# Module -- TMS570 Startup - errata\_SSWF021\_45

# High-Level Description

This module outlines errata functions of the TMS570 for the PLL Slip. Please refer **Hercules PLL Advisory SSWF021#45 Workaround** (SPNA233.pdf) for more details

Refer Section 9 for the deviation from the errata

# Figures

## Diagram – Function Data Sharing

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

No Shared Data

# Variable Data Dictionary

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

|  |  |  |
| --- | --- | --- |
| Module Inputs | Module Outputs | |
| <None> | |  |
|  | |  |

## 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.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Variable Name | Resolution | Legal Range  (min) | Legal Range  (max) | Software Segment |
| <None> |  |  |  |  |
|  |  |  |  |  |

### User defined typedef definition/declaration

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

|  |  |  |
| --- | --- | --- |
| Typedef Name | Element Name | Value |
|  |  |  |

# 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.

|  |
| --- |
| Constant Name |
| <None> |
|  |

## Program(fixed) Constants

### Embedded Constants

#### Local

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

#### 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 |
|  |
|  |

### 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 |
| None |  |  |  |

# Functions/Macros used by the Sub-Modules

## Library Functions / Macros

The library and functions / Macros that are called by the various sub modules are identified below,

1. <None>

## Data Hiding Functions

1. <None>

## Global Functions/Macros Defined by this Module

### Global Function #1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Function Name** | (Exact name used) | Type | Min | Max | UTP Tol. |
| **Arguments Passed** | (if none, write None) |  |  |  |  |
|  | (Insert more rows for additional passed arguments) |  |  |  |  |
| **Return Value** | (if no value returned, write N/A) |  |  |  |  |

#### Description

(Place flowchart/design for local function)

## Local Functions/Macros Used by this MDD only

### Local Function

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Function Name** |  | Type | Min | Max | UTP Tol. |
| **Arguments Passed** | None |  |  |  |  |
|  |  |  |  |  |  |
| **Return Value** | N/A |  |  |  |  |

#### Description

# Software Module Implementation

## Runtime Environment (RTE) Initial Values

This section lists the initial values of data written by this module but controlled by the RTE. After RTE initialization, the data in this table will contain these values.

|  |  |
| --- | --- |
| Data | Value |
| <None> |  |

## Initialization Functions

### Init: \_c\_int00 / Startup

#### Design Rationale

##### TI Recommended Initialization

##### Failed Initialization Diagnostic Strategy

##### Software Initiated Resets

##### Errata Processing

#### Processing

## Periodic Functions

None

## Fault Recovery Functions

None

## Shutdown Functions

None

## Interrupt Functions

None

## Serial Communication Functions

None

# 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 | Calling Frequency | System State(s) in which the function is called |
| \_c\_int00() | power on and during a reset | N/A |

## 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 |
| \_c\_int00() |  |

## Local Functions

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

|  |  |
| --- | --- |
| Name of Sub Module | Software Segment |
|  |  |
|  |  |

# Known Issues / Limitations With Design

There is a deviation in the code implemented from what is suggested in the errata

In the errata the clock control register is set in a single statement

/\* save CLKCNTL, then set VCLK = HCLK, enable peripherals \*/

clkCntlSav = systemREG1->CLKCNTL;

systemREG1->CLKCNTL = SYS\_CLKCNTRL\_PENA;

However as per the Technical reference manual the VCLKR should be set only after VCLK2R in two

different steps. Hence the deviation in the code from the errata



/\* save CLKCNTL, then set VCLK = HCLK, enable peripherals \*/

clkCntlSav\_Cnt\_T\_u32 = (systemREG1->VCLK2R << 24U) | (systemREG1->VCLKR << 16U) | (systemREG1->PENA << 8U);

systemREG1->VCLK2R = 0U;

systemREG1->VCLKR = 0U;

systemREG1->PENA = 1U;

# Revision Control Log

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Item #** | **Rev #** | **Change Description** | **Date** | **Author Initials** |
| 1 | 1 | Initial creation | 06/27/16 | AJM |
| 2 | 2 | Removed the parameter for the number of retries | 08/24/16 | AJM |