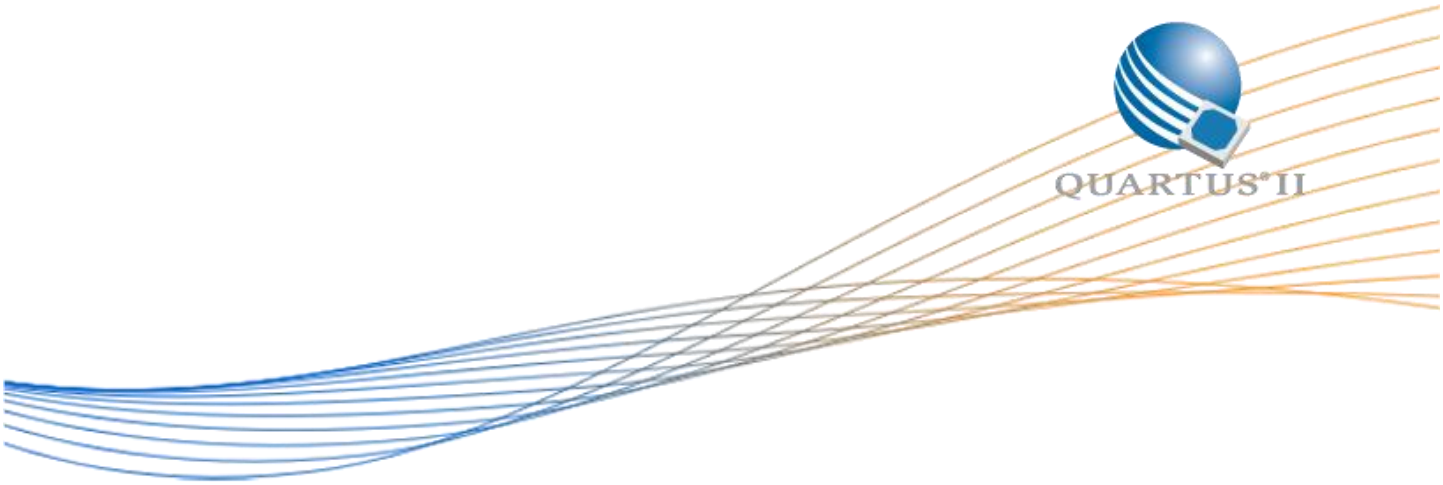


NEEK10 NIOS Design Example

Date: August, 2015

Revision: 1.0

TABLE OF CONTENTS

1 Introduction

1.1 Overview

The configurable Nios II system targeted at the MAX 10 NEEK board, the objective is to provide the customer with a design that tests individual peripherals. This integrated platform includes hardware, intellectual property(IP) and embedded software based on uC/OS-II. This design interfaces with each hardware component on the MAX10 NEEK board.

The design uses the following software:

- Qsys for abstract design capture, parametrizable IP cores and switch fabric generation
- The Quartus II software for pin mapping and to interface IP cores
- The Nios II Software Build Tools for Eclipse for run-time software control and configuration

The design provides the following features:

- Open interface and protocol standards to enable design reuse and connection of custom IP cores with off-the-self IP including:
    - Data streaming interfaces and protocols for transmission of data between IP cores in the system(Avalon-ST Video protocol layers on the Avalon-ST interface).
    - Control interfaces(Avalon-MM master and slave interfaces).
    - Access to external memory(Avalon-MM master and slave interfaces).
- System-level tools and design methodology for rapid system construction, integration and redesign. The Qsys tool uses standard interfaces to present an abstract view of the design and generates an application-specific switch fabric to construct the system.
- Parametrizable IP cores that enable you to quickly construct complete the systems.
- Development kits to rapidly prototype the designs.

1.2 Qsys

The Qsys flow is the primary design flow for the MAX10 NEEK Nios system development. Specifically, Qsys simplifies the process of systemdesign, including the data path, processor core and external memory integration. Qsys enables you to capture the design at an abstract level, with single point-to-point data connections rather than connecting individual data and control interface wires.

All connections in the Qsys system use Avalon-ST and Avalon-MM interaces.

Qsys automatically generates an interconnect switch fabric, including arbitration logic to connect the memory mapped masters and slaves together. A common example is a system that usese a single memory controller but contains multiple Avalon-MM masters that buffer video data in an external memory.

1.3 Quartus II software

The Quartus II software environment describes the top-level system and integrates the Qsys system into the top-level design and makes the appropriate pin assignments. The Quartus II software includes a wide range of tools to help with timing closure and perform hardware compilation to generat an FPGA programming file.

- For more information about the Quartus II software, refer to the Quartus II help.

1.4 Nios II Software Build Tools for Elipse

The Nios II Software Build Tools (SBT) for Eclipse is the primary software development tool for the Nios II family of embedded processors. You can perform all software development tasks within the Nios II SBT for Eclipse, including editing, building, and debugging programs. The Nios II SBT for Eclipse provides a consistent development platform that supports all Nios II processor systems.

You can configure the interfaces and control processing functions in the Nios II SBT for Eclipse software. These features provide a very rapid development cycle for software control code changes, without requiring hardware recompilation. This environment provides you with all the standard software debug tools, including breakpoints, views of memory, variables, and registers, and single stepping.

- For more information about the Nios II SBT for Eclipse software, refer to the Nios II Software Developer's Handbook.

2 Functional Description

Figure 2-1 gives the software block diagram of the reference design.

