Integration Manual – Dma

Table of Contents

[1 Dependencies 2](#_Toc384642160)

[1.1 SWCs 2](#_Toc384642161)

[1.2 Global Functions(Non RTE) to be provided to Integration Project 2](#_Toc384642162)

[2 Configuration 3](#_Toc384642163)

[2.1 Build Time Config 3](#_Toc384642164)

[2.2 Configuration Files to be provided by Integration Project 3](#_Toc384642165)

[2.2.1 Da Vinci Parameter Configuration Changes 3](#_Toc384642166)

[2.2.2 DaVinci Interrupt Configuration Changes 3](#_Toc384642167)

[2.2.3 Manual Configuration Changes 3](#_Toc384642168)

[3 Integration 4](#_Toc384642169)

[3.1 Required Global Data Inputs 4](#_Toc384642170)

[3.2 Required Global Data Outputs 4](#_Toc384642171)

[3.3 Specific Include Path present 4](#_Toc384642172)

[4 Runnable Scheduling 5](#_Toc384642173)

[5 Memory Mapping 6](#_Toc384642174)

[5.1 Mapping 6](#_Toc384642175)

[5.2 Usage 6](#_Toc384642176)

[5.3 Non RTE NvM Blocks 6](#_Toc384642177)

[5.4 RTE NvM Blocks 6](#_Toc384642178)

[6 Compiler Settings 6](#_Toc384642179)

[6.1 Preprocessor MACRO 6](#_Toc384642180)

[6.2 Optimization Settings 6](#_Toc384642181)

[7 Architectural Concerns 7](#_Toc384642182)

[7.1 Overall DMA Architecture 8](#_Toc384642183)

[7.2 Affected Modules 9](#_Toc384642184)

[7.2.1 IoHwAbstractionUsr 9](#_Toc384642185)

[7.2.2 Adc 9](#_Toc384642186)

[7.2.3 MtrCtrl\_Irq 9](#_Toc384642187)

[7.2.4 uDiag 10](#_Toc384642188)

[8 Revision Control Log 12](#_Toc384642189)

# Dependencies

## SWCs

|  |  |
| --- | --- |
| Module | Required Feature |
| Adc | Using ADC triggers for DMA transfers as well as ADC conversion group sizes to calculate buffer sizes. Only required if ADC DMA channels are enabled. Requires version FDD33C\_008\_and\_FDD33E\_002.3 or later. |
| SpiNxt | Using SPI Rx buffer full triggers for DMA transfers as well as SPI transfer group sizes to calculate DMA buffer sizes. Only required if SPI DMA channels are used. Requires version ASR038\_2.3.0\_8 or later. |
| uDiag | If the FlsTst channels are enabled, uDiag is responsible for initializing and enabling these blocks. Requires version FDD32B\_TMS570\_uDiag\_000.24 or later. |
| ePWM | Using NHET triggers for DMA transfers as well as NHET program addresses for destination addresses. Only required if NHET DMA channels are enabled. Requires version FDD34B\_EPWM\_NHETSENT\_005.0 or later. |
| TMS570\_Startup | Note that the DMA parity functionality requires version FDD32B\_TMS570\_Startup\_000.19 or later in the bootloader. Note that DMA MPU startup test functionality requires FDD32B\_TMS570\_startup\_000.21 or later in the application. |
| DiagMgr | Error reporting mechanism required if either of the slow SPI or ADC channels are enabled. |

Note : Referencing the external components should be avoided in most cases. Only in unavoidable circumstance external components should be referred. Developer should track the references.

## Global Functions(Non RTE) to be provided to Integration Project

Dma\_Init

Dma\_SlowADCGroupValidity

Dma\_InvalidateSlowADCGroup

Dma\_SetupMtrCtrlGroups

Dma\_SetupFlsTstBlock

Dma\_EnableFlsTstBlock

Dma\_DisableFlsTstBlock

# Configuration

## Build Time Config

|  |  |  |
| --- | --- | --- |
| Modules | Notes |  |
| None |  |  |

## Configuration Files to be provided by Integration Project

Dma\_Cfg.h should be manually generated based on the template provided in the Tools folder.

