# Module -- HiLoadStall

# High-Level Description

The High Load Stall Thermal Management algorithm protects the system from prolonged intervals of high assist torque at near-stall conditions.

# Figures

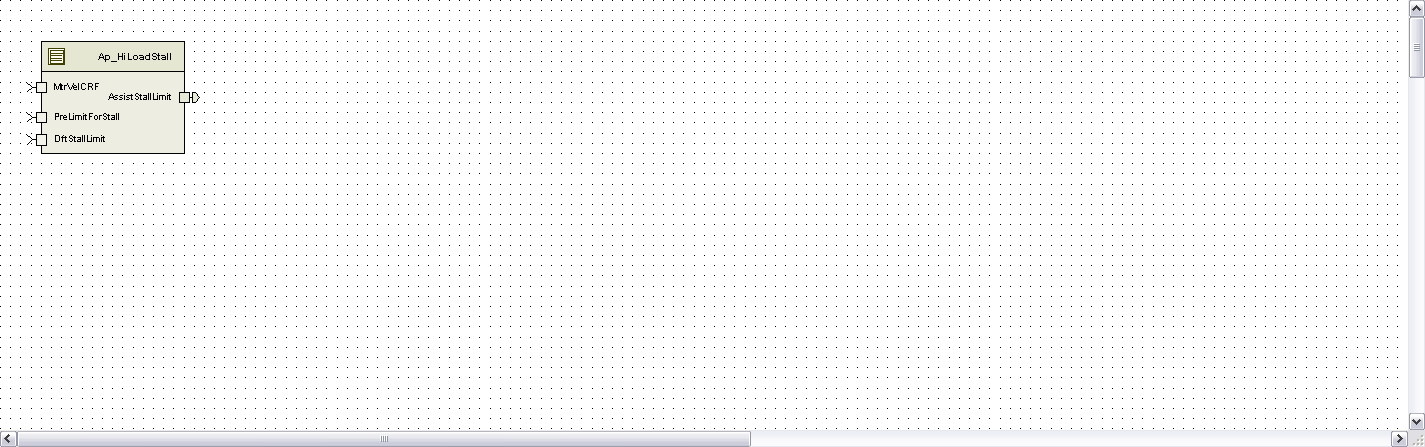
## Diagram – Function Data Sharing

This diagram shows all data that is shared between functions within the module.

None

### Diagram – Ret HiLoadStall \_Per1

This diagram describes the functional characteristics and data flow of a given function.



# Module Inputs and Outputs

For details on module input / output variable, refer to the Data Dictionary for the application. Input / output variable names are listed here for reference.

(Note: Full variable names required in table.)

(Note: All global variables including End Of Line data used should be shown here)

|  |  |
| --- | --- |
| Module Inputs (Global Variable Name) | Module Outputs (Global Variable Name) |
| MtrVelCRF\_MtrRadpS\_f32 | AssistStallLimit\_MtrNm\_f32 |
| PreLimitForStall\_MtrNm\_f32 |  |
| DftStallLimit\_Cnt\_lgc |  |
|  |  |

## Module Internal Variables

This section identifies the name, range and resolutions for module specific data created by this module. If there are no range restrictions on the variable, the term “FULL” is placed into the table for legal range.

(Note: If no module specific variables are used by the design, place the text “None” in the first Variable Name cell in the table)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Variable Name | Resolution | Legal Range  (min) | Legal Range  (max) | Software Segment |
| PrevAssistStallLimit\_MtrNm\_M\_f32 | Single Precision Floating Point | 0 | 8.8 | HILOADSTALL\_START\_SEC\_VAR\_CLEARED\_32 |
| ModPreLimitFiltSV\_MtrNm\_M\_u8p24 | 2^-24 | 0 | 8.8 | HILOADSTALL\_START\_SEC\_VAR\_CLEARED\_32 |
| ModPreLimit\_MtrNm\_D\_u8p8 | 0.00390625 | 0 | 8.8 | HILOADSTALL\_START\_SEC\_VAR\_CLEARED\_16 |
| FiltModPreLimit\_MtrNm\_D\_u8p8 | 0.00390625 | 0 | 8.8 | HILOADSTALL\_START\_SEC\_VAR\_CLEARED\_16 |
| StallLimit\_MtrNm\_D\_u8p8 | 0.00390625 | 0 | 8.8 | HILOADSTALL\_START\_SEC\_VAR\_CLEARED\_16 |

### User defined typedef definition/declaration

This section documents any user types uniquely used for the module.

|  |  |  |  |
| --- | --- | --- | --- |
| Variable Name | Typedef Name | Storage Type | Safety Critical Classification |
| None |  |  |  |

# Constant Data Dictionary

## Calibration Constants

This section lists the calibrations used by the module. For details on calibration constants, refer to the Data Dictionary for the application.