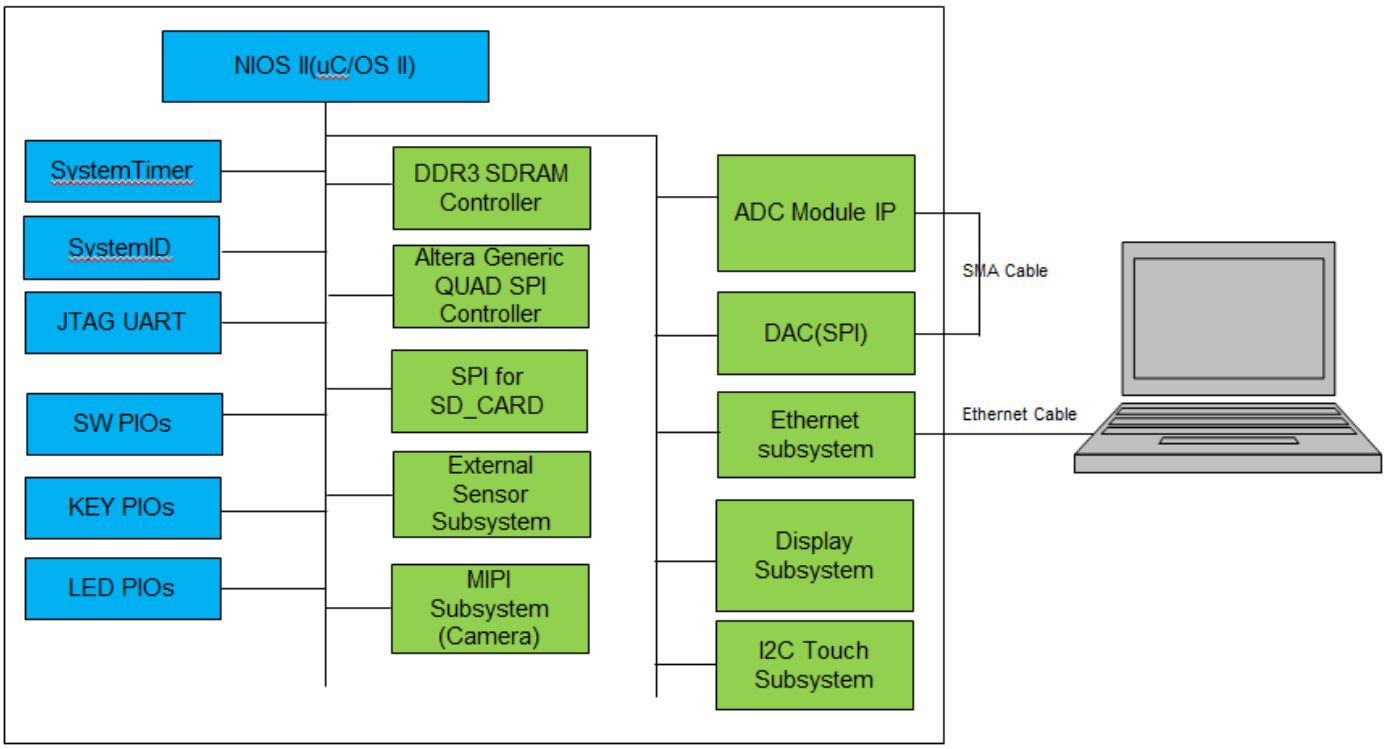


Figure 2-1 NEEK10 NIOS Reference Design Block Diagram

The design comprises the following blocks:

- Nios II processor for real time control
- DDR3 SDRAM controller for external and on-chip memory
- Display and touch system as the external input-output device
- Other interfaces for test,such as QSPI Flash, MIPI Camera, SD-card, DAC/ADC and Ethernet.

2.1 Nios processor

Nios II processor performs the following functions:

- Initializes individual peripherals, such as LCD, touch screen, DDR3,Camera, I2C sensor etc.
- Implement real-time task switch based on MicroC/OS II
- Receives instruction came from touch screen
- Control display content on LCD, picture, string or image by config Video Switch

2.2 DDR3 SDRAM

The Nios II software is in DDR3 SDRAM. It also buffers the video frames and Ethernet frame.

2.3 Display & Touch system

The MAX10 NEEK features a 7-inch capacitive amorphous TFT-LCD panel. The LCD touch screen offers resolution of (800x480) to provide users the best display quality for developing applications. The LCD panel supports 24-bit parallel RGB data interface.

The MAX10 NEEK is also equipped with a Touch controller touch controller, which can read the coordinates of the touch points through the serial port interface of Touch controller.

The MAX10 NEEK Multi-touch IP is provided for developers to retrieve user inputs, including multi-touch gestures and single-touch. The file name of this IP is i2c_touch_config and it is encrypted. To compile projects with the IP, users need to install the IP license first. Interface is a standard I2C interface.

3 System Requirement

This section describes the hardware and software requirements to run the NEEK10 NIOS design example.

3.1 Hardware Requirements

The following external parts are needed to demonstrate the design example.

- MAX10 NEEK
- Mini-USB cable for programming the device
- Ethernet cable
- SMA cable
- A PC as DHCP SERVER
- IMPORTANT: only use the 5V adapter that came with this kit. Do not use other power supplies from other Altera kits, these have higher voltage and may blow out the kit's power circuits.

3.2 Software Requirements

Ensure that you download and install the design neek10_test.par file and install the software provided with the development kit on your PC.

- For information about the software installation, refer to the documentation provided with the NEEK10 Development Kit.

You must install the Quartus II software, version 15.0, which includes the MegaCore IP Library on your PC.

- This application note assumes that you install the software into the default locations.

Ensure that you can extract neek10_test.qar and open neek10_test.qpf in your Quartus II software.

