Definition table.

FD.7.1.7.2 PERIODIC FUNCTION DESCRIPTION: Set the HUD PUAC position.Mnemonic: +FD_HUD_PUAC_P+Output produced:

<u>Item</u>	<u>Type</u>	<u>Access program</u>
PUAC elevation	angle	+DI_S_HUD_PUAC_POSITION+(PUAC elevation,
PUAC azimuth	angle	PUAC azimuth)

Function definition:

Condition Table FD.7.1.7.2-a -- Setting HUD PUAC elevation

<u>MODES</u>	<u>CONDITIONS</u>
BOC	
BOCFlyto0	
BOCOffset	!+sr_ac_gpup+! gt 5000 ft !+sr_ac_gpup+! lseq 5000 ft
CCIP	AND OR
Nattack	!+sr_ac_bpup+! gt 5000 ft !+sr_ac_bpup+! lseq 5000 ft
Noffset	
Walleye	
A/G Guns	!+sr_ac_gpup+! gt 5000 ft !+sr_ac_gpup+! lseq 5000 ft
	AND OR
	(!+sr_ac_bpup+! gt 5000 ft (!+Weapon Class+! = \$RK\$ AND
	OR !+sr_ac_bpup+! lseq 5000 ft)
	(!+Weapon Class+! = \$GN\$)
PUAC	
elevation:	!+FPM elevation+! - 3.5 deg. !!pullup elev!!

Selector Table FD.7.1.7.2-b -- Setting HUD PUAC azimuth

MODES	PUAC azimuth
BOC	
BOCFlyto0	
BOCoffset	
Nattack	
Noffset	!!PUAC_az_on_ASL!!
SBOC	
SBOCFlyto0	
SBOCoffset	
Snattack	
Snoffset	

A/G Guns	
CCIP	!+FPM azimuth+!
Walleye	

Local dictionary:

<u>Term</u>	<u>Definition</u>
!!pullup elev!!	!+FPM elevation+! - (0.7 degrees x MIN(!+sr_ac_bpup+!, !+sr_ac_gpup+!)/1000 feet)
!!PUAC_az_on_ASL!!	The value returned by +SS_SYMBOL_AS_ON_ASL+ when called with the newly-computed !+PUAC elevation+! this is the azimuth angle at which to place the PUAC so that it intersects the ASL.

FD.7.1.8 DEMAND FUNCTION DESCRIPTION: Set the HUD pullup cue mode.

Mnemonic: +FD_HUD_PUC_MODE_D+

Output produced:

<u>Item</u>	<u>Type</u>	<u>Access program</u>
Pullup cue mode	ind_cntrl	+DI_S_HUD_PUC_MODE+

Function definition:

Event table FD.7.1.8-a -- When to start/stop the HUD pullup cue

MODES		EVENTS	
=====			
All modes except *Grtest*	X	@T(!+blast danger+!) OR @T(!+ground danger+!)	@F(!+blast danger+! OR !+ground danger+!) OR @T(!+time PUC blinked!! gteq 2 secs) WHEN(!+low drag release+! AND !!Speci-1!!)
=====			
Output value:	\$On\$	\$Intermittent\$	\$Off\$

Note: The HUD pullup cue cannot be turned on steady.

Local dictionary:

<u>Term</u>	<u>Definition</u>
!!Special!!	!+Weapon Class+! = \$SOD\$ OR \$SSH\$
!!time PUC blinked!!	elapsed time since last call to +DI_S_HUD_PUC_MODE+(\$Intermittent\$)

FD.7.1.9 -- Control the HUD Upper Solution Cue (USC)

FD.7.1.9.1 DEMAND FUNCTION DESCRIPTION: Set the HUD upper solution cue mode.

Mnemonic: +FD_HUD_USC_MODE_D+

Output produced:

<u>Item</u>	<u>Type</u>	<u>Access program</u>
USC mode	ind_cntrl	+DI_S_HUD_USC_MODE+

Function definition:

Event table FD.7.1.9.1-a -- Setting upper solution cue mode (1)

MODES	EVENTS
=====	
NBnotShrike	@F(In mode)
SBOC	@T(!+Special in range+! OR
SBOCFlyto0	AND !+desig+! @F(!+Special in range+!
SBOCOffset	AND NOT !+GAS+! AND NOT !+GAS+!
SHUDdown1	AND NOT !+during AND NOT !+during
SHUDdown2	slewing+!) slewing+!)
Snattack	WHEN(!+low drag release+!) OR
Snoffset	@F(!+low drag release+!)

No mode listed	
above	X @T(In mode)
=====	
Output value:	\$On\$ \$Off\$

Event Table FD.7.1.9.1-b -- Setting the upper solution cue mode (2)

MODES

EVENTS

```
=====
*HUDDown1*
*HUDDown2*
*Nattack*
*Noffset*          @T(!+LSC mode+! =          @F(!+LSC mode+! =
*SHUDDown1*          $Intermittent$)          $Intermittent$)
*SHUDDown2*
*Snattack*
*Snoffset*
=====
```

```
-----
No mode listed
above          X          @T(In mode)
and not *Grtest*
```

```
=====
Output value:          $Intermittent$          !!stale USC mode!!
=====
```

Local dictionary:TermDefinition

!!stale USC mode!!

The value of !+USC mode+! before the call to
+DI_S_HUD_USC_MODE+(\$Intermittent\$)

FD.7.1.9.2 PERIODIC FUNCTION DESCRIPTION: Set the HUD USC position.

Mnemonic: +FD_HUD_USC_POSN_P+

Output produced:

<u>Item</u>	<u>Type</u>	<u>Access program</u>
USC elevation	angle	+DI_S_HUD_USC_POSITION+(elevation,
USC azimuth	angle	azimuth)

Function definition:

Condition Table FD.7.1.9.2-a -- Positioning the upper solution cue

MODES

CONDITIONS

=====

NBnotShrike

SBOC

SBOCFlyto0

!+OTS+!

NOT !+OTS+!

SBOCoffset

SHUDdown1

SHUDdown2

Snattack

Snoffset

=====

USC elevation:

!!ltd OTS pullup!!

!!ltd loft pullup!!

USC azimuth:

!!USC_az_on_ASL!!

!!USC_az_on_ASL!!

Local dictionary:

Term

Definition

!!ltd loft pullup!!

+SS_LIMIT_FN+(!+loft pullup+!, 1/8, 4)

!!ltd OTS pullup!!

+SS_LIMIT_FN+(!+OTS pullup+!, 1/8, 4)

!!USC_az_on_ASL!!

The value returned by +SS_SYMBOL_AS_ON_ASL+ when called with the newly-computed !+USC elevation+!; this is the azimuth angle at which to place the USC so that it intersects the ASL.

FD.7.1.10 FUNCTION DESCRIPTION: Set the HUD symbol blink rate.

Mnemonic: +FD_HUD_BLINKRATE+

Output produced:

<u>Item</u>	<u>Type</u>	<u>Access program</u>
New blink rate	real	+DI_S_HUD_BLINK_RATE+

Function definition:

This function is never performed, because under the current requirements, the blink rate for all HUD symbols is the same, and is equal to the default rate. However, should the requirements change and other blink rates be necessary, this function would have the responsibility to determine the prevailing rate.

FD.7.2 -- HUD Value Indicators

These functions display information on the HUD giving the situation of the aircraft relative to the earth, or numerical indications of the aircraft's velocity. All of these functions control value displays on the HUD, rather than symbol positions.

FD.7.2.1 PERIODIC FUNCTION DESCRIPTION: Set the HUD altitude display.

Mnemonic: +FD_HUD_BARO_ALT_P+

Output produced:

<u>Item</u>	<u>Type</u>	<u>Access program</u>
HUD altitude display	distance	+DI_S_HUD_ALT_DISPLAY+

Initiation/termination events:

None. Always performed.

Function definition:

Condition table FD.7.2.1-a -- Setting the HUD altitude display

MODES		CONDITIONS	
=====			
All alignment and navigation modes	!+adc alt up+!	NOT !+adc alt up+!	X

Grtest	X	X	Always
=====			
Output value:	!+alt ADC+!	4500 ft	!!last pre-test value!!

Local dictionary:TermDefinition

!!last pre-test value!!

The value that was being output when
@T(*Grtest*) occurred.

FD.7.2.2 PERIODIC FUNCTION DESCRIPTION: Set the HUD heading display.

Mnemonic: +FD_HUD_HEADING_P+

Output produced:

<u>Item</u>	<u>Type</u>	<u>Access program</u>
HUD heading	angle	+DI_S_HUD_HEADING_DISPLAY+

Initiation/termination events:

None. Always performed.

Function definition:

Condition table FD.7.2.2-a -- Setting the HUD heading display

MODES	CONDITIONS	
=====		
All alignment and navigation modes, except *IMS fail*	Always	X

IMS fail	!+ims_mode+! noteq \$Off\$	!+ims_mode+! = \$Off\$
=====		
Output value:	!+heading MAG+!	0 degrees (North)

FD.7.2.3 PERIODIC FUNCTION DESCRIPTION: Set the HUD pitch and roll displays.

Mnemonic: +FD_HUD_PITCH_ROLL_P+

Output produced:

<u>Item</u>	<u>Type</u>	<u>Access program</u>
HUD pitch display	angle	+DI_S_HUD_PITCH_DISPLAY+
HUD roll display	angle	+DI_S_HUD_ROLL_DISPLAY+

Initiation/termination events:

None. Always performed.

Function definition:

Selector table FD.7.2.3-a -- Setting HUD pitch and roll displays

<u>MODES</u>	<u>HUD pitch display</u>	<u>HUD roll display</u>
All navigation and alignment modes, except *IMS fail*	!+pitch IMS+!	!+roll IMS+!
IMS fail	!+AOA+!	0 degrees
Grtest	!!pre-test pitch!!	!!pre-test roll!!

Local dictionary:

<u>Term</u>	<u>Definition</u>
!!pre-test pitch!!	The value of !+pitch IMS+! when @T(*Grtest*) occurred.
!!pre-test roll!!	The value of !+roll IMS+! when @T(*Grtest*) occurred.

FD.7.2.4 -- Control the HUD vertical velocity and vertical acceleration displays.

The HUD vertical velocity indicator shows the vertical velocity calculated from the highest-priority reliable sensors. The HUD acceleration display shows vertical (normal) acceleration. These displays differ from other value displays in that they both must be enabled before values may be displayed. Function 7.2.4.1 enables/disables the HUD vertical velocity and acceleration displays.

FD.7.2.4.1 DEMAND FUNCTION DESCRIPTION: Enable the HUD vertical velocity and vertical acceleration displays.

Mnemonic: +FD_HUD_VV_MODE_D+

Output produced:

<u>Item</u>	<u>Type</u>	<u>Access program</u>
vv/accel mode	ind_cntrl	+DI_S_HUD_VV_MODE+

Function definition:

Event Table FD.7.2.4.1-a
Setting the velocity and acceleration displays' mode

MODES		EVENTS	
=====		=====	
Airaln	@F(!+align_stage+!=\$FM\$)	@F(!+AOA valid+!)	
	OR	WHEN(NOT !+FM stage	X
	@T(!+AOA valid+!)	complete+!)	

IMS fail	@T(!+AOA valid+!)	@T(!+in flight+!	
Grid	OR	AND	X
Mag sl	@F(!+in_flight+!)	NOT !+AOA valid+!)	
OLB			

!*None*!	@T(In mode)	X	X
=====			
Output			
value:	\$On\$	\$Off\$	\$Intermittent\$

Local dictionary:

<u>Term</u>	<u>Definition</u>
!*None*!	The state when the system is in none of the other modes listed in the Function Definition table.

FD.7.2.4.2 PERIODIC FUNCTION DESCRIPTION: Set the HUD vertical acceleration display.

Mnemonic: +FD_HUD_ACCEL_P+

Output produced:

<u>Item</u>	<u>Type</u>	<u>Access program</u>
HUD vertical acceleration display	accel	+DI_S_HUD_NACC_DISPLAY+

Initiation/termination events:

Event Table FD.7.2.4.2-a
When the HUD vertical acceleration display is updated

MODES	Initiation events	Termination events
Snattack		
Snoffset	@T(!+low drag release+!	@F(!+low drag release+!
SBOC	AND	AND
SBOCFlyto0	!+VV mode+! = \$On\$)	!+VV mode+! = \$On\$)
SBOCoffset		

Function definition:

Call +DI_S_HUD_NACC_DISPLAY+(!+normal_accel+!).

FD.7.2.4.3 PERIODIC FUNCTION DESCRIPTION: Set the HUD vert. vel. display.

Mnemonic: +FD_HUD_VERTVEL_P+

Output produced:

<u>Item</u>	<u>Type</u>	<u>Access program</u>
HUD vertical velocity display	speed	+DI_S_HUD_VERTVEL_DISPLAY+

Initiation/termination events:

Event Table FD.7.2.4.3-a -- When the HUD vertical velocity is updated

MODES	Initiation events	Termination events
=====	=====	=====
SBOC		
SBOCFlyto0	@T(NOT !+low drag release+!	@T(!+low drag release+!
SBOCoffset	AND	OR
Snattack	!+VV mode+! = \$On\$)	!+VV mode+! = \$Off\$)
Snoffset		
None of the		
above five modes	@T(!+VV mode+! = \$On\$)	@F(!+VV mode+! = \$On\$)
=====	=====	=====

Function definition:

Condition Table FD.7.2.4.3-b
Updating the HUD vertical velocity display

MODES	CONDITIONS
=====	=====
All alignment modes	
DIG	
DI	
I	
PolarDI	Always
PolarI	X
UDI	
Grid	
IMS fail	
Mag Sl	
OLB	
Grtest	X
	Always
=====	=====
Output value:	!+Velocity vertical system+!
	0

The HUD vertical velocity display is also set to 0 at the moment when any alignment mode except *Airaln* is exited.

```

*****
*
*           FD.8           IMS functions
*
*****

```

FD.8.1 DEMAND FUNCTION DESCRIPTION: Switch computer control of the IMS on/off.

Mnemonic: +FD_IMS_COMP_CTRL_D+

Output produced:

<u>Item</u>	<u>Type</u>	<u>Access program</u>
IMS enable	boolean	+DI_S_IMS_ENABLE+

Function definition:

Event Table FD.8.1-a -- IMS computer control setting

MODES	EVENTS	
=====	=====	
Airaln	@T(!!roll lrg!! AND !+align_stage+!= \$FM\$) OR @T(In mode AND !+align_stage+! noteq \$FM\$)	@F(!!roll lrg!!) WHEN (!+align_stage+! = \$FM\$) OR @T(!+align_stage+! = \$FM\$) WHEN(NOT !!roll lrg!!) OR @T(!+align stage+! noteq \$CL\$ AND !+IMS reasonable+!) WHEN(!+in flight+!)
Any align- ment mode but *Airaln*	@T(In mode) @T(!+align stage+! noteq \$CL\$ AND !+IMS reasonable+!) WHEN(!+in flight+!)	
DIG		
DI		
I	@T(In mode) X	
OLB		
PolarDI		
PolarI		
UDI		
Grid		
IMS fail	X @T(In mode)	
Mag Sl		
=====	=====	
Output value:	<u>true</u> <u>false</u>	

Local dictionary:

<u>Term</u>	<u>Definition</u>
!!roll lrg!!	ABS(!+roll IMS+!) gt 5°

FD.8.2 DEMAND FUNCTION DESCRIPTION: Set the IMS velocity measurement scale.

Mnemonic: +FD_IMS_SCALE_D+

Output produced:

<u>Item</u>	<u>Type</u>	<u>Access program</u>
IMS scale	imsscale	+DI_S_IMS_SCALE+

Function definition:

Event Table FD.8.2-a -- Changing the IMS Velocity Measurement Scale

MODES	EVENTS
Landaln	
Lautocal	@T(In mode) X
Ol Update	
HUDaln	@T(In mode) WHEN @T(In mode) WHEN(!+IMS mode+! (!+IMS mode+! = \$Gndal\$) = (\$Norm\$ OR \$Iner\$))
Airaln	
Sautocal	
SINSaln	
DI	
DIG	X @T(In mode)
I	
OLB	
PolarDI	
PolarI	
UDI	
Output value:	\$Fine\$ \$Coarse\$

FD.8.3: Adjust the alignment of the IMS X, Y, and Z axes.

Under certain conditions, the IMS platform axes are assumed to be properly aligned, and the axes are only maintained in alignment; that is, compensations are made for the earth's rotation, a/c movement, etc. The values which are only for maintaining alignment are suffixed with "m".

Under other conditions, misalignments are corrected, by using certain reference information to discover how much the IMS axes are misaligned from their corresponding earth-reference-frame axes. The values which are for correcting misalignment, as well as for compensating for normal motion (i.e., maintaining, as explained above), are suffixed with "mc". In addition, the third term of each such value's identifier denotes the source of the reference information; i.e., SINS, Doppler radar, or the knowledge that the aircraft is not moving.

Only small x and y adjustments are made in navigation modes. Whether small or large x and y adjustments are made in an alignment mode is determined completely by the alignment stage. The system is never in a mode or stage where both large and small x/y corrections should be applied at the same time.

Whether large or small z adjustments are made usually depends on the magnitude of the adjustment to be made in each case. Errors above a certain point (!+ims_cutoff+) are considered to require a large adjustment; errors at or below that cutoff are considered to require a small adjustment.

In the requirements document, there are two functions for adjusting the alignment of the IMS axes: the first applies small adjustments to all three axes, and the second applies large adjustments to all three axes. These functions have been re-shaped so that there is a function to apply small x/y corrections, a function to apply large x/y corrections, and a function to apply all adjustments (i.e., large and small) to the z axis. This was done because the small x/y function is intuitively periodic, while the large x/y function is intuitively demand. The z function was separated because it has very little in common with either of the other two. Further, it is conceivable that deleting z axis alignment from the module would leave a useful functional subset.

FD.8.3.1 PERIODIC FUNCTION DESCRIPTION: Perform small adjustments to the IMS platform X and Y axes.

Mnemonic: +FD_IMS_SMADJ_XY_P+

Output produced:

<u>Item</u>	<u>Type</u>	<u>Access program</u>
IMS small x-axis adjustment	angle	+DI_S_X_FINE_ROTATION+
IMS small y-axis adjustment	angle	+DI_S_Y_FINE_ROTATION+

Initiation/termination events:

None. Always performed.

Function definition:

Condition Table FD.8.3.1-a -- When small x and y adjustments are applied

MODES	CONDITIONS
Lautocal	!+align_stage+! = \$FG\$ OR \$ND\$ OR \$ND2\$ OR \$ED\$ OR \$ED2\$
Sautocal	!+align_stage+! = \$ED\$ OR \$ED2\$ OR \$ND\$ OR \$ND2\$
OUpdate	
HUDaln	!+align_stage+! = \$FG\$
Landaln	
SINSaln	
Airaln	!+align_stage+! = \$HL\$ OR \$FG\$ OR \$HG\$
DI	
DIG	
I	Always
OLB	
PolarDI	
PolarI	
UDI	
Function calls:	+DI_S_X_FINE_ROTATION+(!!IMS adj x error!!) +DI_S_Y_FINE_ROTATION+(!!IMS adj y error!!)

Local dictionary:

<u>Term</u>	<u>Definition</u>
!!IMS adj x error!! and !!IMS adj y error!!	Defined by Table 8.3.1-b below.

