NAND Host Controller IP
Integration Manual

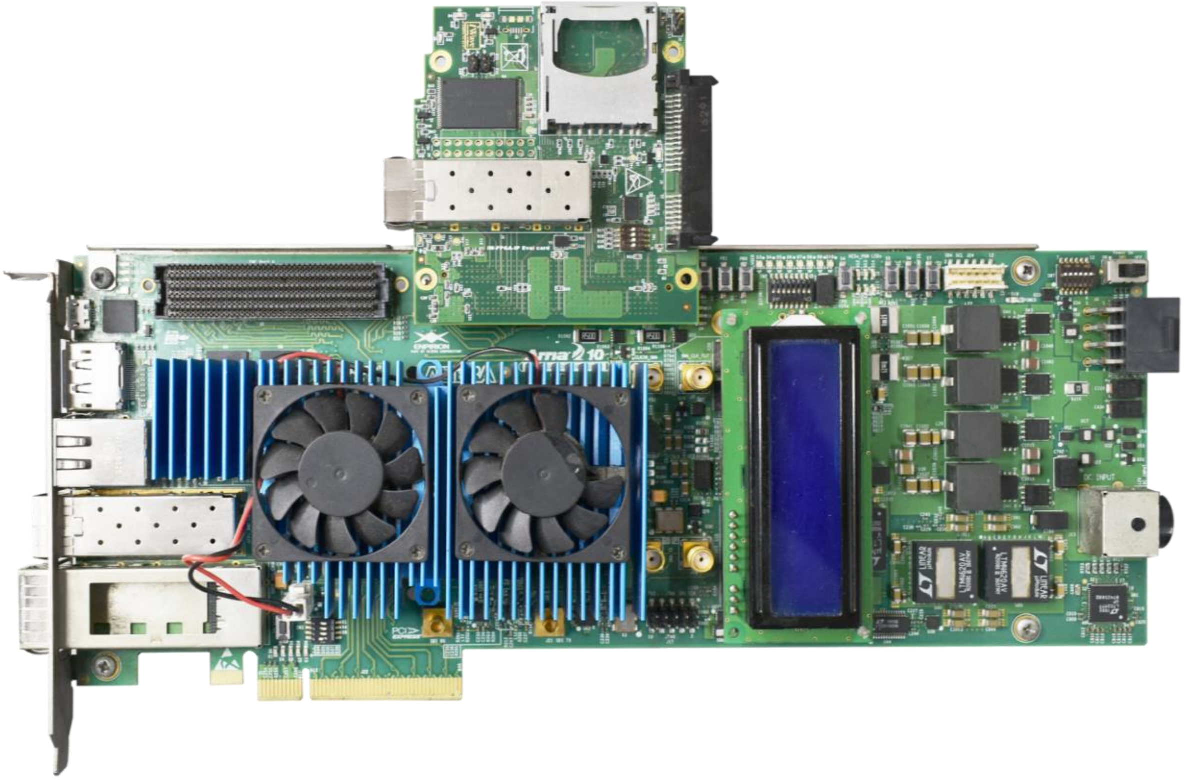


Table of Content

List Of Figures

List Of Tables

1 Introduction

1.1 Purpose

The purpose of this document is to describe Nand Host Controller IP Integration details.

1.2 Reference Document

- iW- EMFBZ-DS-01-R1.0-REL1.2

1.3 Overview

Nand Flash Host Controller interfaces the User and NAND flash via FMC using FPGA GPIO's. So, this IP forms a bridge between the NAND flash and User (Nios processor), enabling the storing, reading and erasing the data in NAND flash. This IP will issue the necessary command address and controls all the necessary actions required to program, erase and read data using NAND flash.

1.4 Acronyms and Abbreviations

Table 1: Acronyms & Abbreviations

Term	Meaning
FPGA	Field Programmable Gate Array
GPIO	General purpose input output
FMC	FPGA Mezzanine Card
LUT	Look Up Table
IO	Input and Output
FF	Flip Flop

2 IP Configuration and Instantiation

2.1 Example design

The NAND host controller IP example design mainly consists of

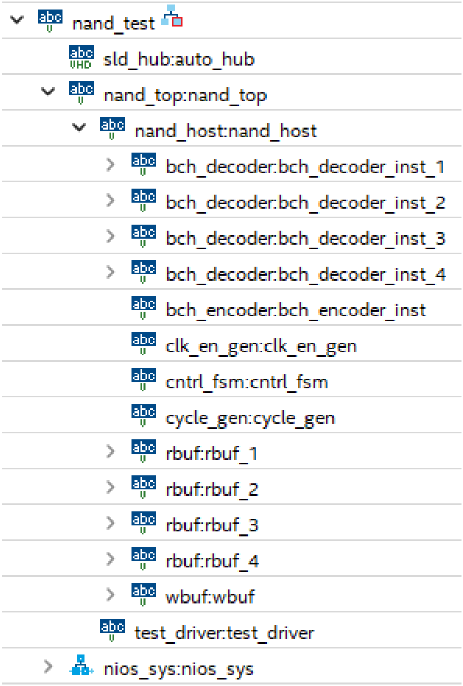
1. **Nios soft core:** User can provide the inputs to run the different tests using the bare metal code running in the Nios Soft processor. This will pass the different control signals to SD host controller based on the user selection.
2. **NAND Test Driver**: NAND test driver is responsible for generating the control signal for the NAND IP. The interface between the test driver and NAND IP is custom interface and these signals will be driven by the test driver based on the test cases selected by user.
3. **NAND IP**: NAND IP is the design under test and controls the access to the external NAND chip based on the inputs from test driver IP Configuration

Note:

- **Example design is compiled for Arria-10 Gx dev. kit board with Quartus Prime Standard Edition 17.0 version.**
- **Example design currently supports single die.**
- **Read ID can be changed in test driver refer chapter 3.**

2.2 IP Configuration

The Nand Flash Host Controller IP is instantiated in the design as shown in the figure below.



```
nand_test
    sld_hub:auto_hub
    nand_top:nand_top
        nand_host:nand_host
            bch_decoder:bch_decoder_inst_1
            bch_decoder:bch_decoder_inst_2
            bch_decoder:bch_decoder_inst_3
            bch_decoder:bch_decoder_inst_4
            bch_encoder:bch_encoder_inst
            clk_en_gen:clk_en_gen
            cntrl_fsm:cntrl_fsm
            cycle_gen:cycle_gen
            rbuf:rbuf_1
            rbuf:rbuf_2
            rbuf:rbuf_3
            rbuf:rbuf_4
            wbuf:wbuf
        test_driver:test_driver
    nios_sys:nios_sys
```

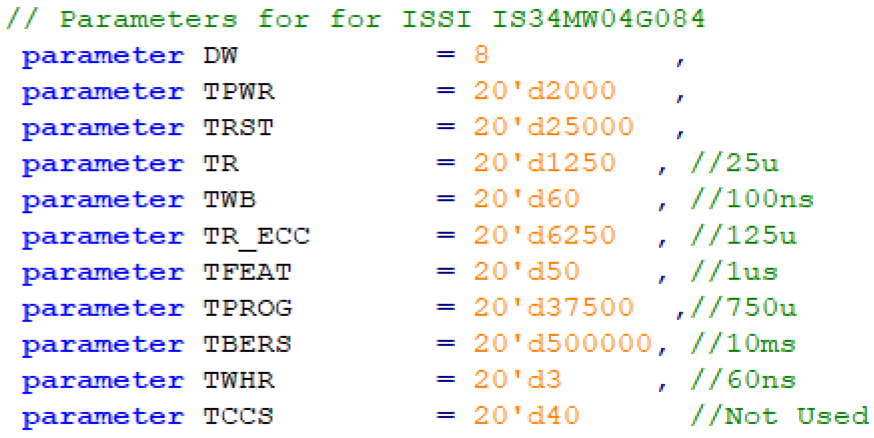
Figure 1 :Design sources where NAND Host IP is instantiated

- The NAND Host Controller’s Example Design Project can be found in the following path

  E:\EMFBZ_Release1.4_NAND_Flash_Controller\iW-EMFBZ-PF-01-R1.0-REL1.4\iW-EMFBZ-FF-01-R1.0-REL1.2\iW-EMFBZ-ED-01-R1.0-REL1.2

Module nand_test is the top module of the project which integrates the top module of NAND IP and Nios Soft Processor. There are few timing parameters which can be modified by the user depending on which Nand flash IC to be used.

