

# TMS320C6455 Chip Support Library (CSL)

Version 3.00.10.01

## Release Notes

02 May 2006

This release of CSL for TMS320C6455 contains peripheral programming (functional and register level) APIs for C6455 modules. The list of modules supported in this release is listed in later sections. This set of APIs provides peripheral abstraction that can be used by higher layers of software.

This release includes:

- Compiled library of supported CSL modules.
- Interrupt Controller (INTC) module
- API reference guides.
- Archived sources and build scripts.

**Copyright © 2006 Texas Instruments Incorporated. All rights reserved.**

Information in this document is subject to change without notice. Texas Instruments may have pending patent applications, trademarks, copyrights, or other intellectual property rights covering matter in this document. The furnishing of this documents is given for usage with Texas Instruments products only and does not give you any license to the intellectual property that might be contained within this document. Texas Instruments makes no implied or expressed warranties in this document and is not responsible for the products based from this document.

---

**TABLE OF CONTENTS**

---

|          |                                    |           |
|----------|------------------------------------|-----------|
| <b>1</b> | <b>System Requirements .....</b>   | <b>3</b>  |
| <b>2</b> | <b>Release Contents .....</b>      | <b>4</b>  |
| <b>3</b> | <b>General information .....</b>   | <b>6</b>  |
| <b>4</b> | <b>Fixed in this release .....</b> | <b>11</b> |
| <b>5</b> | <b>Known Issues.....</b>           | <b>12</b> |
| <b>6</b> | <b>Open Issues .....</b>           | <b>13</b> |

## Release notes for TMS320C6455 CSL Version 3.00.10.01

### 1 System Requirements

- ❖ This version of CSL has been compiled with TI Codegen tools in Code Composer Studio version 3.2.37.12, Codegen compiler version C60\_6\_0\_3
- ❖ Compiler options used:
  - -mv6400+ -al -k

## 2 Release Contents

This release of CSL for TMS320C6455 contains functional and registers level APIs for modules listed in the table below to program the peripherals. This set of APIs provides peripheral abstraction that can be used by higher layers of software.

|                                  |
|----------------------------------|
| <b><i>DSP</i></b>                |
| <b><i>DAT</i></b>                |
| <b><i>DDR2</i></b>               |
| <b><i>DEV</i><sup>3</sup></b>    |
| <b><i>EDMA</i></b>               |
| <b><i>EMAC</i><sup>3</sup></b>   |
| <b><i>EMIFA</i></b>              |
| <b><i>GPIO</i></b>               |
| <b><i>HPI</i></b>                |
| <b><i>I2C</i></b>                |
| <b><i>INTC</i><sup>1,2</sup></b> |
| <b><i>McBSP</i></b>              |
| <b><i>MDIO</i><sup>3</sup></b>   |
| <b><i>PCI</i><sup>3</sup></b>    |
| <b><i>PLL</i></b>                |
| <b><i>SRIO</i></b>               |
| <b><i>TCP2</i></b>               |
| <b><i>TIMER64</i></b>            |
| <b><i>UTOP1A2</i></b>            |
| <b><i>VCP2</i></b>               |
| <b><i>BWMNGMT</i></b>            |
| <b><i>CACHE</i></b>              |
| <b><i>CFG</i></b>                |
| <b><i>CHIP</i></b>               |
| <b><i>ECTL</i><sup>3</sup></b>   |
| <b><i>IDMA</i></b>               |
| <b><i>MEMPROT</i></b>            |
| <b><i>POWERDOWN</i></b>          |
| <b><i>TSC</i></b>                |

**Note:**

1. *Example for INTC module is available only in csl\_c64xplus\_intc package.*
2. *INTC module is available as separate library. Modules using INTC should use “csl\_c64xplus\_intc.lib” (little endian) and “csl\_c64xplus\_intce.lib” (big endian).*
3. *Only the register layer is provided for these modules.*