Selector Table FD.8.3.1-b
Values of x and y adjustments:
!!IMS adj x error!! & !!IMS adj y error!!

MODES	!!IMS adj x error!!	!!IMS adj y error!!
=====		
Landaln		
Lautocal	!+ims_x_const_error_mc+!	!+ims_y_const_error_mc+!
HUDaln		
OlUpdate		

Sautocal		
SINSaln	!+ims_x_sins_error_mc+!	!+ims_y_sins_error_mc+!

Airaln	!+ims_x_dop_error_mc+!	!+ims_y_dop_error_mc+!

I		
UDI	!+ims_x_nav_error_m+!	!+ims_y_nav_error_m+!
OLB		
PolarI		

DIG		
DI	!+ims_x_nav_error_mc+!	!+ims_y_nav_error_mc+!
PolarDI		
=====		

FD.8.3.2 DEMAND FUNCTION DESCRIPTION: Perform large adjustments of the IMS platform X and Y axes.

Mnemonic: +FD_IMS_LGADJ_XY_D+

Output produced:

<u>Item</u>	<u>Type</u>	<u>Access program</u>
IMS large x-axis adjustment	angle	+DI_S_X_COARSE_ROTATION+
IMS large y-axis adjustment	angle	+DI_S_Y_COARSE_ROTATION+

Function definition:

Event Table FD.8.3.2-a -- When large x and y adjustments are applied

MODES

EVENTS

```

=====
*HUDaln*
*Lautocal*      @T(!!IMS adj xy error!! gt !+IMS adj xy tolerance+!)
*Landaln*       WHEN(!+align_stage+! = $CL2$ OR $CA2$)
*Sautocal*
*SINSaln*
=====

```

```

*Airaln*      @T(!!IMS adj xy error!! c !+IMS adj xy tolerance+!)
               WHEN(!+align_stage+! = $CL2$)
=====

```

```

Function call:  +DI_S_X_COARSE_ROTATION+(!!IMS adj x error!!)
                +DI_S_Y_COARSE_ROTATION+(!!IMS adj y error!!)

```

Local dictionary:

<u>Term</u>	<u>Definition</u>
!!IMS adj xy error!!	MAX(!!IMS adj x error!!, !!IMS adj y error!!)
!!IMS adj x error!! and	
!!IMS adj y error!!	Defined by Table 8.3.1-b.

FD.8.3.3 DEMAND FUNCTION DESCRIPTION: Adjust the alignment of the IMS platform z axis.

Mnemonic: +FD_IMS_ADJ_Z_D+

Output produced:

<u>Item</u>	<u>Type</u>	<u>Access program</u>
IMS small z-axis correction	angle	+DI_S_Z_FINE_ROTATION+
IMS large z-axis correction	angle	+DI_S_Z_COARSE_ROTATION+

Function definition:

Like the x and y axes, the IMS platform z axis may be corrected by either a large or a small adjustment. A large adjustment can be either a "preliminary adjustment" or a "subsequent adjustment". The subsequent adjustment is applied at the concluding portion of an alignment stage, while the preliminary adjustment does not depend on the alignment stage at all. There is a third class of large z-axis adjustments, which we have called "radical" adjustments. These involve temporary ninety-degree rotations about the axis to allow the IMS to align itself in another plane. Values defined by the following table.

Event Table FD.8.3.3-a -- When small z adjustments are applied

MODES	EVENTS
Lautocal	@T(!!IMS small z error!! gt !+IMS z adj tolerance+!) WHEN(!+align_stage+! = \$CA2\$ OR \$ND2\$ OR \$ED2\$ AND !!IMS small z error!! lseq cims_cutoffc OR !+align_stage+! = \$FG\$)
Sautocal	@T(!!IMS small z error!! gt !+IMS z adj tolerance+!) WHEN(!+align_stage+! = \$CA2\$ OR \$ND2\$ OR \$ED2\$ AND !!IMS small z error!! lseq cims_cutoffc OR !+align_stage+! = \$FG\$)
HUDaln *Landaln*	@T(!!IMS small z error!! gt !+IMS z adj tolerance+!) WHEN(!+align_stage+! = \$CA2\$ AND !!IMS small z error!! lseq cims_cutoffc OR !+align_stage+! = \$FG\$)
SINSaln	@T(!!IMS small z error!! gt !+IMS z adj tolerance+!) WHEN(!+align_stage+! = \$CA2\$ AND !!IMS small z error!! lseq cims_cutoffc OR !+align_stage+! = \$FG\$)
Airaln	@T(!!IMS small z error!! gt !+IMS z adj tolerance+!) WHEN(!+align_stage+! = \$HL\$ OR \$FG\$)
UUpdate	@T(!!IMS small z error!! gt !+IMS z adj tolerance+!) WHEN(!+align_stage+! = \$FG\$)
DI *DIG*	@T(!!IMS small z error!! gt !+IMS z adj tolerance+!)
I *OLB* *UDI*	@T(!!IMS small z error!! gt !+IMS z adj tolerance+!) WHEN(!+latitude+! lseq 80 ⁰)
Function call: +DI_S_Z_FINE_ROTATION+(!!IMS small z error!!)	

Event Table FD.8.3.3-b -- When large z adjustments are applied:
 !!IMS preliminary z adj!! and !!IMS subsequent z adj!!

MODES	When to apply !!IMS preliminary z adj!!	When to apply !!IMS subsequent z adj!!
HUDaln	@T(!+TD pressed+! WHEN(NOT !+IMS rotating+!)	@T(!+align_stage+! = \$CA2\$) WHEN(!+IMS subsequent z adj!! gt cims_cutoffc)
Lautocal	X	@T(!+align_stage+! = \$CA2\$ OR \$ND2\$ OR \$ED2\$) WHEN(!+IMS subsequent z adj!! gt cims_cutoffc)
Landaln	X	@T(!+align_stage+! = \$CA2\$) WHEN(!+IMS subsequent z adj!! gt cims_cutoffc)
SINSaln	@T(In mode)	@T(!+align_stage+! = \$CA2\$) WHEN(!+IMS subsequent z adj!! gt cims_cutoffc)
Function call:	+DI_S_Z_COARSE_ROTATION+ (!!IMS preliminary z adj!!)	+DI_S_Z_COARSE_ROTATION+ (!!IMS subsequent z adj!!)

Event Table FD.8.3.3-c -- Applying radical IMS z-axis adjustments:
 Value of adjustments and when to apply them

MODES	EVENTS		
Lautocal	X	@F(!+align_ stage+! = \$ED\$ OR \$ED2\$)	@T(!+align_ stage+! = \$ED\$ OR \$ED2\$)
Sautocal	@F(!+align_ stage+! = \$ED\$ OR \$ED2\$)	X	@T(!+align_ stage+! = \$ED\$ OR \$ED2\$)
Output value:	90 deg CCW + !+ims_z_const_ error_mc+!	90 deg CCW + !+ims_z_sins_ error_mc+!	90 deg CW

Local dictionary:TermDefinition

!!az ref error!!

The error (difference in heading) calculated by comparing the IMS heading to the heading calculated from !+az ref hdg pnl+! corrected by the HUD aiming symbol azimuth displacement at the time of the test. The value is updated whenever @T(!+desig+!) WHEN(!+align_mode+! = *HUDaln*) occurs. The value is (!+az ref hdg pnl+! + !+AS azimuth+! - !+heading IMS+!) modulo 360 degrees.

!!IMS small z error!!

The amount of small correction to be applied to the IMS z axis; values defined by table below.

Selector Table FD.8.3.3-d

Values of small IMS z-axis adjustments: !!IMS small z error!!

MODES

!!IMS small z error!!

HUDaln

Landaln

Lautocal

OlUpdate

!+ims_z_const_error_mc+!

Sautocal

SINSaln

!+ims_z_sins_error_mc+!

Airaln

DIG

!+ims_z_dop_error_mc+!

DI

!+ims_z_dop_error_m+!

I

OLB

UDI

!+ims_z_nav_error_m+!

Local dictionary (continued):

<u>Term</u>	<u>Definition</u>
!!sins error!!	Angular difference measured from !+heading IMS+! to (!+SINS heading+! + !+SINS dhdg pnl+!). Positive if that angle is measured clockwise; negative if counterclockwise.
!!IMS preliminary z adj!! and	
!!IMS subsequent z adj!!	The amount of large (coarse) correction to be applied to the IMS z axis during either a preliminary adjustment or a subsequent adjustment, respectively.

Selector Table FD.8.3.3-e -- Values of preliminary and subsequent
IMS z-axis adjustments

MODES	!!IMS preliminary z adj!!	!!IMS subsequent z adj!!
=====	=====	=====
HUDaln	!!az_ref_error!!	!+ims_z_const_error_mc+!

SINSaln	!!sins_error!!	!+ims_z_sins_error_mc+!

Landaln	X	!+ims_z_const_error_mc+!
=====	=====	=====

FD.8.4 DEMAND FUNCTION DESCRIPTION: Initialize the IMS velocities.Mnemonic: +FD_IMS_SET_VELS_D+Output produced:

<u>Item</u>	<u>Type</u>	<u>Access program</u>
New IMS east velocity	speed	+DI_S_IMS_E_VELOCITY+
New IMS north velocity	speed	+DI_S_IMS_N_VELOCITY+

Function definition:

Event Table FD.8.4-a -- Initializing IMS velocities

MODES		EVENTS
SINSaln	@T(!+New align stage+!)	X
Sautocal		
Landaln	X	@T(In mode)
Lautocal		
IMS fail	X	@T(In mode)
=====		
OUTPUT VALUES:		
New IMS east velocity:	!+SINS east vel+!	0 fps
New IMS north velocity:	!+SINS north vel+!	0 fps

FD.8.5 DEMAND FUNCTION DESCRIPTION: Set the IMS reconfiguration values.

Mnemonic: +FD_IMS_RECONFIG_D+

Output produced:

<u>Item</u>	<u>Type</u>	<u>Access program</u>
New X drift	real	+DI_S_X_GYRO_DRIFTRATE+
New Y drift	real	+DI_S_Y_GYRO_DRIFTRATE+
New Z drift	real	+DI_S_Z_GYRO_DRIFTRATE+
New X corr increm	real	+DI_S_X_GYRO_SCALE+
New Y corr increm	real	+DI_S_Y_GYRO_SCALE+
New Z corr increm	real	+DI_S_Z_GYRO_SCALE+
New N coarse scale factor	real	+DI_S_N_COARSE_VEL_SCALE+
New E coarse scale factor	real	+DI_S_E_COARSE_VEL_SCALE+
New V scale factor	real	+DI_S_V_VEL_SCALE+
New N fine scale factor	real	+DI_S_N_FINE_VEL_SCALE+
New E fine scale factor	real	+DI_S_E_FINE_VEL_SCALE+
New N coarse bias	real	+DI_S_N_COARSE_VEL_BIAS+
New E coarse bias	real	+DI_S_E_COARSE_VEL_BIAS+
New V bias	real	+DI_S_V_VEL_BIAS+
New N fine bias	real	+DI_S_N_FINE_VEL_BIAS+
New E fine bias	real	+DI_S_E_FINE_VEL_BIAS+

Function definition:

Event Table FD.8.5-a -- IMS reconfiguration updates

<u>MODES</u>	<u>EVENTS</u>
=====	=====
All modes	@T(!+New !!input item!! entered+!)
except *Grtest*	
=====	=====
Output value:	!+ !!input item!! +!

For instance, one of the values of !!input item!! is "drift pnl" (see local dictionary). Therefore, when @T(!+New X drift entered+!) occurs, this function should output the new value of "drift pnl" via the access program +DI_S_X_GYRO_DRIFTRATE+.

Local dictionary:Term

!!input item!!

Definition

any of the various input items used to update an IMS Reconfiguration value. Enumerated in the list below.

X drift pnl
Y drift pnl
Z drift pnl
X corr increm pnl
Y corr increm pnl
Z corr increm pnl
N coarse scale pnl
E coarse scale pnl
V coarse scale pnl
N fine scale pnl
E fine scale pnl
N coarse bias pnl
E coarse bias pnl
V coarse bias pnl
N fine bias pnl
E fine bias pnl

```

*****
*
*                      FD.9                      Panel functions
*
*****

```

FD.9.1 DEMAND FUNCTION DESCRIPTION: Set the panel's mark window display.

Mnemonic: +FD_PNL_MARK_D+

Output produced:

<u>Item</u>	<u>Type</u>	<u>Access program</u>
To blank out the mark window	--	+DI_CLEAR_MARK+
Display for mark window	char_string(1)	+DI_S_MARK_WINDOW+

Function definition:

Event Table FD.9.1-a -- Clearing the Mark window display

<u>MODES</u>	<u>EVENT</u>
=====	
All modes	@T(!+Init complete+!)
except *Grtest*	

Grtest	@T(In mode)
=====	
Function call:	Call +DI_CLEAR_MARK+

Event Table FD.9.1-b -- Changing the Mark window display

<u>MODES</u>	<u>EVENTS</u>
=====	
All	@T(!+Mark pressed+!) @T(!Nav2 config!!) @T(!Nav2 config!!)
modes	WHEN(NOT !!Nav2 AND AND
except	config!!) NOT !+ADC reasonable+!) !+ADC reasonable+!)
Grtest	
=====	
Output	
Value:	!+Mark+!, converted character '0' character '1'
	to character form

Local dictionary:

Term

!!Nav2 config!!

Definition

true iff calling +SS_G_PNL_CONFIG+(\$Nav diags2\$, p2) returns p2 with a value of true. This means that the panel is in the \$Nav diags2\$ configuration.

AD-A107 922

NAVAL RESEARCH LAB WASHINGTON DC F/G 9/2
FUNCTION SPECIFICATIONS FOR THE A-7E FUNCTION DRIVER MODULE.(U)
NOV 81 P C CLEMENTS

F/G 9/2

UNCLASSIFIED

NL

AD
ALCANTARA

END
DATE
FILMED
1-82
DTIC

FD.9.2 DEMAND FUNCTION DESCRIPTION: Control the panel's enter light.

Mnemonic: +FD_PNL_ENTER_D+

Output produced:

<u>Item</u>	<u>Type</u>	<u>Access program</u>
Enter light mode	boolean	+DI_S_ENTER_LIGHT+

Function definition:

Event Table FD.9.2-a -- Controlling the ENTER light

MODES

EVENTS

```
=====
*FlyUpd*      @T(!+TD pressed+!)      @T(!Panel switch changed!!
*HUDUpd*      WHEN(NOT !+Enter light+!) OR !+Input attempted+!
*MapUpd*              OR              OR !+Fly to num changed+!
*RadarUpd*      @T(!+data enterable+!) OR !+Fly to state changed+!
*TacUpd*              WHEN(!+Enter light+!)
                      OR
                      @T(!+Init complete+!)
                      OR
                      @F(!+data enterable+!)
=====
```

OUTPUT VALUE:

true

false

Local dictionary:

<u>Term</u>	<u>Definition</u>
!!Panel switch changed!!	(!+Panel mode changed+! OR
	!+Update changed+! OR
	!+Pres pos changed+! OR
	!+Map hold changed+! OR
	!+Enter pressed+! OR
	!+Input requested+!)

FD.9.3 DEMAND FUNCTION DESCRIPTION: Display data in the upper or lower windowMnemonic: +FD_PNL_DISPLAY_D+Output producer:

<u>Item</u>	<u>Type</u>	<u>Access program</u>
To control upper and lower window displays	angle	+SS_S_ANGLE_LOWER+
	angle	+SS_S_ANGLE_UPPER+
	charlit	+SS_S_CHARLIT_LOWER+
	charlit	+SS_S_CHARLIT_UPPER+
	latitude	+SS_S_LATITUDE_UPPER+
	longitude	+SS_S_LONGITUDE_LOWER+
	real	+SS_S_REAL_LOWER+
	real	+SS_S_SFRAC_LOWER+
	boolean	+SS_S_SIGN_LOWER+
	integer	+SS_S_SINT_LOWER+
	integer	+SS_S_SINT_UPPER+
	real	+SS_S_UFRAC_LOWER+
	time	+SS_S_TIME_LOWER+
	integer	+SS_S_UINT_UPPER+
	integer	+SS_S_UINT_LOWER+
To clear displays	--	+SS_CLEAR_LOWER+
	--	+SS_CLEAR_UPPER+
To control format lights	boolean	+DI_S_E_LIGHT+
	boolean	+DI_S_N_LIGHT+
	boolean	+DI_S_S_LIGHT+
	boolean	+DI_S_W_LIGHT+

Function definition:WHAT TO DISPLAY ON THE PANEL WINDOWS: Table FD.9.3-a shows:

- (1) all of the various values that can be displayed on the panel. This list is under the "!!item!!" column. The value for each item is either available directly from some other module (if bracketed by "!!+ +!" symbols), or is defined in this function's local dictionary (if bracketed by "!! !!" symbols).
- (2) the data type for the item to be displayed (called "Input Type" in the table). Some values come directly from other modules, and this is the data type of the provided value. On the other hand, some display values are composed by this function driver from several input values rather than being obtained as a whole from one source. If all components are of the same type, that type is given. Otherwise, the type is given as "various".
- (3) the data type that the item must have in order to be displayed (if that data type is different from that item's input data type). If so, a type conversion will be required. In most cases, the value mapping from one type to another is obvious (e.g., from "integer" to "real"), and will not be discussed. In cases where the mapping is not obvious (e.g., from "angle" to "boolean") it will be given in the "Notes" section following the table.
- (4) any value constraints placed on the item; if the produced display value exceeds these constraints, this function driver should display the nearest limiting value.
- (5) the required display format of the item; each window-display access program is of the form +SS_S_format_window+, where "format" is replaced as appropriate by the entry from this column.
- (6) the window in which to display the item: UPPER or LOWER.

WHEN TO DISPLAY AN !!ITEM!!:

A display !!item!! is displayed on the specified window with the specified format each time the following occurs:

```
@T( !+pnl config+! = !!item!! )
```

The !!item!! value is re-displayed on the window everytime its value changes by at least !!resolution!! amount, until the event @T(!+pnl config = !!item!!) occurs for some other !!item!!. The term !!resolution!! is defined in the local dictionary for each !!item!!.

WHEN TO DISPLAY NOTHING:

When @T(!+Pnl config changed+!) occurs, this function should blank out both windows by calling +SS_CLEAR_UPPER+ and +SS_CLEAR_LOWER+.