(Note: If no calibrations are used by the design, place the text “None” in the first location in the table)

|  |
| --- |
| Constant Name |
| k\_AbsMtrVelBkt\_MtrRadps\_f32 |
| k\_EOTThrmPrtLPFKn\_Cnt\_u16 |
| t\_EOTThrmIndptTbl\_MtrNm\_u8p8 |
| t\_EOTThrmDpntTbl\_MtrNm\_u8p8 |
| k\_EOTThrmSlwLmtStp\_MtrNm\_f32 |

## Program(fixed) Constants

### Embedded Constants

All embedded constants whose values are provided in Eng units will be evaluated to the equivalent counts by using the FPM\_InitFixedPoint\_m() macro within the #define statement.

#### Local

|  |  |  |
| --- | --- | --- |
| Constant Name | Resolution | Value |
|  |  |  |

#### Note: RtnLoopTime depends on the rate of the periodic function.

#### Global

This section lists the global constants used by the module. For details on global constants, refer to the Data Dictionary for the application.

|  |
| --- |
| Constant Name |
| D\_MTRTRQCMDHILMT\_MTRNM\_F32 |
| D\_ZERO\_ULS\_F32 |

### Module specific Lookup Tables Constants

(This is for lookup tables (arrays) with fixed values, same name as other tables)

|  |  |  |  |
| --- | --- | --- | --- |
| Constant Name | Resolution | Value | Software Segment |
|  |  |  |  |

# Software Module Implementation

## Initialization Functions

### Init: HiLoadStall\_Init()

#### Design Rationale

None

#### Module Outputs

None

#### Module Internal

None

## Periodic Functions

### Per: HiLoadStall\_Per1

#### Design Rationale

None

#### Program Flow Start

Rte\_Call\_HiLoadStall\_Per1\_CP0\_CheckpointReached()

#### Store Module Inputs to Local copies

MtrVelCRF\_MtrRadpS\_T\_f32 = Rte\_IRead\_HiLoadStall\_Per1\_HighStallLmt \_MtrVelCRF\_MtrRadpS\_f32()

PreLimitForStall\_MtrNm\_T\_f32 = Rte\_Iread\_HiLoadStall\_Per1\_HighStallLmt \_ PreLimitForStall\_MtrNm\_f32()

DftStallLimit\_Cnt\_T\_lgc = Rte\_Iread\_HiLoadStall\_Per1\_HighStallLmt \_DftStallLimit\_Cnt\_lgc()

PreLimitForStall\_MtrNm\_T\_u8p8 = FPM\_FloatToFixed\_m(PreLimitForStall\_MtrNm\_T\_f32, u8p8\_T)

#### Determine EOT Thermal Limiting

##### EOT Thermal Algorithm Enable

##### 

##### Low Pass Filter



##### EOT Thermal Protection Limit Table



##### Slew Limiting



##### Serial Comm Defeat Sub-Function



#### Store Local copy of outputs into Module Outputs

#### Rte\_IWrite\_HiLoadStall\_Per1\_AssistStallLimit\_MtrNm\_f32(PrevAssistStallLimit\_MtrNm\_M\_f32)

#### Program Flow End

Rte\_Call\_HiLoadStall\_Per1\_CP1\_CheckpointReached()

## Fault Recovery Functions

None

## Shutdown Functions

None

## Interrupt Functions

None

## Serial Communication Functions

None

## Local Function/Macro Definitions



# Execution Requirements

## Execution Sequence of the Module

## Execution Rates for sub-modules called by the Scheduler

This table serves as reference for the Scheduler design

|  |  |  |  |
| --- | --- | --- | --- |
| Function Name | Task List | Calling Frequency | System State(s) in which the function is called |
| HiLoadStall \_Per1 |  | 2ms | All |

## Execution Requirements for Serial Communication Functions

|  |  |
| --- | --- |
| Function Name | Sub-Module called by (Serial Comm Function Name) |
| None |  |

# Memory Map Definition Requirements

## Sub Modules (Functions)

This table identifies the software segments for functions identified in this module.

|  |  |
| --- | --- |
| Name of Sub Module | Software Segment |
| HiLoadStall\_Per1() | RTE\_START\_SEC\_AP\_HILOADSTALL\_APPL\_CODE |

## Local Functions

This table identifies the software segments for local functions identified in this module.

|  |  |
| --- | --- |
| Name of Sub Module | Software Segment |
| None |  |

# Known Issues / Limitations With Design

1. Inline functions in globalmacros.h are not unit tested.

# Revision Control Log

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Item #** | **Rev #** | **Change Description** | **Date** | **Author Initials** |
| 1 | 1.0 | Initial release | 15DEC11 | M. Story |
| 2 | 2.0 | Changed ranges on M variables for unsigned. Made ModPreLimitFiltSV\_Cnt\_M\_u16 name match code. | 07Jan12 | M. Story |
| 3 | 3.0 | Changes to correct state variable issue found during review. Updated slew limit logic and replaced local constants with use of global constants. Variable name updates and added output legal range limit. | 27Jan12 | LWW |
| 4 | 4.0 | Check Point added for Periodic program flow | 21-Sep-2012 | Selva |