**Module Design Document**

**For**

**SyncCrc**

**July 5, 2017**

**Prepared For:**

**Software Engineering**

**Nexteer Automotive,**

**Saginaw, MI, USA**

**Prepared By:**

**Kevin Smith,**

**Nexteer Automotive,**

**Saginaw, MI, USAChange History**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Description** | **Author** | **Version** | **Date** | **Approved By** |
| Initial Version | K. Smith | 1 | 07-Oct-15 |  |
| Updates to meet the Rev1 of the FDD | K. Smith | 2 | 11-Jan-16 |  |
| Updates to support OS version and hardware differences | K. Smith | 3 | 14-Jun-17 |  |
| Updates for anomaly EA4#13207 | K. Smith | 4 | 05-Jul-17 |  |

Table of Contents

[1 Introduction 5](#_Toc485325396)

[1.1 Purpose 5](#_Toc485325397)

[1.2 Scope 5](#_Toc485325398)

[2 SyncCrc & High-Level Description 6](#_Toc485325399)

[3 Design details of software module 7](#_Toc485325400)

[3.1 Graphical representation of SyncCrc 7](#_Toc485325401)

[3.2 Data Flow Diagram 7](#_Toc485325402)

[3.2.1 Component level DFD 7](#_Toc485325403)

[3.2.2 Function level DFD 7](#_Toc485325404)

[4 Constant Data Dictionary 8](#_Toc485325405)

[4.1 Program (fixed) Constants 8](#_Toc485325406)

[4.1.1 Embedded Constants 8](#_Toc485325407)

[4.2 Variable Data Dictionary 8](#_Toc485325408)

[4.2.1 User Defined Typedef Definition/Declaration 8](#_Toc485325409)

[4.2.2 User Defined Enumerated Types 9](#_Toc485325410)

[5 Software Component Implementation 10](#_Toc485325411)

[5.1 Sub-Module Functions 10](#_Toc485325412)

[5.1.1 Init: SyncCrcInit0 10](#_Toc485325413)

[5.1.1.1 Design Rationale 10](#_Toc485325414)

[5.1.1.2 Processing 10](#_Toc485325415)

[5.1.1.3 Module Outputs 10](#_Toc485325416)

[5.1.2 Init: SyncCrcInit1 10](#_Toc485325417)

[5.1.2.1 Design Rationale 10](#_Toc485325418)

[5.1.2.2 Processing 10](#_Toc485325419)

[5.1.2.3 Module Outputs 10](#_Toc485325420)

[5.2 Server Runnables 11](#_Toc485325421)

[5.2.1 ResvCrcHwUnit 11](#_Toc485325422)

[5.2.1.1 Design Rationale 11](#_Toc485325423)

[5.2.1.2 (Processing of function)……… 11](#_Toc485325424)

[5.2.1.3 Module Outputs 13](#_Toc485325425)

[5.2.2 CRC API Server Runnables 14](#_Toc485325426)

[5.2.2.1 API Design Rationale 14](#_Toc485325427)

[5.2.2.2 API Processing 14](#_Toc485325428)

[5.3 Interrupt Functions 17](#_Toc485325429)

[5.4 Module Internal (Local) Functions 17](#_Toc485325430)

[5.4.1 RelsCrcHwUnit 17](#_Toc485325431)

[5.4.1.1 Design Rationale 17](#_Toc485325432)

[5.4.1.2 Processing 17](#_Toc485325433)

[5.4.2 NONTRUSTED\_NtWrapS\_SyncCrc\_RelsCrcHwUnit 18](#_Toc485325434)

[5.4.2.1 Design Rationale 18](#_Toc485325435)

[5.4.2.2 Processing 18](#_Toc485325436)

[5.4.3 GetAvlCrcHwUnit 19](#_Toc485325437)

[5.4.3.1 Design Rationale 19](#_Toc485325438)

[5.4.3.2 Processing 19](#_Toc485325439)

[5.4.4 NONTRUSTED\_NtWrapS\_SyncCrc\_GetAvlCrcHwUnit 20](#_Toc485325440)

[5.4.4.1 Design Rationale 20](#_Toc485325441)

[5.4.4.2 Processing 20](#_Toc485325442)

[5.4.5 CrcRegCfg 21](#_Toc485325443)

[5.4.5.1 Design Rationale 21](#_Toc485325444)

[5.4.5.2 Processing 21](#_Toc485325445)

[5.5 GLOBAL Function/Macro Definitions 23](#_Toc485325446)

[6 Known Limitations with Design 24](#_Toc485325447)

[7 UNIT TEST CONSIDERATION 25](#_Toc485325448)

[Appendix A Abbreviations and Acronyms 26](#_Toc485325449)

[Appendix B Glossary 27](#_Toc485325450)

[Appendix C References 28](#_Toc485325451)

# Introduction

## Purpose

This MDD aids in documenting the implementation of CM800A for the synchronous CRC API with the EA4 hardware CRC units.