Selector Table 9.3-a -- Definition of panel displays

<u>!!item!!</u>	<u>Input Type</u>	<u>Output type if different and value constraints</u>	<u>Format</u>	<u>Window</u>
<u>Altitude displays</u>				
!!alt baro AGL!!	distance	integer(0 - 65535 feet)	UINT	UPPER
!!Priority alt display!!	various	char string	CHARLIT	LOWER
<u>Device reconfiguration values -- ADC, IMS, and PML'S</u>				
!!+L-probe+!	boolean	--	SIGN	LOWER
!!+E coarse bias+!	real	-.02 - +.02	SFRAC	LOWER
!!Displ E coarse scale!!	real	0, or .026 - .038	UFRAC	LOWER
!!+E fine bias+!	real	-.01 - +.01	SFRAC	LOWER
!!Displ E fine scale!!	real	0, or .00026-.00038	UFRAC	LOWER
!!+N coarse bias+!	real	-.02 - +.02	SFRAC	LOWER
!!Displ N coarse scale!!	real	0, or .026 - .038	UFRAC	LOWER
!!+N fine bias+!	real	-.01 - +.01	SFRAC	LOWER
!!Displ N fine scale!!	real	0, .00026-.00038	UFRAC	LOWER
!!+V coarse bias+!	real	-.02 - +.02	SFRAC	LOWER
!!Displ V coarse scale!!	real	0, or .026 - .038	UFRAC	LOWER
!!+X corr increm+!	real	.32 - .48	UFRAC	LOWER
!!+X drift+!	real	-1 - +1	SFRAC	LOWER
!!+Y corr increm+!	real	.32 - .48	UFRAC	LOWER
!!+Y drift+!	real	-1 - +1	SFRAC	LOWER
!!+Z corr increm+!	real	.32 - .48	UFRAC	LOWER
!!+Z drift+!	real	-1 - +1	SFRAC	LOWER
!!central long a!!	longitude	-180°0'0" - +180°0'0"	LONGITUDE	LOWER
!!central long b!!	longitude	-180°0'0" - +180°0'0"	LONGITUDE	LOWER
!!low lat ct a!!	integer	-90 - +90	SINT	LOWER
!!low lat ct b!!	integer	-90 - +90	SINT	LOWER
!!map orient a!!	angle	boolean -- See note 2	SIGN	LOWER
!!map orient b!!	angle	boolean -- See note 2	SIGN	LOWER
<u>Doppler-related displays</u>				
!!gndspd filtered!!	speed	integer(0 - 1214 knots)	UINT	LOWER
!!drift angle filtered!!	angle	integer(-180 - +180 deg)	SINT	LOWER

Selector Table 9.3-a (continued)

<u>!!item!!</u>	<u>Input Type</u>	<u>Output type if different and value constraints</u>	<u>Format</u>	<u>Window</u>
<u>Hardware diagnostic displays</u>				
!!IMS diags1!!	boolean	charlit	CHARLIT	UPPER
!!IMS diags2!!	boolean	charlit	CHARLIT	LOWER
!!MFSW diags!!	boolean	charlit	CHARLIT	UPPER
!!Map sw diags!!	boolean	charlit	CHARLIT	LOWER
!!Nav diags1!!	boolean	charlit	CHARLIT	UPPER
!!Nav diags2!!	boolean	charlit	CHARLIT	LOWER
!!SINS valid1!!	boolean	charlit	CHARLIT	UPPER
!!SINS valid2!!	boolean	charlit	CHARLIT	LOWER
!!STARDY diags!!	boolean	charlit	CHARLIT	UPPER
!!Wpn sw diags!!	boolean	charlit	CHARLIT	LOWER
!+ARPINT+!	integer	0 - 990	UINT	LOWER
!+ARPQUANT+!	integer	0 - 99	UINT	UPPER
!+WEAPTYP+!	integer	0 - 99	UINT	LOWER
<u>Heading displays</u>				
!+az ref hdg pnl+!	angle	0° 0' - 360° 0'	ANGLE	LOWER
!+heading IMS+!	angle	0° 0' - 360° 0'	ANGLE	UPPER
!+heading MAG+!	angle	0° 0' - 360° 0'	ANGLE	LOWER
!!hdg system!!	angle	0° 0' - 360° 0'	ANGLE	UPPER
<u>IMS-related displays</u>				
!!align stage!!	astage	charlit	CHARLIT	UPPER
!!drftangl IMS!!	angle	integer(-180 - +180 deg)	SINT	UPPER
!!groundspeed IMS!!	speed	integer(0 - 1214 knots)	UINT	UPPER
!!gyro drift delta n!!	real	-.99 - +.99	SFRAC	LOWER
<u>Position-related displays</u>				
!+Latitude error+!	latitude	N90°0'0" - S90°0'0"	LATITUDE	UPPER
!+Map latitude+!	latitude	N99°59'59 - S99°59'59	LATITUDE	UPPER
!+latitude+!	latitude	N99°59'59 - S99°59'59	LATITUDE	UPPER
!+Longitude error+!	longitude	E180°0'0" - W180°0'0"	LONGITUDE	LOWER
!+Map longitude+!	longitude	E180°0'0" - W180°0'0"	LONGITUDE	LOWER
!+longitude+!	longitude	E180°0'0" - W180°0'0"	LONGITUDE	LOWER

Selector Table 9.3-a (continued)

<u>!!item!!</u>	<u>Input Type</u>	<u>Output type if different and value constraints</u>	<u>Format</u>	<u>Window</u>
<u>Release-time displays</u>				
!!Alt AGL at rls!!	distance	int(0 - 65535 ft)	UINT	LOWER
!!fpangl at rls!!	angle	real (-180 - +180 deg)	REAL	LOWER
!!norm accel at rls!!	accel	real (0 - 32767 g's)	REAL	LOWER
!!Slant range at rls!!	distance	integer(0 - 262141 feet)	UINT	UPPER
!!az miss dist at rls!!	distance	integer(0 - 65535 feet)	UINT	UPPER
!!TAS ADC at rls!!	speed	integer(0 - 32767 knots)	UINT	UPPER
<u>SINS-related displays</u>				
!+SINS dhdg pnl+!	angle	0°0'0" - 360°0'0"	ANGLE	LOWER
!+SINS heading+!	angle	0° 0' - 360° 0'	ANGLE	LOWER
!+SINS lat+!	latitude	N90°0'0" - S90°0'0"	LATITUDE	UPPER
!+SINS long+!	longitude	E180°0'0" - W180°0'0"	LONGITUDE	LOWER
!+SINS x offset pnl+!	distance	integer(-2047 - +2047 ft)	SINT	UPPER
!+SINS y offset pnl+!	distance	integer(-2047 - +2047 ft)	SINT	LOWER
!+SINS z offset pnl+!	distance	integer(-2047 - +2047 ft)	SINT	UPPER
!+SINS east vel+!	speed	integer(-2047 - +2047kts)	SINT	LOWER
!+SINS north vel+!	speed	integer(-2047 - +2047kts)	SINT	LOWER
<u>Software switch displays</u>				
!+data nbr pnl+!	integer	0 - 99	UINT	UPPER
!+Doppler coupled pnl+!	boolean	--	SIGN	LOWER
!+Land based pnl+!	boolean	--	SIGN	LOWER
!+Radalt priority pnl+!	boolean	--	SIGN	LOWER
<u>Static displays</u>				
!+Compfail config+!	--	See note 3.	--	--
!!OFP version upper!!	charlit	--	CHARLIT	UPPER
!!OFP version lower!!	charlit	--	CHARLIT	LOWER
<u>Time-related displays</u>				
!+elapsed navaln time+!	time	0:00:00 - 6:45:00	TIME	LOWER
!+time to ftpt+!	time	See note 1.	TIME	LOWER
<u>Velocity displays</u>				
!!TAS filtered!!	speed	integer(0 - 1214 knots)	UINT	LOWER
!!IMS total vel!!	speed	integer(0 - 1214 knots)	UINT	UPPER
!!vel E!!	speed	integer(-2047 - +2047kts)	SINT	UPPER
!!vel N!!	speed	integer(-2047 - +2047kts)	SINT	UPPER

Selector Table 9.3-a (continued)

<u>!!item!!</u>	<u>Input Type</u>	<u>Output type if different and value constraints</u>	<u>Format</u>	<u>Window</u>
		<u>Wind displays</u>		
!+wind vel+!	integer	0 - 255 knots	UINT	UPPER
!+wind dir+!	angle	000'0" - 36000'0"	ANGLE	LOWER

Items indexed by pilot-chosen number

For all display items in this section, there are several different versions of each that may be displayed. The version that should be displayed is denoted in the specifications by "i", where $1 \leq i \leq \text{cnum destsc}$. The value for "i" is obtained from the data banker item !+dest entry pnl+!.

!+dest altitude pnl+!(i)	distance	integer(-65535 - 65535ft)	SINT	UPPER
!+dest lat+! (i)	latitude	N99059'59"-S99059'59"	LATITUDE	UPPER
!+dest long+! (i)	longitude	E18000'0" - W18000'0"	LONGITUDE	LOWER
!+dest mslp pnl+! (i)	pressure	real (0 - 40.95 in. Hg)	REAL	LOWER
!+mark lat+! (i)	latitude	N99059'59"-S99059'59"	LATITUDE	UPPER
!+mark long+! (i)	longitude	E18000'0" - W18000'0"	LONGITUDE	LOWER
!+offset brg pnl+! (i)	angle	000'0" - 36000'0"	ANGLE	LOWER
!+offset dht pnl+! (i)	distance	integer(-65535 - 65535ft)	SINT	UPPER
!+offset rng pnl+! (i)	distance	integer(0 - 131,070 ft)	UINT	UPPER
!+burst ht pnl+! (i)	distance	integer(0 - 65535 ft)	UINT	LOWER
!+Mag variation pnl+!(i)	angle	18000'0" - W18000'0"	LONGITUDE	LOWER

Notes:

1. If !+time to dest+! is greater than 6:45:00, then 0:00:00 is displayed.
2. The mapping between types "angle" and "boolean" is as follows: The boolean value is true iff the angular measure is 0 degrees.
3. The display for this item consists of a blank panel, with all the format lights turned on. There is no input item associated with this display.

Local dictionary:TermDefinition

Any term of the
form !! item a!!

The value of !+ (item) +!, parameterized by the map
value \$A\$. For instance, !!map orient a!! is
defined as !+map orient+! parameterized by the map
value \$A\$.

Any term of the
form !! item b!!

The value of !+ (item) +! parameterized by the map
value \$B\$. For instance, !!low lat ct b!! is
defined as !+low lat ct+! parameterized by the map
value \$B\$.

!!align stage!!

If the current !+align_stage+! is \$FM\$ or \$HS\$,
this display is blank. Otherwise, the alignment
stage is displayed in positions three and four of
an otherwise-blank six-character string.

!!Alt AGL at rls!!

An altitude measure taken at the time of the first
weapon release in the most recent stik. The value
is updated whenever the event @T(!+rls_pts_passed+!
= 1) occurs. The value to be used in the update
is by the table below.

Definition of !!Alt AGL at rls!!

MODES

CONDITIONS

```

=====
*CCIP*
*HUDDown1*
*HUDDown2*
*NatAttack*      !+sr_
*Noffset*        reason-
*SHUDDown1*      able+!
*SHUDDown2*
*Snattack*
*Snoffset*

```

	NOT !+sr_	NOT !+sr_
	reasonable+!	reasonable+!
	AND	AND
	!+Radalt	NOT !+Radalt
	priority	priority
	pnl+!	pnl+!

```

=====
*A/A Manrip*
*BOC*
*BOCFlyto0*
*BOCOffset*      X
*Manrip*
*SBOC*
*SBOCFlyto0*
*SBOCOffset*

```

	!+Radalt	NOT !+Radalt
	priority	priority
	pnl+!	pnl+!

```

=====
VALUE:      !+alt      !+alt      !!altitude
              from sr+!  RADAR+!    baro above
                                   tgt!!

```

!!altitude aro
above tgt!! Defined by the table below. In the table,
!+dest altitude pnl+! and !+offset dht pnl+! are
indexed by !+Fly to num+!.

MODES	DEFINITION
=====	
BOCOffset	
HUDDown2	
Noffset	!+alt ADC+! - !+dest altitude pnl+!
SBOCOffset	- !+offset dht pnl+!
SHUDDown2	
Snoffset	

Any other	!+alt ADC+! - !+dest altitude pnl+!
weapon mode	
=====	

!!alt baro AGL!! Definition in table below.

MODES	EVENTS	
=====		
AflyUpd	X	@T(!+Dest selected+!) OR @T(!+TD pressed+!)

BOCFlyto0		
HUDDown1		
HUDDown2	@F(!+desig+!)	@T(!+desig+!)
Nattack	OR	OR
Noffset	@T(In mode)	@F(!+Slew displacement non-zero+!)
SBOCFlyto0		WHEN(!+rls_pts_passed+! = 0)
SHUDDown1		
SHUDDown2		
Snattack		
Snoffset		

BOC	@T(In mode	@T(!+gr_ac_tgt+! lseq 30 nmi)
	AND	OR
SBOC	!+gr_ac_tgt+!	@F(!+Slew displacement non-zero+!)
	gt 30 nmi)	WHEN(!+gr_ac_tgt+! lseq 20 nmi
		AND !+rls_pts_passed+! = 0)

BOCOffset	@T(In mode)	@T(!+desig+!) OR
	OR	@T(!+gr_ac_oap+! lseq 30 nmi OR
SBOCOffset	@F(!+desig+!)	@F(!+Slew displacement non-zero+!)
	OR	WHEN(!+gr_ac_oap+! lseq 20 nmi
	@T(!+gr_ac_tgt+!	AND !+rls_pts_passed+! = 0)
	gt 30 nmi)	

CCIP	@T(In mode)	@T(!+ip elev+! lseq 16 deg)
		WHEN(!+rls_pts_passed+! = 0)
=====		
VALUE:	0 feet	!!altitude baro above tgt!!

!!az miss dist at rls!!

The value of !+az miss dist+! at the first release of the most recent stik, or zero. Defined by the table below.

Definition of !!az miss dist at rls!!

MODES	EVENTS	
=====		
BOC		
BOCFlyto0		
BOCOffset		
HUDDown1		
HUDDown2		
Nattack		
Noffset	X	@T(!+rls_pts_passed+! = 1)
SBOC		
SBOCFlyto0		
SBOCOffset		
SHUDDown1		
SHUDDown2		
Snattack		
Snoffset		

A/A Manrip		
CCIP	@T(In	X
Manrip	mode)	
=====		
VALUE:	0 feet	!+az miss dist+!

!!Displ E coarse scale!!

!+E coarse scale+! if value is noteq
çIMSR_init_coarse_vscaleç; 0 otherwise.

!!Displ E fine scale!!

!+E fine scale+! if value is noteq
çIMSR_init_fine_vscaleç; 0 otherwise.

!!Displ N coarse scale!!

!+N coarse scale+! if value is noteq
çIMSR_init_coarse_vscaleç; 0 otherwise.

!!Displ N fine scale!!

!+N fine scale+! if value is noteq
çIMSR_init_fine_vscaleç; 0 otherwise.

!!Displ V coarse scale!!

!+V coarse scale+! if value is noteq
çIMSR_init_coarse_vscaleç; 0 otherwise.

!!drftangl IMS!!

0 degrees if !+nav mode+! = *IMS fail*;
!+drift angle IMS+! otherwise.

!!drift angle filtered!!

!+drift angle DRS+! averaged over 3 seconds.

!!fpangl at rls!!

The value used is !+flt path angle+!. The value is updated when in one of the following weapon modes: *A/A Manrip*, *BOC*, *BOCFlyto0*, *BOCoffset*, *CCIP*, *HUDDown1*, *HUDDown2*, *Manrip*, *Nattack*, *Noffset*, *SBOC*, *SBOCFlyto0*, *SBOCoffset*, *SHUDDown1*, *SHUDDown2*, *Snattack*, or *Snoffset*. The value is updated whenever @T(!+rls_pts_passed+! = 1) occurs.

!!gndspd filtered!!

smoothed !+gnd speed DRS+!.

!!groundspeed IMS!!

0 fps if !+nav_mode+! = *IMS fail*;
!+IMS horiz velocity+! otherwise.

!!gyro drift delta n!!

The difference between the latest value of !+X drift+! and the previous value. The value is updated in *0lUpdate* mode when @F(!+align_stage+! = \$TS\$) occurs. Also, the value is set to 0 when @T(*Landaln*) occurs.

!!hdg system!!

Definition of !!hdg system!!

MODES	CONDITIONS	
=====		
All	!+IMS ready+!	NOT !+IMS ready+!
modes	AND	OR
	!+IMS rel+!	NOT !+IMS rel+!
=====		
VALUE:	!+heading	!+heading MAG+!
	IMS+!	+
		!+magvar IMS+!

!!item!!

Enumerated by Table FD.9.3-a. This tells not only the values to use for display, but also the configuration that the panel must be in for that item to be displayed.

!!IMS diagsl!!

This is a six-element string of character literals. The conditions for each element are given below. If a condition is true, then the element has the value \$1\$; if false, \$0\$.

Element #	Condition
1	!+IMSREL+!
2	!+ACAIRB+!
3	!+MA+!
4	!+IMSMODE eq Gndal+!
5	!+IMSMODE eq Norm+!
6	!+IMSMODE eq Iner+!

!!IMS diags2!!

This is a seven-element string of character literals. The conditions for each element are given below. If a condition is true, then the element is set to \$1\$; if false it is set to \$0\$.

<u>Element #</u>	<u>Condition</u>
1	!+IMSMODE eq Grid+!
2	!+IMSMODE eq Magsl+!
3	NOT !+IMSREDY+!
4	!+IMSAUTOC+!
5	!+ADCFAIL+!
6	!+ARPPAIRS+!
7	Always set to \$blank\$

!!IMS total vel!!

0 fps if !+nav mode+! = *IMS fail*; otherwise, !+IMS total velocity+!.

!!Map sw diags!!

This is a seven-element string of character literals. The conditions for each element are given below. If a condition is true, then the element is set to \$1\$; otherwise it is set to \$0\$.

<u>Element #</u>	<u>Condition</u>
1	!+PMSCAL eq 80+!
2	!+PMHOLD+!
3	!+PMDCTR+!
4	!+PMNORUP+!
5	!+PMLAND+!
6	!+HUDREL+!
7	Always set to \$blank\$

!!MFSW diags!!

This is a six-element string of character literals. The conditions for each element are given below. If a condition is true, then the element is set to \$1\$; otherwise it is set to \$0\$.

<u>Element #</u>	<u>Condition</u>
1	!+BMBDRAG eq High+!
2	!+MFSW eq NATT+! OR !+MFSW eq NATTOFF+!
3	!+MFSW eq NATTOFF+! OR !+MFSW eq BOCOFF+!
4	!+MFSW eq BOC+! OR !+MFSW eq BOCOFF+!
5	!+MFSW eq CCIP+!
6	!+MFSW eq TF+!

!!Nav diags1!!

This is a six-element string of character literals. The conditions for each element are given below. If a condition is true, then the element has the value given after the condition.

Element #	Condition	Value
1	NOT !+ACAIRB+!	\$1\$
2	Always	\$blank\$
3	(!+nav_mode+! = *Mags1* OR *Grid* OR *IMS fail*) OR (!+align_mode+! = *Airln* AND !+align_stage+! = \$FM\$)	\$1\$
4	Always	\$blank\$
5	NOT !+IMSREDY+!	\$1\$
6	Always	\$blank\$

!!Nav diags2!!

This is a seven-element string of character literals. The conditions for each element are given below. If a condition is true, then the element has the value given after the condition.

Element #	Condition	Value
1	NOT !+IMSREDY+! OR NOT !+IMSREL+!	\$1\$
2	Always	\$blank\$
3	NOT !+IMS reasonable+!	\$1\$
4	Always	\$blank\$
5	NOT !+DRSREL+!	\$1\$
6	Always	\$blank\$
7	NOT !+Doppler reasonable+!	\$1\$

!!norm accel at rls!!

The value used is !+normal accel+!. The value is updated when in one of the following weapon modes: *A/A Manrip*, *BOC*, *BOCflyto0*, *BOCoffset*, *CCIP*, *HUDDown1*, *HUDDown2*, *Manrip*, *Nattack*, *Noffset*, *SBOC*, *SBOCflyto0*, *SBOCoffset*, *SHUDDown1*, *SHUDDown2*, *Snattack*, or *Snoffset*. The value is updated whenever @T(!+rls_pts_passed+! = 1) occurs.