Module nand_top. v is the top module of NAND IP which integrates the test_driver. v module and the nand_host. v file.



```
// Parameters for for ISSI IS34MW04G084
 parameter DW          = 8          ,
 parameter TPWR        = 20'd2000   ,
 parameter TRST        = 20'd25000  ,
 parameter TR          = 20'd1250   , //25u
 parameter TWB         = 20'd60     , //100ns
 parameter TR_ECC      = 20'd6250   , //125u
 parameter TFEAT       = 20'd50     , //1us
 parameter TPROG       = 20'd37500  ,//750u
 parameter TBERS       = 20'd500000, //10ms
 parameter TWHR        = 20'd3      , //60ns
 parameter TCCS        = 20'd40       //Not Used
```

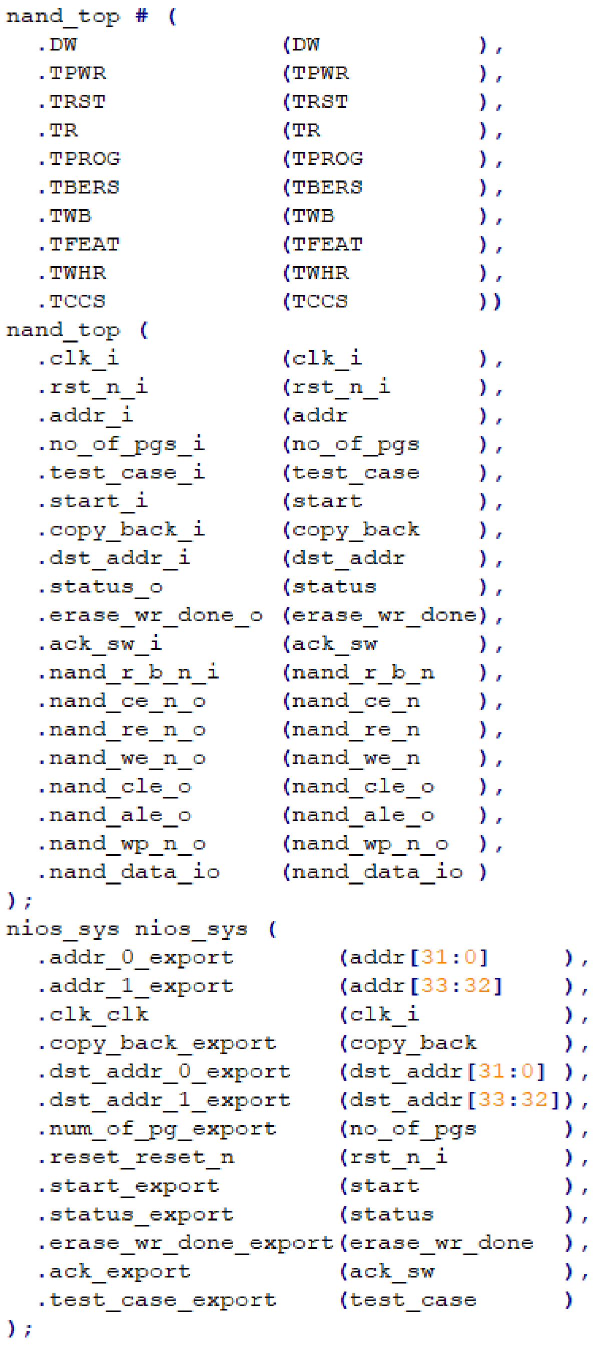
Figure 2 :Parameters for ISSI IS34MW04G084

1. The figure above shows the configurable parameters in nand_test. v file. Parameters like TRST device reset time, TPROG Maximum PROGRAM PAGE time and parameters which decides the timing parameters are controlled directly from the nand_test. v module as shown in the figure IS34MW04G084
2. User need to make the required changes depending upon the NAND flash IC.

NOTE: All the above parameters should be with respect to the frequency of 50MHz and values shown is for ISSI IS34MW04G084 part.

2.3 Steps to Instantiate NAND HOST IP

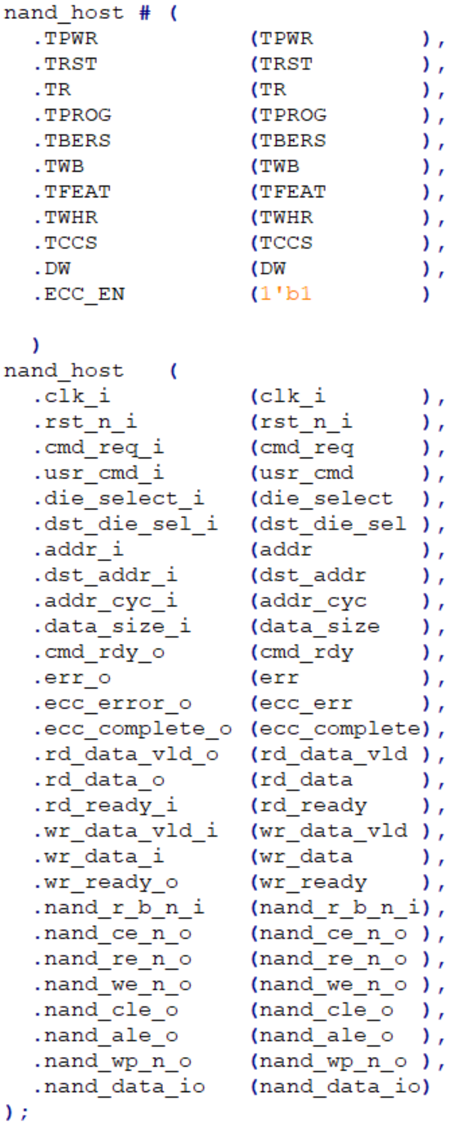
1) Instantiate the top module of Nios processor and Nand IP in the module nand_test. v is shown below,



```
nand_top # (
  .DW              (DW           ),
  .TPWR            (TPWR         ),
  .TRST            (TRST         ),
  .TR              (TR           ),
  .TPROG           (TPROG        ),
  .TBERS           (TBERS        ),
  .TWB             (TWB          ),
  .TFEAT           (TFEAT        ),
  .TWHR            (TWHR         ),
  .TCCS            (TCCS         ))
nand_top (
  .clk_i           (clk_i        ),
  .rst_n_i         (rst_n_i      ),
  .addr_i          (addr         ),
  .no_of_pgs_i     (no_of_pgs    ),
  .test_case_i     (test_case    ),
  .start_i         (start        ),
  .copy_back_i     (copy_back    ),
  .dst_addr_i      (dst_addr     ),
  .status_o        (status       ),
  .erase_wr_done_o (erase_wr_done),
  .ack_sw_i        (ack_sw       ),
  .nand_r_b_n_i    (nand_r_b_n   ),
  .nand_ce_n_o     (nand_ce_n    ),
  .nand_re_n_o     (nand_re_n    ),
  .nand_we_n_o     (nand_we_n    ),
  .nand_cle_o      (nand_cle_o   ),
  .nand_ale_o      (nand_ale_o   ),
  .nand_wp_n_o     (nand_wp_n_o  ),
  .nand_data_io    (nand_data_io )
);
nios_sys nios_sys (
  .addr_0_export       (addr[31:0]     ),
  .addr_1_export       (addr[33:32]    ),
  .clk_clk             (clk_i          ),
  .copy_back_export    (copy_back      ),
  .dst_addr_0_export   (dst_addr[31:0] ),
  .dst_addr_1_export   (dst_addr[33:32]),
  .num_of_pg_export    (no_of_pgs      ),
  .reset_reset_n       (rst_n_i        ),
  .start_export        (start          ),
  .status_export       (status         ),
  .erase_wr_done_export(erase_wr_done  ),
  .ack_export          (ack_sw         ),
  .test_case_export    (test_case      )
);
```

Figure 3: Instantiation of nios and nand top modules

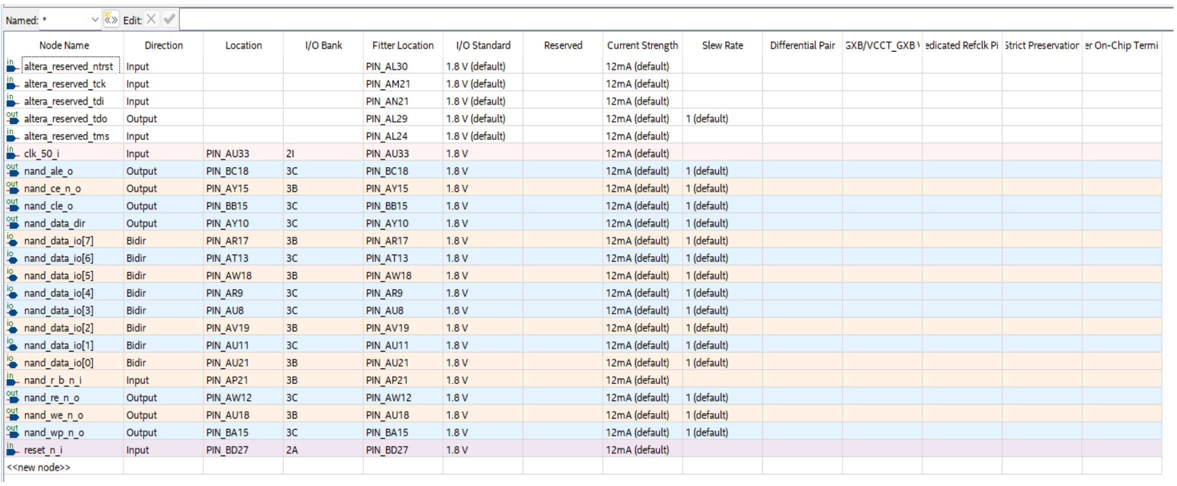
System clock is given to clocking wizard whose output is 50MHz given to both NAND IP and Nios Soft processor.
nand_top. v is the top module of NAND IP which is used to integrate the test_driver. v module with nand_host.v module.



```
nand_host # (
  .TPWR           (TPWR        ),
  .TRST           (TRST        ),
  .TR             (TR          ),
  .TPROG          (TPROG       ),
  .TBERS          (TBERS       ),
  .TWB            (TWB         ),
  .TFEAT          (TFEAT       ),
  .TWHR           (TWHR        ),
  .TCCS           (TCCS        ),
  .DW             (DW          ),
  .ECC_EN         (1'b1        )

  )
nand_host   (
  .clk_i          (clk_i       ),
  .rst_n_i        (rst_n_i     ),
  .cmd_req_i      (cmd_req     ),
  .usr_cmd_i      (usr_cmd     ),
  .die_select_i   (die_select  ),
  .dst_die_sel_i  (dst_die_sel ),
  .addr_i         (addr        ),
  .dst_addr_i     (dst_addr    ),
  .addr_cyc_i     (addr_cyc    ),
  .data_size_i    (data_size   ),
  .cmd_rdy_o      (cmd_rdy     ),
  .err_o          (err         ),
  .ecc_error_o    (ecc_err     ),
  .ecc_complete_o (ecc_complete),
  .rd_data_vld_o  (rd_data_vld ),
  .rd_data_o      (rd_data     ),
  .rd_ready_i     (rd_ready    ),
  .wr_data_vld_i  (wr_data_vld ),
  .wr_data_i      (wr_data     ),
  .wr_ready_o     (wr_ready    ),
  .nand_r_b_n_i   (nand_r_b_n_i),
  .nand_ce_n_o    (nand_ce_n_o ),
  .nand_re_n_o    (nand_re_n_o ),
  .nand_we_n_o    (nand_we_n_o ),
  .nand_cle_o     (nand_cle_o  ),
  .nand_ale_o     (nand_ale_o  ),
  .nand_wp_n_o    (nand_wp_n_o ),
  .nand_data_io   (nand_data_io)
);
```

