**Module Design Document**

**For**

**GuardCfgAndDiagc**

**Apr 10 , 2018**

**Prepared For:**

**Software Engineering**

**Nexteer Automotive,**

**Saginaw, MI, USA**

**Prepared By:**

**Software Group,**

**Nexteer Automotive,**

**Saginaw, MI, USAChange History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Description** | **Author** | **Version** | **Date** |
| Initial Version | Avinash James | 1.0 | 02/16/16 |
| Updates for PBG Register Lock bits and Syncm inclusion | Avinash James | 2.0 | 03/31/16 |
| Updates for design limitation | Avinash James | 3.0 | 04/10/18 |

**Table of Contents**

[1 Introduction 5](#_Toc443402489)

[1.1 Purpose 5](#_Toc443402490)

[1.2 Scope 5](#_Toc443402491)

[2 GuardCfgAndDiagc & High-Level Description 6](#_Toc443402492)

[3 Design details of software module 7](#_Toc443402493)

[3.1 Graphical representation of GuardCfgAndDiagc 7](#_Toc443402494)

[3.2 Data Flow Diagram 7](#_Toc443402495)

[3.2.1 Component level DFD 7](#_Toc443402496)

[3.2.2 Function level DFD 7](#_Toc443402497)

[4 Constant Data Dictionary 8](#_Toc443402498)

[4.1 Program (fixed) Constants 8](#_Toc443402499)

[4.1.1 Embedded Constants 8](#_Toc443402500)

[5 Software Component Implementation 9](#_Toc443402501)

[5.1 Sub-Module Functions 9](#_Toc443402502)

[5.1.1 Init: GuardCfgAndDiagcInit1 9](#_Toc443402503)

[5.1.1.1 Design Rationale 9](#_Toc443402504)

[5.1.1.2 Module Outputs 9](#_Toc443402505)

[5.1.2 Init: GuardCfgAndDiagcInit2 9](#_Toc443402506)

[5.1.2.1 Design Rationale 9](#_Toc443402507)

[5.1.2.2 Module Outputs 9](#_Toc443402508)

[5.1.3 Init: GuardCfgAndDiagcInit3 9](#_Toc443402509)

[5.1.3.1 Design Rationale 9](#_Toc443402510)

[5.1.3.2 Module Outputs 9](#_Toc443402511)

[5.1.4 Per: None 9](#_Toc443402512)

[5.2 Server Runables 9](#_Toc443402513)

[5.3 Interrupt Functions 9](#_Toc443402514)

[5.4 Module Internal (Local) Functions 10](#_Toc443402515)

[5.4.1 ConfigureFilterN 10](#_Toc443402516)

[5.4.1.1 Design Rationale 10](#_Toc443402517)

[5.4.1.2 Processing 10](#_Toc443402518)

[5.4.2 ChkForPBGErr 10](#_Toc443402519)

[5.4.2.1 Design Rationale 10](#_Toc443402520)

[5.4.2.2 Processing 10](#_Toc443402521)

[5.4.3 ChkForECMErr 10](#_Toc443402522)

[5.4.3.1 Design Rationale 10](#_Toc443402523)

[5.4.3.2 Processing 11](#_Toc443402524)

[5.4.4 Vrfy32BitPBGRegAcs 11](#_Toc443402525)

[5.4.4.1 Design Rationale 11](#_Toc443402526)

[5.4.4.2 Processing 11](#_Toc443402527)

[5.4.5 Vrfy16BitPBGRegAcs 11](#_Toc443402528)

[5.4.5.1 Design Rationale 11](#_Toc443402529)

[5.4.5.2 Processing 11](#_Toc443402530)

[5.4.6 Vrfy8BitPBGRegAcs 11](#_Toc443402531)

[5.4.6.1 Design Rationale 11](#_Toc443402532)

[5.4.6.2 Processing 11](#_Toc443402533)

[5.5 GLOBAL Function/Macro Definitions 12](#_Toc443402534)

[5.5.1 GLOBAL Function #1 12](#_Toc443402535)

[5.5.1.1 Design Rationale 12](#_Toc443402536)

[5.5.1.2 Processing 12](#_Toc443402537)

[6 Known Limitations with Design 13](#_Toc443402538)

[7 UNIT TEST CONSIDERATION 14](#_Toc443402539)

[Appendix A Abbreviations and Acronyms 15](#_Toc443402540)

[Appendix B Glossary 16](#_Toc443402541)

[Appendix C References 17](#_Toc443402542)

# Introduction

## Purpose

## Scope

The following definitions are used throughout this document:

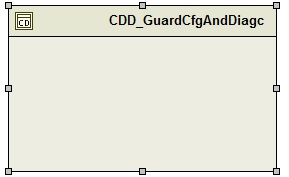
* **Shall**: indicates a mandatory requirement without exception in compliance.
* **Should**: indicates a mandatory requirement; exceptions allowed only with documented justification.
* **May**: indicates an optional action.

