Test Environment for TC39x Aurix Board Application Note Version: 1.01

app0040_v1_01.docx





Designed by XYLON

All rights reserved. This manual may not be reproduced or utilized without the prior written permission issued by Xylon.

Copyright © Xylon d.o.o. logicBRICKS[™] is a registered Xylon trademark.

All other trademarks and registered trademarks are the property of their respective owners.

This publication has been carefully checked for accuracy. However, Xylon does not assume any responsibility for the contents or use of any product described herein. Xylon reserves the right to make any changes to product without further notice. Our customers should ensure to take appropriate action so that their use of our products does not infringe upon any patents.

| LOGIC BRICKS | Test Environment for TC39x Aurix Board | Xylon |
|---|---|----------------|
| July 24 th , 2019 | CONFIDENTIAL | Version: v1.01 |
| 1 INSTALLING AND SET | ING UP THE ENVIRONMENT | 4 |
| 1.1 HIGHTEC TRICORE I 1.2 INSTALLING BIFACES | ENTRY TOOL CHAIN SETUP | 4 |
| 1.3 FLASHING THE AURIX 2 REVISION HISTORY | TC39X BOARD WITH INFINEON MEMTOOL | |





July 24th, 2019

Version: v1.01

1 Installing and Setting Up the Environment

The testing of logiHSSL IP core on FPGA is done by setting up the whole environment where Aurix TC39x board is master, while FPGA side (logiHSSL) is slave. Programming the Aurix TC39x board is done with binary files generated in the special Infineon Eclipse environment, named BIFACES. There are steps before installing and running BIFACES that are necessary to be taken, like installing the free HIGHTEC TriCore Entry Tool chain software.

Depending on the version of the Aurix TriCore, user will have to choose between StepA and StepB version of the software. Programming wrong version of the software will result in probable damage to the TC39x board. TC39x ver 1.0 usually uses StepA, ver 2.0 usually uses StepB – user must take care to select the correct version.

Also, it's important to mention here that the logiHSSL IP inside the FPGA has the internal address remaper for accessing the resources on the FPGA side. This means, if user wants to access all these resources from the Master (AURIX) side, the destination addresses for this are calculated by adding the base address offset of 0x3000 0000.

1.1 HIGHTEC TriCore Entry Tool Chain Setup

These steps are needed for installation and setting up the HIGHTEC TriCore Entry Tool Chain on Windows PC:

- 1. Go to <u>http://free-entry-toolchain.hightec-rt.com/.</u>
- 2. Enter name, email, company name, phone number, MAC address of target machine and click on **Generate License File & Download**.
- 3. Download Installation Package and license.lic file.
- 4. Extract and start setup.exe (administrator rights!) and leave all options on default.
- 5. If not already available a JRE (Java Runtime Environment) must be installed on the target machine.
- 6. Copy the license file (license.lic) to the default installation folder C:\HIGHTEC\licenses.
- 7. Start the Environment.
- 8. Create a Project.
- 9. Enter Project Name => **Next**.
- 10. Select the Kit you are using, e.g. TC29xB all options on default => Finish.
- 11. Modify and build the project.
- 12. Open hello.c inside src-folder.
- 13. Implement a main which does something but runs forever, for example:

```
int main(void) {
    int i=0;
    int a=0;
    for (i=0;i<100;i++) {
        a++;</pre>
```

| LOGIC BRICKS | Test Environment for TC39x Aurix Board | Xylon |
|------------------------------|---|----------------|
| July 24 th , 2019 | CONFIDENTIAL | Version: v1.01 |
| if (a== 50) { | | |
| a= 0 ; | | |
| i= 0 ; | | |
| } | | |
| } | | |
| return EXIT_SUC | CESS; | |
| } | | |

- 14. Project => **Build Active project**.
- 15. Power your kit and connect it via USB connection to target machine (PC).
- 16. Debug the program.
- 17. **DebugSymbol** => **Debug as** => **Universal Debug Engine** wait until debugger is started and software is downloaded.
- 18. Place a breakpoint by double click.
- 19. Start software and debug.

1.2 Installing BIFACES Software

As mentioned before, binaries for flashing TC39x board are built through the BIFACES SDK. Some already created software workspace environments are provided by Infineon in *BIFACES_XILINX_TEST* folder.