Figure 4 : Nand host IP instantiation

2) Give the required clock, reset Pin/IO constraints for NAND Interface in the Quartus Prime Standard Edition Tool’s Assignment→Pin Planner as shown in the Figure Below and compile the custom design with NAND host IP.



Named: * Edit:

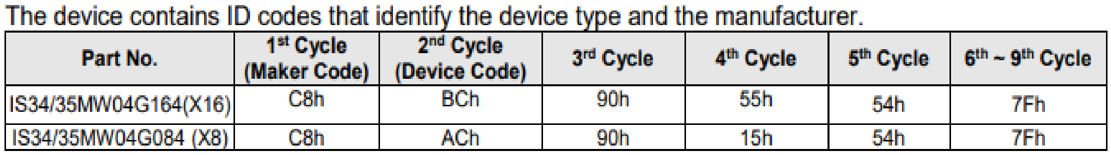
Node Name	Direction	Location	I/O Bank	Fitter Location	I/O Standard	Reserved	Current Strength	Slew Rate	Differential Pair	GXB/VCCT_GXB V	edicated Refclk Pi	Strict Preservatior	er On-Chip Termi
altera_reserved_ntrst	Input			PIN_AL30	1.8 V (default)		12mA (default)						
altera_reserved_tck	Input			PIN_AM21	1.8 V (default)		12mA (default)						
altera_reserved_tdi	Input			PIN_AN21	1.8 V (default)		12mA (default)						
altera_reserved_tdo	Output			PIN_AL29	1.8 V (default)		12mA (default)	1 (default)					
altera_reserved_tms	Input			PIN_AL24	1.8 V (default)		12mA (default)						
clk_50_i	Input	PIN_AU33	2I	PIN_AU33	1.8 V		12mA (default)						
nand_ale_o	Output	PIN_BC18	3C	PIN_BC18	1.8 V		12mA (default)	1 (default)					
nand_ce_n_o	Output	PIN_AY15	3B	PIN_AY15	1.8 V		12mA (default)	1 (default)					
nand_cle_o	Output	PIN_BB15	3C	PIN_BB15	1.8 V		12mA (default)	1 (default)					
nand_data_dir	Output	PIN_AY10	3C	PIN_AY10	1.8 V		12mA (default)	1 (default)					
nand_data_io[7]	Bidir	PIN_AR17	3B	PIN_AR17	1.8 V		12mA (default)	1 (default)					
nand_data_io[6]	Bidir	PIN_AT13	3C	PIN_AT13	1.8 V		12mA (default)	1 (default)					
nand_data_io[5]	Bidir	PIN_AW18	3B	PIN_AW18	1.8 V		12mA (default)	1 (default)					
nand_data_io[4]	Bidir	PIN_AR9	3C	PIN_AR9	1.8 V		12mA (default)	1 (default)					
nand_data_io[3]	Bidir	PIN_AU8	3C	PIN_AU8	1.8 V		12mA (default)	1 (default)					
nand_data_io[2]	Bidir	PIN_AV19	3B	PIN_AV19	1.8 V		12mA (default)	1 (default)					
nand_data_io[1]	Bidir	PIN_AU11	3C	PIN_AU11	1.8 V		12mA (default)	1 (default)					
nand_data_io[0]	Bidir	PIN_AU21	3B	PIN_AU21	1.8 V		12mA (default)	1 (default)					
nand_r_b_n_i	Input	PIN_AP21	3B	PIN_AP21	1.8 V		12mA (default)						
nand_re_n_o	Output	PIN_AW12	3C	PIN_AW12	1.8 V		12mA (default)	1 (default)					
nand_we_n_o	Output	PIN_AU18	3B	PIN_AU18	1.8 V		12mA (default)	1 (default)					
nand_wp_n_o	Output	PIN_BA15	3C	PIN_BA15	1.8 V		12mA (default)	1 (default)					
reset_n_i	Input	PIN_BD27	2A	PIN_BD27	1.8 V		12mA (default)						
<<new node>>													

Figure 5:Pin Pin Constraints in the Quartus Pin Planner

NOTE: These constraints are in accordance to example design. Please change the pin constraints for NAND interface as required for custom design. And also define the constraints for clock, reset pins accordingly.

3 Design modification to be done for Custom Board

- Update the FPGA part number according to the FPGA device available in the custom board
- Update the complete design for the selected FPGA device
- Update the NAND flash timing parameters according to the device selected.
- Make sure that 50Mhz input is provided properly to the design. If the clock available is other than 50Mhz, use the clocking wizard to generate the 50Mhz clock as done in example design.
- Update the pin constraints for NAND interface, clock and reset pins in the Quartus Pin Planner
- Recompile the design to generate the new binaries and use the .sopcinfo file to create the new application project in Nios Eclipse tool.
- Read ID parameters in Example design for ISSI IS34MW04G084 is given in figure below



The device contains ID codes that identify the device type and the manufacturer.

Part No.	1st Cycle (Maker Code)	2nd Cycle (Device Code)	3rd Cycle	4th Cycle	5th Cycle	6th ~ 9th Cycle
IS34/35MW04G164(X16)	C8h	BCh	90h	55h	54h	7Fh
IS34/35MW04G084 (X8)	C8h	ACh	90h	15h	54h	7Fh

Figure 6: Read ID table snap from micron DATA sheet

- User can change these parameters in accordance to the Device ID in test driver.

4 Test setup

4.1 Quartus Prime SJ 17.0

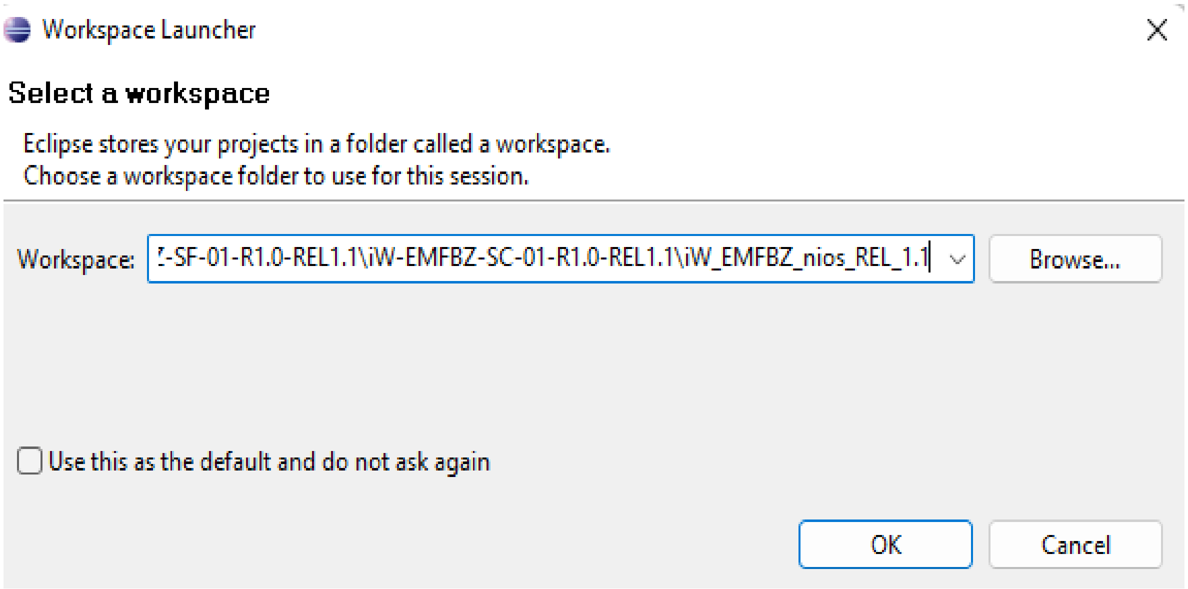
This section explains the procedure and detailed information for installing the Quartus Prime SJ 17.0 in the host PC and import the application project.
Refer the below link to install the Quartus Prime SJ 17.0 in the host PC.
https://www.intel.com/content/www/us/en/software-kit/669514/intel-quartus-prime-standard-edition- design-software-version-17-0-for-windows.html

4.2 Test Requirements

- Intel Arria 10 GX FPGA Development kit- 10AX115S2F45I1SG
- USB type A to Micro B cable
- Host PC
- Power supply
- iWave FMC daughter card

4.3 Launching Eclipse tool

- The project path should be imported to Eclipse workspace.
  For Example: ...\E:\EMFBZ_Release1.4_NAND_Flash_Controller\iW-EMFBZ-PF-01-R1.0-REL1.4\iW-EMFBZ-SF-01-R1.0-REL1.4\iW-EMFBZ-SC-01-R1.0-REL1.4\iW_EMFBZ_nios_REL_1.2
- Launch Nios II Software Build tool for Eclipse 17.0 by selecting workspace path of the project directory and click the OK button as shown in the figure below



Workspace Launcher

Select a workspace

Eclipse stores your projects in a folder called a workspace.
Choose a workspace folder to use for this session.

