example_emif_sdram.c

This example code configures EMIF module to write and execute from SDRAM

Step 1:

Create a new project.

Navigate: -> File -> New -> Project

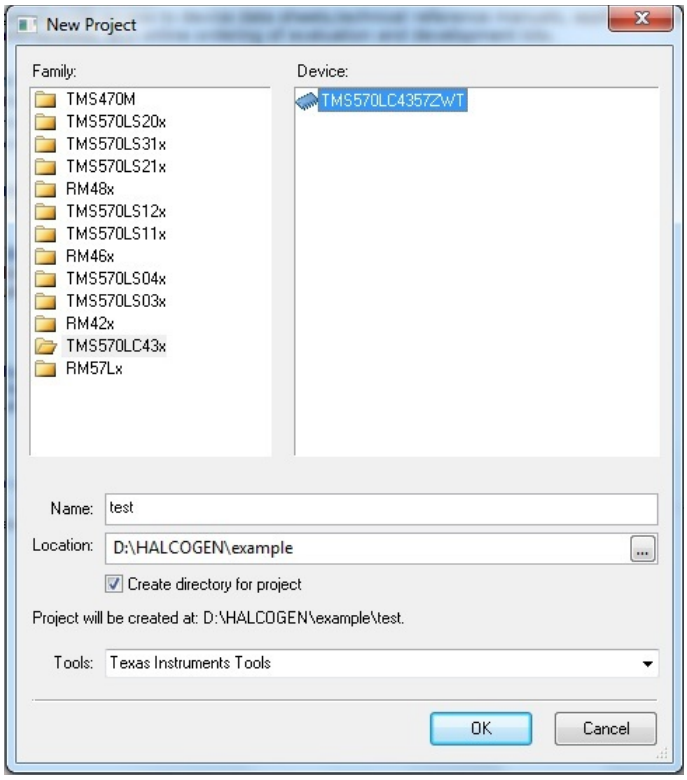

Figure: Create a new Project

Step 2:

Configure driver code generation:

Navigate: -> TMS570LCxx /RM5x -> Enable Drivers

- Enable EMIF driver
- Enbale GIO driver
- Disable others

Step 3:

Configure pinmux:

- Enable all EMIF pins except EMIF_RNW
- Enable all EMIF outputs including EMIF_CLK