These steps are needed for installation and setting up the BIFACES on Windows PC:

- 1. Run the installation file *BIFACES_V1_0_2_Win64.exe* or *BIFACES_V1_0_2_Win32.exe*.
- 2. Select folder C: \Tools \BifacesWin64 for installation.
- 3. Select full installation type.
- 4. After finishing the setup, run StartBifaces.bat in C:\Tools\BifacesWin64 folder.
- During startup there is a prompt for workspace directory, browse through BIFACES_XILINX_TEST folder and select one of them, for example, select BIFACE_STEPA_XILINX_UC01_PING_ANYSPEED_160MBHIGHSPEED and click Launch (Figure 1). Take care to select StepA or StepB version of the software, depending on your TC39x board version.



Figure 1: Selecting SDK Workspace Directory

6. Before starting the application build, change the TriCore GNUC path in /BaseFramework_tC39/1_ToolEnv/0_Build/1_Config/Config_Tricore_Gnuc/Co nfig_Gnuc.mk file. The path is defined in folder where HIGHTEC TriCore is installed (see chapter 1.1 HIGHTECH TriCore Entry Tool Chain Setup). In our case the path is –

B_GNUC_TRICORE_PATH := C: \HIGHTEC \toolchains \tricore \v4.9.1.0.

- 7. The application test source files are located under /BaseFramework_tC39/0_Src/AppSw/Tricore/Main folder. All our tests are created in Cpu0_Main.c file.
- 8. If no editing the source is done, click **Build** to generate output files.
- 9. Output files are located in /BaseFramework_tC39/2_Out/Tricore_Gnuc folder. File that is used for programming the Aurix TC39x board has the extension .hex (BaseFramework_TC39A_Tc.hex).

1.3 Flashing the AURIX TC39x Board with Infineon MemTool

After successfully building test files and generating output files, next step is to actually download the test binary to Aurix TC39x board. This is done with another Infineon software tool called Infineon MemTool. Installation of this tool is straight forward Unzip Infineon-Memtool-DT-v04_75-EN.zip and run Infineon-Memtool-DT-v04_75-EN.exe. After finishing installation, run the Infineon MemTool application on PC.

Steps to program the board:

- 1. Connect the board's power supply.
- 2. Connect the board to PC with Micro-B USB cable.
- 3. Power up the board.
- 4. Set the target configuration (Figure 2):

| LOGIC BRICKS | Test Environment for TC39x Aurix Board | Xylon |
|------------------------------|---|----------------|
| July 24 th , 2019 | CONFIDENTIAL | Version: v1.01 |
| | File Target Device Log Help | |

Figure 2: Setting Target Configuration

5. Create a new configuration (Figure 3):

| | : | | 27.1 | | | | _ |
|---------------------------------------|---------------|----------------|-----------|------|------|---------|------------|
| C:\Users\Abler\ Additional Filter: | Documents \In | tineon \IM I 4 | /\largets | \ | | | |
| Files in folder : | | | | | | Show de | escription |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| Dofout | Now | Ser Cor | | Edit | Bemo | ve | |

Figure 3: Creating New Configuration

6. Select configuration depending on the TriCore version, StepA or StepB, here StepA is selected (Figure 4):



Test Environment for TC39x Aurix Board



July 24th, 2019

CONFIDENTIAL

Version: v1.01

| Create or use default | x |
|--|----------|
| C Create a new target configuration step by step | |
| Se a default target configuration | |
| | <u> </u> |
| | E |
| a most native and the second s | |
| Triboard with TC38x A-Step (DAS) | |
| Triboard with TC39x A-Step (DAS) | |
| millionard with TC30x B-Step (DAS) | |
| i | |
| н 🖶 VC2000Ц | - |
| Triboard with TC39x A-Step (DAS) Init TLE35584 C-Step on connect switch off FLASH error traps | |
| | |
| Keack Finish Cancel Help | |