Workspace: Z-SF-01-R1.0-REL1.1\iW-EMFBZ-SC-01-R1.0-REL1.1\iW_EMFBZ_nios_REL_1.1

□ Use this as the default and do not ask again

Figure 7 : Workspace launcher window of Eclipse tool

- The new Eclipse tool window opens as illustrated below

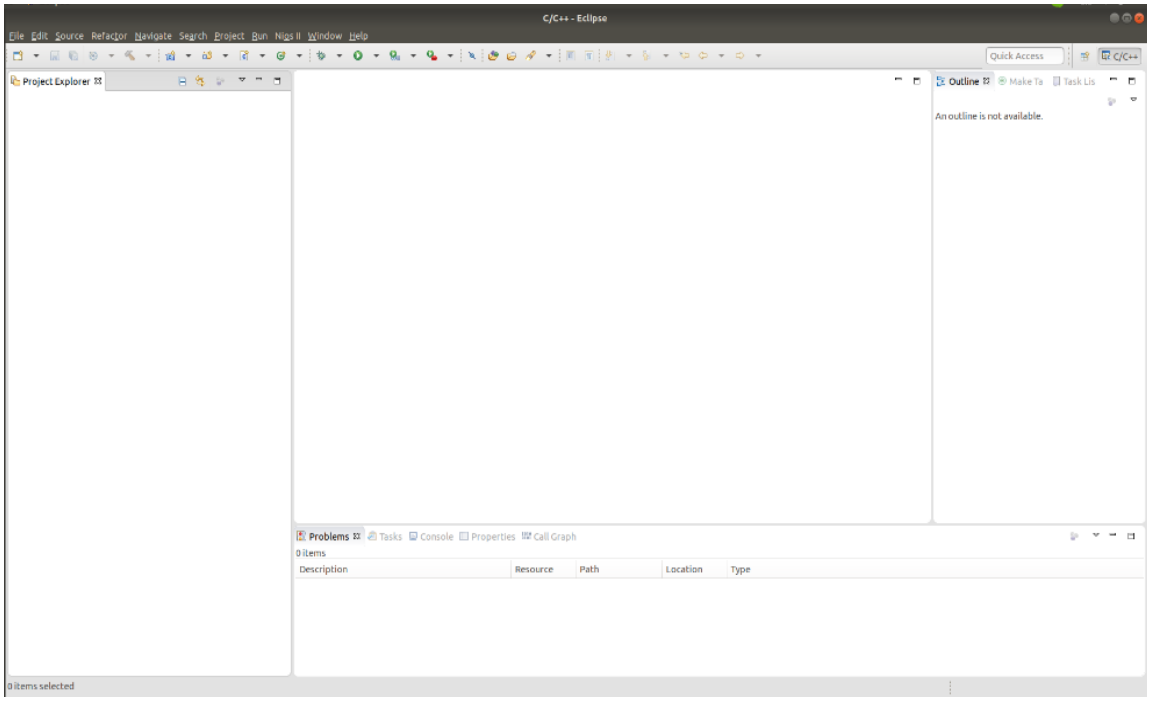
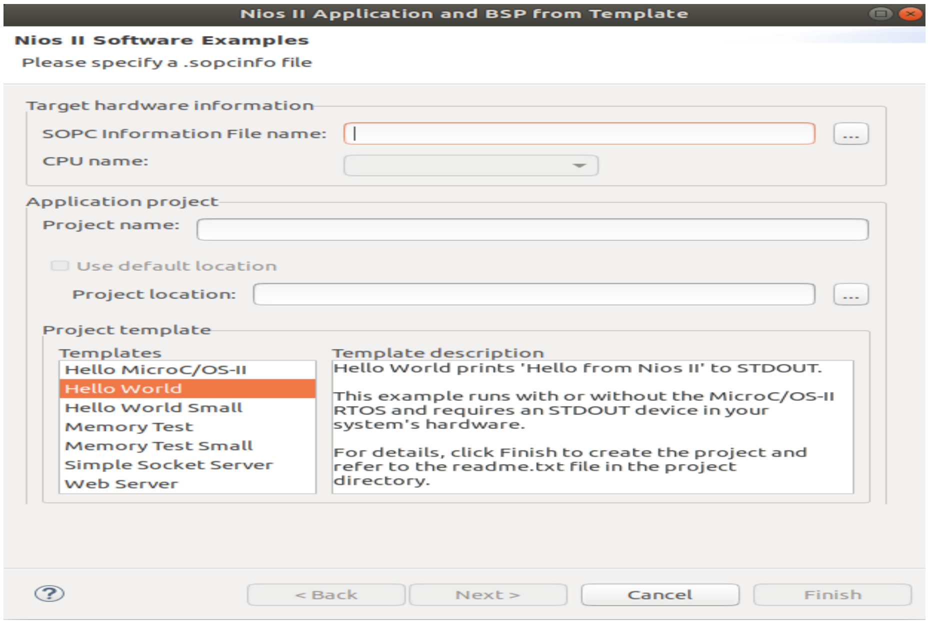


Figure 8: NiosII Eclipse tool window

- Once Nios Eclipse is launched go to File --> New--> Nios II Application and BSP from Template. The new Eclipse tool window opens as shown below


Nios II Application and BSP from Template

Nios II Software Examples

Please specify a .sopcinfo file

Target hardware information

SOPC Information File name:

CPU name:

Application project

Project name:

□ Use default location

Project location:

Project template

Templates
- Hello MicroC/OS-II
- Hello World
- Hello World Small
- Memory Test
- Memory Test Small
- Simple Socket Server
- Web Server

Template description

Hello World prints 'Hello from Nios II' to STDOUT.

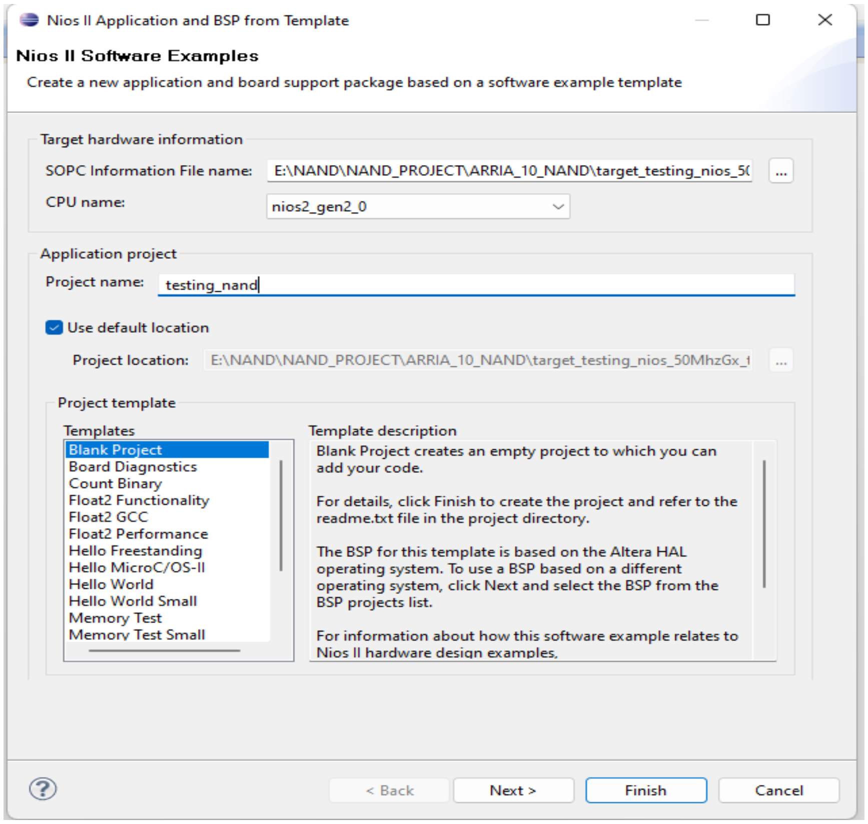
This example runs with or without the MicroC/OS-II RTOS and requires an STDOUT device in your system's hardware.

For details, click Finish to create the project and refer to the readme.txt file in the project directory.

Figure 9: NiosII Application and BSP template -1

- The FPGA (. sopcinfo) file can be found in the following path

  …...\E:\EMFBZ_Release1.4_NAND_Flash_Controller\iW-EMFBZ-PF-01-R1.0-REL1.4\iW-EMFBZ-SF-01-R1.0-REL1.4\iW-EMFBZ-SC-01-R1.0-REL1.4\ nios_sys. sopcinfo
- As shown in the figure, fill the "Target Hardware information" and "Application project" sections. In the project templates click on Blank project and click finish. This creates the Nios Application and BSP template directory on the Host PC



Nios II Application and BSP from Template

Nios II Software Examples

Create a new application and board support package based on a software example template

Target hardware information

SOPC Information File name: E:\NAND\NAND_PROJECT\ARRIA_10_NAND\target_testing_nios_5(

CPU name: nios2_gen2_0

Application project

Project name: testing_nand

☑ Use default location

Project location: E:\NAND\NAND_PROJECT\ARRIA_10_NAND\target_testing_nios_50MhzGx_1

Project template

Templates

- Blank Project
- Board Diagnostics
- Count Binary
- Float2 Functionality
- Float2 GCC
- Float2 Performance
- Hello Freestanding
- Hello MicroC/OS-II
- Hello World
- Hello World Small
- Memory Test
- Memory Test Small

Template description

Blank Project creates an empty project to which you can add your code.