## Scope

The following definitions are used throughout this document:

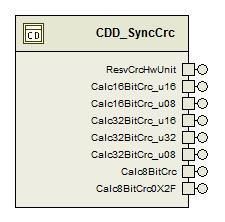
* **Shall**: indicates a mandatory requirement without exception in compliance.
* **Should**: indicates a mandatory requirement; exceptions allowed only with documented justification.
* **May**: indicates an optional action.

# SyncCrc & High-Level Description

Provides an API interface for other BSW and application level software components to calculate a synchronous CRC calculation using the EA4 hardware peripherals.

# Design details of software module

## Graphical representation of SyncCrc



## Data Flow Diagram

### Component level DFD

### Function level DFD

# Constant Data Dictionary

## Program (fixed) Constants

### Embedded Constants

#### Local Constants

Values between the brackets [] are the ranges that the configurable constants could be defined as for a given integration. These values are generated by Configurator before the software build.

|  |  |  |  |
| --- | --- | --- | --- |
| Constant Name | Resolution | Units | Value |
| CRCININVAL8BIT\_CNT\_U08 | Uint8 | Cnt | 0xFF |
| CRCININVAL16BIT\_CNT\_U16 | Uint16 | Cnt | 0xFFFF |
| CRCININVAL32BIT\_CNT\_U32 | Uint32 | Cnt | 0xFFFFFFFF |
| CRCERRVAL\_CNT\_U08 | Uint8 | Cnt | 0 |
| CRCHWRESVCFGRNG\_CNT\_U08 | Uint8 | Cnt | 7 |
| INVLDTASKID\_CNT\_U16 | Uint16 | Cnt | 0xFFFF |
| NROFCRCHWUNIT\_CNT\_U08 | Uint8 | Cnt | [0 – 8]\* |
| NROFACTVCRCHWUNIT\_CNT\_U08 | Uint8 | Cnt | [0 – 8]† |
| ARWRPRENAD\_CNT\_U08 | Uint8 | Cnt | [STD\_OFF – STD\_ON]\*\* |
| CRCOSREF\_CNT\_U08 | Uint8 | Cnt | [0-255]\*\*\* |
| MAXNROFCRCHWUNIT\_CNT\_U08 | Uint8 | Cnt | 8 |

\* Based on the value of “Device Type” as defined in Configurator.  
\*\* Based on the value of “Autosar Wrapper Enable” as defined in Configurator.  
\*\*\*Based on the value of “Crc Os Application Reference” as defined in Configurator.   
† The number is calculated by NROFCRCHWUNIT\_CNT\_U08 – “Reserved CRC Hardware Units” as defined in Configurator.

## Variable Data Dictionary

The following type definitions are found in the private header of this component.

### User Defined Typedef Definition/Declaration

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Typedef Name | Element Name | User Defined Type | Legal Range  (min) | Legal Range  (max) |
| NtCrcIdRec | CrcHwIdx | Uint8 | 0 | 255 |
| NtResvCallRec | ResvCall | Boolean | FALSE | TRUE |

### User Defined Enumerated Types

|  |  |  |
| --- | --- | --- |
| Enum Name | Element Name | Value |
| CrcDataAcsWidth1 | CRCDATAACSWIDTH\_32BIT | 0 |
|  | CRCDATAACSWIDTH\_16BIT | 1 |
|  | CRCDATAACSWIDTH\_8BIT | 2 |
| CrcAlg1 | CRCALG\_32BITETH | 0 |
|  | CRCALG\_16BIT | 1 |
|  | CRCALG\_8BIT | 2 |
|  | CRCALG\_8BITH2F | 3 |

# Software Component Implementation

## Sub-Module Functions

## Init: SyncCrcInit0

## Design Rationale

This function initializes the PIM with the proper status for the CRC hardware units for use by the application components. This function is defined in the CDD\_SyncCrcNonRte.c file as it shall be called prior to the RTE Init functions.