4 Opening the Example Design

This section describes how to open the NEEK10 NIOS Design Example. This section includes the following sections:

- Opening the Quartus II Top-Level Project
- Opening the Qsys System
- Viewing the Parameters
- Examining and Recompiling the NIOS Code
- Building the Software in the Nios II SBT for Eclipse

4.1 Opening the Quartus II Top-Level Project

To open the top-level Quartus II project, perform the following steps:

1. Launch the Quartus II software.
2. On the File menu, click Open Project, browse to <design example install directory>, and select the neek10_test.qpf Quartus II project file.
3. On the File menu, click Open, browse to <design example install directory>\verilog, and select the neek10_test.v top-level Verilog HDL design file.

4.2 Opening the Qsys System

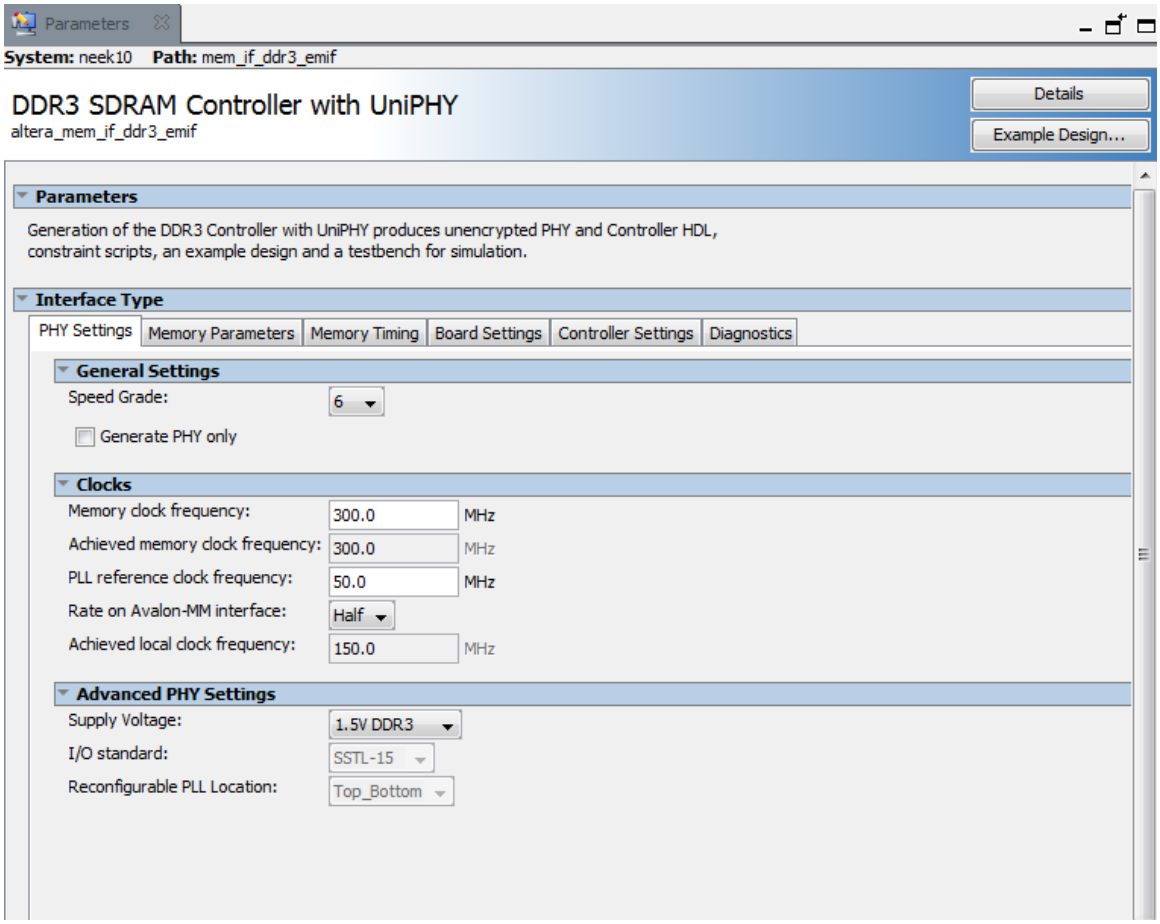
To open the design example Qsys system, with the **neek10_test.qpf** project open in the Quartus II software, click **Qsys** on the **Tools** menu.

- When you launch Qsys, a dialog box displays for you to select the design file. Always select the **neek10.qsys** file for this design.

4.3 Viewing the parameters

You can view the parameters of each component, to see what parameters are available to set for customized designs. At this stage, do not change any of the parameters. To view the parameters, perform the following steps:

In Qsys, in the Name column, double-click **mem_if_ddr3_emif** to display the MegaWizard™ interface for the DDR3 SDRAM controller with UniPHY(Figure 4-1).



Firgure 4-1 Parameter Settings for the DDR3 SDRAM controller with UniPHY

Table 4-1 shows the main component names to click to view the parameters of each IP core.

Table 4-1 IP Cores in Design Example IP Cores

Name	IP Core
Mem_if_ddr3_emif	DDR3 SDRAM Controller with UniPHY
Generic_quad_spi_controller	Altera Generic QUAD SPI controller
Alt_vip_vfr	Video Frame Reader
Alt_vip_swi_0	Video Switch
Alt_vip_vfb_0	Video Frame Buffer
Alt_vip_cl_cvo	Clocked Video Output II(4K Ready)