!!OFP version upper!!
!!OFP version lower!!

Two character strings to be displayed in the panel display windows denoting information about this OFP. Defined at system generation time by the sysgen parameters çOFP version upperç and çOFP version lowerç, respectively.

!!Priority alt display!!

The character form of !+alt priority stale+!, prefixed by the character form of !+alt priority source+!.

!!resolution!!

The required display resolution for each display item is given in the following table.

<u>Display item:</u>	<u>Display resolution:</u>
!+ADC smoothed TAS+!	1 knot
!!Alt AGL at rls!!	1 foot
!+ARPINT+!	10 feet
!+ARPQUANT+!	1
!!az miss dist at rls!!	.01 feet
!+az ref hdg pnl+!	1 minute
!+alt baro AGL+!	1 foot
!+burst ht pnl+!	2 feet (rounded down value)
!+central long a pnl+!	1 second
!+central long b pnl+!	1 second
!+Data nbr pnl+!	1 (integer)
!+dest altitude pnl+!	2 feet (rounded down value)
!+dest lat+!	1 second
!+dest long+!	1 second
!+dest mslp pnl+!	.01 inches
!+drift angle IMS+!	1 degree
!+DRS treated gndspd+!	1 knot
!+DRS avgd drift angle+!	1 degree
!+E coarse bias pnl+!	.0003 feet/sec/sec
!+E coarse scale pnl+!	.00003 feet/sec/pulse
!+E fine bias pnl+!	.0003 feet/sec/sec
!+E fine scale pnl+!	.000001 feet/sec/pulse
!+E vel IMS+!	1 knot
!+elapsed navaln time+!	1 second
!!fpangl at rls!!	.01 degrees
!+groundspeed IMS+!	1 knot
!!gyro drift delta n!!	.001 deg/hour
!+hdg IMS system+!	1 minute
!+heading IMS+!	1 minute
!+heading MAG+!	1 minute
!+IMS total vel+!	1 knot
!+latitude+!	1 second
!+Latitude error+!	1 second
!+longitude+!	1 second
!+Longitude error+!	1 second
!+low lat ct a pnl+!	10 seconds
!+low lat ct b pnl+!	10 seconds
!+Mag variation pnl+!	1 second
!+Map latitude+!	1 second
!+Map longitude+!	1 second
!+mark lat+!	1 second
!+mark long+!	1 second
!+N coarse bias pnl+!	.0003 feet/sec/sec
!+N coarse scale pnl+!	.00003 feet/sec/pulse
!+N fine bias pnl+!	.0003 feet/sec/sec
!+N fine scale pnl+!	.000001 feet/sec/pulse
!+N vel IMS+!	1 knot

Display item:

!!norm accel at rls!!
 !+offset brg pnl+!
 !+offset dht pnl+!
 !+offset rng pnl+!
 !+SINS dhdg pnl+!
 !+SINS east vel+!
 !+SINS heading+!
 !+SINS lat+!
 !+SINS long+!
 !+SINS north vel+!
 !+SINS x offset pnl+!
 !+SINS y offset pnl+!
 !+SINS z offset pnl+!
 !!Slant range at rls!!
 !!TAS ADC at rls!!
 !+time to dest+!
 !+V coarse bias pnl+!
 !+V coarse scale pnl+!
 !+WEAPTYP+!
 !+wind dir+!
 !+wind vel+!
 !+X corr increm pnl+!
 !+X drift pnl+!
 !+Y corr increm pnl+!
 !+Y drift pnl+!
 !+Z corr increm pnl+!
 !+Z drift pnl+!

Display resolution:

.01 g
 1 second
 2 feet (rounded down value)
 4 feet (rounded down value)
 1 second
 1 knot
 1 minute
 1 second
 1 second
 1 knot
 1 foot
 1 foot
 1 foot
 8 feet
 1 knot
 1 second
 .0003 feet/sec/sec
 .00003 feet/sec/pulse
 1 (integer)
 1 second
 1 knot
 .0004 sec/pulse
 .001 deg/hour
 .0004 sec/pulse
 .001 deg/hour
 .0004 sec/pulse
 .001 deg/hour

Term

!!SINS valid!!!

Definition

This is a six-element string of character literals. The conditions for each element are given below. If a condition is true, then the element has the value given after the condition.

<u>Element #</u>	<u>Condition</u>	<u>Value</u>
1	!+SINS heading valid+!	\$0\$
2	Always	\$Blank\$
3	!+SINS north vel valid+!	\$0\$
4	Always	\$Blank\$
5	!+SINS roll valid+!	\$0\$
6	Always	\$Blank\$

!!SINS valid2!!

This is a seven-element string of character literals. The conditions for each element are given below. If a condition is true, then the element has the value given after the condition.

Element #	Condition	Value
1	!+SINS east vel valid+!	\$0\$
2	Always	\$Blank\$
3	!+SINS pitch valid+!	\$0\$
4	Always	\$Blank\$
5	!+SINS lat valid+!	\$0\$
6	Always	\$Blank\$
7	!+SINS long valid+!	\$0\$

!!Slant range at rls!!

Definition of !!Slant range at rls!!

MODES

EVENTS

BOC

BOCFlyto0

BOCOffset

HUDDown1

HUDDown2

Nattack

Noffset

SBOC

SBOCFlyto0

SBOCOffset

SHUDDown1

SHUDDown2

Snattack

Snoffset

CCIP

X

X

@T(!+rls_pts
_passed+!=1)

A/A Manrip @T(In

Manrip mode)

X

X

VALUE:

0 feet

!+sr_ac_tgt+! !+sr_ac_ip+!

!!STARDY diags!!

This is a six-element string of character literals. The conditions for each element are given below. If a condition is true, then the element has the value \$1\$; otherwise, \$0\$.

Element #	Condition
1	!+STA1RDY+!
2	!+STA2RDY+!
3	!+STA3RDY+!
4	!+STA6RDY+!
5	!+STA7RDY+!
6	!+STA8RDY+!

!!TAS ADC at rls!!

The value used is !+TAS ADC+!. The value is updated when in one of the following weapon modes: *A/A Manrip*, *BOC*, *BOCflyto0*, *BOCoffset*, *CCIP*, *HUDDown1*, *HUDDown2*, *Manrip*, *Nattack*, *Noffset*, *SBOC*, *SBOCflyto0*, *SBOCoffset*, *SHUDDown1*, *SHUDDown2*, *Snattack*, or *Snoffset*. The value is updated whenever @T(!+rls_pts_passed+! = 1) occurs.

!!TAS filtered!!

!+TAS ADC+!, after smoothing.

!!vel E!!

!!vel N!!

MODES		EVENTS	
=====			
SINSaln	@T(!+new align	@T(!!disp	
Sautocal	stage+!)	diff!!)	@F(In mode)

Landaln		@T(!!disp	
Lautocal	X	diff!!)	@F(In mode)

IMS fail	X	X	@T(In mode)

All other			
navigation	X	@T(!!disp	X
or align.		diff!!)	
modes			
=====			
!!vel E!!:	!+SINS E vel+!	!+E vel IMS+!	0 fps
!!vel N!!:	!+SINS N vel+!	!+N vel IMS+!	0 fps

!!Wpn sw diags!!

This is a seven-element string of character literals. The conditions for each element are given below. If a condition is true, then the element has the value given after the condition.

<u>Element #</u>	<u>Condition</u>	<u>Value</u>
1	!+GUNSEL+!	\$1\$
2	!+RE+!	\$1\$
3	!+MULTRACK+!	\$1\$
4	!+TD+!	\$1\$
5	Always	\$blank\$
6	Always	\$blank\$
7	Always	\$blank\$

Design issues:

1. At one time, this module was responsible for receiving input from the panel as well as displaying data on it. Now, however, the input operations reside completely in the Shared Services Panel i/o submodule. A parallel may be drawn to the SINS input: a module is responsible for collecting the input and reporting it to the outside world, while hiding the method of input.

2. In a previous version, there was a class of functions which controlled both windows at once. This class included the diagnostic displays, the OFP version display, the error display, and a few more. Most of the other displays had already been divorced from their other-window partners, and the error display has been internalized to the module that handles panel inputs. It was decided to hide the pairing of window displays completely, and so even the limited class of "dual-window" displays was broken up into single-window constituents. It was felt that a future requirements change could easily break up any dual-window display into two unrelated displays. It would be much easier to handle such a change if there were no displays that were considered to use both windows.

```
*****
*
*          FD.10          PMDS functions          *
*
*****
```

FD.10.1 PERIODIC FUNCTION DESCRIPTION: Set the map indicator.

Mnemonic: +FD_MAP_IND_P+

Output produced:

<u>Item</u>	<u>Type</u>	<u>Access program</u>
Map indicator value	angle	+DI_S_MAP_INDICATOR+

Initiation/termination events:

None. Always performed.

Function definition:

This function always outputs the value !+heading MAG+! + !+grtk+!
!+heading MAG+!.

FD.10.2 PERIODIC FUNCTION DESCRIPTION: Set the map orientation angle.

Mnemonic: +FD_MAP_ORIENT_P+

Output produced:

<u>Item</u>	<u>Type</u>	<u>Access program</u>
Map orientaton angle	angle	+DI_S_MAP_ROTATION+

Initiation/termination events:

None. Always performed.

Function definition:

Condition Table FD.10.2-a -- Map orientation control

MODES		CONDITIONS	
=====			
All modes	!+Map north-up+!	NOT !+Map north-up+!	!+Map hold+!
except	AND	AND	
Grtest	NOT !+Map hold+!	NOT !+Map hold+!	
=====			
Output			
value:	0 degrees	!+grtk+!	!!stale orient!!

Local dictionary:

<u>Term</u>	<u>Definition</u>
!!stale orient!!	last value of map orientation angle before @T(!+Map hold+!)

FD.10.3 PERIODIC FUNCTION DESCRIPTION: Set the map pointer.Mnemonic: +FD_MAP_PTR_P+Output produced:

<u>Item</u>	<u>Type</u>	<u>Access program</u>
Map pointer angle	angle	+DI_S_MAP_POINTER_ANGLE+

Initiation/termination events:

None. Always performed.

Function definition:

MODES	CONDITIONS	
=====		
All modes	!+Fly to num+! = 0	!+Fly to num+! noteq 0
=====		
Output value:	0 degrees	!+brg_grtk_ftpt+!

FD.10.4 DEMAND FUNCTION DESCRIPTION: Set the map reference point.

Mnemonic: +FD_MAP_REF_PT_D+

Output produced:

<u>Item</u>	<u>Type</u>	<u>Access program</u>
Map reference point	ref_pt	+DI_S_MAP_REFERENCE_PT+

Function definition:

Event Table FD.10.4-a -- Map reference point control

MODES

EVENTS

```

=====
All modes      @T(!+Init complete+!)      @T(!+Init complete+!)
except         WHEN(!+Map north-up+! OR  WHEN(NOT !+Map north-up+!
*Grtest*      NOT !+Map decenter+!)      AND !+Map decenter+!)
                                   OR
                                   OR
                                   @F(NOT !+Map north-up+!
                                   AND !+Map decenter+!)      @T(NOT !+Map north-up+!
                                   AND !+Map decenter+!)
=====

```

```

=====
Output value:      $center$                      $bottom-center$
=====

```

FD.10.5 PERIODIC FUNCTION DESCRIPTION: Position the map.Mnemonic: +FD_MAP_DISPLAY_P+Output produced:

<u>Item</u>	<u>Type</u>	<u>Access program</u>
To display the map warning	--	+DI_DISPLAY_MAP_WARNING+
To display a location or remove the map warning	--	+DI_S_MAP_POSITION+

Initiation/termination events: None. Always performed.Function definition:

Condition Table FD.10.5-b: Setting the map position

MODES	CONDITIONS	
=====	=====	
All navigation and alignment modes;	!!Posn displayable!!	NOT !!Posn displayable!!
MapUpd		
=====	=====	

FUNCTION

RESULT: Call +DI_S_MAP_POSITION+ Call +DI_DISPLAY_MAP_WARNING+

Note that the Requirements do not specify what to do if a desired display location is not displayable with the current map set. The condition is described in the Requirements as "Error -- undefined". NWC-2 does not perform displayability checks. Since our virtual PMDS does perform this check, we have chosen to have the map display a warning in case a location is not displayable.

Local dictionary:

<u>Term</u>	<u>Definition</u>
!!refpt!!	Defined by the table below.
!!refpt lat!!	Defined by the table below.
!!refpt long!!	Defined by the table below.

Condition Table FD.10.5-c -- Definition of !!refpt!!,
!!refpt lat!!, and !!refpt long!!

MODESCONDITIONS

```

=====
All alignment  NOT !!Dest/Mark!!  NOT !+Map hold+!  !+Map hold+!
& navigation      AND NOT                AND NOT                OR
modes;            !+Map hold+!    !+during slewing+! !+during slewing+!
                  AND NOT                AND
Not *MapUpd*      !+during slewing+! !!Dest/Mark!!
=====

```

```

=====
*MapUpd*        NOT !+desig+!                !+desig+!
                  AND NOT                X                OR
                  !+during slewing+!    !+during slewing+!
=====
!!refpt!!:      a/c present posn  !!Recalled pt!!    !!Slewed-to point!!

!!refpt lat!!:  !+latitude+!      See local dict.    !!slewed map lat!!

!!refpt long!!: !+longitude+!     See local dict.    !!slewed map long!!
=====

```

TermDefinition

!!Dest/Mark!! !!Dest displayed!! OR !!Mark displayed!!.

!!Dest displayed!! true iff calling +SS_G_PNL_CONFIG+(\$dest lat\$,p2)
OR calling +SS_G_PNL_CONFIG+(\$dest long\$,p2)
returns p2 with a value of true. This means that
the panel is in the \$dest lat\$ or \$dest long\$
configuration.

!!Mark displayed!! true iff calling +SS_G_PNL_CONFIG+(\$mark lat\$,p2)
OR calling +SS_G_PNL_CONFIG+(\$mark long\$,p2)
returns p2 with a value of true. This means that
the panel is in the \$mark lat\$ or \$mark long\$
configuration.

Local dictionary (continued):

<u>Term</u>	<u>Definition</u>
!!Posn displayable!!	<u>true</u> iff the map is capable of displaying the next location required to be displayed. Value is obtained from the DIM by calling +DI_G_MAP_DISPLAYABLE+(!!refpt long!!, !!refpt lat!!, !!Posn displayable!!).
!!Recalled pt!!	Defined by table below. MODES

FD.10.6 DEMAND FUNCTION DESCRIPTION: Set the map scale.

Mnemonic: +FD_MAP_SCALE_D+

Output produced:

<u>Item</u>	<u>Type</u>	<u>Access program</u>
New map scale	map_scale	+DI_S_MAP_SCALE+

Function definition:

Event Table FD.10.6-d -- Setting the map scale

<u>MODES</u>	<u>EVENTS</u>
=====	=====
All modes	@T(!+Init complete+!)
	WHEN(NOT !+Map scale sw+!)
	OR @F(!+Map scale sw+!)
	OR @T(!+Map scale sw+!)
=====	=====
Map scale:	cMap_scale_arrayc(1)
	cMap_scale_arrayc(2)

FD.10.7 DEMAND FUNCTION DESCRIPTION: Set the map reconfiguration values.

Mnemonic: +FD_MAP_RECONFIG_D+

Output produced:

<u>Item</u>	<u>Type</u>	<u>Access program</u>
New central long.	longitude	+DI_S_MAP_LONGITUDE+
New lower lat. count	integer	+DI_S_MAP_LATITUDE_CT+
New map orient.	angle	+DI_S_MAP_ORIENTATION+

Function definition:

Event Table FD.10.7-a -- Map reconfiguration updates

<u>MODES</u>	<u>EVENTS</u>
=====	=====
All modes except *Grtest*	@T(!+New !!input item!! entered+!)
=====	=====
Output value:	!+ !!input item!! +! and !!mapid!!

For instance, one of the values of !!input item!! is "central long a pni" (see local dictionary). Therefore, when @T(!+central long a pni entered+!) occurs, this function should output the new value of !+central long a pni+!, along with the corresponding !!mapid!! which is \$\$\$, via the access program +DI_S_MAP_LONGITUDE+.

Local dictionary:TermDefinition

!!input item!!

any of the various input items used to update an IMS Reconfiguration value. Enumerated in Table FD.10.7-b below.

!!mapid!!

the parameter used in the DIM PMDS Reconfiguration access program to specify which map is being updated. Either \$A\$ or \$B\$. Each !!input item!! has a corresponding !!mapid!!. The correspondence is given in Table FD.10.7-b below.

Table FD.10.7-b -- PMDS Reconfiguration values

!!input item!!	type	corresponding !!mapid!!
=====	=====	=====
!+central long a pnl+!	longitude	(\$A\$)
!+central long b pnl+!	longitude	(\$B\$)
!+low lat ct a pnl+!	integer	(\$A\$)
!+low lat ct b pnl+!	integer	(\$B\$)
!+map orient a pnl+!	angle	(\$A\$)
!+map orient b pnl+!	angle	(\$B\$)
=====	=====	=====

```

*****~*****
*
*          FD.11          SINS functions
*
*****

```

FD.11.1 DEMAND FUNCTION DESCRIPTION: Start and stop the SINS.

Mnemonic: +FD_SINS_CNTRL_D+

Output produced:

<u>Item</u>	<u>Type</u>	<u>Access program</u>
To start the SINS	--	+DI_START_SINS+
To stop the SINS	--	+DI_STOP_SINS+

Function definition:

Event Table FD.11.1-a -- Starting/stopping SINS

<u>MODES</u>	<u>EVENTS</u>	
=====	=====	=====
All assignment and navigation modes	@T(!+In flight+!) OR @T(!+Land based+!)	@T(!+Init complete+!) WHEN (NOT !+Land based pnl+!) AND NOT !+In flight+!) OR @F(!+Land based+!) WHEN (NOT !+In flight+!) OR @F(!+In flight+!) WHEN (NOT !+Land based pnl+!)
=====	=====	=====
Function call:	+DI_STOP_SINS+	+DI_START_SINS+

```

*****
*
*           FD.12           Visual indicator functions
*
*****

```

FD.12.1 DEMAND FUNCTION DESCRIPTION: Control the visual indicator labelled "Auto-cal".

Mnemonic: +FD_AUTOCAL_IND_D+

Output produced:

<u>Item</u>	<u>Type</u>	<u>Access program</u>
AUTOCAL indicator	ind_cntrl	+DI_S_AUTO_CAL_INDICATOR+

Function definition:

Event Table FD.12.1-a -- Auto-cal indicator control

MODES	EVENTS	
=====		
Lautocal	@T(In mode)	@F(In mode)
Sautocal		

All other alignment or navigation modes	X	@T(In mode)
=====		
Output value:	\$On\$	\$Off\$

FD.12.2 DEMAND FUNCTION DESCRIPTION: Control the visual indicator labelled "IMS Non-aligned".