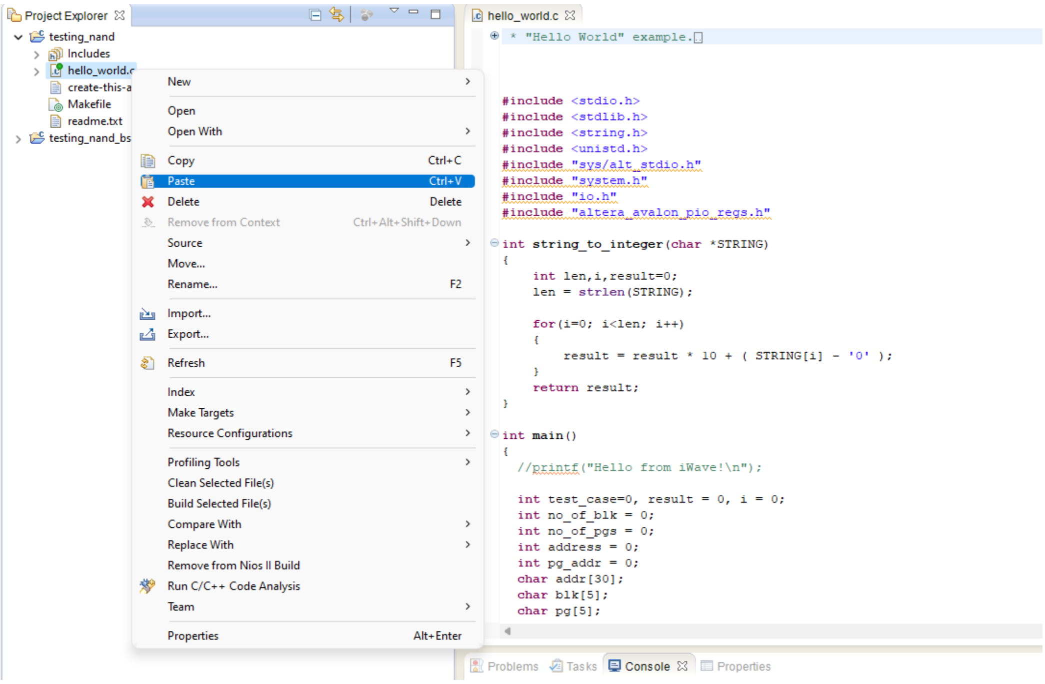
For details, click Finish to create the project and refer to the readme.txt file in the project directory.

The BSP for this template is based on the Altera HAL operating system. To use a BSP based on a different operating system, click Next and select the BSP from the BSP projects list.

For information about how this software example relates to Nios II hardware design examples,

Figure 10: NiosII Application and BSP template -2

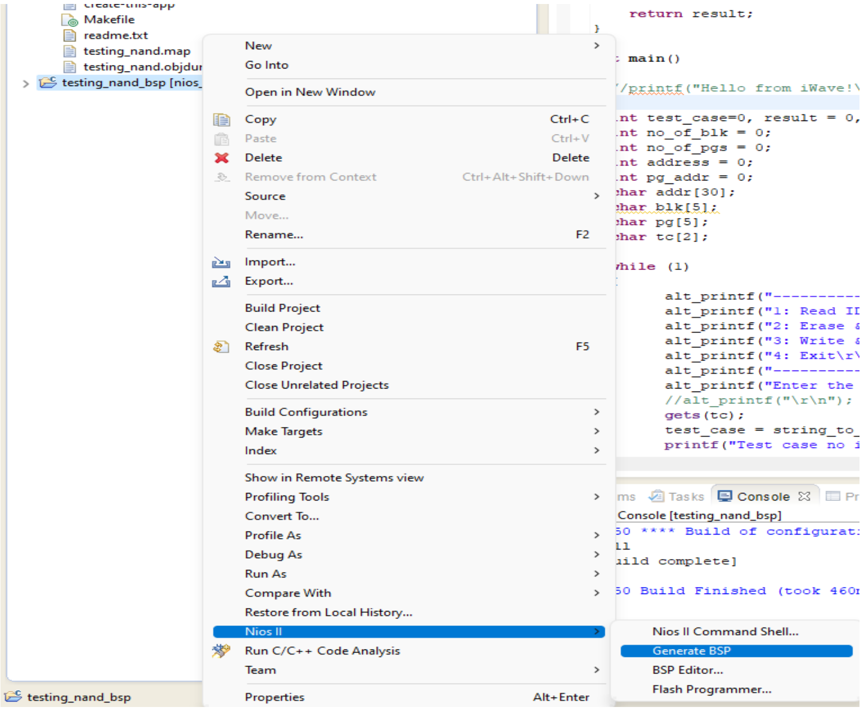
- Copy the source code to application directory
  (For eg: E:\EMFBZ_Release1.4_NAND_Flash_Controller\iW-EMFBZ-PF-01-R1.0-REL1.4\iW-EMFBZ-SF-01-R1.0-REL1.4\iW-EMFBZ-SC-01-R1.0-REL1.4\iW_EMFBZ_nios_REL_1.2\hello_world.c and save it by pressing Ctrl+S on your keyboard



```
 * "Hello World" example.


#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include "sys/alt_stdio.h"
#include "system.h"
#include "io.h"
#include "altera_avalon_pio_regs.h"

int string_to_integer(char *STRING)
{
    int len,i,result=0;
    len = strlen(STRING);

    for(i=0; i<len; i++)
    {
        result = result * 10 + ( STRING[i] - '0' );
    }
    return result;
}

int main()
{
  //printf("Hello from iWave!\n");

  int test_case=0, result = 0, i = 0;
  int no_of_blk = 0;
  int no_of_pgs = 0;
  int address = 0;
  int pg_addr = 0;
  char addr[30];
  char blk[5];
  char pg[5];
```

Figure 11: Copy the source code to application directory

- Build the Nios Application by pressing Ctrl-B on your keyboard
- Generate the BSP by right clicking on application.bsp→ Nios II→Generate BSP. The Nios Application takes some duration to generate it



```
    return result;
}

: main()

'/printf("Hello from iWave!\

.nt test_case=0, result = 0,
.nt no_of_blk = 0;
.nt no_of_pgs = 0;
.nt address = 0;
.nt pg_addr = 0;
:har addr[30];
:har blk[5];
:har pg[5];
:har tc[2];

vhile (1)

        alt_printf("----------
        alt_printf("1: Read ID
        alt_printf("2: Erase &
        alt_printf("3: Write &
        alt_printf("4: Exit\r\
        alt_printf("----------
        alt_printf("Enter the
        //alt_printf("\r\n");
        gets(tc);
        test_case = string_to_
        printf("Test case no i
```

```
Console [testing_nand_bsp]
50 **** Build of configurat:
11
uild complete]

50 Build Finished (took 460r
```

Figure 12: Generate the BSP

4.4 Nios Programming

This section explains the step-by-step procedure to program the binaries into platform using USB blaster cable

NOTE:

The connection shown in the figure below is made for the Intel Arria 10 GX FPGA Development kit to the HOST PC and it varies for other user boards.

- Connect the USB blaster cable from the host PC to the Micro USB connector of the Intel Arria 10 GX FPGA Development kit and power on the board as shown in the figure below

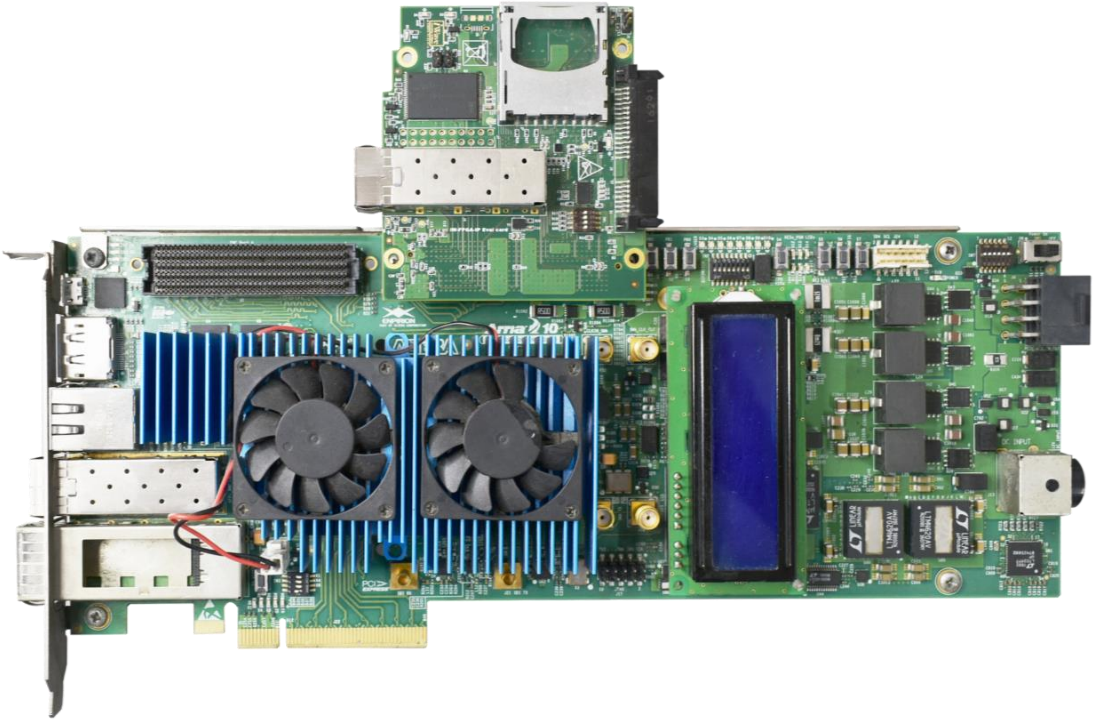


Figure 13: Test Setup of Intel Arria 10 GX FPGA Development kit with iW_FMC card

- Click on Nios II→Quartus Prime Programmer in the tool bar of Nios Eclipse window