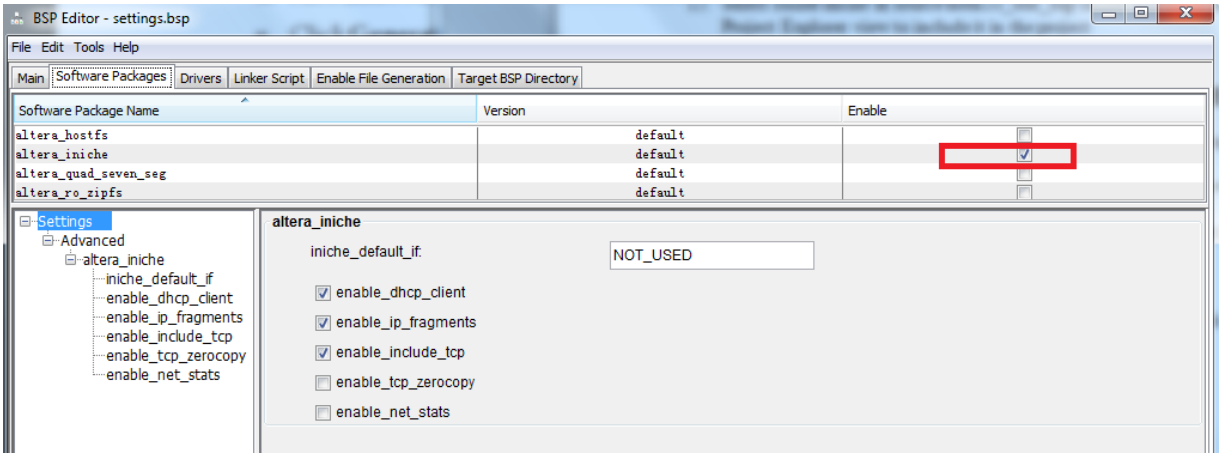
I2c_opencores_mipi	I2C master(opencores.org)
TERASIC_CAMERA_0	TERASIC_CAMERA
Tse_mac	Triple-Speed Ethernet
Sgdma_tx	Scatter-Gather DMA Controller
Sgdma_rx	Scatter-Gather DMA Controller
Adc_wave_dual_ram	On-chip memory(RAM or ROM)

4.4 Examining and Recompiling the NIOS Code

To use the Nios II SBT for Eclipse to examine and recompile the Nios II code and associated board support package(BSP), perform the following steps:

1. On the Windows Start menu, point to **All Programs**, point to **Altera**, point to **Nios II EDS** <version>, and then click **Nios II** <version> **Software Build Tools for Eclipse**. The **Workspace Launche**r dialog box appears.
2. Select workspace folder by browsing to the design example \software directory and specifying a workspace subdirectory. For example,<install directory>\software
3. Click **OK** to create a new workspace.
4. On the File menu, point to **New**, and then click **Nios II Application and BSP from Template**. The **Nios II Application and BSP from Template** dialog box appears.
5. From the **SOPC Information File name** box, browse the the verilog/**neek10.sopcinfo** file. The Nios II SBT for Eclipse fills in **CPU name** with the processor name found in the **.sopcinfo** file.
6. In the **Project name** box, type **neek10_test**.
7. Select **Hello MicroC/OS-II** from the **Templates** list and then click **Next**.
8. Select **Create a new BSP prject based on the application project template**.
9. Accept the default project name, neek10_test_bsp
10. Ensure that **Use default location** is turned on.
11. Click Finish to create the application and the BSP based on the **neek10.sopcinfo** file.
12. After the BSP generates, the neek10_test and neek10_test_bsp projects appear in the Project Explorer view.
13. In windows Explorer, Delete the hello_ucosii.c from neek10_test project and navigate to design's software\source directory.

14. Select all files in source\neek10_test folder and drag them onto neek10_test in the Nios II SBT for Eclipse Project Explorer view to include them in the project
15. Select folder **iniche** in source\neek10_test_bsp folder and drag it onto neek10_test_bsp in the Nios II SBT for Eclipse Project Explorer view to include it in the project
16. Before building the neek10_test project:
    a. Right-click neek10_test_bsp in the Project Explorer view, point to Nios II and then click BSP Editor. The Nios II BSP Editor dialog box appears.
    b. Select the Software Packages tab at the top of this dialog and enable the altera_iniche as below.



    c. Click Generate.
    d. Click Exit.
17. Right-click neek10_test_bsp in the Project Explorer view, point to **NiosII**, click **Generate BSP**.
18. Right-click neek10_test_bsp in the Project Explorer view, click **Build Project**.
19. Again drag folder source\neek10_test_bsp\iniche onto neek10_test_bsp in the Nios II SBT for Eclipse Project Explorer view to include it in the project

    Important: this step couldn't be skipped.

    Because we modified some files related to the folder 'iniche' in the design. Once we click 'genereate BSP ' , Eclipse will recover these files from <Quartus install directory>\nios2eds\components\altera_iniche> directory. So we need drag the folder again.
20. Right-click neek10_test in the Project Explorer view, click **Build Project**. The Nios II SBT for Eclipse compiles the neek10_test_bsp and neek10_test software projects and builds the **neek10_test.elf** software executable.

5 Simple Demo Setup

To demonstrate the design, perform the following steps:

1. Connect an Ethernet cable between MAX10 NEEK and a PC
2. Connect the SMA cable between the DAC SMA OUT (J16) and the ADC SMA IN dedicated ANAIN1 channel (J14) .
3. Connect the power cord to the power plug, and connect a mini USB from your PC to the J8 UB2 connector.