### 3 General information

#### Installation Information

This contains the API reference guide for the CSL and four directories:

|  | File name                          | Contents  |
|--|------------------------------------|---|
| API reference guide                          | C6455_CSL_APIREFERENCE.pdf         | C6455 API Reference document  |
| Product directory                            | csl_c6455                          | <ul style="list-style-type: none"> <li>❖ All header files.</li> <li>❖ CSL libraries in big endian and little endian mode.</li> <li>❖ Release example - Each device has its own example folders.</li> </ul>  |
| Interrupt Controller (INTC) module directory | csl_c64xplus_intc                  | <ul style="list-style-type: none"> <li>❖ INTC module header files.</li> <li>❖ INTC module library file in big endian and little endian mode.</li> <li>❖ Release examples for INTC.</li> </ul> <p>This module is provided as separate library to facilitate use of CSL with an operating system.</p> |
| C6455 Source directory                       | csl_c6455_src                      | <ul style="list-style-type: none"> <li>❖ All C6455 CSL source files.</li> <li>❖ All header files.</li> <li>❖ Build scripts to build the little endian and big endian libraries</li> </ul>   |
| INTC Source directory                        | csl_c64xplus_intc_src              | <ul style="list-style-type: none"> <li>❖ INTC CSL source files.</li> <li>❖ INTC CSL header files.</li> <li>❖ Build scripts to build the little endian and big endian libraries</li> </ul>   |
| Release Notes                                | Releasenotes_C6455_v3_00_10_01.doc | <ul style="list-style-type: none"> <li>❖ Releasenotes document</li> </ul>   |

## Installation guidelines

The steps to be followed for installation of the release package are as follows:

1. Download the release zip file.
2. Unzip the files with command "unzip <zip file name>" at the command prompt. Alternatively, use the WinZip wizard to extract the files.

## Build guidelines

The steps to be followed to build the C6455 CSL library from the release are as follows:

1. At the command prompt go to the csl\_c6455\_src directory
2. Compile the code with the batch file given. Set the environment variable "CSL3X\_ROOT\_DIR = <folder path>" in your system. Different batch files for creating libraries for the devices are listed below. It will be found in following path csl\_c6455\_src\.
3. To clean the built object files invoke the corresponding batch file with the argument cleanall.

.\build\_c6455.bat cleanall

| Library   | Batch file to be invoked    | Library name   |
|-----------|-----------------------------|----------------|
| C6455 CSL | build_c6455 (little endian) | csl_c6455.lib  |
|           | build_c6455e (big endian)   | csl_c6455e.lib |

The steps to be followed to build the INTC CSL library from the release are as follows:

1. At the command prompt go to the csl\_c64xplus\_intc\_src directory
2. Compile the code with the batch file given. Set the environment variable "CSL3X\_ROOT\_DIR = <folder path>" in your system. Different batch files for creating libraries for the devices are listed below. It will be found in following path csl\_c64xplus\_intc\_src\.
3. To clean the built object files invoke the corresponding batch file with the argument cleanall.

.\build\_c64xplus\_intc.bat cleanall

| Library  | Batch file to be invoked            | Library name           |
|----------|-------------------------------------|------------------------|
| INTC CSL | build_c64xplus_intc (little endian) | csl_c64xplus_intc.lib  |
|          | build_c64xplus_intce (big endian)   | csl_c64xplus_intce.lib |

---

## Directory structure

The directory structures of product C6455 CSL library release is as shown below:

```

csl_c6455
|
|_____example →
|
|_____edma
|_____i2c
|_____mcbsp
|_____ddr2
|_____emifa
|_____vcp
|
|_____
|_____
|_____tcp
|
|_____inc      CSL module Header files
|_____lib
|_____csl_c6455.lib (little endian CSL library)
|_____csl_c6455e.lib (big endian CSL library)