### Da Vinci Parameter Configuration Changes

|  |  |  |
| --- | --- | --- |
| Parameter | Notes | SWC |
| None |  |  |

### DaVinci Interrupt Configuration Changes

|  |  |  |  |
| --- | --- | --- | --- |
| ISR Name | VIM # | Priority Dependency | Notes |
| MtrCtrl\_Irq | 40 |  | If using the Fast ADC channel, update this to use IRQ40. |

### Manual Configuration Changes

|  |  |  |
| --- | --- | --- |
| Constant | Notes | SWC |
| D\_DMAFLSTSTENABLED\_CNT\_ENUM | STD\_ON or STD\_OFF – used to enable the FlsTst DMA channels. | Dma\_Cfg.h |
| D\_FASTSPIGROUPENABLED\_CNT\_ENUM | STD\_ON or STD\_OFF – used to enable the MtrCtrl ISR SPI DMA channels. | Dma\_Cfg.h |
| D\_FASTADCGROUPENABLED\_CNT\_ENUM | STD\_ON or STD\_OFF – used to enable the MtrCtrl ISR ADC DMA channel. | Dma\_Cfg.h |
| D\_FASTPWMGROUPENABLED\_CNT\_ENUM | STD\_ON or STD\_OFF – used to enable the MtrCtrl ISR NHET and ePWM DMA channels. | Dma\_Cfg.h |
| D\_SLOWADCGROUPENABLED\_CNT\_ENUM | STD\_ON or STD\_OFF – used to enable the 2ms ADC DMA channel. | Dma\_Cfg.h |
| DMA\_PARITY\_ENABLE | STD\_ON or STD\_OFF – used to enable the parity check. This should be enabled on all programs – note that there is a required change in the bootloader to accommodate this. | appinit\_cfg.h |
| DMA\_REPORTERRORSTATUS | Set to NxtrDiagMgrX\_ReportNTCStatus, where X is the application calling the group validity check functions. | Dma\_Cfg.h |
| DMA\_NHET1PROGHFILE | Header file containing the definition of HET\_PRD\_BUF1\_0 should be used | Dma\_Cfg.h |

# Integration

## Required Global Data Inputs

None

## Required Global Data Outputs

DMAData\_G\_str

Dma\_DmaRstFail\_Cnt\_G\_lgc

## Specific Include Path present

Yes

# Runnable Scheduling

This section specifies the required runnable scheduling.

|  |  |  |
| --- | --- | --- |
| Init | Scheduling Requirements | Trigger |
| Dma\_Init | Must be scheduled before FlsTst\_Init (uDiag) | Init |
| Dma\_SetupMtrCtrlGroups | Must be scheduled after Dma\_Init. Should be scheduled after other used peripherals are enabled (SPI, ADC, ePWM, NHET, CRC). Should be scheduled before the MtrCtrl ISR is enabled. Only required if any of the SPI, ADC, or PWM groups are enabled. | Init |

|  |  |  |
| --- | --- | --- |
| Runnable | Scheduling Requirements | Trigger |
| Dma\_SlowADCGroupValidity | Must be scheduled before the data is collected from the 2ms ADC buffer. Only required if the slow ADC channel is used. | 2ms |
| Dma\_InvalidateSlowADCGroup | Must be scheduled after the data is collected from the 2ms ADC buffer. Only required if the slow ADC channel is used. | 2ms |

# Memory Mapping

## Mapping

|  |  |  |
| --- | --- | --- |
| Memory Section | Contents | Notes |
| DMA\_START\_SEC\_VAR\_CLEARED\_UNSPECIFIED | All global DMA variables in RAM (source and destination). | This section can be mapped to an application only available in supervisor mode. In this case, Dma\_InvalidateSlowADCGroup will need to be accessed with a trusted function call. |
| DMA\_START\_SEC\_VAR\_CLEARED\_BOOLEAN | DMA reset fail flag |  |

\* Each …START\_SEC… constant is terminated by a …STOP\_SEC… constant as specified in the AUTOSAR Memory Mapping requirements.

## Usage

|  |  |  |
| --- | --- | --- |
| Feature | RAM | ROM |
| None |  |  |

Table 1: ARM Cortex R4 Memory Usage

## Non RTE NvM Blocks

|  |
| --- |
| Block Name |
| None |

Note : Size of the NVM block if configured in developer

## RTE NvM Blocks

|  |
| --- |
| Block Name |
| None |

Note : Size of the NVM block if configured in developer

# Compiler Settings

## Preprocessor MACRO

None

## Optimization Settings

Dma.c should use the same optimization settings as other functions called from the MtrCtrl ISR (generally, optimize = 3, optimize for speed = 5).

# Architectural Concerns

DMA is a module that largely impacts the system architecture, particularly at the MtrCtrl ISR time domain. As such, this section includes some considerations for integrating DMA into an integration project for the first time.

The examples show integration of every DMA channel; if only a subset is used, not all of these examples may apply.