## Processing



## Module Outputs

None

## Init: SyncCrcInit1

## Design Rationale

Stub function for mapping the server runnable functions within a memory region.

## Processing

None

## Module Outputs

None

## Server Runnables

## ResvCrcHwUnit

## Design Rationale

This function allows the caller to reserve a single CRC hardware unit until it is released. This will allow features such as the DMA to perform a CRC calculation over a large portion of data and does not need to permanently reserve a CRC hardware unit. This function can be called from within or outside a task. This server runnables must be defined with “can be invoked concurrently” enabled in Developer.

## (Processing of function)………

##### Top Level Logic



##### Reserve CRC



##### Release CRC



## Module Outputs

None

## CRC API Server Runnables

## API Design Rationale

The following flow chart is applicable to all CRC API functions in this document. However, the highlighted green squares vary by the API. These differences will be pointed out in each sub-function section. All API client calls must be called from within a task. These server runnables must be defined with “can be invoked concurrently” enabled in Developer.

## API Processing





##### Calc8BitCrc\_Oper

CRCREG\_REC[CrcHwIdx\_Cnt\_T\_u08].Reg\_rec->Ctrl\_rec.Bit\_b08.ISZ = CRCDATAACSWIDTH\_8BIT;

CRCREG\_REC[CrcHwIdx\_Cnt\_T\_u08].Reg\_rec->Ctrl\_rec.Bit\_b08.POL = CRCALG\_8BIT;

CRCREG\_REC[CrcHwIdx\_Cnt\_T\_u08].Reg\_rec->Outp\_u32 = CRCININVAL8BIT\_CNT\_U16;

##### Calc8BitCrc0X2F\_Oper

CRCREG\_REC[CrcHwIdx\_Cnt\_T\_u08].Reg\_rec->Ctrl\_rec.Bit\_b08.ISZ = CRCDATAACSWIDTH\_8BIT;

CRCREG\_REC[CrcHwIdx\_Cnt\_T\_u08].Reg\_rec->Ctrl\_rec.Bit\_b08.POL = CRCALG\_8BITH2F;

CRCREG\_REC[CrcHwIdx\_Cnt\_T\_u08].Reg\_rec->Outp\_u32 = CRCININVAL8BIT\_CNT\_U16;

##### Calc16BitCrc\_u08\_Oper

CRCREG\_REC[CrcHwIdx\_Cnt\_T\_u08].Reg\_rec->Ctrl\_rec.Bit\_b08.ISZ = CRCDATAACSWIDTH;

CRCREG\_REC[CrcHwIdx\_Cnt\_T\_u08].Reg\_rec->Ctrl\_rec.Bit\_b08.POL = CRCALG\_16BIT;

CRCREG\_REC[CrcHwIdx\_Cnt\_T\_u08].Reg\_rec->Outp\_u32 = CRCININVAL16BIT\_CNT\_U16;

##### Calc16BitCrc\_u16\_Oper

CRCREG\_REC[CrcHwIdx\_Cnt\_T\_u08].Reg\_rec->Ctrl\_rec.Bit\_b08.ISZ = CRCDATAACSWIDTH\_16BIT;

CRCREG\_REC[CrcHwIdx\_Cnt\_T\_u08].Reg\_rec->Ctrl\_rec.Bit\_b08.POL = CRCALG\_16BIT;

CRCREG\_REC[CrcHwIdx\_Cnt\_T\_u08].Reg\_rec->Outp\_u32 = CRCININVAL16BIT\_CNT\_U16;

##### Calc32BitCrc\_u08\_Oper

CRCREG\_REC[CrcHwIdx\_Cnt\_T\_u08].Reg\_rec->Ctrl\_rec.Bit\_b08.ISZ = CRCDATAACSWIDTH;

CRCREG\_REC[CrcHwIdx\_Cnt\_T\_u08].Reg\_rec->Ctrl\_rec.Bit\_b08.POL = CRCALG\_32BITETH;

CRCREG\_REC[CrcHwIdx\_Cnt\_T\_u08].Reg\_rec->Outp\_u32 = CRCININVAL32BIT\_CNT\_U16;

##### Calc32BitCrc\_u16\_Oper

CRCREG\_REC[CrcHwIdx\_Cnt\_T\_u08].Reg\_rec->Ctrl\_rec.Bit\_b08.ISZ = CRCDATAACSWIDTH\_16BIT;

CRCREG\_REC[CrcHwIdx\_Cnt\_T\_u08].Reg\_rec->Ctrl\_rec.Bit\_b08.POL = CRCALG\_32BITETH;