Navigate PINMUX->Pin Muxing

Enable checkbox EMIF and disable EMIF_RNW at ball D17

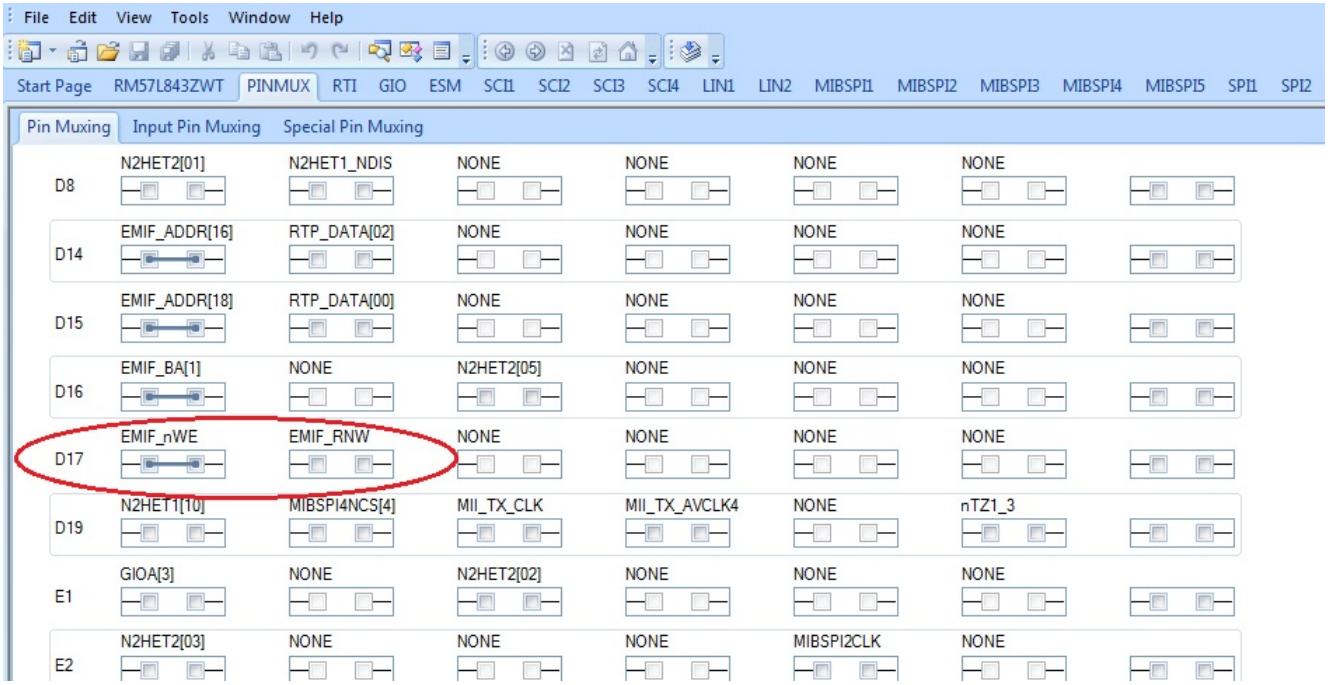

Figure: Pinmux configuration

Navigate PINMUX->Special Pin Muxing

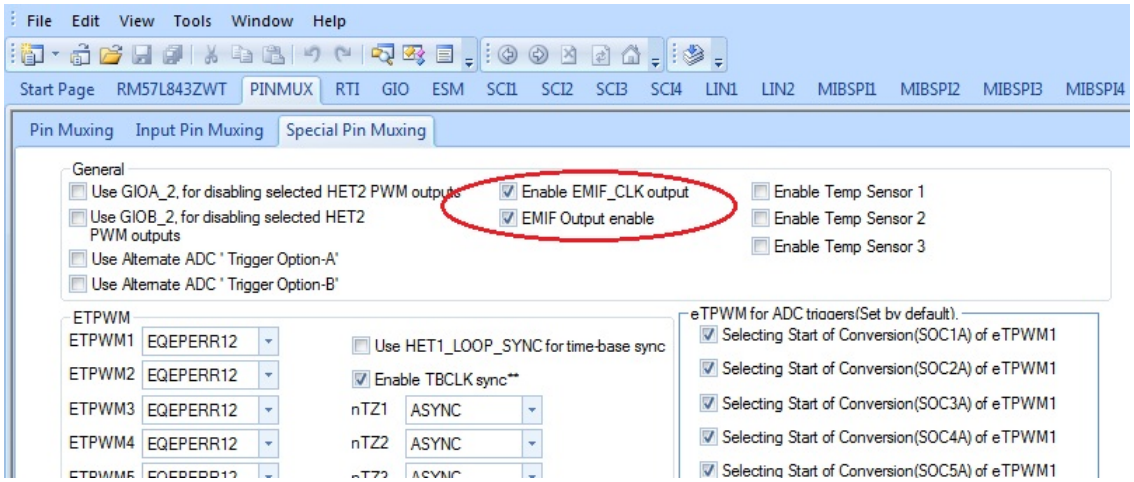

Figure: Special Pinmux configuration

Step 4:

Configure EMIF

Navigate: -> EMIF -> EMIF General

- Enable EMIF SDRAM
- Disable others

Navigate: -> EMIF -> EMIF SDRAM

- Fill in the timing parameters

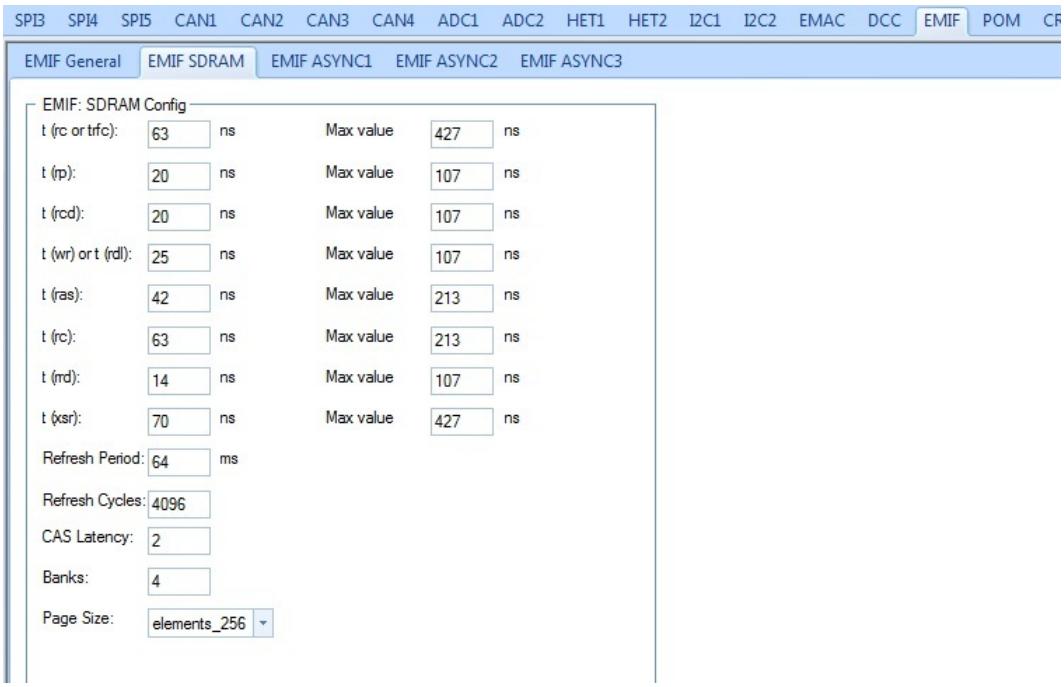

Figure: Configuration for ISSI IS42S16400F

Step 5:

Generate code

Navigate: -> File -> Generate Code

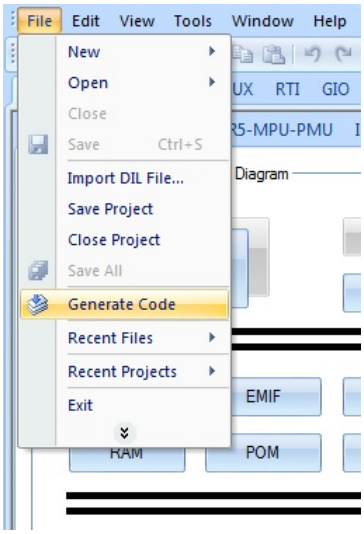

Figure: Generate Code

Step 6:

Modify the linker command file

- Add the follwing under MEMORY
  - SDRAM (RWX) : origin=0x80000000 length=0200000000
- Add the follwing under SECTIONS
  - .blinky_section : RUN = SDRAM, LOAD = FLASH0 | FLASH1 LOAD_START(BlinkyLoadStart), LOAD_END(BlinkyLoadEnd), LOAD_SIZE (BlinkySize), RUN_START(BlinkyStartAddr), RUN_END(BlinkyEndAddr)

```
/*----------------------------------------------------------------------------*/
/* Linker Settings                                                            */

--retain="*(.intvecs)"

/* USER CODE BEGIN (1) */
--retain="*(.blinky_section)"
/* USER CODE END */

/*----------------------------------------------------------------------------*/
/* Memory Map                                                                 */

MEMORY
{
/* USER CODE BEGIN (2) */
/* USER CODE END */
    VECTORS (X)  : origin=0x00000000 length=0x00000020
    FLASH0  (RX) : origin=0x00000020 length=0x001FFFE0
    FLASH1  (RX) : origin=0x00200000 length=0x00200000
    STACKS  (RW) : origin=0x08000000 length=0x00001500
    RAM     (RW) : origin=0x08001500 length=0x0007EB00

/* USER CODE BEGIN (3) */
    SDRAM  (RWX) : origin=0x80000000 length=0200000000
/* USER CODE END */
}

/* USER CODE BEGIN (4) */
/* USER CODE END */

/*----------------------------------------------------------------------------*/
/* Section Configuration                                                      */

SECTIONS
{
/* USER CODE BEGIN (5) */
/* USER CODE END */
    .intvecs : {} > VECTORS
    .text   align(8) : {} > FLASH0 | FLASH1
    .const  align(8) : {} > FLASH0 | FLASH1
    .cinit  align(8) : {} > FLASH0 | FLASH1
    .pinit  align(8) : {} > FLASH0 | FLASH1
    .bss     : {} > RAM
    .data    : {} > RAM
    .sysmem  : {} > RAM


/* USER CODE BEGIN (6) */
    .blinky_section :  RUN = SDRAM, LOAD = FLASH0 | FLASH1
                       LOAD_START(BlinkyLoadStart), LOAD_END(BlinkyLoadEnd),  LOAD_SIZE(BlinkySize),
                       RUN_START(BlinkyStartAddr ), RUN_END(BlinkyEndAddr )
/* USER CODE END */
}