Mnemonic: +FD_IMS_NA_IND_D+

Output produced:

<u>Item</u>	<u>Type</u>	<u>Access program</u>
IMS NA indicator signal	ind_cntrl	+DI_S_NON_ALIGN_INDICATOR+

Function definition:

Event Table FD.12.2-a -- IMS non-aligned indicator control

MODES		EVENTS		
=====				
Lautocal	@T(In mode)	@F(In mode)	X	X
=====				
Landaln	@T(In mode)	@T(!+Land vel. test passed+!)	X	X
=====				
SINSaln	@T(In mode)	@T(!+SINS vel. test passed+!)	@T(In mode AND NOT !!SINS up!!)	@T(!!SINS up!!)
=====				
Airaln	@F(!+align stage=\$CL\$) OR @T(In mode) WHEN(!+CL stage complete+!)	@T(In mode) WHEN (NOT !+CL stage complete+!)	@T(In mode AND NOT !!All IMS cks passed!!)	@T(!!All IMS cks passed!!)
=====				
01 Update	@T(!+Land vel. test failed+!)	@T(!+Land vel. test passed+!)	X	X
=====				
HUDaln	@F(!+align stage=\$HS\$) WHEN (!+IMS mode+! = \$Gndal\$)	@T(In mode) OR @F(In mode) OR @F(!+align stage+! = \$CL\$)	X	X
=====				
DI	@T(!+Nav vel. test failed+!)	X	@T(In mode AND NOT !!All IMS cks passed!!)	@T(!!All IMS cks passed!!)
DIG				
PolarDI				
=====				
Mag sl	X	@T(In mode)	X	X
=====				
I			@T(In mode AND	@T(!!All
PolarI	X	X	NOT !!All IMS	IMS cks
UDI			cks passed!!)	passed!!)
=====				
Output				!!stale
value:	\$On\$	\$Off\$	\$Intermittent\$	mode!!

Local dictionary:

<u>Term</u>	<u>Definition</u>
!!All IMS cks passed!!	(!!IMS-Dop Reasonable!! OR NOT !+Doppler up+!) AND (!!IMS-ADC Reasonable!! OR NOT !+adc tas up+!) AND !+IMS Reasonable+!
!!stale mode!!	The value of !+Non-align+! before the last call to +DI_S_NON_ALIGN_INDICATOR+(\$Intermittent\$)
!!IMS-ADC Reasonable!!	ABS(!+IMS total velocity+! - !+TAS ADC+!) lseq 191 knots.
!!IMS-Dop Reasonable!!	ABS(!+IMS horiz velocity+! - !+gnd speed DRS+!) lseq 62 knots.
!!SINS up!!	At least one validity boolean must be <u>true</u> AND the SINS velocity cannot be invalid for more than one second AND the SINS attitude data cannot be invalid for more than three seconds. If any of these conditions are violated, then the value is <u>false</u> .

FD.12.3 FUNCTION DESCRIPTION: Set the visual indicator blink rates.

Mnemonic: +FD_VISIND_RATE_D+

Output produced:

<u>Item</u>	<u>Type</u>	<u>Access program</u>
New autocal rate	real	+DI_S_AUTOCAL_BLINK_RATE+
New IMS nonalign rate	real	+DI_S_NON_ALIGN_BLINK_RATE+

Function definition:

This function is never performed, because under the current requirements, the visual indicator blink rates never need to be changed from their default values. However, should the requirements change and another blink rate be called for, this function would have the responsibility for setting the new value(s).

```

*****
*
*           FD.13           Weapon release functions
*
*****

```

FD.13.1 DEMAND FUNCTION DESCRIPTION: Prepare and release a weapon.

Mnemonic: +FD_WEAP_RLS_D+

Output produced:

<u>Item</u>	<u>Type</u>	<u>Access program</u>
To prepare a	--	+DI_PREPARE_WEAPON+
weapon for release		
To release a weapon	time	+DI_RELEASE_WEAPON+

Function definition:

Event Table FD.13.1-a -- Preparing a weapon for release

<u>MODES</u>	<u>EVENTS</u>
=====	
All weapon	@T(!+time to prepare+!)
modes	WHEN(!+RE pressed+!)
=====	
Function call:	Call +DI_PREPARE_WEAPON+

Function definition (continued):

Event Table FD.13.1-b -- Releasing the active weapon

MODES

EVENTS

NBnotShrike

NBShrike

HiNuke

LoNuke

@T(!+Computed rls+!) WHEN(!+RE pressed+!)

A/A Manrip

CCIP

Manrip

Walleye

@T(!+RE delay!!) WHEN(!+wpns rlsd+! = 0)

OR

@T(!+Computed rls+!) WHEN(!+RE pressed+!)

Output value:

!+release pulse width+!

Local dictionary:Term

!!RE delay!!

Definition

elapsed time since the last occurrence of
 @T(!+RE pressed+!) for the current weapon
 gteq !+preparation time+! for that weapon.

```

*****
*
*           FD.14           Ground test functions           *
*
*****

```

FD.14.1 DEMAND FUNCTION DESCRIPTION: Conduct the ground test.

Mnemonic: +FD_GRTEST_D+

Output produced:

<u>Item</u>	<u>Type</u>	<u>Access program</u>
Send EC to fail state	--	+EC_S_FAIL_STATE+

Function definition:

This function uses the Extended Computer diagnostic programs to decide whether to set the EC state to "failed". The table below specifies when to call each diagnostic. If any test fails, then +EC_S_FAIL_STATE+ is called, and subsequent tests are not performed.

Event Table FD.14.1-a -- Conducting the ground tests

MODE	EVENT	DIAGNOSTIC PROGRAM(S) CALLED
Grtest	@T(!+test_stage+! = \$CS\$)	+EC_START_MEMTEST+ Wait for @F(!+Memtest occurring+!) Call +EC_G_MEMTEST+
	@T(!+test_stage+! = \$TM\$)	+EC_TEST_TIMER+ +EC_TEST_INTERRUPTS+
	@T(!+test_stage+! = \$GA\$)	+EC_TEST_XACC+ +EC_TEST_YACC+ +EC_TEST_ZACC+
	@T(!+test_stage+! = \$DIO\$)	+EC_TEST_DIOW1+ +EC_TEST_DIOW2+ +EC_TEST_DIOW3+
	@T(!+test_stage+! = \$SC\$)	+EC_TEST_CSA+ +EC_TEST_CSB+
	@T(!+test_stage+! = \$DC\$)	+EC_TEST_DC+
	@T(!+test_stage+! = \$AC\$)	+EC_TEST_AC+

Appendix 1

Cross-Reference between functions of the Function Driver module and the Requirements Document

The functions of the Function Driver module are descended from the functions in Section 4 of the A-7 Requirements Document. This appendix names the Requirements function from which each Function Driver module is derived. Not all function drivers are descended from the Requirements functions; some are created in order to correctly use the device abstract interfaces.

FUNCTION DRIVER MODULE		REQUIREMENTS FUNCTION
FD.1 <u>ADC Functions</u>		
FD.1.1	Set ADC sea level pressure	4.3.3
FD.1.2	Set ADC reconfiguration values	None.
FD.2 <u>Audible Signal Functions</u>		
FD.2.1	Control audible signal	4.4.3
FD.2.2	Set audible signal beep rate	None.
FD.3 <u>Computer Fail Signal Functions</u>		
FD.3.1	Signal tactical computer failure	4.1.3
FD.4 <u>Doppler Radar Functions</u>		
FD.4.1	Start and stop Doppler radar	None.

FD.5 Flight Information Display Functions

FD.5.1	ADI functions	
FD.5.1.1	Set ADI azimuth display	4.8.1
FD.5.1.2	Set ADI elevation indicator	4.8.1
FD.5.2	HSI functions	
FD.5.2.1	Set HSI pointer 1 and DME	4.7.1
FD.5.2.2	Set HSI pointer 2	4.7.2

FD.6 Forward-Looking Radar (FLR) Functions

FD.6.1	Set FLR mode	4.2.1, 3
FD.6.2	Control FLR range and azimuth cursors	
FD.6.2.1	Position FLR cursors	4.2.2
FD.6.2.2	Set azimuth cursor display mode	4.2.2
FD.6.3	Set FLR direction	4.2.4 - 5
FD.6.4	Set FLR symbol blink rate	None.

FD.7 Head-Up Display Functions

FD.7.1	HUD Location-indicator Functions	
FD.7.1.1	Control HUD Aiming Symbol (AS)	
FD.7.1.1.1	Set HUD AS mode	4.3.1
FD.7.1.1.2	Set HUD AS position	4.3.1
FD.7.1.2	Control HUD Azimuth Steering Line (ASL)	
FD.7.1.2.1	Set HUD ASL mode	4.3.2
FD.7.1.2.2	Set HUD ASL position	4.3.2
FD.7.1.3	Control HUD Flight Director (FD)	
FD.7.1.3.1	Set HUD FD mode	4.3.4
FD.7.1.3.2	Set HUD FD position	4.3.4
FD.7.1.4	Control HUD Flight Path Marker (FPM)	
FD.7.1.4.1	Set HUD FPM mode	4.3.5
FD.7.1.4.2	Set HUD FPM position	4.3.5

FD.7.1.5	Set HUD in-range cue mode	4.3.11.0 - 3
FD.7.1.6	Control HUD Lower Solution Cue (LSC)	
FD.7.1.6.1	Set HUD LSC mode	4.3.11.0, 3
FD.7.1.6.2	Set HUD LSC position	4.3.11.1
FD.7.1.7	Control HUD Pullup Anticipation Cue (PUAC)	
FD.7.1.7.1	Set HUD PUAC mode	4.3.8
FD.7.1.7.2	Set HUD PUAC position	4.3.8
FD.7.1.8	Set HUD pullup cue mode	4.3.9
FD.7.1.9	Control HUD Upper Solution Cue (USC)	
FD.7.1.9.1	Set HUD USC mode	4.3.11.0, 3
FD.7.1.9.2	Set HUD USC position	4.3.11.2
FD.7.1.10	Set HUD symbol blink rate	None.
FD.7.2	HUD Value Indicators	
FD.7.2.1	Set HUD altitude display	4.3.3
FD.7.2.2	Set HUD heading display	4.3.6
FD.7.2.3	Set HUD pitch & roll displays	4.3.7, 10
FD.7.2.4	-- Control HUD vertical velocity/acceleration displays.	
FD.7.2.4.1	Enable vert. vel. and accel. displays	4.3.12.0
FD.7.2.4.2	Set vert. accel. display	4.3.12.1, 2
FD.7.2.4.3	Set vert. vel. display	4.3.12.1
FD.8	<u>IMS Functions</u>	
FD.8.1	Switch IMS computer control on/off	4.1.2
FD.8.2	Set IMS velocity measurement scale	4.1.5
FD.8.3	Adjust alignment of IMS platform x, y, and z axes	
FD.8.3.1	Perform small X and Y axis adjustments	4.1.4, 4.1.6
FD.8.3.2	Perform large X and Y axis adjustments	4.1.4, 4.1.6
FD.8.3.3	Adjust z axis	4.1.4, 4.1.6
FD.8.4	Initialize IMS velocities	4.6.48
FD.8.5	Set IMS reconfiguration values	None.

FD.9 Panel Functions

FD.9.1	Change panel's mark window display	4.6.3
FD.9.2	Control panel's enter light	4.6.4
FD.9.3	Display data in upper or lower window	4.6.1-54

FD.10 PMDS Functions

FD.10.1	Set map indicator	4.5.1
FD.10.2	Set map orientation angle	4.5.3
FD.10.3	Set map pointer	4.5.2
FD.10.4	Set map reference point	4.5.4
FD.10.5	Set map display	4.5.4
FD.10.6	Set map scale	4.5.4
FD.10.7	Set map reconfiguration values	None.

FD.11 SINS Functions

FD.11.1	Start and stop SINS	None.
---------	---------------------	-------

FD.12 Visual Indicator Functions

FD.12.1	Control "Auto-cal" indicator	4.1.1
FD.12.2	Control "IMS Non-aligned" indicator	4.1.8
FD.12.3	Set visual indicator blink rates	None.

FD.13 Weapon Release Functions

FD.13.1	Prepare and release a weapon	4.4.1, 4.4.2
---------	------------------------------	--------------

FD.14 Ground Test Functions

FD.14.1	Conduct ground test	None.
---------	---------------------	-------

Appendix 2

Timing Requirements

The numbers and descriptions of the FD functions appear on the left; each function's timing requirement appears on the right. On the far right is a "D" (if the function is demand) or "P" (if the function is periodic). If the function is demand then the timing requirement represents the maximum allowable delay to completion of the function. If the function is periodic then the timing requirement is the maximum length of one period. When a maximum and minimum acceptable times are known, then the frequency requirement is given as a range. In cases where two or more requirements functions have been combined, the more demanding time requirement has been adopted.

FD.1 ADC Functions

FD.1.1	Set ADC sea level pressure	Not specified	D
FD.1.2	Set ADC reconfiguration values	Not specified	D

FD.2 Audible Signal Functions

FD.2.1	Control audible signal	2 ms	D
FD.2.2	Set audible signal beep rate	Not specified	D

FD.3 Computer Fail Signal Functions

FD.3.1	Signal computer failure	Not significant	
--------	-------------------------	-----------------	--

FD.4 Doppler Radar Functions

FD.4.1 Start/stop Doppler radar

Not specified D

FD.5 Flight Information Display Functions

FD.5.1 ADI functions

FD.5.1.1 Set ADI azimuth display

67 - 80 ms P

FD.5.1.2 Set ADI elevation indicator

200 ms P

FD.5.2 HSI functions

FD.5.2.1 Set HSI pointer 1 and DME

200 ms P

FD.5.2.2 Set HSI pointer 2

200 ms P

FD.6 Forward-Looking Radar (FLR) Functions

FD.6.1 Set FLR mode

40 ms D

FD.6.2 Control FLR range and azimuth cursors

FD.6.2.1 Position cursors

40 - 80 ms P

FD.6.2.2 Set azimuth cursor display mode

40 - 80 ms D

FD.6.3 Set FLR direction

40 ms P

FD.6.4 Set FLR symbol blink rate

Not specified D

FD.7 Head-Up Display Functions

FD.7.1 HUD Location-indicator Functions

FD.7.1.1 Control HUD Aiming Symbol (AS)

FD.7.1.1.1 Set AS mode

200 ms D

FD.7.1.1.2 Set AS position

40 ms P

FD.7.1.2 Control HUD Azimuth Steering Line (ASL)

FD.7.1.2.1 Set ASL mode

200 ms D

FD.7.1.2.2 Set ASL position

40 - 80 ms P

Appendix 2

Timing Requirements

FD.7.1.3 Control HUD Flight Director (FD)		
FD.7.1.3.1	Set FD mode	Not specified D
FD.7.1.3.2	Set FD position	40 - 80 ms P
FD.7.1.4 Control HUD Flight Path Marker (FPM)		
FD.7.1.4.1	Set FPM mode	40 ms D
FD.7.1.4.2	Set FPM position	40 - 80 ms P
FD.7.1.5	Set HUD in-range cue mode	40 ms D
FD.7.1.6 Control HUD Lower Solution Cue (LSC)		
FD.7.1.6.1	Set LSC mode	40 ms D
FD.7.1.6.2	Set LSC position	40 - 200 ms P
FD.7.1.7 Control HUD Pullup Anticipation Cue (PUAC)		
FD.7.1.7.1	Set PUAC mode	Not specified D
FD.7.1.7.2	Set PUAC position	40 - 200 ms P
FD.7.1.8	Set HUD pullup cue mode	40 ms D
FD.7.1.9 Control HUD Upper Solution Cue (USC)		
FD.7.1.9.1	Set USC mode	40 ms D
FD.7.1.9.2	Set USC position	40 - 200 ms P
FD.7.1.10	Set HUD symbol blink rate	Not specified D
FD.7.2 HUD Value Indicators		
FD.7.2.1	Set Altitude display	Not specified P
FD.7.2.2	Set Heading display	40 - 200 ms P
FD.7.2.3	Set Pitch/roll displays	40 ms P
FD.7.2.4 Control HUD vertical velocity/acceleration displays.		
FD.7.2.4.1	Enable vert. vel./accel. display	1000 ms D
FD.7.2.4.2	Set HUD vertical accel. display	40 ms P
FD.7.2.4.3	Set HUD vertical velocity display	40 ms P
FD.8 <u>IMS Functions</u>		
FD.8.1	Switch computer control of IMS on/off	200 ms D
FD.8.2	Set IMS velocity msrmt. scale	200 ms D

Appendix 2

Timing Requirements

FD.8.3	Adjust alignment of IMS platform axes		
FD.8.3.1	Perform small X and Y axis adjustments	200 ms	P
FD.8.3.2	Perform large X and Y axis adjustments	200 ms	D
FD.8.3.3	Adjust Z axis	200 ms	D
FD.8.4	Initialize IMS velocities	Not specified	D
FD.8.5	Set IMS reconfiguration values	Not specified	D
FD.9	<u>Panel Functions</u>		
FD.9.1	Set Mark window display	200 ms	D
FD.9.2	Panel enter light	40 ms	D
FD.9.3	Display data in upper or lower window	300 ms *	D
	*This represents a desired limit only; the current OFP actually takes 1000 ms to change the window displays.		
FD.10	<u>PMDS Functions</u>		
FD.10.1	Set Map indicator	40 - 200 ms	P
FD.10.2	Set Map orientation angle	40 - 200 ms	P
FD.10.3	Set Map pointer	40 - 200 ms	P
FD.10.4	Set Map reference point	Not specified	D
FD.10.5	Set Map display	40 - 200 ms	P
FD.10.6	Set Map scale	40 - 200 ms	D
FD.10.7	Set Map reconfiguration values	Not specified	D
FD.11	<u>SINS Functions</u>		
FD.11.1	Start/stop SINS	Not specified	D
FD.12	<u>Visual Indicator Functions</u>		
FD.12.1	Control Auto-cal indicator	200 ms	D
FD.12.2	Control IMS Non-aligned indicator	200 ms	D
FD.12.3	Set visual indicator blink rates	Not specified	D
FD.13	<u>Weapon Release Functions</u>		
FD.13.1	Prepare and release a weapon	2 ms	D
FD.14	<u>Ground Test Functions</u>		
FD.14.1	Conduct ground test	Not specified	D

Appendix 3

List of Required Events (Ordered Alphabetically)

This appendix lists those events used by the Function Driver module. The events are listed alphabetically by the module that signals them. Following each event is a list of the function drivers that use it.