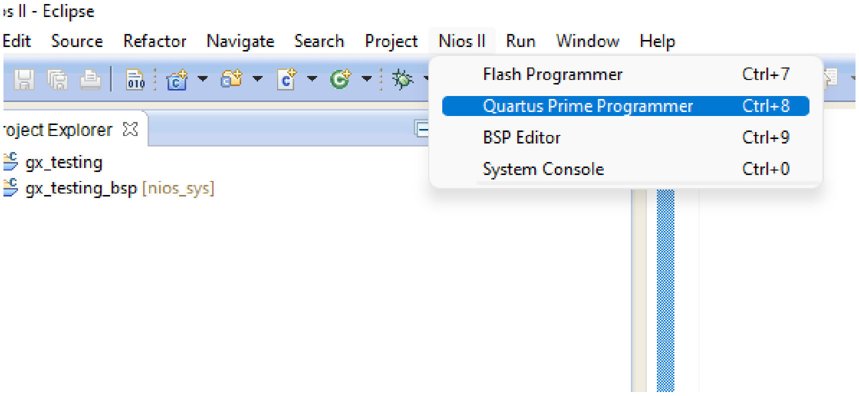
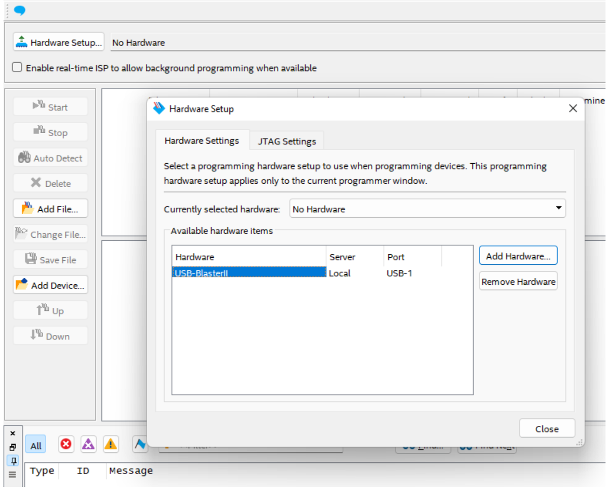

›s II - Eclipse
Edit Source Refactor Navigate Search Project Nios II Run Window Help
Flash Programmer Ctrl+7
Quartus Prime Programmer Ctrl+8
BSP Editor Ctrl+9
System Console Ctrl+0
›oject Explorer
gx_testing
gx_testing_bsp [nios_sys]

Figure 14: Programming .sof file in Quartus Prime Programmer

- Click on hardware setup on the left most side of Quartus Prime Programmer and select the On-board USB Blaster II as shown in the figure below and give close.



Hardware Setup

Select a programming hardware setup to use when programming devices. This programming hardware setup applies only to the current programmer window.

Currently selected hardware: No Hardware

Available hardware items

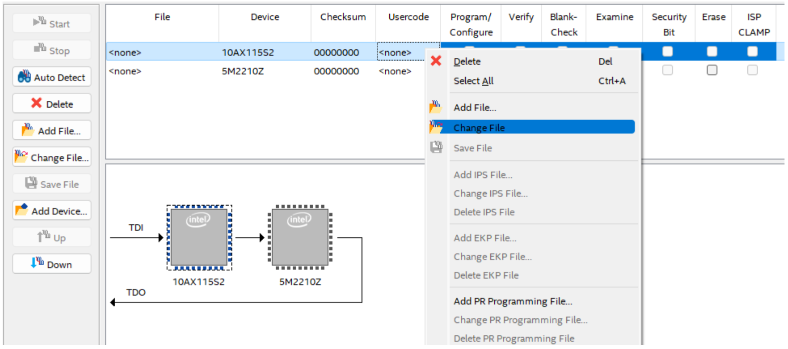
Hardware	Server	Port
USB-BlasterII	Local	USB-1

☐ Enable real-time ISP to allow background programming when available

Type	ID	Message

Figure 15: Selecting On-board USB Blaster II

- Once the USB Blaster II is selected, Click Autodetect present on the left most end of Quartus Prime Programmer and select the required device part number→ 10AX115S2
- Chang the .sof file by right clicking on the 1st device as shown in the figure below

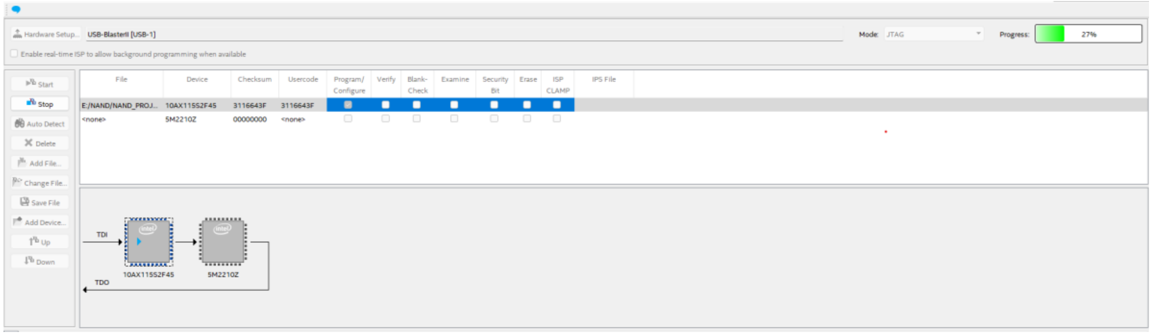


File	Device	Checksum	Usercode	Program/ Configure	Verify	Blank-Check	Examine	Security Bit	Erase	ISP CLAMP
<none>	10AX115S2	00000000	<none>					□	□	□
<none>	5M2210Z	00000000	<none>					□	□	□

Figure 16:Programming with .sof file

.sof file is present in the following path …E:\EMFBZ_Release1.4_NAND_Flash_Controller\iW-EMFBZ-PF-01-R1.0-REL1.4\iW-EMFBZ-FF-01-R1.0-REL1.2\iW-EMFBZ-BN-01-R1.0-REL1.2\ nand_test.sof

- Tick the 1st device and give start for Nios Programming.



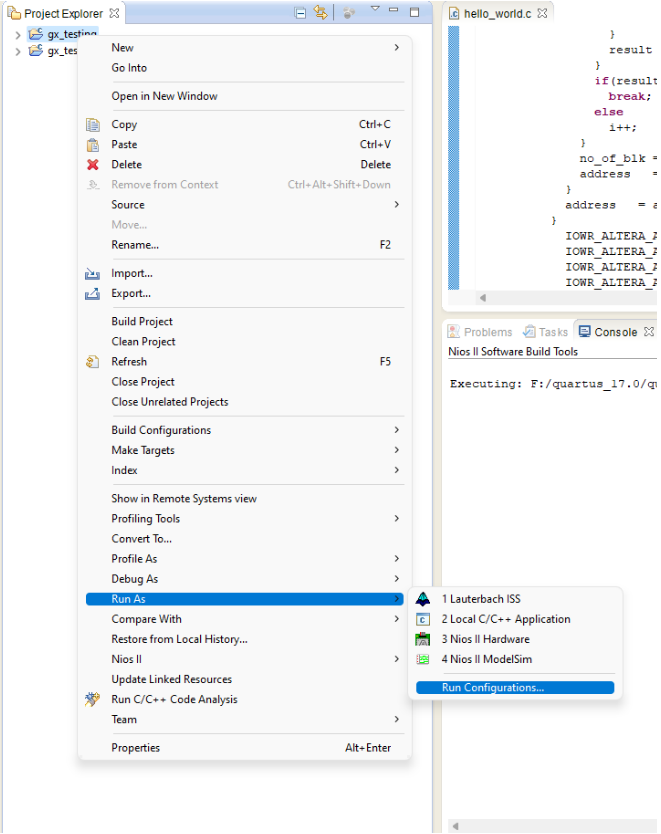
USB-BlasterII [USB-1]

☐ Enable real-time ISP to allow background programming when available

File	Device	Checksum	Usercode	Program/ Configure	Verify	Blank-Check	Examine	Security Bit	Erase	ISP CLAMP	IPS File
E:/NAND/NAND_PROJ...	10AX115S2F45	3116643F	3116643F	☑	☐	☐	☐	☐	☐	☐	
<none>	5M2210Z	00000000	<none>	☐	☐	☐	☐	☐	☐	☐	

Figure 17: Nios Programming in Quartus Prime Programmer

- In the Nios Eclipse window, right click on the Application and click on Run → Run Configurations as shown below to open the Run Configurations dialog box.