Figure 4: Selecting New Configuration

7. Save configuration and close the dialogue window (Figure 5):





Figure 5: Saving New Configuration

8. Connect to target (Figure 6):

| File : | | FLASH/OTP - Memory Device DF_UCBS: 24 Kbyte Data Flash 0 UCB (not ready) | Enable |
|----------|---|--|---------------------------------|
| | Open File Select All Add Sel. >> Save As Read Edit | Index Start End Size Erase 0 0x4F400000 0x4F4001FF 512 F 1 0x4F400200 0x4F4003FF 512 Program 2 0x4F400000 0x4F4003FF 512 Verity 4 0x4F400000 0x4F4003FF 512 Verity 5 0x4F400000 0x4F4003FF 512 Verity 6 0x4F400000 0x4F4000BFF 512 SW Protect 7 0x4F401000 0x4F4013FF 512 HW Protect 9 0x4F401200 0x4F4013FF 512 Info 10 0x4F401200 0x4F4013FF 512 Verity 9 0x4F401200 0x4F4013FF 512 Info 10 0x4F401200 0x4F4013FF 512 Verity 4 IIII Verity State State | Program all Verify all st |
| infineon | Connect | arget not connected Help | Exit |

Figure 6: Connecting PC to Target (Board)



9. The result should be as presented in Figure 7:

| Tool | | |
|------------|---------------------------|---|
| Disconnect | Ready for Memtool Command | _ |
| | | |

Figure 7: Connection to Target Established Info

10. If the user is not sure if the status of the configuration (target) is really correct, to avoid it being accidentally used (then connection fails), it would be the best to remove it and repeat previous steps in order to apply correct target configuration, taking care to select StepA or StepB TriCore (Figure 8):

| C:\Users\Abler\E | ocuments\Infi | neon \IMT 4.7\" | Targets \ | | • |
|--------------------|---------------|-----------------|-----------|-------|---------------|
| Additional Filter: | | | | | - |
| Files in folder : | | | | 🔽 Sho | w description |
| | | | | | |

Figure 8: Removing Potentially Incorrect Target Configuration

11. After step 9, user selects the generated BIFACES . hex file (Figure 9):

| Logic BRICKS [™] | | Test Enviro Au | Χί | noly | | |
|------------------------------|--|---|--|-------------------|---|-----------|
| July 24 th , 2019 | | CON | IFIDENTIAL | | Version: v1.01 | |
| | Infineon - Memtool 4.7 on File Target Device Log File : D:\/MShared\BaseFramework D:\/MShared\BaseFramework D:\/MShared\BaseFramework Open Hex File Organize | Triboard with TC39x A-Step (DAS) Help ork_TC39A_MASTER_Tc.hex @ Open File ework_TC39 > 2_Out > Tricore r Name @ Gnuc_Files @ BaseFramework_TC39_tc.he | FLASH/OTP - Memory Device PFLASH0: 3 MByte OnChip Program Index Start End Gnuc > Date modified 10.9.2018.10:06 x 14.6.2018.16:25 | FLASH (not ready) | ✓ ✓ Enat Erase Search Tricore_Gnuc Size Size 68 KB | Ne 23 |
| | Subversion Videos Computer System (C:) Data (D:) File na | ime: BaseFramework_TC39_tc.hex | : | · | HEX-Files (*.hex) Open Car | v icel |

Figure 9: Selecting Programming File

12. After opening the file, click **Select All** (Figure 10):

| File : | | |
|---|----------|--------------|
| D:\HSSL_IP_IFX\BIFACES_XILIN | X_TEST | BIFACE_STEP. |
| 0x80000020 - 0x800048C3 0x800048C8 - 0x800048C3 | <u> </u> | Open File |
| 0x802FE780 - 0x802FE78D 0x802FE7A0 - 0x802FE7AD | = | Select All |
| 0x802FE7C0 - 0x802FE7CD 0x802FE7E0 - 0x802FE7ED | | Add Sel. >> |
| 0x802FE800 - 0x802FE80D 0x802FE820 - 0x802FE82D 0x802FE840 - 0x802FE84D | | Save As |

Figure 10: Address Selection

- 13. Slide down and unselect all the values starting with 0xAF4X XXXX, the last 8 values (Figure 10).
- 14. Now, click **Add Sel**, and then **Program All** button. After programming is done, message is issued. Power-cycle the board, and at that step, board is flashed and ready for testing with the Slave side (FPGA).



Test Environment for TC39x Aurix Board



CONFIDENTIAL

Version: v1.01

2 Revision History

| Version | Date | Author | Approved by | Note |
|---------|-------------|------------|-------------|---|
| 1.00 | 08.05.2019. | A. Popović | R. Končurat | Initial Xylon release |
| 1.01 | 24.07.2019. | S. Opačić | R. Končurat | Added information on selecting between StepA and StepB software versions with regards to the Aurix TriBoard platform version (TC39x Auriox Board). |