Security Accelerator Low Level Driver

User Guide

Contributors to this document

Texas Instruments, Incorporated
20450 Century Boulevard
Germantown, MD 20874 USA

VP00102-Form-1
Revision D

Contents

User Guide

User Guide for SA LLD and Examples

Overview

This document provides guideline for compiling and executing Security Accelerator Low Level Device Driver (SA LLD) with examples included in the release package.

As a pre-requisite refer release note for the corresponding C66X PDK in order to find recommended version of compiler and dependent tools. PDK and SA LLD would need to be installed by running the installer. For rest of the document keyword

- <*PDK_INSTALL_PATH*> refers to the location where PDK is installed.
- <*SALLD_INSTALL_PATH*> refers to the location where SA LLD is installed

Build guidelines for SA LLD package

Release package comes with a pre-built set of libraries for SA LLD module. Below are the steps in case if source release is provided and the SA LLD would need to be rebuilt. Note that makefile support is now added so that SA LLD libraries can be built using makefile.
NOTE: ***SA LLD package for ARM Linux user space has to be built in Linux environment***.

1) Building SA LLD using `gmake` in Windows environment

a) Setting Environment Variables:

- **C6X_GEN_INSTALL_PATH**
  - Install location for Code Gen Tool
  - Example for default location of Code Gen Tool 7.2 GA. Note use of forward slash for the path in the case of windows environment.

set C6X_GEN_INSTALL_PATH="C:/Program Files/Texas Instruments/C6000 Code Generation Tools 7.2.4"

- **XDC_INSTALL_PATH** [Optional - See below]

- Install location for XDC Tool
- This is only required if gmake is available through XDC tool
- Example for Default location in the case of xdctools_3_22_01_21 [Illustration Purpose] :

```
set XDC_INSTALL_PATH=C:/Program Files/Texas Instruments/xdctools_3_22_01_21
```

- **PDK_INSTALL_PATH**
  - Install location for PDK package
  - Example for Default location of PDK TCI6614 1.0.0.13:

```
set PDK_INSTALL_PATH=C:ti/pdk_tci6614_1_00_00_13/packages
```

- Additional environment variables are optional. Refer sasetupenv.bat for the details.

b) Once the environment variables are initialized correctly, run `sasetupenv.bat` located in the <*SALLD_INSTALL_PATH*>*packages*> directory to set the environment. Refer to the batch file in case if any modifications are required for the target environment.

```
sasetupenv.bat
```

c) To build the SA LLD libraries, execute `sabuild.bat` from the SA LLD packages directory <*SALLD_INSTALL_PATH*>*packages*

```
sabuild.bat
```

2) Building SA LLD using `gmake` in MSYS environment

a) Setting Environment Variables

- `C6X_GEN_INSTALL_PATH`
  - Install location for Code Gen Tool
  - Examples for default location of Code Gen Tool 7.2.4 GA

```
export C6X_GEN_INSTALL_PATH="C:/Program Files/Texas Instruments/C6000 Code Generation Tools 7.2.4"
```

- `PDK_INSTALL_PATH`
  - Install location for PDK package

- Change the shell prompt directory to PDK installation directory. Example for a default PDK installation at `C:/ti/pdk_tci6614_1_00_00_13`, the commands would be

```
$cd "C:/ti/pdk_tci6614_1_00_00_13/packages"

$export PDK_INSTALL_PATH=$PWD
```

- `SALLD_INSTALL_PATH`
  - Install location for SA LLD package
  - Change the shell prompt directory to SA LLD installation directory. Example for a default SA LLD installation at `C:/ti/salld_1_0_5_4/packages`, the commands would be

```
$cd "C:/ti/salld_1_0_5_4/packages"

$export SALLD_INSTALL_PATH=$PWD
```

b) To build all the SA LLD libraries, run `sabuild.sh` from the top level SA LLD packages directory *<SALLD_INSTALL_PATH>/packages*

```
$./sabuild.sh
```

The shell script cleans and rebuilds all SA LLD libraries.

3) Building SA LLD using gmake in Linux bash environment

a) Setting Environment Variables
Set the build environment by sourcing the environment variable setup script "sasetupenv_lnx.sh".
source sasetupenv_lnx.sh

Following are the environment variables in the script which needs to be modified according the build environment.

- C66X_GEN_INSTALL_PATH
  Install location for c66x processor Code Generation Tool
- CROSS_TOOL_INSTALL_PATH
  Install location for GNU ARM Cortex-A8 tool.
- PDK_INSTALL_PATH
  Install location for TI PDK package which contains CSL

- SA_INSTALL_DIR
  Install location for SA LLD package

b) To build the SA LLD components, run sabuild.sh from the top level SA LLD packages directory <SALLD_INSTALL_PATH>/packages
The shell script cleans and rebuilds SA LLD libraries.

Steps to run example and/or unit test projects

The "`example`" directory in SA LLD contains several examples[1] which demonstrate usage of API's. The "test" directory in SA LLD contains unit test module[2].

1. Check Prerequisites

Please ensure that all dependent/pre-requisite packages are installed before proceeding with the examples and/or unit test.

2. Configure CCS Environment

The CCS environment configuration step needs to be done only once for a workspace as these settings are saved in the workspace preferences. These settings only need to be modified if:

- New workspace is selected
- Newer version of the component is being used. In that case modify the paths of the upgraded component to the newer directory.