```



The directory structure of INTC library release is as shown below:

```
csl_C64xplus_intc
|_____ example          Example for INTC module
|
|_____ inc              INTC module Header files
|
|_____ lib
|           |_____ csl_c64xplus_intc.lib (little endian INTC CSL library)
|           |_____ csl_c64xplus_intce.lib (big endian INTC CSL library)
```

The directory structure of C6455 CSL source directory is as shown below:

```
csl_c6455_src
|_____ inc ALL Header files
|
|_____ src
|           |_____ chip
|           |_____ common
|           |_____ edma
|           |_____ .
|           |_____ .
|           |_____ .
|           |_____ mcbasp
|_____ build_c6455_bc.bat (to build little endian library)
|_____ build_c6455e_bc.bat (to build big endian library)
|_____ Makefile
|_____ Makefile.inc
```

The directory structure of INTC CSL source directory is as shown below:

```
csl_c64xplus_intc_src
|_____ inc INTC Header files
|
|_____ src
|           |_____ intc
|           |_____ common
|_____ build_c64xplus_intc.bat (to build little endian library)
|_____ build_c64xplus_intce.bat (to build big endian library)
|_____ Makefile
|_____ Makefile.inc
```

## **Label in clearcase**

C6455\_CSL\_BASE\_REL\_03\_00\_10\_01

## **Examples**

The peripheral examples are designed to run on the TMS320C6455 EVM. Examples may have hardware dependencies that prevent them from running to completion on the CCS Simulator.

#### 4 Fixed in this release

|   |   |
|---|---|
| <b>1</b>  | <b>CACHE: bug in L1D cache controller when parallel load instructions are predicated and are accessing the same cache line.</b> |
| <b>Release Note</b><br>This has been fixed by installing new Codegen compiler v6.0.3. |   |

|  |  |
|--|--|
| <b>2</b>   | <b>COMPILER: New compiler v6.0.3 for TCI6455</b> |
| <b>Release Note</b><br>This has been tested using the new compiler v6.0.3. |  |

## 5 Known Issues

|                                   |  |
|-----------------------------------|--|
| <b>1</b>                          | <b>VCP2: Offset of VCP decisions read FIFO register(VCPRDECS) is mentioned wrong in the User guide</b> |
| <b>Release Note</b>               |  |
| None                              |  |
| <b>Workaround</b> (If one exists) |  |
| None                              |  |

|                                   |  |
|-----------------------------------|--|
| <b>2</b>                          | <b>SRIO:The default values of the certain SRIO registers mentioned in the SRIO datasheet are "undefined"</b> |
| <b>Release Note</b>               |  |
| None                              |  |
| <b>Workaround</b> (If one exists) |  |
| None                              |  |

|                                   |  |
|-----------------------------------|--|
| <b>3</b>                          | <b>TIMER: Timer user guide mention timer internal clock frequency as SYSCLK2</b> |
| <b>Release Note</b>               |  |
| None                              |  |
| <b>Workaround</b> (If one exists) |  |
| None                              |  |

## 6 Open Issues

|                                   |  |
|-----------------------------------|--|
| <b>1</b>                          | <b>DAT Module Functionality Reduced From CSL 2.x</b> |
| <b>Release Note</b>               |  |
| None                              |  |
| <b>Workaround</b> (If one exists) |  |
| None                              |  |

|                                   |   |
|-----------------------------------|---|
| <b>2</b>                          | <b>Description for "tcp_state" field of the TCPSTAT register is not mentioned in the latest HW Spec for TCP</b> |
| <b>Release Note</b>               |   |
| None                              |   |
| <b>Workaround</b> (If one exists) |   |
| None                              |   |

|                                   |  |
|-----------------------------------|--|
| <b>3</b>                          | <b>Add support for C6454 device in C6455 CSL</b> |
| <b>Release Note</b>               |  |
| None                              |  |
| <b>Workaround</b> (If one exists) |  |
| None                              |  |