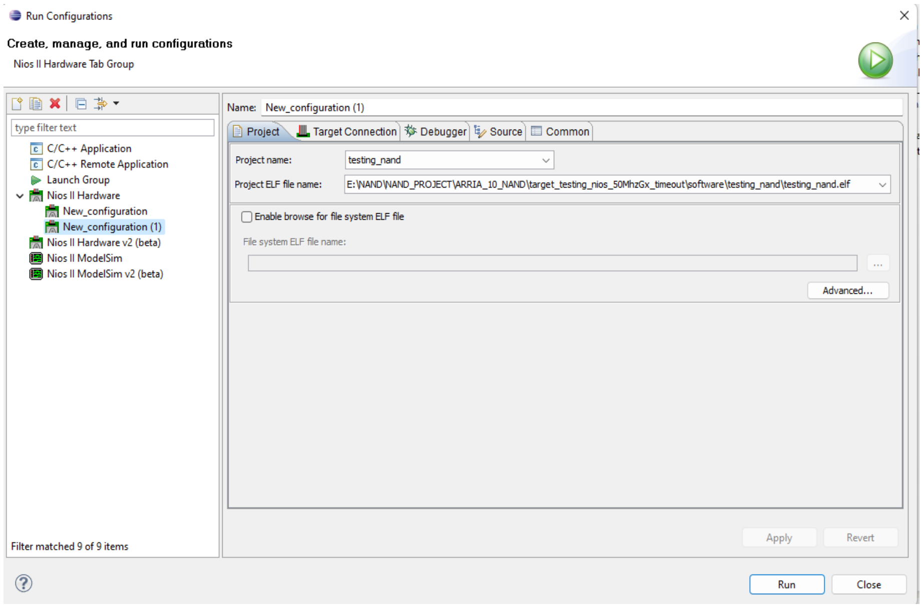


```
            }
            result
          }
          if(result
            break;
          else
            i++;
        }
        no_of_blk =
        address   =
      }
      address    = a
    }
    IOWR_ALTERA_A
    IOWR_ALTERA_A
    IOWR_ALTERA_A
    IOWR_ALTERA_A
```

```
Executing: F:/quartus_17.0/qu
```

Figure 18: Run the Application

- Select the project name and .elf file name as shown in the figure below



Run Configurations

Create, manage, and run configurations

Nios II Hardware Tab Group

Name: New_configuration (1)

Project name: testing_nand

Project ELF file name: E:\NAND\NAND_PROJECT\ARRIA_10_NAND\target_testing_nios_50MhzGx_timeout\software\testing_nand\testing_nand.elf

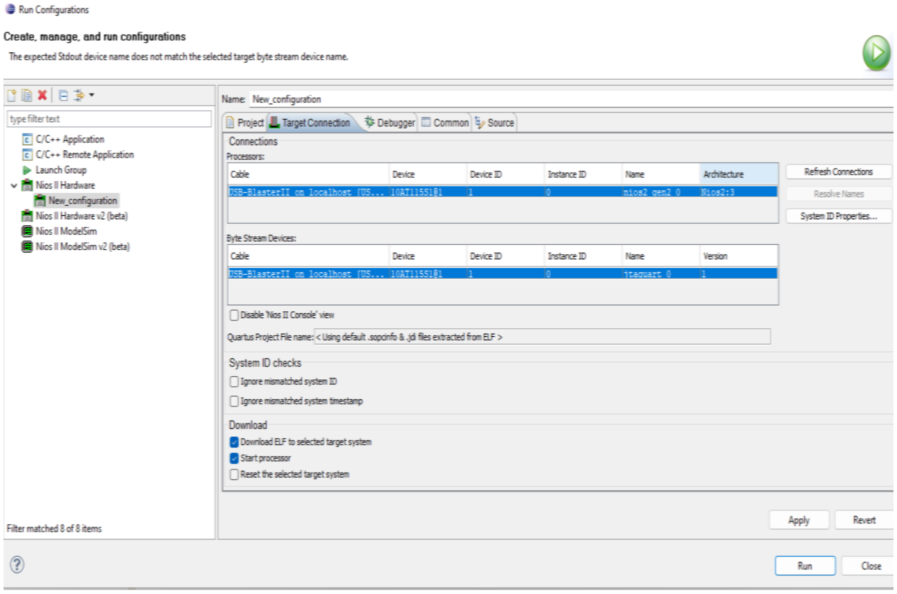
☐ Enable browse for file system ELF file

File system ELF file name:

Filter matched 9 of 9 items

Figure 19: Selecting the project name and .elf file name

- Go to the "Target Connection" tab and make sure there is USB Blaster connection in the "Connection" section by clicking on the Refresh Connections button as shown below



Run Configurations

Create, manage, and run configurations

The expected Stdout device name does not match the selected target byte stream device name.

Name: New_configuration

Connections

Processors:

Cable	Device	Device ID	Instance ID	Name	Architecture
USB-BlasterII on localhost [US...	10AT115S1@1	1	0	nios2_gen2_0	Nios2:3

Byte Stream Devices:

Cable	Device	Device ID	Instance ID	Name	Version
USB-BlasterII on localhost [US...	10AT115S1@1	1	0	jtaguart_0	1

☐ Disable 'Nios II Console' view

Quartus Project File name: < Using default .sopcinfo & .jdi files extracted from ELF >

System ID checks

☐ Ignore mismatched system ID

☐ Ignore mismatched system timestamp

Download

☑ Download ELF to selected target system

☑ Start processor

☐ Reset the selected target system

Filter matched 8 of 8 items

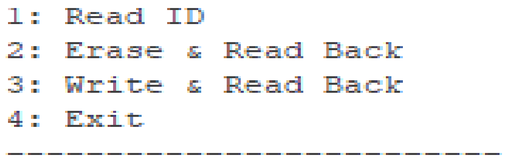
Figure 20: Target Connection tab setting

- Click Apply and Run button in the “Run Configuration” window to start the application.

5 Test Procedure

This section provides the information about how to run the different Nand operations.

- Connect the micro-USB cable (On-board USB Blaster) from the Development kit to the PC
- Power ON the Board, program the binaries into platform using micro-USB cable and give Run Configurations (Refer 2 section). Wait until the Nios II Console appears as shown below

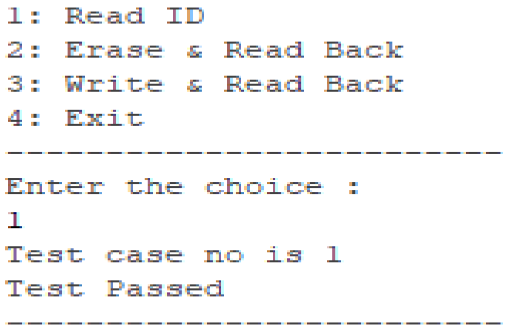


```
1: Read ID
2: Erase & Read Back
3: Write & Read Back
4: Exit
--------------------------
```

Figure 21: Nios II Console prints

5.1 Read ID

- Enter ‘1’ in Nios Console to perform the Read ID operation



```
1: Read ID
2: Erase & Read Back
3: Write & Read Back
4: Exit
--------------------------
Enter the choice :
1
Test case no is 1
Test Passed
--------------------------
```

Figure 22: Read ID

5.2 Erase block

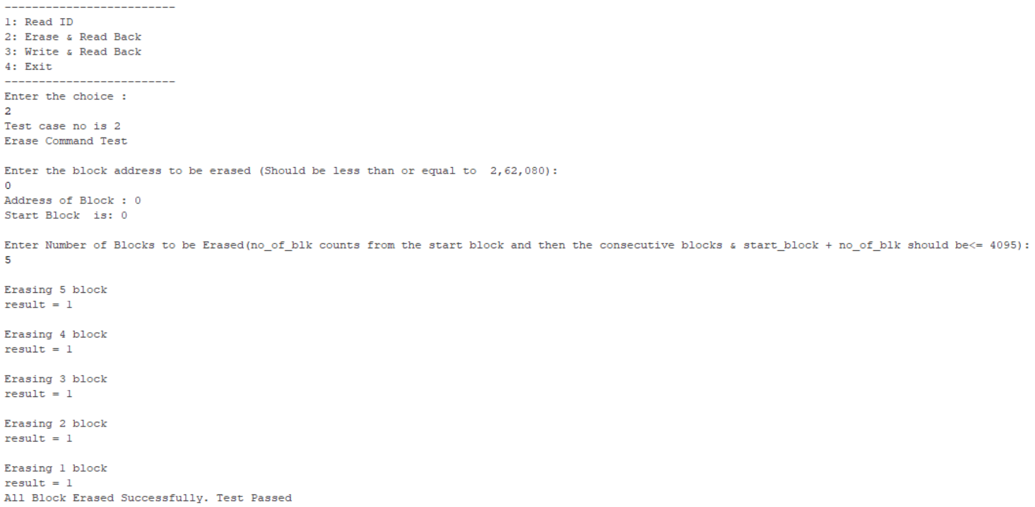
- Enter ‘2’ in Nios Console to perform the NAND erase operation
- Enter the block address to be erased in Hex
- Enter the number of blocks to be erased

Note:

- The number of blocks to be erased includes the start block also. If the start block is 0th block and number of blocks =5, the erase operation happens for the blocks 0,1,2,3 and 4.
- To understand how the start block is calculated using the block address, user can refer the array addressing table in ISSI NAND flash

memory datasheet (ISSI IS34MW04G084).