# GuardCfgAndDiagc & High-Level Description

*See FDD*

# Design details of software module

## Graphical representation of GuardCfgAndDiagc

**

## Data Flow Diagram

### Component level DFD

*See FDD*

### Function level DFD

*See FDD*

# Constant Data Dictionary

## Program (fixed) Constants

### Embedded Constants

#### Local Constants

|  |  |  |  |
| --- | --- | --- | --- |
| Constant Name | Resolution | Units | Value |
| **PBGPROTNCMN\_CNT\_U32** | **1** | **uint32** | **0x0405FE1FU** |
| **PBGUSRMODENA\_CNT\_U32** | **1** | **uint32** | **0x02000000U** |
| **PBGUSRMODDI\_CNT\_U32** | **1** | **uint32** | **0x00000000U** |
| **PBGSPID321ENA\_CNT\_U32** | **1** | **uint32** | **0x000001C0U** |
| **PBGSPID31ENA\_CNT\_U32** | **1** | **uint32** | **0x00000140U** |
| **PBGSPID21ENA\_CNT\_U32** | **1** | **uint32** | **0x000000C0U** |
| **PBGSPID1ENA\_CNT\_U32** | **1** | **uint32** | **0x00000040U** |
| **PBGSETNOREADWRACS\_CNT\_U32** | **1** | **uint32** | **0x405FE5CU** |
| **NROF8BITREG\_CNT\_U08** | **1** | **uint8** | **((uint8)0x09)** |
| **NROF32BITREG\_CNT\_U08** | **1** | **uint8** | **((uint8)0x02)** |
| **READERRBIT\_CNT\_U32** | **1** | **uint32** | **((uint32)1U<<6U)** |
| **WRERRBIT\_CNT\_U32** | **1** | **uint32** | **((uint32)1U<<7U)** |
| **CFGERRBIT\_CNT\_U32** | **1** | **uint32** | **((uint32)1U<<8U)** |
| **PBGERRBIT\_CNT\_U32** | **1** | **uint32** | **((uint32)1U<<9U)** |
| **ECMERRBIT\_CNT\_U32** | **1** | **uint32** | **((uint32)1U<<10U)** |
| **REGTYPE8BIT\_CNT\_U32** | **1** | **uint32** | **((uint32)0U<<4U)** |
| **REGTYPE16BIT\_CNT\_U32** | **1** | **uint32** | **((uint32)1U<<4U)** |
| **REGTYPE32BIT\_CNT\_U32** | **1** | **uint32** | **((uint32)2U<<4U)** |
| **PBGSTRTUPTESTNOFAILR\_CNT\_U32** | **1** | **uint32** | **0x0U** |
| **PBGPROTNLOCKENA\_CNT\_U32** | **1** | **uint32** | **0x80000000U** |

# Software Component Implementation

## Sub-Module Functions

### Init: GuardCfgAndDiagcInit1

## Design Rationale

*Non-RTE function for Guard configuration initialization of PEG, IPG, and PBG so that guard protection can be initialized and enabled before the RTE is started*

## Module Outputs

*Configuration registers for PEG, IPG, and PBG*

### Init: GuardCfgAndDiagcInit2

## Design Rationale

*RTE Empty function for purposes of memory mapping*

*See FDD for more.*

## Module Outputs

*None*

### Init: GuardCfgAndDiagcInit3

## Design Rationale

*Non-RTE function for Start Up Initialization test of PBG of Group 3A*

*See FDD for more.*

## Module Outputs

*None*

## Per: None

## Server Runables

*None*

## Interrupt Functions

*None*

## Module Internal (Local) Functions

### ConfigureFilterN

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | ConfigureFilterN | Type | Min | Max |
| **Arguments Passed** | PbgProtReg | volatile uint32\* | 0 | 0xFFFFFFFF |
|  | Val | uint32 | 0 | 0xFFFFFFFF |
|  | PbgStrtUpTestFailSts | Uint32 \* | 0 | 0xFFFFFFFF |
| **Return Value** | None |  |  |  |

## Design Rationale

This local function sets the value **Val** to the register address **PbgProtReg** passed as the arguments and verifies the write operation was successful. If not a diagnostic is set.

