TEXAS INSTRUMENTS

Release Notes

SPRUGQ6–October 2009

Scan Converter Software

The purpose of this document is to provide information about the contents, installation and usage of the scan converter release 1.0.1. Application programming interface (API), sources, and links to test procedure documentation are provided.

Project collateral discussed in this document can be downloaded from the following URL: http://www-s.ti.com/sc/techlit/sprugq6.zip.

Contents

List of Figures

List of Tables

1 Directory Structure

Table 1 shows the directory structure used for building the scan converter library and test application. The <platform> supported here is TMS320DM6437, TMS320DM6446, OMAP3530 and TMS320C6455. The C6455 is not optimized; scu is only ported on this platform to be able to run and see functional behavior as other ultrasound modules can run on the common C6455 platform.

Table 1. Directory Structure

Directory/File	Content
scu\c64P	Contains scu.lib, scu_dbg.lib, release and debug version of scan converter library
scu\docs\doxygen	Documentation
scu\make_c64P_lib	Contains make_c64P_lib.pjt, Code Composer Studio™ project file for building scan converter library
scu\src	Contains scan converter specific c source files
scu\src\c64P	Contains scan converter files that are C64x+™ DSP specific
scu\test	Root directory for files used for testing scan converter
scu\test\ccsProj_<platform>	Directory containing Code Composer Studio project files for <platform> for testing scan converter
scu\test\ccsProj_<platform>\scutest_<platform>.tcf	DSP/BIOS™ configuration file for test project for <platform>

Code Composer Studio, C64x+, DSP/BIOS are trademarks of Texas Instruments.
All other trademarks are the property of their respective owners.

Table 1. Directory Structure (continued)

Directory/File	Content
scu\test\ccsProj_<platform>\testscu.pjt	Code Composer Studio project for test
scu\test\ccsProj_<platform>\Memory.cmd	Contains declaration of sections needed by scan converter library: SCU_COEFFS_EXT_MEM SCU_COEFFS_L1DSRAM SCU_OUT_L1DSRAM
scu\test\gen	Ignore
scu\test\vectors\inp	Contains input test vectors that are named with the mode of scu (refer to test section) and test ID list file
scu\test\vectors\out	Contains output of test runs and Matlab file display scuimage.m to display a file
scu\test\vectors\refOut	Contains reference output that should be bit-exact with those in vectors\out

Additionally, a C model project is provided under test/ccsProj_C6455_Cmodel/ for source release. This has a Code Composer Studio project that builds the reference C sources (in scu/src) and the test, and can be run on the C6455 evaluation module (EVM) or on the C64x+ CPU cycle accurate simulator (little endian). Some sources, referred to by this project and test, are provided for ease of understanding the core processing logic in scan conversion without the usage of DMA resources and without OS (BIOS). This project build may be useful for debugging production cases as it is plain C processing and produces bit-exact results with the production cases.

2 Dependencies

Most of the infrastructure software, like BIOS and XDAIS in subsequent sections, can be downloaded from the following link. You will need to provide the login and password that was obtained after registering with Code Composer Studio (bundled with an EVM purchase).

https://www-a.ti.com/downloads/sds_support/targetcontent/index.html

2.1 Code Composer Studio

Version 3.3, build 3.3.81.6 was used for verification. It is encouraged to use the latest Code Composer Studio 3.3 release.

2.2 DSP BIOS

Bios version 5.33.03 was used. BIOS release can be downloaded from the link referred to in Section 2.

This version of BIOS may not be the default that comes bundled in Code Composer Studio. Code composer Studio’s Help->About->Component Manager can be used to select the installed BIOS version.

2.3 XDAIS (IALG)

The scu has been made to be IALG compliant as well as IDMA3 compliant, see Section 2.4). IALG is a requirement of the XDAIS standard of TI for components. The IALG version used for testing is 6.24, which comes with the Framework Components download and may not be the same as that bundled in Code Composer Studio. This version (or future versions) can be downloaded from the link referred to earlier.

This release has been tested with QualiTI tool for XDAIS compliance. The test report is available in the package at docs\doxygen\qualiti--SCU_TI_C64P--report.html.