The procedure mentioned in this section is provided using <Managed Build Macro> option in CCS. Following are the steps:

a. Create a macro file if not available from the SA LLD or PDK release. For the SA LLD release, sample file: <*SALLD_INSTALL_PATH*>\packages\macros.ini may be modified and used.

b.
Following environment would need to be available in the macros.ini file

```
PDK_INSTALL_PATH    = <PDK_INSTALL_PATH>\packages
CSL_INSTALL_PATH    = <PDK_INSTALL_PATH>\packages
CPPI_INSTALL_PATH   = <PDK_INSTALL_PATH>\packages
QMSS_INSTALL_PATH   = <PDK_INSTALL_PATH>\packages
PASS_INSTALL_PATH   = <PDK_INSTALL_PATH>\packages
SA_INSTALL_PATH     = <SALLD_INSTALL_PATH>\packages
```

3. Create CCS Project

[1] The example projects without "_w3gpp" can be run with standard SA LLD package and the example projects with "_w3gpp" can be run only if the SA 3GPP Enabler is installed.
[2] The unit test project without "_w3gpp" can be run with standard SA LLD package and the unit test project with "_w3gpp" can be run only if the SA 3GPP Enabler is installed.

SA LLD package includes a batch file *saProjectCreate.bat* under *<SA_INSTALL_PATH>\packages*. The batch file allows creation of projects based on the different dependent tool version for all examples and unit tests included in SA LLD. The batch file also allows additional executable types eg: Endianess. Batch file is supported for CCSv5.x environment.

Alternatively, projects can be created using the CCS wizard and importing required files from test and example directories.

.

Additional details on using *saProjectCreate.bat* :

- Prerequisite: All dependent components need to be installed. After the components are installed, start CCS once and wait for eclipse plugin to get into effect before running the batch file.

- Modify environment variables in `saProjectCreate.bat` under "*<SALLD_INSTALL_PATH>*\packages" directory to reflect project options. This would also include
  - `IS_SIMULATOR_SUPPORT_NEEDED`: For running projects in simulator environment
  - `ENDIAN`: To select "`little`" or "`big`" endian
  - Set CCS_INSTALL_PATH to CCS 5.x install location
  - Refer additional environment settings in the batch file

  Run the batch file: saProjectCreate.bat

- The command line above will create projects for all SA LLD examples and unit tests. In order to create projects for a single example, for instance the SA LLD basic example, the command line should be modified as follows:
  `saProjectCreate.bat`
  "*<SALLD_INSTALL_PATH>*`\packages\ti\drv\sa\example\SaBasicExample`"

The CCS projects will be located under the directory selected for environment variable <MY_WORKSPACE> which by default points to C:\MyPDKWorkspace<PART_NUMBER>

4. Import Project

Below are the steps for importing project assumes that CCS project is already available.

a. Select C/C++ Development perspective

b. Click on File → Import

c. On the Import Dialog Box select CCS → Existing CCS/CCE Eclipse Project

d. Click on Next

e. This will pop up a new dialog box; ensure that 'Select Root Directory' option is selected

f. Click on Browse and select the top level directory where the project is present. For example

```
C:\MyPDKWorkspace<PART_NUMBER>
```

<PART_NUMBER> reflects the device part number for the PDK being installed.

g. Under the projects section you should see the project. For example

```
SA_BasicExample[_w3gpp]_exampleProject
```

h. Click Finish

5. Build Project

To build the project; ensure that the project you want to build, i.e., *"SA_BasicExample_exampleProject"* is set as the active project. Click on Project → Build Active Project.

6. Run Project

a. Launch the Debugger and switch to the Debug Perspective.

b. To execute the project ensure the following is done:

i. Click on Target → Reset CPU

ii. Click on Target → Load Program

iii. Select the executable file to be loaded. Example:

"C:\MyPDKWorkspace<PART_NUMBER>\SA_BasicExample`[_w3gpp]`*_examplePr oject\Debug\SA_BasicExamplet_exampleProject.out"*

iv. Click on OK.

v. Once the project is loaded; click on Target → Run to execute it.

Steps to run example project in Linux user space

1. Build example

- Change to the ARM Linux build directory "`"<SALLD_INSTALL_PATH>/packages`
- Modify the build environment setup script to match the installation environment. Setup the build environment by sourcing the script "armv7setupenv.sh".
  source ./ armv7setupenv.sh

- Change to "*ti/drv/sa*" folder
- Build the example by issuing the make command:
  make –f makefile_armv7 clean all
- The executable will be placed in the directory as specified by "ARMV7SABINDIR" variable in the script "armv7setupenv.sh".

Note:

- Please make sure the dependent ARMV7 libraries for PA, QMSS, and CPPI LLDs are pre-built in the ARMV7LIBDIR as specified in the "armv7setupenv.sh" script.
- Any application that needs SA LLD ARMV7 library needs to export "ARMV7SALIBDIR" variable and update the library include path in the applications makefile.
  - Linux:
    - export ARMV7SALIBDIR =<SALLD_INSTALL_PATH>/packages/ti/drv/sa/lib
  - Windows:
    - set ARMV7SALIBDIR =<SALLD_INSTALL_PATH>/packages/ti/drv/sa/lib

2. Install the executable to Linux filesystem

Install/copy the "bin" directory and it sub-directories to the target linux filesystem.

3. Execute the example

Execute the example from on the target ARM processor. At the end of execution the example prints the test summary.

Additional notes for Linux user space

1. Resource sharing with Linux Kernel

Following multicore navigator related resources are shared with the Linux kernel in the case of LLD examples.

- CPPI Flow id
- General purpose Queues
- QMSS memory regions

Application and the Linux kernel need to ensure that there is no conflict.of these resources.