## Processing

Figure 4.5.3 from SAN ver 1.20

### ChkForPBGErr

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | ChkForPBGErr | Type | Min | Max |
| **Arguments Passed** | PbgStrtUpTestFailSts | Uint32 \* | 0 | 0xFFFFFFFF |
|  |  |  |  |  |
| **Return Value** | None |  |  |  |

## Design Rationale

This local function checks PBG access violation error is captured. If not set diagnostic, clear the error and if the error doesn’t clear set diagnostic.

## Processing

Figure 4.5.3 from SAN ver 1.20

### ChkForECMErr

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | ChkForECMErr | Type | Min | Max |
| **Arguments Passed** | PbgStrtUpTestFailSts | Uint32 \* | 0 | 0xFFFFFFFF |
|  |  |  |  |  |
| **Return Value** | None |  |  |  |

## Design Rationale

This local function checkscwhether ECM captures the error sets diagnostic message and clears the ECM errors after the check else set diagnostic.

## Processing

Refer FDD 4.5.3 Implementation

### Vrfy32BitPBGRegAcs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | Vrfy32BitPBGRegAcs | Type | Min | Max |
| **Arguments Passed** | PbgStrtUpTestFailSts | Uint32 \* | 0 | 0xFFFFFFFF |
|  |  |  |  |  |
| **Return Value** | None |  |  |  |

## Design Rationale

This is defined to reduce the path count and modularizes the check for the 32 bit Access registers alone.

## Processing

### Vrfy16BitPBGRegAcs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | Vrfy16BitPBGRegAcs | Type | Min | Max |
| **Arguments Passed** | PbgStrtUpTestFailSts | Uint32 \* | 0 | 0xFFFFFFFF |
|  |  |  |  |  |
| **Return Value** | None |  |  |  |

## Design Rationale

This is defined to reduce the path count and modularizes the check for the 16 bit Access registers alone.

## Processing

### Vrfy8BitPBGRegAcs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | Vrfy8BitPBGRegAcs | Type | Min | Max |
| **Arguments Passed** | PbgStrtUpTestFailSts | Uint32 \* | 0 | 0xFFFFFFFF |
|  |  |  |  |  |
| **Return Value** | None |  |  |  |

## Design Rationale

This is defined to reduce the path count and modularizes the check for the 8 bit Access registers alone.

## Processing

## GLOBAL Function/Macro Definitions

## GLOBAL Function #1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** |  | Type | Min | Max |
| **Arguments Passed** |  |  |  |  |
|  |  |  |  |  |
| **Return Value** |  |  |  |  |

## Design Rationale

## Processing

# Known Limitations with Design

In the design for the error injection code, the reads make use of variables mapped in the global shared memory. Since we already have global variables mapped to global shared memory in the error injection code which are of type uint32, an explicit cast of uint32 is done for registers that are not unit32.Also the error injection code accesses register in the sys\_regs.h file which is not currently included in the DF003A. This is being conditionally included in the component c file as this will not be part of the production code.

# UNIT TEST CONSIDERATION

None

Abbreviations and Acronyms

| **Abbreviation or Acronym** | **Description** |
| --- | --- |
|  |  |
|  |  |

Glossary

**Note**: Terms and definitions from the source “Nexteer Automotive” take precedence over all other definitions of the same term. Terms and definitions from the source “Nexteer Automotive” are formulated from multiple sources, including the following:

* ISO 9000
* ISO/IEC 12207
* ISO/IEC 15504
* Automotive SPICE® Process Reference Model (PRM)
* Automotive SPICE® Process Assessment Model (PAM)
* ISO/IEC 15288
* ISO 26262
* IEEE Standards
* SWEBOK
* PMBOK
* Existing Nexteer Automotive documentation

| **Term** | **Definition** | **Source** |
| --- | --- | --- |
| MDD | Module Design Document |  |
| DFD | Data Flow Diagram |  |

References

| **Ref. #** | **Title** | **Version** |
| --- | --- | --- |
| 1 | AUTOSAR Specification of Memory Mapping (Link:[AUTOSAR\_SWS\_MemoryMapping.pdf](http://www.autosar.org/download/R4.0/AUTOSAR_SWS_MemoryMapping.pdf)) | v1.3.0 R4.0 Rev 2 |
| 2 | MDD Guideline | EA4 01.00.01 |
| 3 | [Software Naming Conventions.doc](http://misagweb01.nexteer.com/eRoomReq/Files/erooms8/NextGeneration/0_fc55f/Software%20Naming%20Conventions%2003x(In%20Work).doc) | 2.0 |
| 4 | [Software Design and Coding Standards.doc](http://eroom1.nexteer.com/eRoomReq/Files/erooms8/NextGeneration/0_1a67a9/Software%20Design%20and%20Coding%20Standards.doc) | 2.1 |