```

Figure: Linker command file

Step 7:

Copy the source code below into your sys_main.c (or) replace sys_main.c with this file.

The example file **example_mibspi_loopback.c** can also be found in the examples folder: ../HALCoGen/examples

Note

HALCoGen generates an empty main function in sys_main.c.

```
/*
* Copyright (C) 2009-2015 Texas Instruments Incorporated - www.ti.com
*
*
*  Redistribution and use in source and binary forms, with or without
*  modification, are permitted provided that the following conditions
*  are met:
*
*    Redistributions of source code must retain the above copyright
*    notice, this list of conditions and the following disclaimer.
*
*    Redistributions in binary form must reproduce the above copyright
*    notice, this list of conditions and the following disclaimer in the
*    documentation and/or other materials provided with the
*    distribution.
*
*    Neither the name of Texas Instruments Incorporated nor the names of
*    its contributors may be used to endorse or promote products derived
*    from this software without specific prior written permission.
*
*  THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
```

```
*  "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
*  LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
*  A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
*  OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
*  SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES INCLUDING, BUT NOT
*  LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
*  DATA, OR PROFITS; OR BUSINESS INTERRUPTION HOWEVER CAUSED AND ON ANY
*  THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
*  INCLUDING NEGLIGENCE OR OTHERWISE ARISING IN ANY WAY OUT OF THE USE
*  OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
*
*/

/* USER CODE BEGIN (0) */
/* USER CODE END */

/* Include Files */

#include "HL_sys_common.h"
#include "HL_system.h"

/* USER CODE BEGIN (1) */
#include "HL_emif.h"
#include "HL_sys_mpu.h"
#include "HL_gio.h"
#include "HL_het.h"
#include "HL_mibspi.h"
/* USER CODE END */

/* USER CODE BEGIN (2) */
#pragma SET_CODE_SECTION(".blinky_section")
void blinky()
{
    int i;
    gioSetDirection(hetPORT1, 1);
    while(1)
    {
        gioToggleBit(hetPORT1, 0);
        for(i=0;i<1000000;i++);
    }
}
#pragma SET_CODE_SECTION()

extern uint32 BlinkyLoadStart;
extern uint32 BlinkyLoadEnd;
extern uint32 BlinkySize;
extern uint32 BlinkyStartAddr;
extern uint32 BlinkyEndAddr;
/* USER CODE END */

uint8   emacAddress[6U] =   {0xFFU, 0xFFU, 0xFFU, 0xFFU, 0xFFU, 0xFFU};
uint32  emacPhyAddress  =   1U;

void main(void)
{
/* USER CODE BEGIN (3) */
    int i;
    uint32 size=(uint32)&BlinkySize;

    emif_SDRAMInit();

    for(i=0;i<size;i++)
    {
        ((char *)&BlinkyStartAddr)[i] =((char *)&BlinkyLoadStart)[i];
    }

    blinky();
    while(1);
/* USER CODE END */
}

/* USER CODE BEGIN (4) */
/* USER CODE END */
```