|                                   |   |
|-----------------------------------|---|
| <b>4</b>                          | <b>TCP2_getMaxMinErr() does not return error status</b> |
| <b>Release Note</b>               |   |
| None                              |   |
| <b>Workaround</b> (If one exists) |   |
| None                              |   |

|                                   |  |
|-----------------------------------|--|
| <b>5</b>                          | <b>TCP2_getFrameLenErr() does not return the correct Value</b> |
| <b>Release Note</b>               |  |
| None                              |  |
| <b>Workaround</b> (If one exists) |  |
| None                              |  |

|                                   |   |
|-----------------------------------|---|
| <b>6</b>                          | <b>TCP2_getProLenErr() Does not return the correct status</b> |
| <b>Release Note</b>               |   |
| None                              |   |
| <b>Workaround</b> (If one exists) |   |
| None                              |   |

|                                   |   |
|-----------------------------------|---|
| <b>7</b>                          | <b>For PLLDIV4 and PLLDIV5 the divider configuration is failing</b> |
| <b>Release Note</b>               |   |
| None                              |   |
| <b>Workaround</b> (If one exists) |   |
| None                              |   |

|                                   |                                     |
|-----------------------------------|-------------------------------------|
| <b>8</b>                          | <b>VCP2_unpause() does not work</b> |
| <b>Release Note</b>               |                                     |
| None                              |                                     |
| <b>Workaround</b> (If one exists) |                                     |
| None                              |                                     |

|                                   |   |
|-----------------------------------|---|
| <b>9</b>                          | <b>VCP2_emuEnable() API does not work</b> |
| <b>Release Note</b>               |   |
| None                              |   |
| <b>Workaround</b> (If one exists) |   |
| None                              |   |

|                                   |                                  |
|-----------------------------------|----------------------------------|
| <b>10</b>                         | <b>EDMA3 symbol name changes</b> |
| <b>Release Note</b>               |                                  |
| None                              |                                  |
| <b>Workaround</b> (If one exists) |                                  |
| None                              |                                  |

|                                   |   |
|-----------------------------------|---|
| <b>11</b>                         | <b>General: GetBaseAddress API for some of the modules not handling invalid pointer for base address.</b> |
| <b>Release Note</b>               |   |
| None                              |   |
| <b>Workaround</b> (If one exists) |   |
| None                              |   |

|                                   |   |
|-----------------------------------|---|
| <b>12</b>                         | <b>SRIO: When the API srioHwSetupRaw() is called with valid arguments, the system hangs</b> |
| <b>Release Note</b>               |   |
| None                              |   |
| <b>Workaround</b> (If one exists) |   |
| None                              |   |

|                                   |  |
|-----------------------------------|--|
| <b>13</b>                         | <b>SRIO: When the API srioHwSetup() is called with default arguments, the system hangs</b> |
| <b>Release Note</b>               |  |
| None                              |  |
| <b>Workaround</b> (If one exists) |  |
| None                              |  |

|                                   |  |
|-----------------------------------|--|
| <b>14</b>                         | <b>GPIO: More example required to explain the usage of a pin configured as an input for generating interrupts or EDMA events</b> |
| <b>Release Note</b>               |  |
| None                              |  |
| <b>Workaround</b> (If one exists) |  |
| None                              |  |

|   |  |
|---|--|
| <b>15</b>                                 | <b>Unable to open c64xx_APIREFERENCE.doc</b> |
| <b>Release Note</b><br>None               |  |
| <b>Workaround</b> (If one exists)<br>None |  |

|   |   |
|---|---|
| <b>16</b>                                 | <b>CHIP: CSR register is not getting initialized to default value</b> |
| <b>Release Note</b><br>None               |   |
| <b>Workaround</b> (If one exists)<br>None |   |

|   |  |
|---|--|
| <b>17</b>                                 | <b>INTC: certain registers are not getting initialized to default values</b> |
| <b>Release Note</b><br>None               |  |
| <b>Workaround</b> (If one exists)<br>None |  |