

J5Eco Application Note

CortexM3 access

.1.Introduction

This document shows take M3 RTOS and M3 ISS out of reset. To the M3 SecSS core you should have access by default. Although this User Guide refers to J5Eco-GP, the same procedure is applied to J5Eco-HS.

.2. Requirements

J5Eco environment This document assumes that you have a working environment with J5Eco and that you are aware on how to make a Hello World project on a J5Eco board (Cortex-A8 core). If you are not familiar with this please follow the J5Eco Startup Guide.

JTAG Emulator In this guide XDS560V2 emulator was used but any emulator compatible with TI811x / TI814x will work.

Code Composer Studio In this guide CCS 5.3.0 was used. Other versions should work as well.

http://processors.wiki.ti.com/index.php/Download_CCS

.3.Procedure

In order to access the M3- ISS and M3-RTOS we need to enable the Ducati clocks. There are two ways that this can be achieved.

Procedure 1 – No GEL file

If no GEL file is used than we need to enable the Ducati clocks .This can be done by calling the following function when running the Cortex-A8 core:

```
void TakeM3OutOfReset(void){  
    // Enable the Ducati Logic  
    RD_M_WR_MEM_32(RM_DEFAULT_RSTCTRL, 0x0C, 0xFFFFFFFFE3);  
  
    // Enable Power Domain Transition  
    WR_MEM_32(CM_DEFAULT_DUCATI_CLKSTCTRL, 2);  
  
    // Enable Ducati Clocks  
    WR_MEM_32(CM_DEFAULT_DUCATI_CLKCTRL, 2);  
  
    // Check CLKIN200TR & CLKINTR are active  
    while (((RD_MEM_32(CM_DEFAULT_DUCATI_CLKSTCTRL) & 0x300) >> 8) != 3) ;  
  
    //Write Ducati IRAM Boot Image - address not accessible  
    //WR_MEM_32(DUCATI_BASE_ADDR, 0x10000);  
    //WR_MEM_32(DUCATI_BASE_ADDR + 0x4, 0x9);  
    //WR_MEM_32(DUCATI_BASE_ADDR + 0x8, 0xE7FEE7FE);  
  
    // Reset both CPU's  
    RD_M_WR_MEM_32(RM_DEFAULT_RSTCTRL, 0x00, 0xFFFFFFFFE3);  
    // Check for Ducati M3_0 & M3_1 out of Reset  
    while (((RD_MEM_32(RM_DEFAULT_RSTST) & 0x1C)) != 0x1C) ;  
    // Check Module is in Functional Mode  
    while (((RD_MEM_32(CM_DEFAULT_DUCATI_CLKCTRL) & 0x30000) >> 16) != 0) ;  
}
```

Once this function is executed on Cortex A8, you can connect to the M3 cores via debugger. Test example:

```
/*
 * This Software takes the M3 cores out of reset
 */

//Include files
#include <stdlib.h>
#include <stdint.h>
#include <stdbool.h>

#define PRCM_BASE_ADDR                0x48180000

#define CM_DEFAULT_DUCATI_CLKSTCTRL    (PRCM_BASE_ADDR + 0x0518)
#define CM_DEFAULT_DUCATI_CLKCTRL      (PRCM_BASE_ADDR + 0x0574)
#define RM_DEFAULT_RSTCTRL             (PRCM_BASE_ADDR + 0x0B10)
#define RM_DEFAULT_RSTST               (PRCM_BASE_ADDR + 0x0B14)

// RD/RW Macros
#define UWORD32                        unsigned int
#define WR_MEM_32(addr, data)          *(UWORD32*)(addr) = (UWORD32)(data)
#define RD_MEM_32(addr)                *(UWORD32*)(addr)
#define RD_M_WR_MEM_32(addr, data, mask) WR_MEM_32((addr), (RD_MEM_32(addr) & (mask)) | (data))

void TakeM3OutOfReset(void);

int main(void) {
    int debug_var=1;

    while(debug_var!=2){};

    TakeM3OutOfReset();

    while(debug_var!=3){};

    //return 0;
}

void TakeM3OutOfReset(void){

    //UWORD32 i;

    // Enable the Ducati Logic
    RD_M_WR_MEM_32(RM_DEFAULT_RSTCTRL, 0x0C, 0xFFFFFE3);

    // Enable Power Domain Transition
    WR_MEM_32(CM_DEFAULT_DUCATI_CLKSTCTRL, 2);

    // Enable Ducati Clocks
    WR_MEM_32(CM_DEFAULT_DUCATI_CLKCTRL, 2);

    // Check CLKIN200TR & CLKINTR are active
    while (((RD_MEM_32(CM_DEFAULT_DUCATI_CLKSTCTRL) & 0x300) >> 8) != 3) ;

    //Write Ducati IRAM Boot Image - address not accesible
    //WR_MEM_32(DUCATI_BASE_ADDR, 0x10000);
    //WR_MEM_32(DUCATI_BASE_ADDR + 0x4, 0x9);
    //WR_MEM_32(DUCATI_BASE_ADDR + 0x8, 0xE7FEE7FE);
}
```

```

// Reset both CPU's
RD_M_WR_MEM_32(RM_DEFAULT_RSTCTRL, 0x00, 0xFFFFFFFFE3);
// Check for Ducati M3_0 & M3_1 out of Reset
while (((RD_MEM_32(RM_DEFAULT_RSTST) & 0x1C)) != 0x1C) ;

// Check Module is in Functional Mode
while (((RD_MEM_32(CM_DEFAULT_DUCATI_CLKCTRL) & 0x30000) >> 16) != 0) ;
}