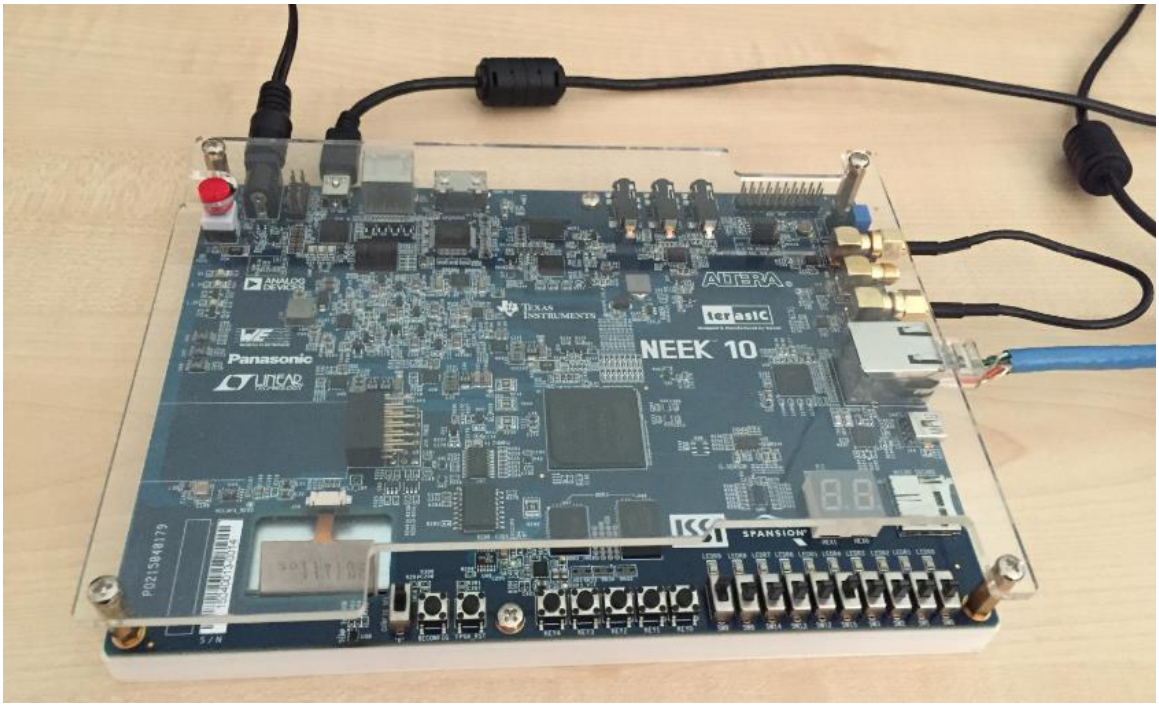


Figure 5-1 NEEK10 Board Hardware connection

4. Open the Quartus II programmer
    a. Make sure the board is powered (push the Red power) and USB calbe connected.
    b. Click Hardware Setup
    c. Select the NEEK10 and select Add Hardware.

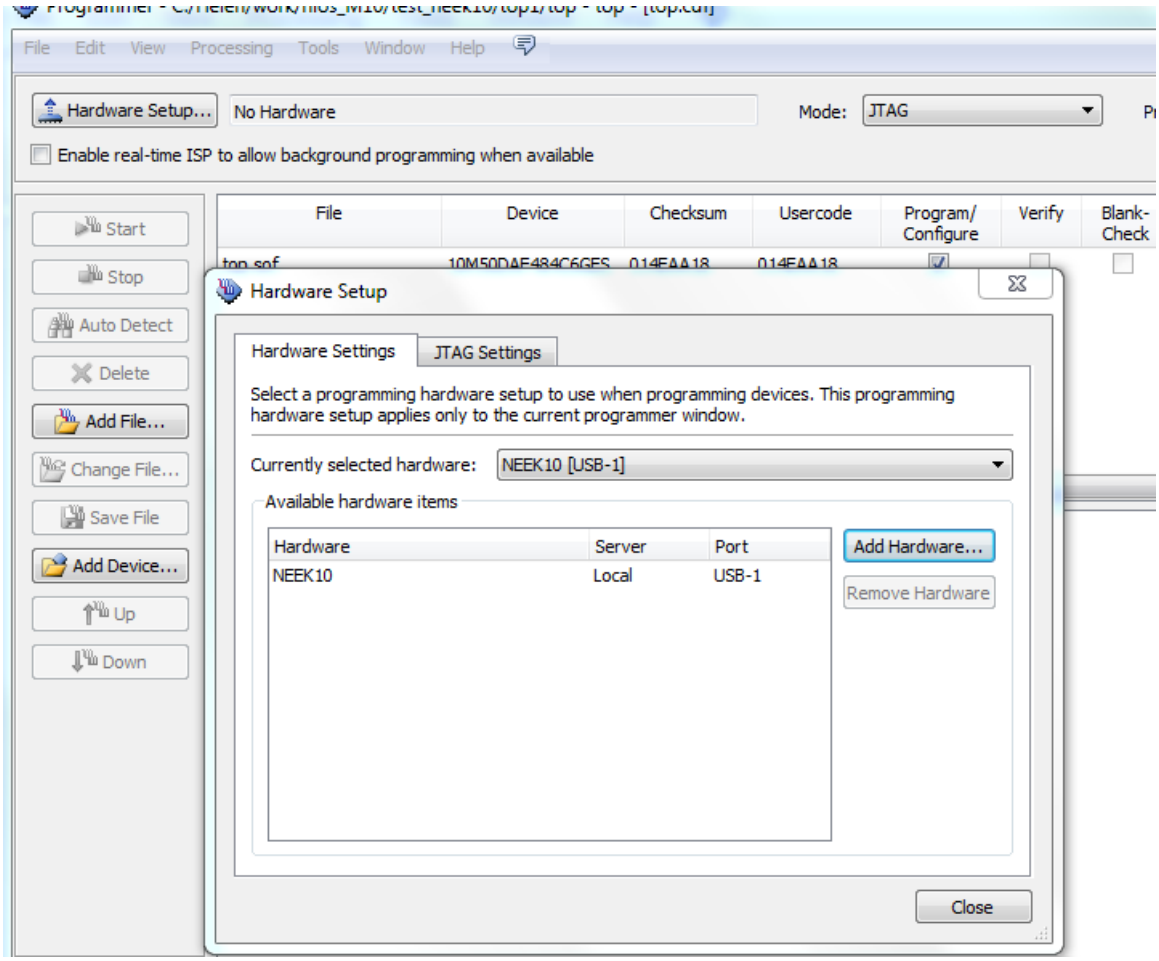


Figure 5-2 NEEK10 Programmer JTAG SELECTION

5. Before using your fresh board, you must erase the flash to satisfy the ES Guidelines item “Full Chip Erase Prior to Initial Device Programming”. In the programmer, select Auto-Detect and select the 10M50DAES device. Then select the Erase option and click Start.

