## Lab 02: HyperLink Example

### Purpose

The purpose of this lab is to demonstrate how to build and run a HyperLink loopback application on the C6678 EVM using the example code as delivered with MCSDK. In addition, you will make modifications to the application parameters to very the transfer rate. Optionally, you will run the application on two boards and modify the transfer rate to determine the maximum throughput allowed with this example configuration.

### Project Files

The following project files are used in this lab:

PDK\_INSTALL\_PATH\pdk\_C6678\_1\_0\_0\_19\packages\ti\drv\exampleProjects\hyplnk\_exampleproject

### TASK 1: Import the Example Project

1. Open CCS.
2. Set the Perspective to CCS Edit.
3. Import the project.
   * Project | Import Existing CCS/CCE Eclipse Project
   * Select search\_directory: \pdk\_C6678\_1\_0\_0\_19\packages\ti\drv\exampleProjects
   * From the list of Discovered projects, choose hyplnk\_exampleproject and then click Finish.
4. hyplnk\_exampleproject should now appear in your Project Explorer.

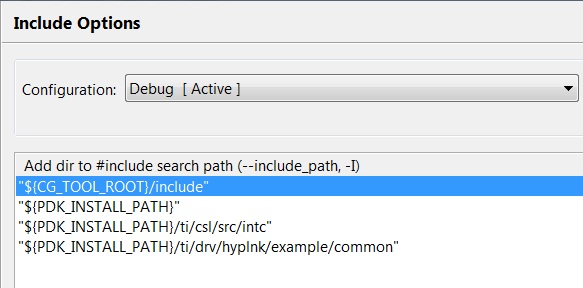
How many lanes are configured?

What is the baud rate?

1. Refer to hyplnkLLDCfg.h

### TASK 2: Set the Project Properties

1. Select the hyplnk\_exampleproject.
2. Right click and select Properties.
3. Select General and choose the Main Tab.
4. Set the following Device Properties.
   * Device Family = C6000
   * Variant = Generic C66x Device
5. Under Build/C6000 Compiler, select Basic Options and set the following compiler debug properties:
   * Target processor verison = 6600
   * Debugging model = Full symbolic debug
   * Optimization level = 0
   * Optimize for code size = 0
6. Click OK.
7. Under Build/C6000 Compiler, select Include Options and verify the following paths:



### TASK 3: Build the Project

1. Select hyplnk\_exampleproject.
2. Build the project.
   * Project | Build Project  
     OR
   * Right Click and select Build Project
3. Verify that the build was successful.

Was the file hyplnk\_exampleproject.out generated?

1. From the CCS Edit perspective, check the Binaries or Debug directory. From the CCS Debug perspective, check the Console.

### TASK 4: Connect to the Target EVM

1. Set the Perspective to CCS Debug.
2. Create a new User-Defined Target:
   * View | Target Configurations
   * Select User Defined
   * Click the New Target button or Right-click and select New Target Configuration
3. Define the C6678L/LE EVM as a new target:
   * File name = EVM6678L or EVM6678LE
   * Location = <local>\ti\CCSTargetConfigurations
   * Click Finish
4. Make sure the EVM is powered ON and connect your PC/laptop to the emulator port on the EVM using the provided USB cable.
5. Launch the target configuration (e.g., EVM6678LE.ccxml).
   * Select the target.
   * Right click and select Launch Selected Configuration.
6. Select Core 0, right click, and select Connect Target.

### TASK 5: Load and Run the Program

1. Load the .out file created earlier in the lab.
   * Run | Load | Load Program
   * Click Browse Project
   * Select hyplnk\_exampleproject.out and Click OK.
   * Click OK to load the application to the target.
2. Run the application.

Did the application execute successfully?

1. Check the console.

### TASK 6: Increase the Transfer Rate

1. Set the Perspective to CCS Edit.
2. Modify the example code for hyplnk\_exampleproject
   * Open the readme.txt and determine which file contains parameters that define the reference clock, the serial data rate, the number of lanes, and loopback.
   * Open the file within the project.
   * Change the Baud Rate to 12.5 Gbaud
3. Build the code, load to Core 0, and run.

Did the application execute successfully at the new rate?

1. Check the console.

### 

### TASK 7 (optional): Board-to-board HyperLink Example

Modify the example to run the HyperLink application on two EVMs.

Hardware requirements:

* + Two C66x EVMs
  + One HyperLink cable
  + Connector Board

1. Modify the example code for hyplnk\_exampleproject
   * Open hyplnkLLDCfg.h
   * Search for “#define hyplnk\_EXAMPLE\_LOOPBACK”
   * Comment out this command.
   * Change the Baud Rate back to 6.25 Gbaud
2. Build the code, load to both targets, and run only on Core 0.

Did the application execute successfully?

1. Check the console.
2. Modify the example code for hyplnk\_exampleproject
   * Open hyplnkLLDCfg.h
   * Change the Baud Rate to a higher rate.
3. Build the code, load to both targets, and run only on Core 0.

Did the application execute successfully?

What is the highest transfer rate that can be achieved using this example?

*Maintain Blank Page Here*