2.4 DMA

The current release of the scan converter algorithm carries data movement between external memory and internal memory using the XDAIS compliant DMAN3/ACPY3/IDMA3 components, which are part of the XDAIS Framework Components.

XDAIS Framework Components version 2.24.01 was used in building the SCU algorithm and test application.

This version (and future versions) of XDAIS Framework Components can be downloaded from the TI web site at:

https://www-a.ti.com/downloads/sds_support/targetcontent/index.html

2.5 3.5 DSKT2

The test application for the current release of the scan converter is using the XDAIS compliant DSKT2 for memory management. This is part of the XDAIS Framework Components download.

This version (and future versions) of XDAIS Framework Components can be downloaded from TI web site at:

https://www-a.ti.com/downloads/sds_support/targetcontent/index.html

2.6 IQ Math Library

The interpolation coefficient table is computed as part of initialization. This initialization is carried out by using functions or macros from TI's IQ Math Library version 2.1.3. This library can be downloaded from:

http://focus.ti.com/docs/toolsw/folders/print/sprc542.html

3 Environment Variables

Table 2 shows the environment variables used in the Code Composer Studio project file used to build the SCU library alone, and the SCU Test application.

Table 2. Code Composer Studio Environment Variables

Environment Variable	Example and Comments
BIOS_INSTALL_DIR	C:\CCStudio_v3.3\bios_5_33_03
FC_INSTALL_DIR	C:\fw_comp\framework_components_2_24_01
IQMATH_LIB_DIR	C:\CCStudio_v3.3\c64plus\IQmath_v213

4 Features Supported in This Release

- B-mode:
  - 8-bit unsigned input
  - 8-bit unsigned output
- Color scan conversion:
  - 8-bit signed input
  - 8-bit unsigned output
- B-mode scan conversion, color mapping, 422 video format output
  - 8-bit unsigned input
  - 422 video format output
- Color scan conversion, arbitration, color mapping, and 422 video format output
  - Input:
    - B-mode scan converted image
    - Color flow input: 8-bit signed flow velocity and 8-bit unsigned flow turbulence
  - Output: 422 video format
- Interpolation: bilinear 2x2 interpolation
- Sector and linear scanheads

5 Implementation Limitations and Features Not Supported

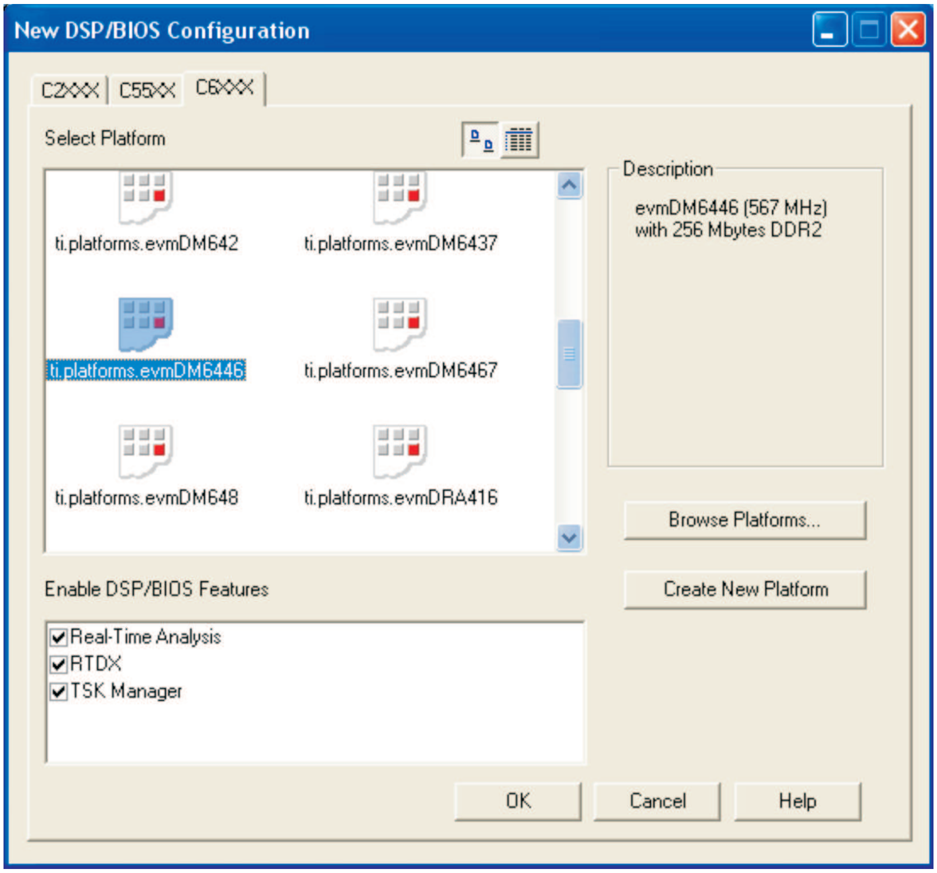
- Sector angles greater than ± 50° relative to vertical axis not supported
- Angle for linear scan heads is not supported in 0° to 180° range, only supported in -180 to 0 range
- Rectangular ROI cannot be positioned such that top of the rectangle is above the apex
- Only counter clockwise sector orientation supported
- The number of samples per scan line must be an integer multiple of 32
- The number of scan lines for B-mode must be an integer multiple of 16
- The offsetY parameter which defines a vertical offset in the output image is not supported.
- The lateralStride parameter which defines the sample points to skip along the input scan lines before the next scan line is not supported.