```

After we run the software the CPU will stop before executing the M3 access function. At this point we can check and see that we do not have access to ISS – M3 and RTOS – M3 cores. After we modify the debug_var to 2 we can run the software until the next while loop. At this point we should be able to access both M3 cores.

The screenshot displays the Code Composer Studio (CCS) interface for a project named "Test M3 Access". The main window shows the C source code for `main.c`. The code includes macros for memory-mapped I/O and a `main` function that sets `debug_var` to 1 and enters a loop where it calls `TakeM3OutOfReset()` until `debug_var` becomes 3. The `TakeM3OutOfReset` function is defined to write to the `RM_DEFAULT_RSTCTRL` register and then to the `CM_DEFAULT_DUCATI_CLKCTRL` register.

On the right side, the "Test M3 Access" window shows the "Memory Map" tab. It displays the memory map for the Cortex-M3, with a table showing the start and end addresses of memory regions. The first entry is for RAM, starting at 0x0 and ending at 0x7FFFFFFF.

Start Address	End Address	Attributes
0x0	0x7FFFFFFF	RAM

Procedure 2 – with GEL file

Another way to take the M3 cores out of reset is to use a predefined script of the GEL file provided by Spectrum Digital software package. The GEL file can be found here: <http://support.spectrumdigital.com/boards/evm811x/revc/>

To add the GEL file to your Target Configuration, select Advance Setup and set the initialization script of Cortex A8:

Basic

General Setup

This section describes the general configuration about the target.

ConnectionSpectrum Digital XDS560V2 STM USB Emulator

Board or Device

type filter text

☐ Stellaris LM4FS270MPDT

☐ Stellaris LM4FS270MZRB

☐ Stellaris LM4FS99H5BB

☐ Stellaris LM4FSXAH5BB

☐ Stellaris LM4FSXLH5BB

☐ TCI6608

☐ TCI6614

☐ TCI6616

☐ TCI6618

☒ TI814x

TI814x

Advanced Setup

Target Configuration: lists the configuration options for the target

Save Configuration

Save

Test Connection

To test a connection, all changes must have been saved, the configuration file contains no errors and the connection type supports the function.

Test Connection

Note: Support for more devices may be available from the update manager.

All Connections

IVAHD0_iCont2

ARM9_ICont2

TPPSS

ARM9_TPPSS

Ducati_M0

CS_DAP_RTOS

subpath_1

Cortex_M3_RTOS

Ducati_M1

CS_DAP_ISS

subpath_2

Cortex_M3_ISS

SECSS

CS_DAP_SECSS

subpath_3

Cortex_M3_SECSS

DAP

CS_DAP_DebugSS

ModenaSS

CortexA8

ICECrusherCS

ETM

Trace_STM

CSSTM_0

CSETB_0

Import...

New...

Add...

Delete

Up

Down

Test Connection

Save

Cpu Properties

Cortex_A8 CPU

Set the properties of the selected cpu.

☐ Bypass

initialization script

Browse...