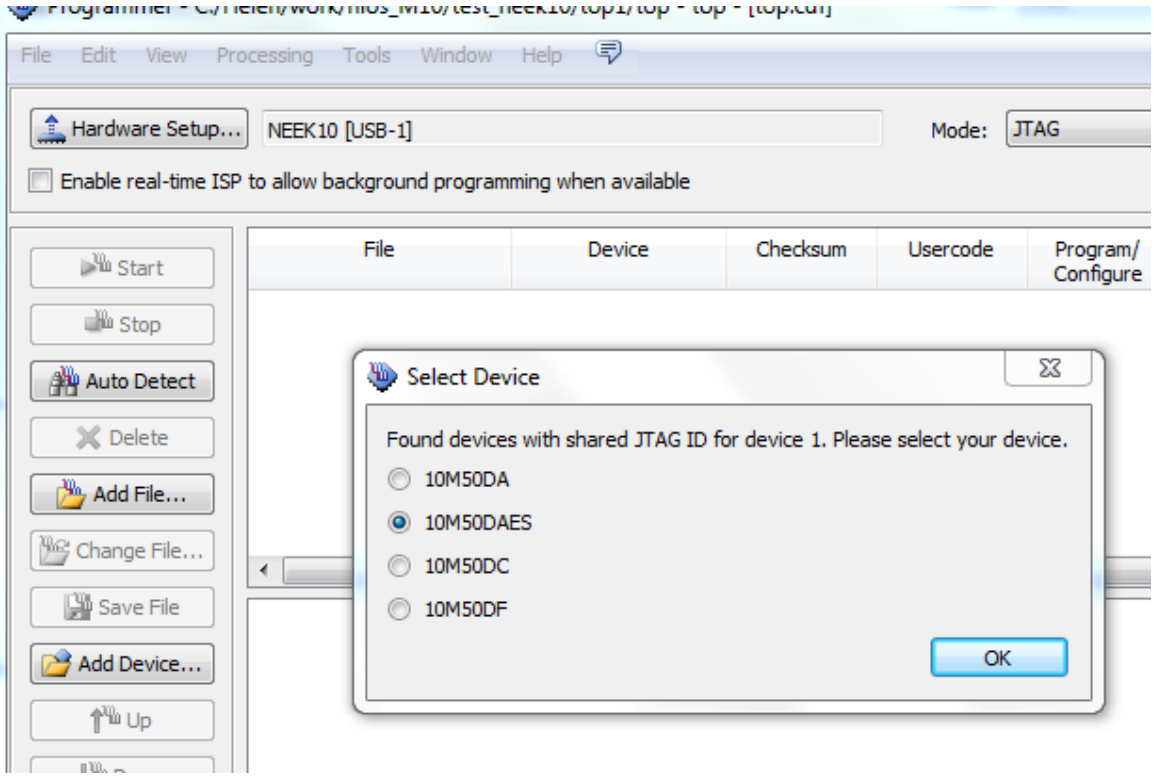


Figure 5-3 NEEK10 Programmer JTAG Select Device

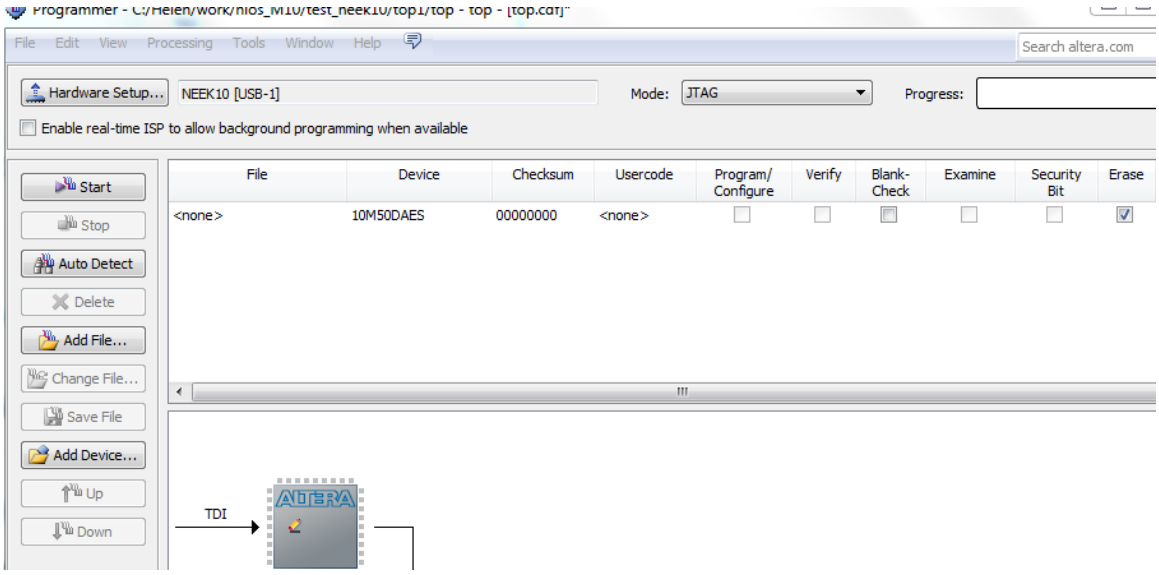


Figure 5-4 NEEK10 Programmer Erase Device

6. Double click on the File column "<none>" next to the 10M50DAES line in the programmer window. Then select"top.sof" file for programming, select Program/Configure and hit Start.