- The IS34MW04G084 consist of 2 planes with page size of 2K in 1 plane making it a 4G NAND Device
- To select the appropriate block for Erase operation, user needs to send the appropriate address to NAND.NAND addressing table is given below
- Start block starts from 0 to 4095.Addition of start block and number of blocks should not exceed total number of blocks i.e., start_block + number_blocks <= 4095

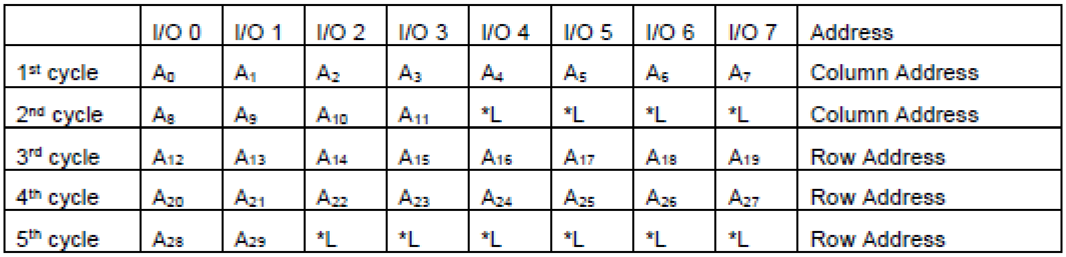


```
-------------------------
1: Read ID
2: Erase & Read Back
3: Write & Read Back
4: Exit
-------------------------
Enter the choice :
2
Test case no is 2
Erase Command Test

Enter the block address to be erased (Should be less than or equal to  2,62,080):
0
Address of Block : 0
Start Block  is: 0

Enter Number of Blocks to be Erased(no_of_blk counts from the start block and then the consecutive blocks & start_block + no_of_blk should be<= 4095):
5

Erasing 5 block
result = 1

Erasing 4 block
result = 1

Erasing 3 block
result = 1

Erasing 2 block
result = 1

Erasing 1 block
result = 1
All Block Erased Successfully. Test Passed
```

Figure 23: Block Erase

Explaining Erase and Read Back test case with example:

The Block address given by user will be decoded to know which block has to be erased and read back.



	I/O 0	I/O 1	I/O 2	I/O 3	I/O 4	I/O 5	I/O 6	I/O 7	Address
1st cycle	A_0	A_1	A_2	A_3	A_4	A_5	A_6	A_7	Column Address
2nd cycle	A_8	A_9	A_{10}	A_{11}	*L	*L	*L	*L	Column Address
3rd cycle	A_{12}	A_{13}	A_{14}	A_{15}	A_{16}	A_{17}	A_{18}	A_{19}	Row Address
4th cycle	A_{20}	A_{21}	A_{22}	A_{23}	A_{24}	A_{25}	A_{26}	A_{27}	Row Address
5th cycle	A_{28}	A_{29}	*L	*L	*L	*L	*L	*L	Row Address

Figure 24: Address decoding in ISSI IS34MW04G084

NOTE: The page size of the ISSI IS34MW04G084 is 2K.

The address that the user provides is 3rd,4th,5th address as shown in above figure.

1. If the user enters block address as 598d i.e.,256h - Bit [16:6] becomes the block address i.e., 9th block (block ranges 0-4095) required for

erase operation.

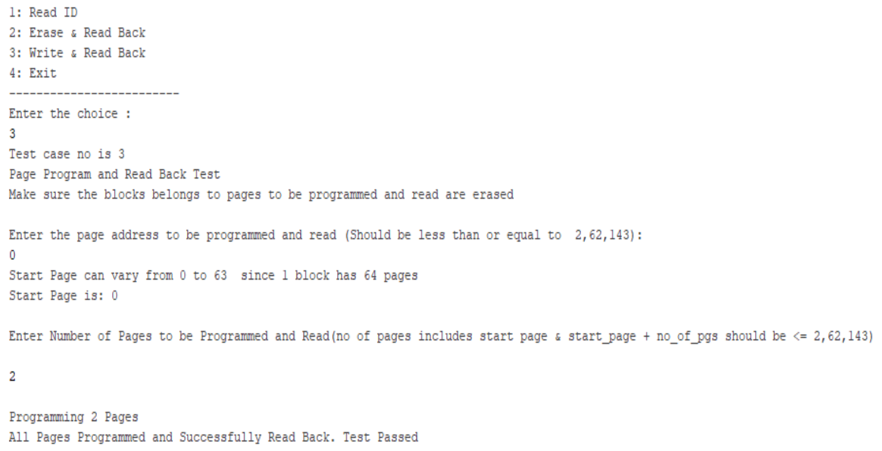
Number of blocks includes start block (9th block):
start block (9th block) + no_of_blocks should be <= 4095
All the 64 pages belonging to the erased block will be read and the result of the operation will be displayed after the 2nd test case is completed.

2. If the user enters page address as 64d i.e.,40h - Bit [16:6] becomes the block address i.e., 1st block (block ranges 0-4095) required for erase operation.

   Number of blocks includes start block (1st block):
   start block (1st block) + no_of_blocks should be <= 4095
   All the 64 pages belonging to the erased block will be read and the result of the operation will be displayed after the 2nd test case is completed.

5.3 Page Write

- Enter '3' in Nios Console to perform the Nand write operation
- Enter the page address to be Programmed and Read in Hex
- Enter the number of pages to be Programmed and Read
- Incremental 8-bit data (00 - FF) is written to a particular page



```
1: Read ID
2: Erase & Read Back
3: Write & Read Back
4: Exit
-------------------------
Enter the choice :
3
Test case no is 3
Page Program and Read Back Test
Make sure the blocks belongs to pages to be programmed and read are erased

Enter the page address to be programmed and read (Should be less than or equal to  2,62,143):
0
Start Page can vary from 0 to 63  since 1 block has 64 pages
Start Page is: 0

Enter Number of Pages to be Programmed and Read(no of pages includes start page & start_page + no_of_pgs should be <= 2,62,143)

2

Programming 2 Pages
All Pages Programmed and Successfully Read Back. Test Passed
```

Figure 25: Page Write

Note:

- In case of pages of a particular block targeted for write operation, make sure that the block is erased 1st (erase operation is not taken care internally in case of write).
- If the page has to rewritten, Block belonging to the particular page has to be erased again and then written.
- Start page starts from 0 to 63**.** Addition of start page and number of pages should not exceed total number of pages (i.e., 64*4096= 2,62,144 which is 0

to 2,62,143) i.e., start_page + number_pages <= 2,62,143

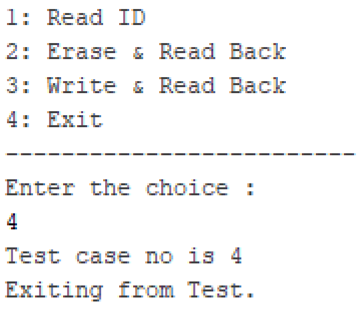
Explaining Write and Read Back test case with example:

The page address given by user will be decoded to know page has to be written and read back. The address that the user provides is 3rd,4th,5th address as shown in Figure 18.

1. If the user enters page address as 64d i.e.,40h - bit [5:0] becomes the page address (cannot go beyond 2,62,143value) indicating the start page as 0th page (page ranges from 0 – 63). Bit [16:6] becomes the block address i.e., 1st block (block ranges 0-4095).
   Number of pages includes start page (0th page):
   start page (0th page) + no_of_pgs should be <= 2,62,143
   All the 64 pages belonging to that block will be read and the result will be displayed after the 3rd test case is completed.
2. If the user enters page address as 2,62,143i.e., 3ffff bit [5:0] indicates the start page as 63rd page. Bit [16:6] becomes 4095th block.
   Number of pages includes start page (63rd page):
   start page (63rd page) + no_of_pgs should be <= 2,62,143
   All the 64 pages belonging to that block will be read and the result will be displayed after the 3rd test case is completed.

5.4 Exit

Enter '4' in Nios Console to quit the application



```
1: Read ID
2: Erase & Read Back
3: Write & Read Back
4: Exit
-------------------------
Enter the choice :
4
Test case no is 4
Exiting from Test.
```

Figure 26: Exit

Each test case mentioned above indicates the status(result) as follows. Test case 4 is simply to exit out and doesn't have status.

1. Timeout error = Failed with Time out error
2. Data mismatch error = Failed due to data mismatch
3. Read Status Failed = Failed due to Read Status Failure
4. Ecc error = Failed Due to ECC Error
5. Test passed = Test Passed without any error

NOTE:

- **The NIOS tool hangs very often. There is a possibility that once the test case is finished, it will hang there and prevent the user from performing the subsequent test case. User needs to close the NIOS tool and run the application again.**
- **Avoid switching tabs in between test case. You can use stop button present in the NIOS Console if you have to switch the tab to avoid the hang in the NIOS tool**
- **In case the Nios tool hangs in between the operation (for ex: after any operation , if the user cannot provide any input in the NIOS Console)user needs to program and run the NIOS application again.**

6 Resource Utilization

The table below shows the resource utilization summary for Arria-10 Gx dev. kit. for NAND flash Controller IP with BCH encoder and decoder.

Table 2: Resource Utilization for device for Arria-10 Gx dev. Kit.

Resource	Utilization	Available
LUT	14386	427200
LOGIC REGISTERS	7737	854400
M20K BLOCK	624	2713

NOTE: Above Resource Utilization table includes the BCH Algorithm implemented for NAND flash Host Controller IP.