Events Signalled by the Extended Computer Module

@T(!+Failed state+!)	FD.3.1
@T(!+Init complete+!)	FD.1.1 FD.3.1 FD.7.1.3.1 FD.9.1 FD.9.2 FD.10.4 FD.10.6 FD.11.1

Events Signalled by the Device Interface Module

@F(!+ADI elev avail+!)	FD.5.1.2
@F(!+AOA valid+!)	FD.7.2.4.1
@T(!+AOA valid+!)	FD.7.2.4.1
@T(!+Enter pressed+!)	FD.9.2
@T(!+Fly to num changed+!)	FD.1.1 FD.9.2
@T(!+Fly to state changed+!)	FD.9.2
@T(!+Map hold changed+!)	FD.9.2
@T(!+Mark pressed+!)	FD.9.1
@T(!+Panel mode changed+!)	FD.9.2
@T(!+Pres pos changed+!)	FD.9.2
@T(!+RE pressed+!)	FD.2.1 FD.7.1.1.1 FD.13.1
@F(!+RE pressed+!)	FD.2.1
@F(!+Slew displacement non-zero+!)	FD.9.3
@T(!+TD pressed+!)	FD.8.3.3 FD.9.2 FD.9.3
@T(!+Update changed+!)	FD.9.2

Mode Transition Events signalled by SS.MODE

@T(*OlUpdate*)	FD.8.2 FD.12.1 FD.5.2.1 FD.8.1
@T(*A/A Guns*)	FD.7.1.1.1
@F(*A/A Guns*)	FD.7.1.1.1
@T(*A/A Manrip*)	FD.7.1.1.1 FD.9.3
@F(*A/A Manrip*)	FD.7.1.1.1
@T(*A/G Guns*)	FD.6.1 FD.7.1.1.1 FD.7.1.7.1
@F(*A/G Guns*)	FD.6.1 FD.7.1.1.1 FD.7.1.5 FD.7.1.6.1
@T(*Airaln*)	FD.12.1 FD.12.2 FD.8.2
@F(*Airaln*)	FD.7.2.4.1
@T(*BOC*)	FD.6.1 FD.6.2.2 FD.7.1.1.1 FD.7.1.2.1 FD.7.1.7.1
	FD.9.3
@F(*BOC*)	FD.6.1 FD.7.1.1.1 FD.7.1.2.1
@T(*BOCFlyto0*)	FD.6.1 FD.7.1.1.1 FD.7.1.2.1 FD.7.1.7.1
@F(*BOCFlyto0*)	FD.6.1 FD.7.1.1.1 FD.7.1.2.1
@T(*BOCOffset*)	FD.6.1 FD.6.2.2 FD.7.1.1.1 FD.7.1.2.1 FD.7.1.7.1
	FD.9.3
@F(*BOCOffset*)	FD.6.1 FD.7.1.1.1 FD.7.1.2.1
@T(*CCIP*)	FD.6.1 FD.7.1.2.1 FD.7.1.7.1 FD.9.3
@F(*CCIP*)	FD.6.1 FD.7.1.2.1 FD.7.1.6.1
@T(*DI*)	FD.12.1 FD.8.1 FD.8.3.3
@F(*DI*)	FD.8.3.1 FD.8.3.3
@T(*DIG*)	FD.12.1 FD.8.1 FD.8.2 FD.8.3.3 FD.12.1
@F(*DIG*)	FD.8.3.3
@T(*Grid*)	FD.8.1
@F(*Grid*)	FD.7.2.4.1
@T(*Grtest*)	FD.7.1.1.1 FD.9.1
@T(*HUDaln*)	FD.12.1 FD.12.2 FD.5.2.1 FD.7.1.1.1 FD.8.1 FD.8.2
@F(*HUDaln*)	FD.12.2 FD.7.1.1.1
@T(*HUDdown1*)	FD.6.1 FD.7.1.1.1 FD.7.1.2.1
@F(*HUDdown1*)	FD.6.1 FD.7.1.1.1 FD.7.1.2.1 FD.7.1.6.1 FD.7.1.9.1
@T(*HUDdown2*)	FD.6.1 FD.7.1.1.1 FD.7.1.2.1
@F(*HUDdown2*)	FD.6.1 FD.7.1.1.1 FD.7.1.2.1 FD.7.1.6.1 FD.7.1.9.1
@T(*HUDUpd*)	FD.6.1 FD.7.1.1.1
@F(*HUDUpd*)	FD.6.1 FD.7.1.1.1
@T(*I*)	FD.12.1 FD.12.2 FD.8.1 FD.8.2
@T(*IMS fail*)	FD.5.2.1 FD.8.1 FD.9.3 FD.12.1
@F(*IMS fail*)	FD.7.2.4.1
@T(*Landaln*)	FD.9.3
@F(*Landaln*)	FD.12.1 FD.12.2 FD.5.2.1 FD.8.1 FD.8.2
@T(*Lautocal*)	FD.12.1 FD.12.2 FD.5.2.1 FD.8.1 FD.8.2
@F(*Lautocal*)	FD.9.3 FD.12.1 FD.12.2
@T(*Mag sl*)	FD.12.1 FD.12.2 FD.8.1
@F(*Mag Sl*)	FD.7.2.4.1
@T(*Manrip*)	FD.9.3
@T(*Nattack*)	FD.6.1 FD.7.1.1.1 FD.7.1.2.1 FD.7.1.7.1
@F(*Nattack*)	FD.6.1 FD.7.1.1.1 FD.7.1.2.1 FD.7.1.6.1
@F(**NBnotShrike**)	FD.7.1.6.1 FD.7.1.9.1

@T(**NBShrike**)	FD.5.2.1
@F(**NBShrike**)	FD.7.1.6.1
@T(*Noffset*)	FD.6.1 FD.7.1.1.1 FD.7.1.2.1 FD.7.1.7.1
@F(*Noffset*)	FD.6.1 FD.7.1.1.1 FD.7.1.2.1 FD.7.1.6.1
@T(*OLB*)	FD.12.1 FD.8.1 FD.8.2
@F(*OLB*)	FD.7.2.4.1
@T(*PolarDi*)	FD.12.1 FD.8.1 FD.8.2
@T(*PolarI*)	FD.12.1 FD.12.2 FD.8.1 FD.8.2
@T(*RadarJpd*)	FD.6.1 FD.6.2.2 FD.7.1.1.1
@F(*RadarUpd*)	FD.6.1 FD.7.1.1.1
@T(*Sautocal*)	FD.12.1 FD.12.2 FD.5.2.1 FD.8.1 FD.8.2
@F(*Sautocal*)	FD.9.3 FD.12.1 FD.12.2
@T(*SBOC*)	FD.6.1 FD.6.2.2 FD.7.1.1.1 FD.7.1.2.1 FD.7.1.7.1 FD.9.3
@F(*SBOC*)	FD.6.1 FD.7.1.1.1 FD.7.1.2.1 FD.7.1.6.1 FD.7.1.9.1 FD.7.2.4.3
@T(*SBOCFlyto0*)	FD.6.1 FD.7.1.1.1 FD.7.1.2.1 FD.7.1.7.1
@F(*SBOCFlyto0*)	FD.6.1 FD.7.1.1.1 FD.7.1.2.1 FD.7.1.6.1 FD.7.1.9.1 FD.7.2.4.3
@T(*SBOCOffset*)	FD.6.1 FD.6.2.2 FD.7.1.1.1 FD.7.1.2.1 FD.7.1.7.1 FD.9.3
@F(*SBOCOffset*)	FD.6.1 FD.7.1.1.1 FD.7.1.2.1 FD.7.1.6.1 FD.7.1.9.1 FD.7.2.4.3
@T(*SHUDDown1*)	FD.7.1.1.1 FD.7.1.2.1 FD.6.1
@F(*SHUDDown1*)	FD.7.1.1.1 FD.7.1.2.1 FD.7.1.6.1 FD.7.1.9.1 FD.6.1
@T(*SHUDDown2*)	FD.6.1 FD.7.1.1.1 FD.7.1.2.1
@F(*SHUDDown2*)	FD.6.1 FD.7.1.1.1 FD.7.1.2.1 FD.7.1.6.1 FD.7.1.9.1
@T(*SINSaln*)	FD.12.2 FD.5.2.2 FD.8.1 FD.8.2 FD.8.3.3
@F(*SINSaln*)	FD.9.3 FD.12.2
@T(*Snattack*)	FD.6.1 FD.7.1.1.1 FD.7.1.2.1 FD.7.1.7.1
@F(*Snattack*)	FD.6.1 FD.7.1.1.1 FD.7.1.2.1 FD.7.1.6.1 FD.7.1.9.1 FD.7.2.4.3
@T(*Snoffset*)	FD.6.1 FD.7.1.1.1 FD.7.1.2.1 FD.7.1.7.1
@F(*Snoffset*)	FD.6.1 FD.7.1.1.1 FD.7.1.2.1 FD.7.1.6.1 FD.7.1.9.1 FD.7.2.4.3
@T(*UDI*)	FD.12.1 FD.12.2 FD.8.1 FD.8.2
@T(*Walleye*)	FD.7.1.1.1 FD.7.1.7.1
@F(*Walleye*)	FD.7.1.1.1 FD.7.1.5 FD.7.1.6.1

Events Signalled by SS.EVENT

@F(ABS(!+roll IMS+!) gt 5 deg)	FD.8.1
@T(ABS(!+roll IMS+!) gt 5 deg)	FD.8.1
@F(ABS(!+LSC elevation+!) gt 4 deg)	FD.5.1.2
@T(ABS(!+LSC elevation+!) gt 4 deg)	FD.5.1.2
@F(ABS(!+PUAC elevation+!) gt 4 deg)	FD.5.1.2
@T(ABS(!+PUAC elevation+!) gt 4 deg)	FD.5.1.2
@T(!+AS mode+! = \$Off\$)	FD.7.1.1.2
@T(!+AS mode+! = \$On\$)	FD.7.1.1.2
@T(!+ASL mode+! = \$Intermittent\$)	FD.7.1.2.2
@T(!+ASL mode+! = \$Off\$)	FD.7.1.2.2
@T(!+ASL mode+! = \$On\$)	FD.7.1.2.2
@T(!+FLR az cursor mode+! = \$On\$)	FD.6.2.1
@T(!+FLR az cursor mode+! = \$Off\$)	FD.6.2.1
@T(!+FLR mode+! = \$CDCE\$)	FD.6.2.1 FD.6.2.2
@F(!+FLR mode+! = \$CDCE\$)	FD.6.2.1 FD.6.2.2
@F(!+FLR mode+! = \$Ranging\$)	FD.6.3
@T(!+FLR mode+! = \$Ranging\$)	FD.6.3
@T(!+FLTDIR mode+! = \$Off\$)	FD.7.1.3.2
@T(!+FLTDIR mode+! = \$On\$)	FD.7.1.3.2
@F(!+Fly to num+! = 0)	FD.1.1
@T(!+Fly to num+! = 0)	FD.1.1
@T(!+Fly_to_state+! = \$Dest\$)	FD.1.1
@T(!+Fly_to_state+! = \$Mark\$)	FD.1.1
@T(!+FPM mode+! = \$Off\$)	FD.7.1.4.2
@T(!+FPM mode+! = \$On\$)	FD.7.1.4.2
@T(!+Gun Enable+!)	FD.6.1 FD.7.1.5
@F(!+Gun Enable+!)	FD.7.1.5
@T(!+LSC mode+! = \$Intermittent\$)	FD.7.1.6.2 FD.7.1.9.1
@F(!+LSC mode+! = \$Intermittent\$)	FD.7.1.9.1
@T(!+LSC mode+! = \$Off\$)	FD.7.1.6.2
@T(!+LSC mode+! = \$On\$)	FD.7.1.6.2
@F(!+Map decenter+!)	FD.10.4
@T(!+Map decenter+!)	FD.10.4
@T(!+Map hold+!)	FD.10.2
@F(!+Map north-up+!)	FD.10.4
@T(!+Map north-up+!)	FD.10.4
@F(!+Map scale sw+!)	FD.10.6
@T(!+Map scale sw+!)	FD.10.6
@F(!+Master Arm+!)	FD.7.1.7.1
@T(!+Master Arm+!)	FD.7.1.7.1

Appendix 3

Event List (Alphabetical)

@T(!+pitch IMS+! = -30 deg)	FD.7.1.7.1
@T(!+pitch IMS+! gt 15 deg)	FD.2.1
@T(!+PUAC mode+! = \$Intermittent\$)	FD.7.1.7.2
@T(!+PUAC mode+! = \$Off\$)	FD.7.1.7.2
@T(!+PUAC mode+! = \$On\$)	FD.7.1.7.2
@T(!+Rel in Progress+!)	FD.2.1
@T(!+USC mode+! = \$Intermittent\$)	FD.7.1.9.2
@T(!+USC mode+! = \$Off\$)	FD.7.1.9.2
@T(!+USC mode+! = \$On\$)	FD.7.1.9.2
@T(!+VV mode+! = \$On\$)	FD.7.1.4.1 FD.7.2.4.2 FD.7.2.4.3
@F(!+VV mode+! = \$On\$)	FD.7.2.4.2 FD.7.2.4.3
@T(!+Weapon Class+! = \$GN\$)	FD.7.1.7.1
@T(!+Weapon Class+! = \$RK\$)	FD.6.1
@T(!+Weapon Mode+! = \$None\$)	FD.7.1.3.1 FD.7.1.7.1
@F(!+Weapon Mode+! = \$None\$)	FD.7.1.3.1 FD.7.1.7.1

Events Signalled by other Shared Services submodules

@F(ABS(!+ip az+!) lseq 12 deg)	FD.7.1.6.1
@T(ABS(!+ip az+!) lseq 12 deg)	FD.7.1.6.1
@F(ABS(!+ip elev+!) lseq 16 deg)	FD.7.1.6.1
@T(ABS(!+ip elev+!) lseq 16 deg)	FD.7.1.6.1
@T(1 second before !+target in range+!)	FD.7.1.6.1
@F(!+ADC reasonable+!)	FD.9.1
@T(!+ADC reasonable+!)	FD.9.1
@T(!+adc tas up+!)	FD.12.2
@F(!+adc tas up+!)	FD.12.2
@T(!+Air velocity test passed+!)	FD.12.2
@F(!+align_stage+! = \$CA\$)	FD.5.2.2
@T(!+align_stage+! = \$CA2\$)	FD.8.3.3
@F(!+align_stage+! = \$CL\$)	FD.8.1 FD.12.2
@F(!+align_stage+! = \$ED\$)	FD.8.3.3
@T(!+align_stage+! = \$ED\$)	FD.8.3.3
@F(!+align_stage+! = \$ED2\$)	FD.8.3.3
@T(!+align_stage+! = \$ED2\$)	FD.8.3.3
@F(!+align_stage+! = \$FM\$)	FD.7.2.4.1 FD.8.1
@T(!+align_stage+! = \$FM\$)	FD.8.1
@F(!+align_stage+! = \$HS\$)	FD.12.2
@T(!+align_stage+! = \$HS\$)	FD.12.2
@T(!+align_stage+! = \$ND2\$)	FD.8.3.3
@F(!+align_stage+! = \$TS\$)	FD.9.3
@T(!+ap ahead+!)	FD.6.2.2
@F(!+ap ahead+!)	FD.6.2.2
@T(!+blast danger+!)	FD.7.1.8
@F(!+blast danger+!)	FD.7.1.8
@T(!+Computed rls+!)	FD.13.1
@T(!+cup ahead+!)	FD.6.2.2
@F(!+cup ahead+!)	FD.6.2.2
@T(!+data enterable+!)	FD.9.2
@F(!+data enterable+!)	FD.9.2
@T(!+desig+!)	FD.5.2.1 FD.6.2.2 FD.7.1.5
	FD.7.1.1.2 FD.7.1.6.1 FD.7.1.9.1
	FD.8.3.3 FD.9.3
@F(!+desig+!)	FD.7.1.6.1 FD.7.1.5 FD.5.2.1
	FD.7.1.9.1 FD.9.3

Appendix 3

Event List (Alphabetical)

@T(!+Dest selected+!)	FD.9.3
@T(!+Doppler up+!)	FD.12.2
@F(!+Doppler up+!)	FD.12.2
@T(!+Drift test failed+!)	FD.12.2
@T(!+Drift test passed+!)	FD.12.2
@T(!+during slewing+!)	FD.7.1.6.1 FD.7.1.9.1
@F(!+during slewing+!)	FD.7.1.6.1 FD.7.1.9.1
@T(!+ftpt ahead+!)	FD.6.2.2
@F(!+ftpt ahead+!)	FD.6.2.2
@T(!+CAS+!)	FD.7.1.6.1 FD.7.1.9.1
@F(!+CAS+!)	FD.7.1.9.1 FD.7.1.6.1
@T(!+gr_ac_ftpt+! lseq 20 nmi)	FD.6.1
@F(!+gr_ac_ftpt+! gteq 10 nmi)	FD.5.2.1 FD.7.1.7.1
@T(!+gr_ac_ftpt+! gteq 10 nmi)	FD.5.2.1
@F(!+gr_ac_ftpt+! gteq 1000 nmi)	FD.5.2.1
@T(!+gr_ac_ftpt+! gteq 1000 nmi)	FD.5.2.1
@F(!+gr_ac_ftpt+! lseq 30 nmi)	FD.7.1.7.1
@T(!+gr_ac_ftpt+! lseq 30 nmi)	FD.7.1.7.1
@T(!+gr_ac_fxpt+! lseq 22 nmi)	FD.6.1
@F(!+gr_ac_HUDrefpt+! lseq 20 nmi)	FD.7.1.1.1
@T(!+gr_ac_HUDrefpt+! lseq 20 nmi)	FD.7.1.1.1
@F(!+gr_ac_HUDrefpt+! lseq 22 nmi)	FD.7.1.1.1
@T(!+gr_ac_HUDrefpt+! lseq 22 nmi)	FD.7.1.1.1
@F(!+gr_ac_HUDrefpt+! lseq 30 nmi)	FD.7.1.1.1
@T(!+gr_ac_HUDrefpt+! lseq 30 nmi)	FD.7.1.1.1
@F(!+gr_ac_HUDrefpt+! lseq 42 nmi)	FD.7.1.1.1
@T(!+gr_ac_HUDrefpt+! lseq 42 nmi)	FD.7.1.1.1
@F(!+gr_ac_oap+! gteq 10 nmi)	FD.5.2.1
@T(!+gr_ac_oap+! gteq 10 nmi)	FD.5.2.1
@F(!+gr_ac_oap+! gteq 1000 nmi)	FD.5.2.1
@T(!+gr_ac_oap+! gteq 1000 nmi)	FD.5.2.1
@T(!+gr_ac_oap+! lseq 30 nmi)	FD.9.3
@F(!+gr_ac_tgt+! gteq 10 nmi)	FD.5.2.1
@T(!+gr_ac_tgt+! gteq 10 nmi)	FD.5.2.1
@F(!+gr_ac_tgt+! gteq 1000 nmi)	FD.5.2.1
@T(!+gr_ac_tgt+! gteq 1000 nmi)	FD.5.2.1
@T(!+gr_ac_tgt+! lseq 30 nmi)	FD.9.3
@F(!+gr_ac_tgt+! lseq 30 nmi)	FD.9.3
@T(!+ground danger+!)	FD.7.1.8
@F(!+ground danger+!)	FD.7.1.8
@T(!+high drag release+!)	FD.7.1.7.1
@F(!+IMS Reasonable+!)	FD.12.2
@T(!+IMS Reasonable+!)	FD.8.1 FD.12.2