## Overall DMA Architecture

The following diagram is taken from FDD52, and shows how the DMA channels fit into the MtrCtrl ISR process:



## Affected Modules

### IoHwAbstractionUsr

Historically, ADC conversions were triggered by the 2ms loop through the MtrCtrl ISR (the MtrCtrl ISR would perform a software trigger to the “slow” ADC groups directly). With DMA, the “slow” ADC group conversion is now triggered by ePWM SOCB, driven by ePWM4 using CMPB.

In this new architecture, IoHwAbstractionUsr must perform the following:

1. Before CDDInterface\_Per1, an IoHwAbstractionUsr runnable must update the value of ePWM4 CMPB (refer to the ePWM component for the interface). It must also clear the ADC1 G2 Conversions Ended bit as follows:

adcREG1->GxSR[2u] = 1u;

1. After the conversion completes, an IoHwAbstractionUsr runnable must update the value of ePWM4 CMPB to disable the ADC conversion (refer to the ePWM component).
2. After the conversion completes but before the ADC values are required to be used, an IoHwAbstractionUsr runnable must read and process the data from the DMA destination buffer. The buffer can optionally be cleared at this point. Note that this can be done from the same runnable as disabling the ADC conversion.
3. An IoHwAbstraction runnable can be called from the MtrCtrl ISR to capture any data from the “fast” ADC conversion needed by the 2ms loop (ex. a redundant read of switched battery voltage). The DMA buffer should be copied to a standard CDDInterface global variable and read by CDDInterface\_Per1 along with the rest of the MtrCtrl ISR data used by the 2ms loop.

The final IoHwAbstractionUsr component could then look something like this:

IoHwAb\_CaptureADC – reads Switched Battery Voltage from ADC2 DMA buffer and writes it to a CDD variable.

IoHwAb\_StartADC – enables the ADC1 conversion and clears the ADC1 G2 Conversions Ended bit. Scheduled in the 2ms loop just before CDDInterface\_Per1.

IoHwAb\_ReadADC – checks that the ADC1 conversion completed and disables the ADC1 conversion. Reads the values from the ADC1 DMA buffer as well as the redundant Switched Battery Voltage from CDDInterface. Converts all of these signals to engineering units and outputs them to the rest of the RTE. Scheduled in the 2ms loop at the end of the forward path. See ePWM component for minimum elapsed time between enabling and disabling the ADC conversion via the write of the ePWM4 CMPB value.

### Adc

The intended DMA architecture uses two ADC conversion groups: ADC1 Group 2 (“slow” ADC) and ADC2 Group 1 (“fast” ADC). Historically, ADC2 Event Group was used instead of ADC2 Group 1 – this may mean that integrating DMA will require a program to switch the ADC group being used.

### MtrCtrl\_Irq

If this is the first integration of DMA into a program, the existing ADC triggering mechanism will need to be removed and replaced with the method described above. This will largely involve calling an IoHwAbstractionUsr runnable. In addition, the ADC Conversion Complete flag will still need to be cleared (though it may have changed, see above), as well as the DMA interrupt that calls the MtrCtrl ISR. It can be cleared as follows:

DMACTRLREG->BTCFLAG = (1 << 3)

### uDiag

The following entries should be added to the Static Register Check list. Note that these values are subject to change depending on which channels are enabled.