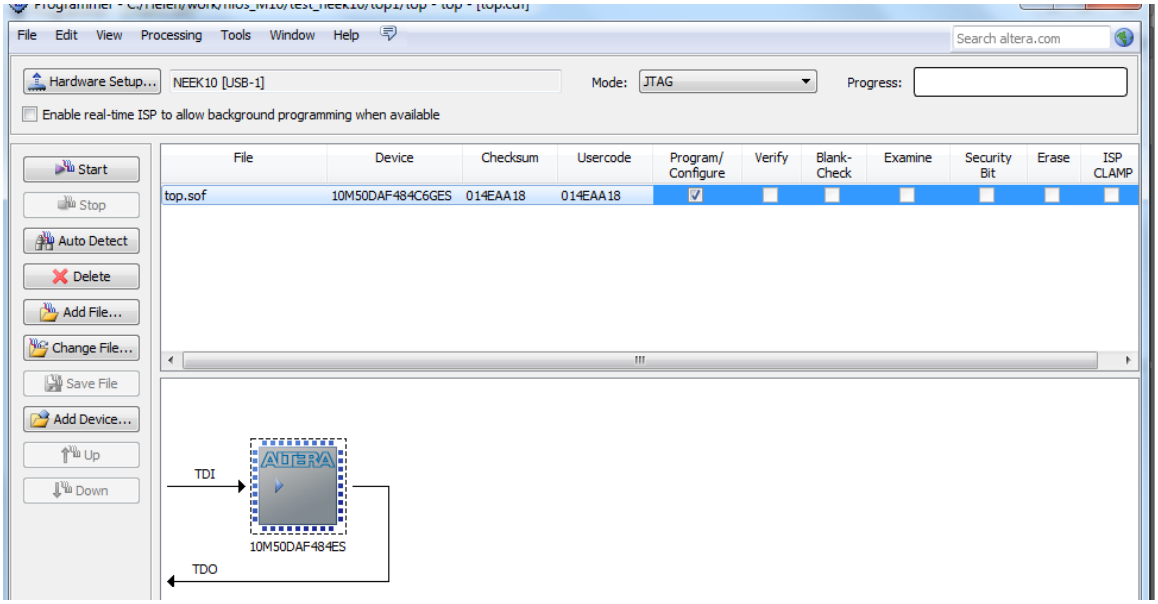


Figure 5-5 NEEK10 Programmer Program Device

7. Open the Nios II terminal by going to Start→Altera→Nios II EDS→Nios II Command Shell
8. Enter the following command:

   Nios2-download <file path>/nios.elf –g; nios2-terminal

Now you can start to run the demo.

6 Design Example Overview

Figure 6-1 gives a snapshot of the default image at power-up. After about 5 second, it will enter the main screen shown in Figure 6-2.