@T(!+ims_x_const_error_mc+! gt !+IMS adj xy tolerance+!)	FD.8.3.2
@T(!+ims_x_dop_error_mc+! gt !+IMS adj xy tolerance+!)	FD.8.3.2
@T(!+ims_x_sins_error_mc+! gt !+IMS adj xy tolerance+!)	FD.8.3.2
@T(!+ims_y_const_error_mc+! gt !+IMS adj xy tolerance+!)	FD.8.3.2
@T(!+ims_y_dop_error_mc+! gt !+IMS adj xy tolerance+!)	FD.8.3.2
@T(!+ims_y_sins_error_mc+! gt !+IMS adj xy tolerance+!)	FD.8.3.2
@T(!+ims_z_const_error_mc+! gt !+ims z adj tolerance+!)	FD.8.3.3
@T(!+ims_z_dop_error_m+! gt !+ims z adj tolerance+!)	FD.8.3.3
@T(!+ims_z_dop_error_mc+! gt !+ims z adj tolerance+!)	FD.8.3.3
@T(!+ims_z_nav_error_m+! gt !+ims z adj tolerance+!)	FD.8.3.3
@T(!+ims_z_sins_error_mc+! gt !+ims z adj tolerance+!)	FD.8.3.3
@T(!+In flight+!)	FD.11.1 FD.4.1 FD.5.2.2
	FD.7.2.4.1
@F(!+In flight+!)	FD.11.1 FD.4.1 FD.5.2.2
	FD.7.2.4.1
@T(!+Input attempted+!)	FD.9.2
@T(!+Input requested+!)	FD.9.2
@T(!+ip elev+! lseq 16 deg)	FD.9.3
@T(!+Land based+!)	FD.11.1
@F(!+Land based+!)	FD.11.1
@T(!+Land velocity test failed+!)	FD.12.2
@T(!+Land velocity test passed+!)	FD.12.2
@T(!+low drag release+!)	FD.7.1.6.1 FD.7.2.4.2
	FD.7.2.4.3
@F(!+low drag release+!)	FD.7.1.6.1 FD.7.1.9.1
	FD.7.2.4.2 FD.7.2.4.3
@T(!+Nav velocity test failed+!)	FD.12.2
@T(!+new align stage+!)	FD.9.3
@T(!+New central long a pnl entered+!)	FD.10.7
@T(!+New central long b pnl entered+!)	FD.10.7
@T(!+New dest mslp pnl entered+!)	FD.1.1
@T(!+New E coarse bias pnl entered+!)	FD.8.5
@T(!+New E coarse scale pnl entered+!)	FD.8.5
@T(!+New E fine bias pnl entered+!)	FD.8.5
@T(!+New E fine scale pnl entered+!)	FD.8.5
@T(!+New L-probe pnl entered+!)	FD.1.2
@T(!+New low lat ct a pnl entered+!)	FD.10.7
@T(!+New low lat ct b pnl entered+!)	FD.10.7
@T(!+New map orient a pnl entered+!)	FD.10.7
@T(!+New map orient b pnl entered+!)	FD.10.7
@T(!+New N coarse bias pnl entered+!)	FD.8.5
@T(!+New N coarse scale pnl entered+!)	FD.8.5
@T(!+New N fine bias pnl entered+!)	FD.8.5
@T(!+New N fine scale pnl entered+!)	FD.8.5
@T(!+New V coarse bias pnl entered+!)	FD.8.5
@T(!+New V coarse scale pnl entered+!)	FD.8.5
@T(!+New X corr increm pnl entered+!)	FD.8.5
@T(!+New X drift pnl entered+!)	FD.8.5

Appendix 3

Event List (Alphabetical)

@T(!+New Y corr increm pnl entered+!)	FD.8.5
@T(!+New Y drift pnl entered+!)	FD.8.5
@T(!+New Z corr increm pnl entered+!)	FD.8.5
@T(!+New Z drift pnl entered+!)	FD.8.5
@T(!+oap ahead+!)	FD.6.2.2
@F(!+oap ahead+!)	FD.6.2.2
@T(!+OTS+!)	FD.7.1.6.1
@T(!+pnl config+! = ANY POSSIBLE VALUE)	FD.9.3
@T(!+pnl config+! = \$Nav diags2\$)	FD.9.1
@T(!+R65+!)	FD.2.1
@T(!+rls_pts_passed+! = 1)	FD.9.3
@T(!+Rmax+!)	FD.2.1 FD.7.1.6.1
@T(!+Rmax+6000+!)	FD.7.1.7.1
@T(!+Rmin+!)	FD.2.1
@T(!+Rmin+6000+!)	FD.2.1 FD.7.1.7.1
@T(!+SINS attitude valid+!)	FD.12.2
@F(!+SINS attitude valid+!)	FD.12.2
@T(!+SINS velocity valid+!)	FD.12.2
@F(!+SINS velocity valid+!)	FD.12.2
@T(!+SINS velocity test passed+!)	FD.12.2
@T(!+Special in range+!)	FD.7.1.6.1 FD.7.1.9.1
@F(!+Special in range+!)	FD.7.1.6.1 FD.7.1.9.1
@T(!+Special solution+!)	FD.13.1
@T(!+sr reasonable+!)	FD.7.1.5 FD.7.1.6.1
@F(!+sr reasonable+!)	FD.7.1.6.1 FD.7.1.5
@T(!+stik created+!)	FD.7.1.7.1
@T(!+stik empty+!)	FD.7.1.4.1 FD.7.1.7.1
@T(!+target in range+!)	FD.7.1.5 FD.7.1.6.1
@F(!+target in range+!)	FD.7.1.5 FD.7.1.6.1
@T(!+test_stage+! = \$AC\$)	FD.14.1
@T(!+test_stage+! = \$CS\$)	FD.14.1
@T(!+test_stage+! = \$DC\$)	FD.14.1
@T(!+test_stage+! = \$DIO\$)	FD.14.1
@T(!+test_stage+! = \$GA\$)	FD.14.1
@T(!+test_stage+! = \$SC\$)	FD.7.1.1.1 FD.14.1
@T(!+test_stage+! = \$TM\$)	FD.14.1
@T(!+tgt ahead+!)	FD.6.2.2 FD.7.1.5 FD.7.1.6.1
@F(!+tgt ahead+!)	FD.6.2.2 FD.7.1.5 FD.7.1.6.1
@T(!+time to prepare+!)	FD.2.1 FD.13.1

Appendix 4

Dictionary of Input Items

For each input item required by a function driver module, there is an entry in this dictionary specifying the software submodule that produces that value, and all function driver modules in which it is used.

A '*' means that the specifications for the producing module are not yet complete.

Appendix 4

Input Source Reference

<u>TERM</u>	<u>PRODUCING MODULE</u>	<u>WHERE USED</u>
!+ACAIRB+!	SS.DIAGIO	FD.9.3
!+adc alt up+!	SS.SYSVAL	FD.7.1.4.2 FD.7.2.1
!+adc tas up+!	SS.SYSVAL	FD.7.1.4.2
!+ADCFAIL+!	SS.DIAGIO	FD.9.3
!+after_slewing+!	SS.SYSVAL	FD.5.2.1 FD.6.2.1 FD.7.1.1.2
!+align_mode+!	SS.MODE	FD.1.1 FD.11.1 FD.12.1 FD.12.2 FD.4.1 FD.5.1.1 FD.5.1.2 FD.5.2.1 FD.5.2.2 FD.7.1.1.1 FD.7.1.1.2 FD.7.1.3.1 FD.7.1.3.2 FD.7.1.4.1 FD.7.1.4.2 FD.7.2.1 FD.7.2.2 FD.7.2.3 FD.7.2.4 FD.7.2.4.3 FD.8.1 FD.8.2 FD.8.3.1 FD.8.3.2 FD.8.3.3 FD.9.3
!+align_stage+!	SS.STAGE	FD.9.3 FD.8.1 FD.8.3.1 FD.8.3.2 FD.8.3.3
!+alt ADC+!	DI.ADC	FD.7.2.1 FD.9.3
!+alt from sr+!	PM*	FD.9.3
!+alt priority stale+!	SS.SYSVAL	FD.9.3
!+alt priority source+!	SS.SYSVAL	FD.9.3
!+alt RADAR+!	DI.RADALT	FD.9.3
!+AOA+!	DI.AOA	FD.7.1.4.2 FD.7.2.3
!+ARPPAIRS+!	SS.DIAGIO	FD.9.3
!+ARPINT+!	SS.DIAGIO	FD.9.3
!+ARPQUANT+!	SS.DIAGIO	FD.9.3
!+AS azimuth+!	DI.HUD	FD.6.3 FD.7.1.2.2 FD.8.3.3
!+AS elevation+!	DI.HUD	FD.6.3
!+ASL elevation+!	DI.HUD	7.1.6.2
!+Az cursor lft max+!	DI.FLR	FD.6.2.1
!+Az cursor rgt max+!	DI.FLR	FD.6.2.1
!+az miss dist+!	PM*	FD.9.3

Appendix 4

Input Source Reference

!+az ref hdg pnl+!	SS.SYSVAL	FD.8.3.3 FD.9.3
!+before_slewing+!	SS.SYSVAL	FD.5.2.1 FD.6.2.1 FD.7.1.1.2
!+BMBDRAG eq High+!	SS.DIAGIO	FD.9.3
!+bomb fall line+!	PM*	FD.7.1.2.2
!+boresight azimuth+!	DI.WCM	FD.7.1.1.2
!+boresight elevation+!	DI.WCM	FD.7.1.1.2
!+brg_ac_ftpt+!	SS.SYSVAL	FD.7.1.3.2
!+brg_grtk_ap+!	SS.SYSVAL	FD.6.2.1
!+brg_grtk_cup+!	SS.SYSVAL	FD.6.2.1
!+brg_grtk_ftpt+!	SS.SYSVAL	FD.10.3 FD.5.1.1 FD.5.2.1 FD.6.2.1
!+brg_grtk_oap+!	SS.SYSVAL	FD.5.2.1 FD.6.2.1
!+brg_grtk_tgt+!	SS.SYSVAL	FD.5.2.1 FD.6.2.1 FD.7.1.2.2
!+burst ht pnl+!	SS.PNL.INPUT	FD.9.3
!+central long+!	DI.PMDSR	FD.9.3
!+central long a pnl+!	SS.PNL.INPUT	FD.10.7
!+central long b pnl+!	SS.PNL.INPUT	FD.10.7
!+CL stage complete+!	SS.STAGE	FD.12.2
!+data nbr pnl+!	SS.PNL.INPUT	FD.9.3
!+desig+!	SS.SYSVAL	FD.10.5 FD.2.1 FD.5.1.1 FD.5.2.1 FD.6.2.1 FD.7.1.1.2 FD.7.1.2.2 FD.7.1.6.1
!+dest altitude pnl+!	SS.PNL.INPUT	FD.9.3
!+dest entry pnl+!	SS.PNL.INPUT	FD.1.1 FD.9.3 FD.10.5
!+dest lat+!	SS.SYSVAL	FD.9.3
!+dest long+!	SS.SYSVAL	FD.9.3

Appendix 4

Input Source Reference

!+dest mslp pnl+!	SS.PNL.INPUT	FD.1.1 FD.9.3
!+dive_pullup+!	PM*	FD.7.1.6.2
!+Doppler coupled pnl+!	SS.PNL.INPUT	FD.9.3
!+Doppler reasonable+!	SS.SYSVAL	FD.9.3
!+drift angle+!	SS.SYSVAL	FD.7.1.2.2
!+drift angle DRS+!	DI.DRS	FD.9.3.
!+drift angle IMS+!	*	FD.9.3
!+DRSREL+!	SS.DIAGIO	FD.9.3
!+during slewing+!	SS.SYSVAL	FD.6.2.1 FD.7.1.1.2 FD.10.5
!+E coarse bias+!	DI.IMSR	FD.9.3
!+E coarse bias pnl+!	SS.PNL.INPUT	FD.8.5
!+E coarse scale+!	DI.IMSR	FD.9.3
!+E coarse scale pnl+!	SS.PNL.INPUT	FD.8.5
!+E fine bias+!	DI.IMSR	FD.9.3
!+E fine bias pnl+!	SS.PNL.INPUT	FD.8.5
!+E fine scale+!	DI.IMSR	FD.9.3
!+E fine scale pnl+!	SS.PNL.INPUT	FD.8.5
!+E vel IMS+!	DI.IMS	FD.9.3
!+elapsed navaln time+!	SS.SYSVAL	FD.9.3
!+Enter light+!	DI.PNL	FD.9.2
!+flt path angle+!	PM.ACM	FD.9.3
!+Fly to num+!	DI.SWB	FD.1.1 FD.5.1.1 FD.5.2.1 FD.7.1.3.2 FD.9.3 FD.10.3
!+Fly to state+!	DI.SWB	FD.1.1
!+FM stage complete+!	SS.STAGE	FD.7.1.4.2 FD.7.2.4

Appendix 4

Input Source Reference

!+FPM azimuth+!	DI.HUD	FD.7.1.2.2 FD.7.1.7.2
!+FPM elevation+!	DI.HUD	FD.7.1.1.2 FD.7.1.2.2 FD.7.1.6.2 FD.7.1.7.2
!+GAS+!	SS.SYSVAL	FD.7.1.2.2
!+gnd spd DRS+!	DI.DRS	FD.9.3 FD.12.2
!+gr_ac_ftpt+!	SS.SYSVAL	FD.5.2.1
!+gr_ac_HUDrefpt+!	SS.SYSVAL	FD.7.1.1.2 FD.7.1.2.2
!+gr_ac_oap+!	SS.SYSVAL	FD.5.2.1 FD.9.3
!+gr_ac_rmax+!	*	FD.7.1.6.2
!+gr_ac_tgt+!	SS.SYSVAL	FD.5.2.1 FD.7.1.2.2 FD.7.1.6.2 FD.9.3
!+grtk+!	PM*	FD.10.1 FD.10.2 FD.5.2.2
!+Gun Enable+!	DI.WRM	FD.7.1.9.2 FD.7.1.5
!+GUNSSSEL+!	SS.DIAGIO	FD.9.3
!+heading IMS+!	DI.IMS	FD.8.3.3 FD.9.3 FD.10.1
!+heading MAG+!	DI.IMS	FD.7.2.2 FD.9.3 FD.10.1
!+HUDrefpt_az+!	SS.SYSVAL	FD.7.1.1.2
!+HUDrefpt_elev+!	SS.SYSVAL	FD.7.1.1.2
!+HUDREL+!	SS.DIAGIO	FD.9.3
!+IMS horiz velocity+!	*	FD.9.3 FD.12.2
!+IMS mode+!	DI.IMS	FD.7.2.2 FD.8.2 FD.12.2
!+IMS ready+!	DI.IMS	FD.9.3
!+IMS reasonable+!	SS.SYSVAL	FD.9.3
!+IMS rel+!	DI.IMS	FD.9.3

Appendix 4

Input Source Reference

!+IMS rotating+!	DI.IMS	FD.8.3.3
!+IMS total velocity+!	*	FD.9.3 FD.12.2
!+ims_x_const_error_mc+!	*	FD.8.3.1 FD.8.3.2
!+ims_x_dop_error_mc+!	*	FD.8.3.1 FD.8.3.2
!+ims_x_nav_error_m+!	*	FD.8.3.1 FD.8.3.2
!+ims_x_nav_error_mc+!	*	FD.8.3.1 FD.8.3.2
!+ims_x_sins_error_mc+!	*	FD.8.3.1 FD.8.3.2
!+ims_y_const_error_mc+!	*	FD.8.3.1 FD.8.3.2
!+ims_y_dop_error_mc+!	*	FD.8.3.1 FD.8.3.2
!+ims_y_nav_error_mc+!	*	FD.8.3.1 FD.8.3.2
!+ims_y_sins_error_mc+!	*	FD.8.3.1 FD.8.3.2
!+ims_z_const_error_mc+!	*	FD.8.3.3
!+ims_z_dop_error_m+!	*	FD.8.3.3
!+ims_z_dop_error_mc+!	*	FD.8.3.3
!+ims_z_nav_error_m+!	*	FD.8.3.3
!+ims_z_sins_error_mc+!	*	FD.8.3.3
!+IMSAUTOC+!	SS.DIAGIO	FD.9.3
!+IMSMODE eq Gndal+!	SS.DIAGIO	FD.9.3
!+IMSMODE eq Grid+!	SS.DIAGIO	FD.9.3
!+IMSMODE+! eq Iner+!	SS.DIAGIO	FD.9.3
!+IMSMODE eq Magsl+!	SS.DIAGIO	FD.9.3
!+IMSMODE eq Norm+!	SS.DIAGIO	FD.9.3
!+IMSREDY+!	SS.DIAGIO	FD.9.3
!+IMSREL+!	SS.DIAGIO	FD.9.3
!+in_flight+!	SS.SYSVAL	FD.5.1.1 FD.8.1 FD.7.1.4.2 FD.11.1

Appendix 4

Input Source Reference

!+ip_elev+!	*	FD.7.1.2.2 FD.7.1.6.2
!+L-probe+!	DI.ADCR	FD.9.3
!+L-probe pnl+!	SS.PNL.INPUT	FD.1.2
!+Land based pnl+!	SS.PNL.INPUT	FD.9.3 FD.11.1
!+latitude+!	SS.SYSVAL	FD.8.3.3 FD.9.3 FD.10.5
!+latitude_cup+!	SS.SYSVAL	FD.10.5
!+Latitude error+!	SS.SYSVAL	FD.9.3
!+loft pullup+!	*	FD.7.1.9.2
!+longitude+!	SS.SYSVAL	FD.9.3 FD.10.5
!+longitude_cup+!	SS.SYSVAL	FD.10.5
!+Longitude error+!	SS.SYSVAL	FD.9.3
!+low drag release+!	SS.SYSVAL	FD.7.1.6.2 FD.7.1.7.1 FD.7.1.8 FD.7.1.9.1 FD.7.1.9.2
!+low lat ct+!	DI.PMDSR	FD.9.3
!+low lat ct a pnl+!	SS.PNL.INPUT	FD.10.7
!+low lat ct b pnl+!	SS.PNL.INPUT	FD.10.7
!+LSC azimuth+!	DI.HUD	FD.6.3
!+LSC elevation+!	DI.HUD	FD.5.1.2 FD.6.3 FD.7.1.1.2
!+LSC mode+!	DI.HUD	FD.7.1.9.2
!+MA+!	SS.DIAGIO	FD.9.3
!+Mag variation pnl+!	SS.PNL.INPUT	FD.9.3
!+magvar IMS+!	DI.IMS	FD.9.3
!+Map decenter+!	DI.SWB	FD.10.4
!+Map hold+!	DI.SWB	FD.10.2 FD.10.5

Appendix 4

Input Source Reference

!+Map latitude+!	DI.PMDS	FD.9.3 FD.10.5
!+Map longitude+!	DI.PMDS	FD.9.3 FD.10.5
!+Map north-up+!	DI.SWB	FD.10.4
!+map orient+!	DI.PMDSR	FD.9.3
!+map orient a pnl+!	SS.PNL.INPUT	FD.10.7
!+map orient b pnl+!	SS.PNL.INPUT	FD.10.7
!+Map position valid+!	DI.PMDS	FD.10.5
!+Map scale sw+!	DI.SWB	FD.10.7
!+Mark+!	SS.SYSVAL	FD.9.1
!+mark lat+!	SS.SYSVAL	FD.9.3 FD.10.5
!+mark long+!	SS.SYSVAL	FD.9.3 FD.10.5
!+MFSW eq BOC+!	SS.DIAGIO	FD.9.3
!+MFSW eq BOCOFF+!	SS.DIAGIO	FD.9.3
!+MFSW eq CCIP+!	SS.DIAGIO	FD.9.3
!+MFSW eq NATT+!	SS.DIAGIO	FD.9.3
!+MFSW eq NATTOFF+!	SS.DIAGIO	FD.9.3
!+MFSW eq TF+!	SS.DIAGIO	FD.9.3
!+MULTRACK+!	SS.DIAGIO	FD.9.3
!+N coarse bias+!	DI.IMSR	FD.9.3
!+N coarse bias pnl+!	SS.PNL.INPUT	FD.8.5
!+N coarse scale+!	DI.IMSR	FD.9.3
!+N coarse scale pnl+!	SS.PNL.INPUT	FD.8.5
!+N fine bias+!	DI.IMSR	FD.9.3
!+N fine bias pnl+!	SS.PNL.INPUT	FD.8.5
!+N fine scale+!	DI.IMSR	FD.9.3