CRCREG\_REC[CrcHwIdx\_Cnt\_T\_u08].Reg\_rec->Outp\_u32 = CRCININVAL32BIT\_CNT\_U16;

##### Calc32BitCrc\_u32\_Oper

CRCREG\_REC[CrcHwIdx\_Cnt\_T\_u08].Reg\_rec->Ctrl\_rec.Bit\_b08.ISZ = CRCDATAACSWIDTH\_32BIT;

CRCREG\_REC[CrcHwIdx\_Cnt\_T\_u08].Reg\_rec->Ctrl\_rec.Bit\_b08.POL = CRCALG\_32BITETH;

CRCREG\_REC[CrcHwIdx\_Cnt\_T\_u08].Reg\_rec->Outp\_u32 = CRCININVAL32BIT\_CNT\_U16;

## Interrupt Functions

None

## Module Internal (Local) Functions

## RelsCrcHwUnit

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | RelsCrcHwUnit | Type | Min | Max |
| **Arguments Passed** | CrcHwIdx\_Cnt\_T\_u08 | Uint8 | 0 | 8 |
| **Return Value** | N/A |  |  |  |

## Design Rationale

To minimize time in Exclusive areas, the Enter and Exit calls were placed within this function. All API server runnables that use this function are defined in Developer to have access to the exclusive area.

## Processing



## NONTRUSTED\_NtWrapS\_SyncCrc\_RelsCrcHwUnit

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | NONTRUSTED\_NtWrapS\_SyncCrc\_RelsCrcHwUnit | Type | Min | Max |
| **Arguments Passed** | FunctionIndex | Uint16 | 0 | 65535 |
|  | FunctionParams | Void\* | 0 | 2^32-1 |
| **Return Value** | N/A |  |  |  |

## Design Rationale

Function is required to prevent MPU violations when the API modifies the per-instance-memory used by all API functions.

## Processing



## GetAvlCrcHwUnit

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | GetAvlCrcHwUnit | Type | Min | Max |
| **Arguments Passed** | N/A |  |  |  |
| **Return Value** | N/A |  |  |  |

## Design Rationale

## Processing



## NONTRUSTED\_NtWrapS\_SyncCrc\_GetAvlCrcHwUnit

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | NONTRUSTED\_NtWrapS\_SyncCrc\_GetAvlCrcHwUnit | Type | Min | Max |
| **Arguments Passed** | FunctionIndex | Uint16 | 0 | 65535 |
|  | FunctionParams | Void\* | 0 | 2^32-1 |
| **Return Value** | N/A |  |  |  |

## Design Rationale

Function is required to prevent MPU violations when the API modifies the per-instance-memory used by all API functions.

## Processing



## CrcRegCfg

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function Name** | CrcRegCfg | Type | Min | Max |
| **Arguments Passed** | CrcHwIdx\_Arg | Uint8 | 0 | 255 |
|  | CrcCfg\_Arg | CrcHwResvCfg1 | 0 | 6 |
|  | StrtVal\_Arg | Uint32 | 0 | 4294967295 |
| **Return Value** | N/A |  |  |  |

## Design Rationale

Function created to reduce the complexity of the ResvCrcHwUnit\_Oper function.

## Processing



## GLOBAL Function/Macro Definitions

None

# Known Limitations with Design

1. API client calls, except ResvCrcHwUnit, must be called from within a task.
2. To meet design and coding standards, ‘For’ loops called out by the FDD were implemented with ‘While’ loops to break out of the loops without using the ‘break’ keyword.
3. The GetTaskID function call provided by the OS between different versions have different arguments. One version, the argument is a uint16, where another is a pointer to an enumerated type. The design simply casts the function call to a uint16. There are no expectations that there are going to be more than 65,535 tasks, so the uint16 width should provide adequate room for growth.
4. In the function ResvCrcHwUnit, the else condition in the main logic path is considered dead code by Polyspace. This is because it is limiting the function to the bounds of the input of the Mod and configuration argument. However, the code is present to provide error handling in the event of an invalid input being passed into the function.

# UNIT TEST CONSIDERATION

The constants NROFACTVCRCHWUNIT\_CNT\_U08, ARWRPRENAD\_CNT\_U08, and CRCOSREF\_CNT\_U08, and NROFCRCHWUNIT\_CNT\_U08 shall be tested to their full range as defined in the constant section.

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) | 1.0 |
| 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 |