6 Test Application

The test application included here for <PLATFORM> is built in the directory scu\test\ccsProj_<platform>\ using the project testscu.pjt.

To create support for a target other than those supported in this release, the following changes should be carried out:

1. Use device-specific BIOS configuration using equivalent settings (menu File → New → DSP/BIOS Configuration).
   For example, DM6446 can be setup using the standard template as shown in Figure 1:



New DSP/BIOS Configuration

C6XXX

Select Platform: ti.platforms.evmDM6446

Description: evmDM6446 (567 MHz) with 256 Mbytes DDR2

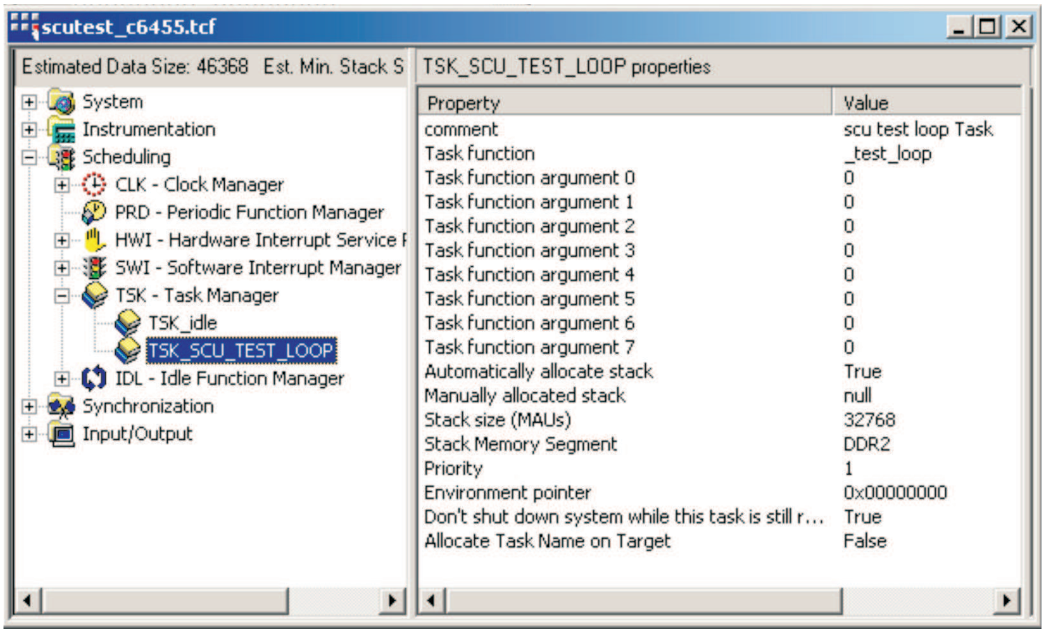
Enable DSP/BIOS Features

☑ Real-Time Analysis
☑ RTDX
☑ TSK Manager

Figure 1. New DSP/BIOS Configuration

2. Scan conversion takes place in a task context In the test application.

   Figure 2 shows an example of BIOS configuration for the task in which scan conversion takes place.