Appendix 4

Input Source Reference

!+N fine scale pnl+!	SS.PNL.INPUT	FD.8.5
!+N vel IMS+!	DI.IMS	FD.9.3
!+nav_mode+!	SS.MODE	FD.10.5 FD.1.1 FD.11.1 FD.12.1 FD.12.2 FD.4.1 FD.5.1.1 FD.5.1.2 FD.5.2.1 FD.5.2.2 FD.7.1.3.1 FD.7.1.3.2 FD.7.1.4.1 FD.7.1.4.2 FD.7.2.1 FD.7.2.2 FD.7.2.3 FD.7.2.4 FD.7.2.4.3 FD.8.1 FD.8.2 FD.8.3.3 FD.9.3
!+Non-align+!	DI.VISIND	FD.12.2
!+normal_accel+!	PM.ACM	FD.7.2.4.2 FD.9.3
!+offset brg pnl+!	SS.PNL.INPUT	FD.9.3
!+offset dht pnl+!	SS.PNL.INPUT	FD.9.3
!+offset rng pnl+!	SS.PNL.INPUT	FD.9.3
!+OTS+!	SS.SYSVAL	FD.7.1.6.2 FD.7.1.2.2 FD.7.1.9.2
!+OTS pullup+!	PM*	FD.7.1.9.2
!+pitch IMS+!	DI.IMS	FD.7.1.2.2 FD.7.1.6.1 FD.7.1.9.2 FD.7.2.3
!+PMDCTR+!	SS.DIAGIO	FD.9.3
!+PMHOLD+!	SS.DIAGIO	FD.9.3
!+PMLAND+!	SS.DIAGIO	FD.9.3
!+PMNORUP+!	SS.DIAGIO	FD.9.3
!+PMSCAL eq 80+!	SS.DIAGIO	FD.9.3
!+pnl config+!	SS.PNL.CONFIG	FD.9.1 FD.9.3 FD.10.5
!+preparation time+!	DI.WCM	FD.13.1
!+PUAC elevation+!	DI.HUD	FD.5.1.2
!+Radalt priority pnl+!	SS.PNL.INPUT	FD.9.3
!+RE+!	SS.DIAGIO	FD.9.3
!+RE pressed+!	DI.SWB	FD.13.1
!+release pulse width+!	DI.WCM	FD.13.1

Appendix 4

Input Source Reference

!+rls pts passed+!	SS.SYSVAL	FD.9.3
!+roll IMS+!	DI.IMS	FD.5.1.2 FD.7.1.2.2 FD.7.2.3 FD.8.1
!+SINS dhdg pnl+!	SS.PNL.INPUT	FD.8.3.3 FD.9.3
!+SINS east vel+!	DI.SINS	FD.9.3
!+SINS east vel valid+!	DI.SINS	FD.9.3
!+SINS heading+!	DI.SINS	FD.8.3.3 FD.9.3
!+SINS heading valid+!	DI.SINS	FD.9.3
!+SINS lat+!	DI.SINS	FD.9.3
!+SINS lat valid+!	DI.SINS	FD.9.3
!+SINS long+!	DI.SINS	FD.9.3
!+SINS long valid+!	DI.SINS	FD.9.3
!+SINS north vel+!	DI.SINS	FD.9.3
!+SINS north vel valid+!	DI.SINS	FD.9.3
!+SINS pitch valid+!	DI.SINS	FD.9.3
!+SINS roll valid+!	DI.SINS	FD.9.3
!+SINS x offset pnl+!	SS.PNL.INPUT	FD.9.3
!+SINS y offset pnl+!	SS.PNL.INPUT	FD.9.3
!+SINS z offset pnl+!	SS.PNL.INPUT	FD.9.3
!+Slew moving+!	SS.SYSVAL	FD.10.5
!+slew FLR delta az+!	SS.SYSVAL	FD.6.2.1 FD.7.1.1.2
!+slew FLR delta rng+!	SS.SYSVAL	FD.6.2.1 FD.7.1.1.2
!+slew HUD delta az+!	SS.SYSVAL	FD.7.1.1.2
!+slew HUD delta elev+!	SS.SYSVAL	FD.7.1.1.2
!+slew map delta lat+!	SS.SYSVAL	FD.10.5

Appendix 4

Input Source Reference

!+slew map delta long+!	SS.SYSVAL	FD.10.5
!+Slew right-left+!	DI.SLEW	FD.6.2.1 FD.7.1.1.2 FD.10.5
!+Slew up-down+!	DI.SLEW	FD.6.2.1 FD.7.1.1.2 FD.10.5
!+sr_ac_ap+!	SS.SYSVAL	FD.6.2.1
!+sr_ac_bpup+!	SS.SYSVAL	FD.7.1.7.2
!+sr_ac_cup+!	SS.SYSVAL	FD.6.2.1
!+sr_ac_ftpt+!	SS.SYSVAL	FD.6.2.1
!+sr_ac_gpup+!	SS.SYSVAL	FD.7.1.7.2
!+sr_ac_ip+!	SS.SYSVAL	FD.9.3
!+sr_ac_oap+!	SS.SYSVAL	FD.6.2.1
!+sr_ac_rls+!	SS.SYSVAL	FD.7.1.6.2
!+sr_ac_tgt+!	SS.SYSVAL	FD.6.2.1 FD.9.3
!+sr reasonable+!	SS.SYSVAL	FD.9.3
!+STA1RDY+!	SS.DIAGIO	FD.9.3
!+STA2RDY+!	SS.DIAGIO	FD.9.3
!+STA3RDY+!	SS.DIAGIO	FD.9.3
!+STA6RDY+!	SS.DIAGIO	FD.9.3
!+STA7RDY+!	SS.DIAGIO	FD.9.3
!+STA8RDY+!	SS.DIAGIO	FD.9.3
!+steering error to rls+!	PM*	FD.7.1.2.2
!+steering error to tgt+!	SS.SYSVAL	FD.5.1.1 FD.7.1.2.2
!+steering to tgt+!	SS.SYSVAL	FD.7.1.2.2
!+symbol_az_on_AS!+!	SS.SUBRTN	FD.7.1.6.2 FD.7.1.7.2 FD.7.1.9.2
!+target in range+!	SS.SYSVAL	FD.7.1.5
!+TAS ADC+!	DI.ADC	FD.9.3 FD.12.2

Appendix 4

Input Source Reference

!+TD+!	SS.DIAGIO	FD.9.3
!+test_mode+!	SS.MODE	FD.6.1 FD.7.1.1.2 FD.7.2.1 FD.7.2.3 FD.7.2.4.3 FD.14.1
!+test result+!	EC.DIAG	FD.14.1
!+time to ftpt+!	SS.SYSVAL	FD.9.3
!+TOS+!	SS.SYSVAL	FD.7.1.6.2
!+update mode+!	SS.MODE	FD.6.1 FD.6.2.1 FD.6.2.2 FD.6.3 FD.7.1.1.1 FD.7.1.1.2 FD.9.2 FD.9.3 FD.10.5
!+USC elevation+!	DI.HUD	FD.7.1.1.2
!+V coarse bias+!	DI.IMSR	FD.9.3
!+V coarse bias pnl+!	SS.PNL.INPUT	FD.8.5
!+V coarse scale+!	DI.IMSR	FD.9.3
!+V coarse scale pnl+!	SS.PNL.INPUT	FD.8.5
!+Velocity east system+!	SS.SYSVAL	FD.7.1.4.2
!+Velocity north system+!	SS.SYSVAL	FD.7.1.4.2
!+Velocity vertical system+!	SS.SYSVAL	FD.7.1.4.2 FD.7.2.4.3
!+weap_mode+!	SS.MODE	FD.7.1.1.1 FD.13.1 FD.2.1 FD.5.1.1 FD.5.1.2 FD.5.2.1 FD.6.1 FD.6.2.1 FD.6.2.2 FD.6.3 FD.7.1.1.2 FD.7.1.2.1 FD.7.1.2.2 FD.7.1.4.1 FD.7.1.4.2 FD.7.1.5 FD.7.1.6.1 FD.7.1.6.2 FD.7.1.7.1 FD.7.1.7.2 FD.7.1.9.1 FD.7.1.9.2 FD.7.2.4.2 FD.7.2.4.3 FD.9.3
!+Weapon Class+!	DI.WCM	FD.2.1 FD.6.1 FD.7.1.2.2 FD.7.1.1.2 FD.7.1.7.1 FD.7.1.7.2 FD.7.1.8
!+WEAP TYP+!	SS.DIAGIO	FD.9.3
!+wind dir+!	SS.SYSVAL	FD.9.3
!+wind vel+!	SS.SYSVAL	FD.9.3

Appendix 4

Input Source Reference

!+wmode class+!	SS.MODE	FD.13.1 FD.2.1 FD.5.1.2 FD.7.1.6.1 FD.7.1.6.2 FD.7.1.9.1 FD.7.1.9.2
!+wpns rlsd+!	SS.SYSVAL	FD.13.1
!+X corr increm+!	DI.IMSR	FD.9.3
!+X corr increm pnl+!	SS.PNL.INPUT	FD.8.5
!+X drift+!	DI.IMSR	FD.9.3
!+X drift pnl+!	SS.PNL.INPUT	FD.8.5
!+Y corr increm+!	DI.IMSR	FD.9.3
!+Y corr increm pnl+!	SS.PNL.INPUT	FD.8.5
!+Y drift+!	DI.IMSR	FD.9.3
!+Y drift pnl+!	SS.PNL.INPUT	FD.8.5
!+Z corr increm+!	DI.IMSR	FD.9.3
!+Z corr increm pnl+!	SS.PNL.INPUT	FD.8.5
!+Z drift+!	DI.IMSR	FD.9.3
!+Z drift pnl+!	SS.PNL.INPUT	FD.8.5

Appendix 5

Function Driver Review Process

I. REVIEWERS

Each function driver specification should be reviewed from the four different points of view described below. Although four different people will review each specification, no one person will be asked to review every specification. Instead, each reviewer will be asked to review a set of closely-related function driver specifications.

<u>Point of view</u>	<u>Expertise required</u>
NWC-2 EXPERTS	Detailed familiarity with significant parts of NWC-2 program. These people need to understand the behavior of the current software. Programming knowledge is not necessary.
DEVICE EXPERTS	Familiarity with the devices used on the A-7 and with similar devices found on other aircraft. These people should be familiar as well with the architecture of the virtual devices provided by the Device Interface Module. They should know about several devices of a particular type, about the technology used to build them, and about past changes and future trends in devices.
COMPARERS	Good logical minds. A certain familiarity with avionics programs would be helpful but not mandatory. These people will be asked to perform certain checks for consistency and completeness; consequently, a lack of information on the specific application may be advantageous.
AVIONICS PROGRAMMERS	Programming. These people must be able to sketch a pseudo-code program implementing a given functional specification along with interface specifications for support modules.

REVIEW A: Accuracy Review

Each function driver specification should produce output that will result in system behavior that is identical to that of NWC-2 at all times, and it should not specify values in situations in which the values are irrelevant.

Each reviewer should answer the following questions for each function driver specification reviewed. Because the behavior of the function drivers is dependent upon the other modules that it uses, reviewers will be supplied with drafts of interface specifications of the Shared Services and Device Interface modules.

- A1: Answer this question for each table that appears in the function driver specification being reviewed: For each box of the table, is the information contained there clear and unambiguous? Explain any "no" answers on the back of the form.
- A2: Answer this question for each table that appears in the function driver specification being reviewed: For each box of the table, does the information contained there specify the current behavior of NWC-2? Explain any "no" answers on the back of the form.
- A3: For each value designated by the "!" brackets that appears in the specification, write down the name of the value and answer the following questions:
- a. Where did you find its definition? Please give the document name and relevant section number. If you failed to locate its definition, answer with "X".
 - b. For an item whose definition you located, is the definition of that item such that it will allow the function driver to meet its requirements? Explain any "no" answers on the back of the form.
 - c. Is the mnemonic suggestive of the proper meaning? Explain any "no" answers on the back of the form.
- A4: Are there situations when NWC-2 will produce output, and we have not specified it?
- A5: Are there situations in which the value of the output is irrelevant in NWC-2, but we have specified a value?

REVIEW B: DIM appropriateness review

Each function driver module depends on the Device Interface Module in two ways. First, the function driver sends output to a virtual device via that device's access programs. In addition, the DIM produces sensor information that the function driver evaluates and uses to determine its output.

If a hardware device is replaced by a new device with the same capabilities, then the function drivers should not change. In addition, the Device Interface Module must provide capabilities that are consistent with those assumed by the function driver module. Finally, no function driver module should depend on any information that is a secret of a device interface module.

Each reviewer should answer the following questions for each function driver specification reviewed:

- B1: List the virtual devices that produce data used by this function driver. For each device listed, answer the following questions:
 - a. What kind of replacements for these devices, if any, would cause a change in this function driver module?
 - b. Are there times when the device or the values it produces are not available when needed by this function driver?
- B2: Determine the virtual device that this function driver controls by its output. What sort of replacement device, if any, would cause a change in this function driver?
- B3: Consider the set of virtual devices used by this function driver, and the undesired events specified in the Device Interface Module as being associated with each device. List the undesired events that this function driver can possibly cause to occur.
- B4: For each undesired event you listed above, answer the following question: Can this function driver always avoid causing this undesired event? Or are there times when this function driver depends on the action of some other module to avoid giving rise to the undesired event?

REVIEW C: Programmability review

The function driver specifications must contain enough information to be unambiguously implemented, yet they must contain no software design decisions.

For each function driver specification reviewed, each reviewer should sketch the program (in pseudo-code) that implements it. Then the reviewer should answer each of the following questions:

- C1: For each line of your pseudo-code program, note in the left margin what information in the specification was used. Use the following code:

<u>Symbol</u>	<u>Meaning</u>
Rn	Used the entire nth row of a table;
Cm	Used the entire mth column of a table; for instance, C1 means that the entire mode list was utilized by this line of the program;
Rn,Cm	Used information in the box defined by row n and column m of the table;
LD	Used information in the specification's local dictionary;
P	Used information contained in prose paragraph;
O	Used information from some other section of the specification;

- C2: What sections of the specification were not used, according to your list of utilized sections created by Question C1?
- C3: Is it possible to write another program that is not functionally equivalent to the first one that also meets the specification? If so, describe the ambiguity in the specification that allows this.
- C4: Is there any necessary information that was not provided by the specification? What was it?
- C5: List all possible failures of services and devices that the function driver assumes are available. For each one, how does the specification determine what to do in that case?

REVIEW D: Comparison review

The function drivers should be relatively independent; that is, a minor change to one should not change any other.

Each review should answer the following questions for each group of function driver specifications reviewed:

- D1: Is there a compound condition or event or expression that appears in more than one function driver? If so, what is it, and in which function drivers does it appear?
- D2: What small change would cause a change in more than one function driver in the group? Are there things common to more than one function driver that are likely to change together?
- D3: Is there more than one function driver in this group that determines the same output to the same virtual device? If so, which function drivers and which device?

REVIEW A: Accuracy Review

Answer Form for Questions A1 and A2

Reviewer: _____

Function Driver table being reviewed: Table FD.

Question being answered (circle one): A1 A2

For each box in the table being reviewed, answer "yes" or "no"; abbreviate "y" or "n" if you like. Explain each "no" answer on the back of this form.

	mode list							
	Col. 1	Col. 2	Col. 3	Col. 4	Col. 5	Col. 6	Col. 7	Col. 8
Row 1								
Row 2								
Row 3								
Row 4								
Row 5								
Row 6								
Row 7								
Row 8								

REVIEW A: Accuracy Review

Answer Form for Question A3

Reviewer: _____

Function Driver being reviewed: FD. _____

Explain any "no" answers on the back of this form. Use as many forms as you need for this function driver.

Name of !+ +! value	Where did you find its definition? (Document and section #, or "X")	Is the definition such that it will allow the FD to meet its requirements?	Is the mnemonic proper?
!+	+!		
!+	+!		
!+	+!		
!+	+!		
!+	+!		
!+	+!		
!+	+!		
!+	+!		
!+	+!		
!+	+!		
!+	+!		

REVIEW A: Accuracy Review

Answer Form for Questions A4 and A5

Reviewer: _____

Function Driver being reviewed: FD.

A4: Are there situations when NWC-2 will produce output, and we have not specified it?

A5: Are there situations in which the value of the output is irrelevant, but we have specified a value?

REVIEW B: DIM appropriateness review

Answer Form for Question B1

Reviewer: _____

Function Driver being reviewed: FD.

B1: List the virtual devices that produce data used by this function driver. For each device listed, answer the following questions:

a. What kind of replacements for these devices, if any, would cause a change in this function driver module?

b. Are there times when the device or the values it produces are not available when needed by this function driver?

REVIEW B: DIM appropriateness review

Answer Form for Questions B2 through B4

Reviewer: _____

Function Driver being reviewed: FD.

B2: Determine the virtual device that this function controls by its output. What sort of replacement device, if any, would cause a change in this function driver?

B3: Consider the set of virtual devices used by this function driver, and the undesired events specified in the Device Interface Module as being associated with each device. List the undesired events that this function driver can possibly cause to occur.

B4: For each undesired event you listed above, answer the following question: Can this function driver always avoid causing this undesired event? Or are there times when this function driver depends on the action of some other module to avoid giving rise to the undesired event?

Use back of form if necessary.

REVIEW C: Programmability review

Answer Form for Psuedo-Code Program Sketch

Reviewer: _____

Function Driver being reviewed: FD.

For each function driver specification reviewed, sketch the program (in a pseudo-code) that implements it. Use the back of this form if necessary.

REVIEW C: Programmability review

Answer Form for Questions C1 and C2

Reviewer: _____

Function Driver being reviewed: FD.

C1: For each line of your pseudo-code program, note in the left margin what information in the specification was used. Use the following code:

<u>Symbol</u>	<u>Meaning</u>
Rn	Used the entire nth row of a table;
Cm	Used the entire mth column of a table; for example, C1 means that the entire mode list was utilized by this line of the program;
Rn,Cm	Used information in the box defined by row n and column m of the table;
LD	Used information in the specification's local dictionary;
P	Used information contained in prose paragraph;
O	Used information from some other section of the specification;

C2: What sections of the specification were not used, according to your list of utilized sections created by Question C1?

REVIEW C: Programmability review

Answer Form for Questions C3 through C5

Reviewer: _____

Function Driver being reviewed: FD.

C3: Is it possible to write another program that is not functionally equivalent to the first one that also meets the specification? If so, describe the ambiguity in the specification that allows this.

C4: Is there any necessary information that was not provided by the specification? What was it?

C5: List all possible failures of services and devices that the function driver assumes are available. For each one, how does the specification determine what to do in that case?

REVIEW D: Comparison review

Answer Form for Questions D1 through D3

Reviewer: _____

Answer the following questions for each group of function driver specifications reviewed:

D1: Is there a compound condition or event or expressions that appears in more than one function driver? If so, what is it, and in which function drivers does it appear?

D2: What small change would cause a change in more than one function driver in the group? Are there things common to more than one function driver that are likely to change together?

D3: Is there more than one function driver in this group that determines the same output to the same virtual device? If so, which function drivers and which device?

**DAT
FILM**