☐ Slave Processor

Address

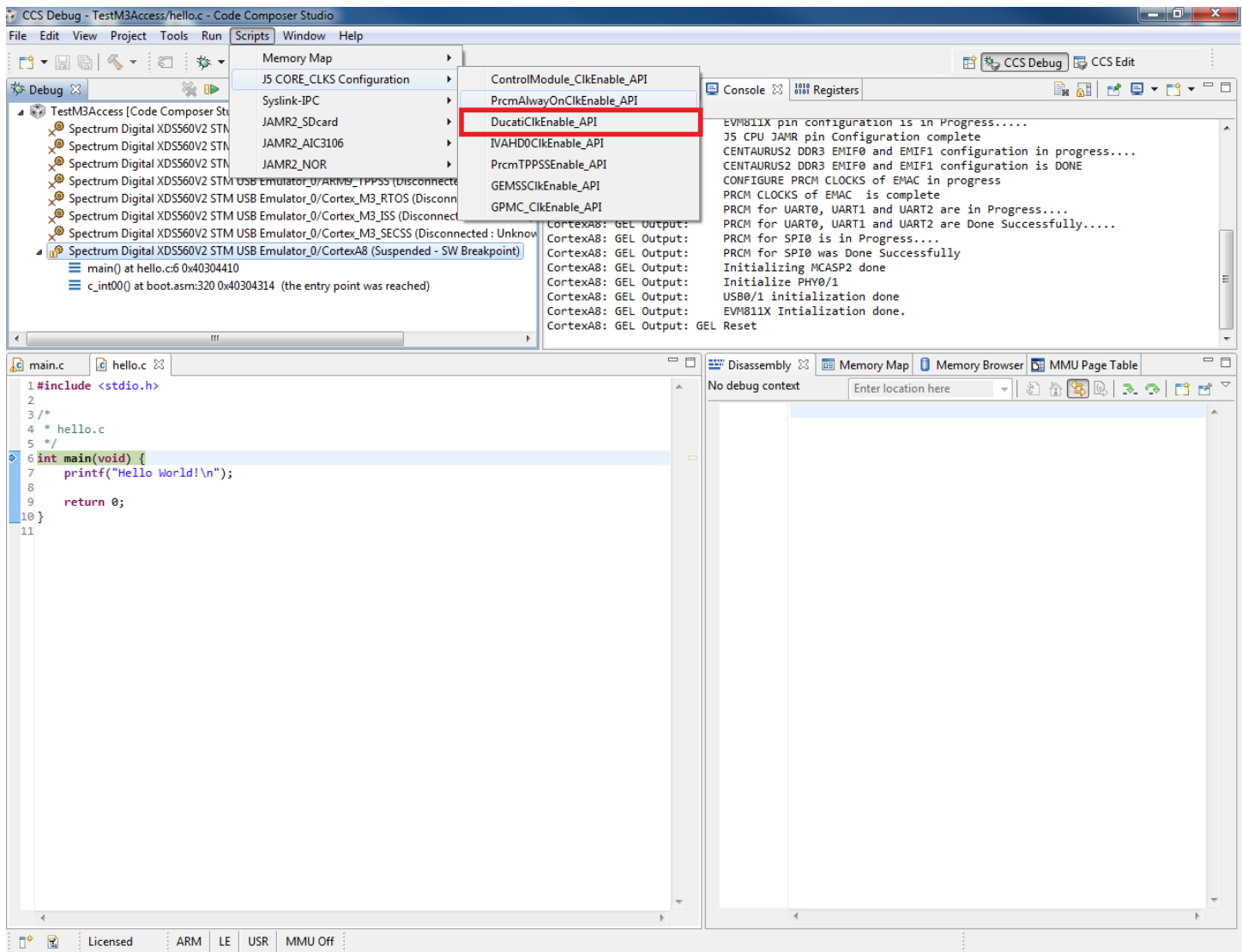
0x80001000

TraceDeviceId

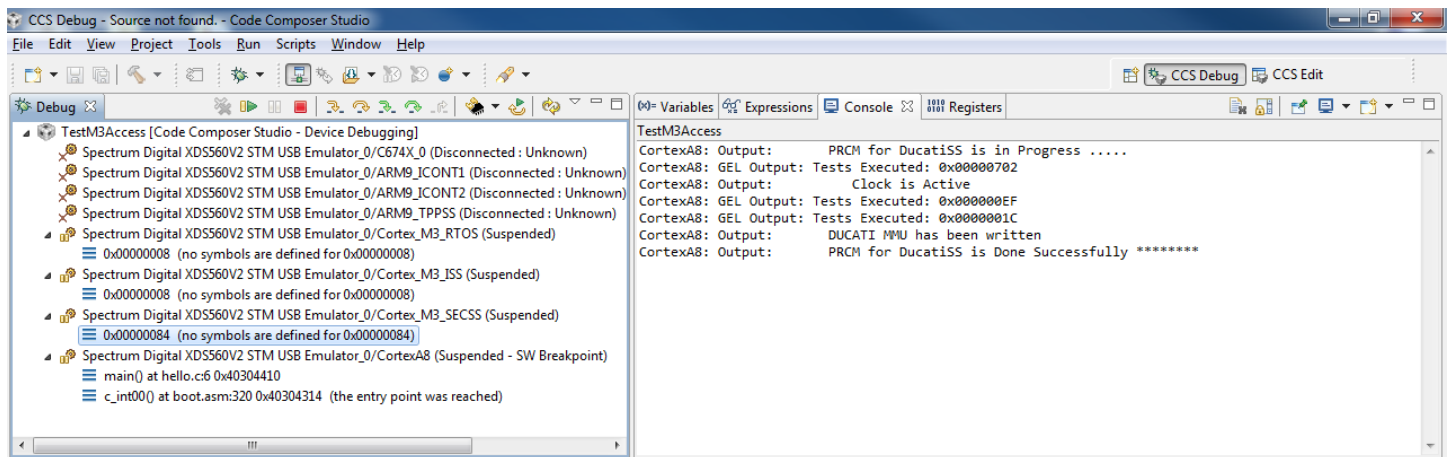
0x0

After you go in Debug Mode you connect to CortexA8 and use the DucatiClkEnable_API script:

CCS Upper Toolbar -> Scripts -> J5 Core_CLKS Configuration -> DucatiClkEnable_API



On the Console you should see the confirmation that your script is being executed.



Now we can access all M3 cores.