scutest_c6455.tcf
Estimated Data Size: 46368 Est. Min. Stack S
System
Instrumentation
Scheduling
CLK - Clock Manager
PRD - Periodic Function Manager
HWI - Hardware Interrupt Service F
SWI - Software Interrupt Manager
TSK - Task Manager
TSK_idle
TSK_SCU_TEST_LOOP
IDL - Idle Function Manager
Synchronization
Input/Output

TSK_SCU_TEST_LOOP properties

Property	Value
comment	scu test loop Task
Task function	_test_loop
Task function argument 0	0
Task function argument 1	0
Task function argument 2	0
Task function argument 3	0
Task function argument 4	0
Task function argument 5	0
Task function argument 6	0
Task function argument 7	0
Automatically allocate stack	True
Manually allocated stack	null
Stack size (MAUs)	32768
Stack Memory Segment	DDR2
Priority	1
Environment pointer	0x00000000
Don't shut down system while this task is still r...	True
Allocate Task Name on Target	False

Figure 2. BIOS Configuration for Scan Conversion

3. Note how defined memory sections for various BIOS objects are mapped. In a real application, this mapping has to be done judiciously.

   Test application uses standard file I/O for reading test vectors and storing scan converter output. Use of standard I/O library requires a higher level of stack allocation.

7 Test Configuration and Test Vectors

This package includes 21 sets of test vectors.

In all of the following text, <mode> is referred to as: bmode, bmode422, color, bmodecolor422. These correspond to scuMode_e definition in scu.h. The <testID> is just a number identifier for the test.

Input configuration file in vectors\inp : *scu_param_<mode>_<testID>.txt*

Input scan converter data file in vectors\inp: *scu_<mode>_<testID>.bin*

Output scan converter file containing output produced by running the test in vectors\out: *scu_<mode>_<testID>.bin*

Reference output scan converter file containing reference (pre-computed) output in vectors\refOut\: *scu_<mode>_<testID>.txt*. The vectors\out\ files should be bit-exact with vectors\refOut\ directory.

For bmodecolor422 mode, there are pairs of input text configuration files and pairs of output and reference output binary files representing the bmode and color parts (suffixed by _bmodePart and _colorPart) for corresponding parts.

The output files can be viewed by video/image display software that can read raw image files and files in 422 video format. One such software is YUV Player that can be downloaded at http://www.elverils.com/products.html. To view non-422 image files (those files that don't have *422*"in their names), the YUV pixel format of the software should be set to *Y*. To view image files in 422 format, the YUV pixel format of the software should be set to *UYVY*.

Alternatively, the generated files can be viewed in Matlab by running the Matlab program in vectors\out\displayscuimage.m. Once ran, it will prompt you to enter the file name. Enter the binary output file name and it will display corresponding to the <mode> indicated through the file name. The Matlab program automatically references the corresponding configuration file in vectors\inp directory to find the X and Y dimensions of the image to be able to display.

IMPORTANT NOTICE

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products		**Applications**	
Amplifiers	amplifier.ti.com	Audio	www.ti.com/audio
Data Converters	dataconverter.ti.com	Automotive	www.ti.com/automotive
DLP® Products	www.dlp.com	Broadband	www.ti.com/broadband
DSP	dsp.ti.com	Digital Control	www.ti.com/digitalcontrol
Clocks and Timers	www.ti.com/clocks	Medical	www.ti.com/medical
Interface	interface.ti.com	Military	www.ti.com/military
Logic	logic.ti.com	Optical Networking	www.ti.com/opticalnetwork
Power Mgmt	power.ti.com	Security	www.ti.com/security
Microcontrollers	microcontroller.ti.com	Telephony	www.ti.com/telephony
RFID	www.ti-rfid.com	Video & Imaging	www.ti.com/video
RF/IF and ZigBee® Solutions	www.ti.com/lprf	Wireless	www.ti.com/wireless

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265