|  |  |  |
| --- | --- | --- |
| Name | Address | Expected Value |
| DMA\_PAR0 \* | 0xFFFFF094 | 0x44440440 |
| DMA\_PAR1 \* | 0xFFFFF098 | 0x04004000 |
| DMA\_DMAPCR | 0xFFFFF1A8 | 0x0000000A |
| DMA\_DMAMPCTRL \* | 0xFFFFF1B0 | 0x0F090909 |
| DMA\_DMAMPR0S | 0xFFFFF1B8 | 0xFF0A0000 |
| DMA\_DMAMPR0E | 0xFFFFF1BC | 0xFF473FFF |
| DMA\_DMAMPR1S | 0xFFFFF1C0 | 0xFCF78C00 |
| DMA\_DMAMPR1E | 0xFFFFF1C4 | 0xFE0001FF |
| DMA\_DMAMPR3S | 0xFFFFF1D0 | 0x00200000 |
| DMA\_DMAMPR3E | 0xFFFFF1D4 | 0xFFFFFFFF |
| DMA\_CP00\_IDADDR \* | 0xFFF80004 | 0xFE000068 |
| DMA\_CP00\_ITCOUNT \* | 0xFFF80008 | 0x00010001 |
| DMA\_CP00\_CHCTRL \* | 0xFFF80010 | 0x0000F000 |
| DMA\_CP01\_IDADDR \* | 0xFFF80024 | 0xFE000060 |
| DMA\_CP01\_CHCTRL \* | 0xFFF80030 | 0x0000F108 |
| DMA\_CP02\_ISADDR \* | 0xFFF80040 | 0xFF0C0202 |
| DMA\_CP02\_ITCOUNT \* | 0xFFF80048 | 0x00010003 |
| DMA\_CP02\_CHCTRL \* | 0xFFF80050 | 0x0000511F |
| DMA\_CP02\_EIOFF \* | 0xFFF80054 | 0x00020004 |
| DMA\_CP03\_ISADDR \* | 0xFFF80060 | 0xFF3A0002 |
| DMA\_CP03\_ITCOUNT \* | 0xFFF80068 | 0x00010004 |
| DMA\_CP03\_CHCTRL \* | 0xFFF80070 | 0x0000511F |
| DMA\_CP03\_EIOFF \* | 0xFFF80074 | 0x00020004 |
| DMA\_CP05\_IDADDR \* | 0xFFF800A4 | 0xFF4600E8 |
| DMA\_CP05\_ITCOUNT \* | 0xFFF800A8 | 0x00010001 |
| DMA\_CP05\_CHCTRL \* | 0xFFF800B0 | 0x0007A101 |
| DMA\_CP06\_IDADDR \* | 0xFFF800C4 | 0xFCF78C10 |
| DMA\_CP06\_CHCTRL \* | 0xFFF800D0 | 0x0000511F |
| DMA\_CP09\_ISADDR \* | 0xFFF80120 | 0xFF3E00F2 |
| DMA\_CP09\_ITCOUNT \* | 0xFFF80128 | 0x00010004 |
| DMA\_CP09\_CHCTRL \* | 0xFFF80130 | 0x0000511F |
| DMA\_CP09\_EIOFF \* | 0xFFF80134 | 0x00020004 |
| DMA\_CP12\_ISADDR \* | 0xFFF80180 | 0xFF0A0202 |
| DMA\_CP12\_ITCOUNT \* | 0xFFF80188 | 0x00010003 |
| DMA\_CP12\_CHCTRL \* | 0xFFF80190 | 0x0000511F |
| DMA\_CP12\_EIOFF \* | 0xFFF80194 | 0x00020004 |

**\*** These values are dependent on whether certain channels are enabled in the configuration.

The following entries should be defined as link-time determined.

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Address | Expected Value | Offset |
| DMA\_DMAMPR2S \* | 0xFFFFF1C8 | DMAData\_G\_str | 0 |
| DMA\_DMAMPR2E \* | 0xFFFFF1CC | DMAData\_G\_str | sizeof(DMAData\_G\_str) – 1U |
| DMA\_CP02\_IDADDR \* | 0xFFF80044 | DMAData\_G\_str.FastSPI\_Cnt\_u16[0] | 0 |
| DMA\_CP03\_IDADDR \* | 0xFFF80064 | DMAData\_G\_str.FastADC\_Cnt\_u16[0] | 0 |
| DMA\_CP05\_ISADDR \* | 0xFFF800A0 | DMAData\_G\_str.PWMPeriod\_Cnt\_u32 | 0 |
| DMA\_CP06\_ISADDR \* | 0xFFF800C0 | DMAData\_G\_str.PWMCmp\_Cnt\_u16[0][0] | 0 |
| DMA\_CP09\_IDADDR \* | 0xFFF80124 | DMAData\_G\_str.SlowADC\_Cnt\_u16[0] | 0 |
| DMA\_CP12\_IDADDR \* | 0xFFF80184 | DMAData\_G\_str.SlowSPI\_Cnt\_u16[0] | 0 |

**\*** These values are dependent on whether certain channels are enabled in the configuration.

# Revision Control Log

|  |  |  |  |
| --- | --- | --- | --- |
| **Rev #** | **Change Description** | **Date** | **Author** |
| 1 | Initial version | 10-Apr-14 | OT |
| 2 | Updates for safety mechanisms, data integrity mechanisms (FDD 52 v001) | 30-Apr-14 | OT |
| 3 | Removed slow SPI integrity scheme functions (FDD 52 v002) | 02-May-14 | OT |
| 4 | Added DMA Reset Failed global output per ES-52 v004; updated section 7.2.\* per ES 52 v004 and latest updates to associated components | 31-Jan-2015 | KMC |
| 5 | Anomaly fix EA3#2825 -Made the NHET header file configurable | 24-Sep-14 | JK |