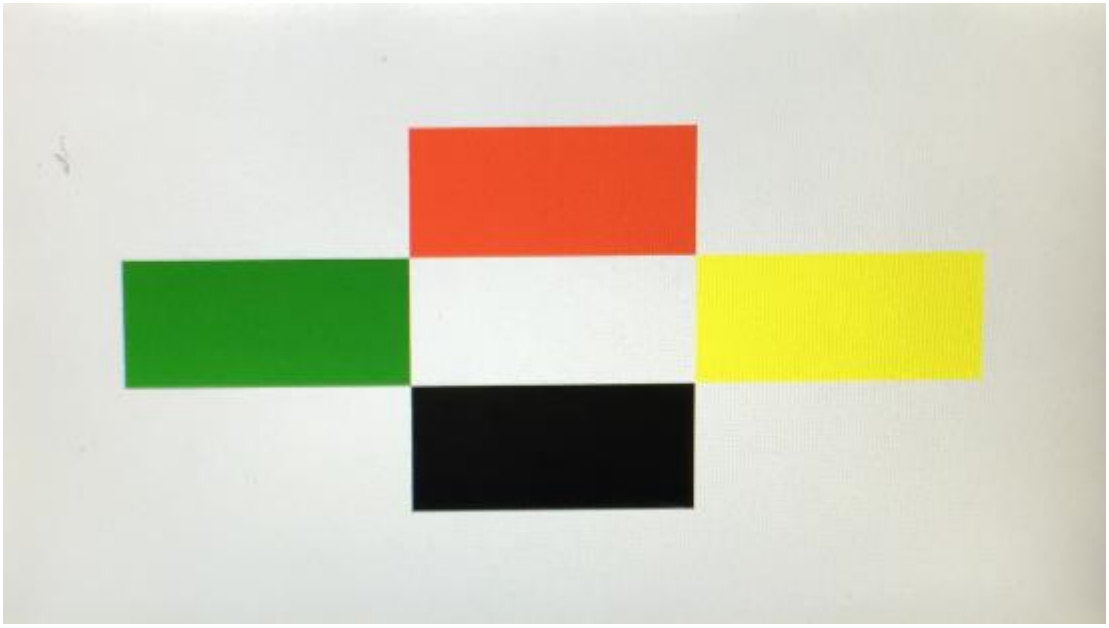


Figure 6-1 NEEK10 splash screen

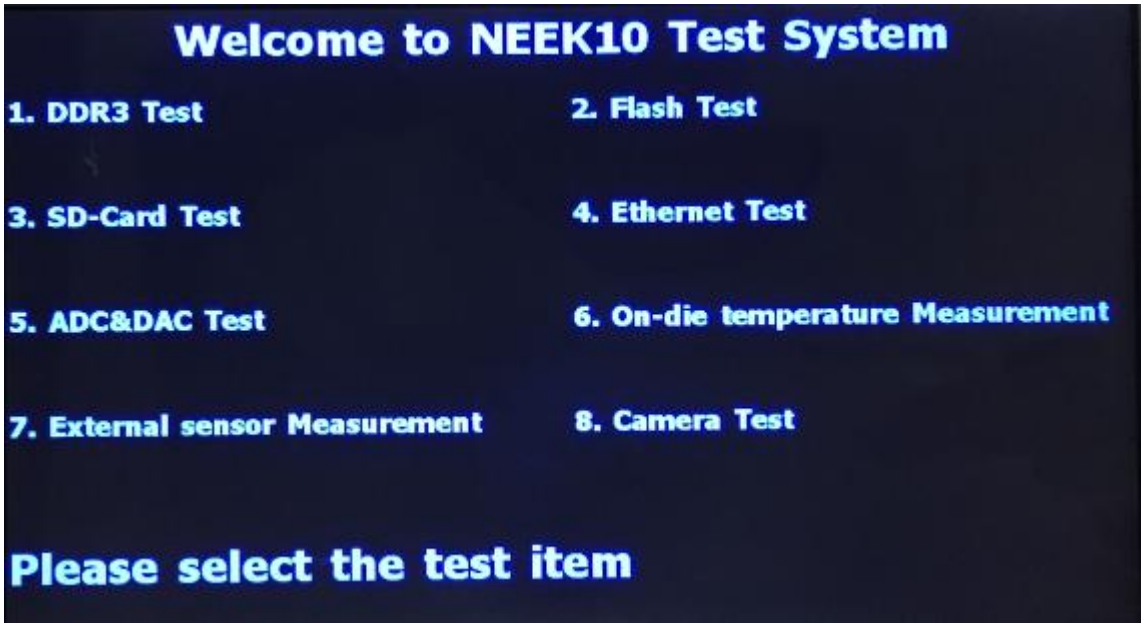


Figure 6-2 NEEK10 main screen

On the NEEK10 main screen, the user may select any item to test by touching specific area on the screen.

6.1 DDR3 Test

When the user selects to test DDR3 interface, it will enter DDR3 test sub-screen. DDR3 test module gives the result by reading and writing to some amount of addresses, and verifying the data. After finishing the test, the result is displayed on the LCD shown in Figure 6-3. Once the user touches any area on the screen, it will return the main screen.

The figure 6-4 shows the block diagram of DDR3 test module. Several status signals of DDR3 SDRAM controller, such as ddr3_local_cal_fail, ddr3_local_cal_success and ddr3_local_init_done, are shown on LEDR[2:0]. The specific meaning of the LED is as follows.

	Description
LEDR[0]	ON--memory initialization is complete OFF-- memory initialization is not complete
LEDR[1]	ON -- memory calibration completes successfully OFF--memory calibration does not complete
LEDR[2]	ON—memory calibration failed OFF—memory calibration successfully

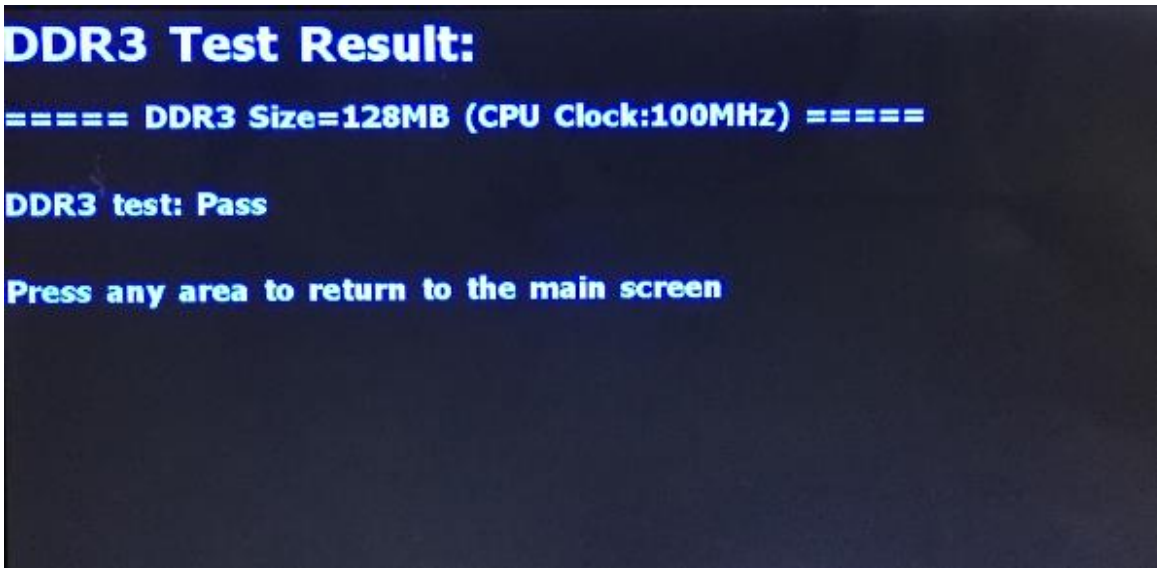


Figure 6-3 Screen Shot of DDR3 Test Module

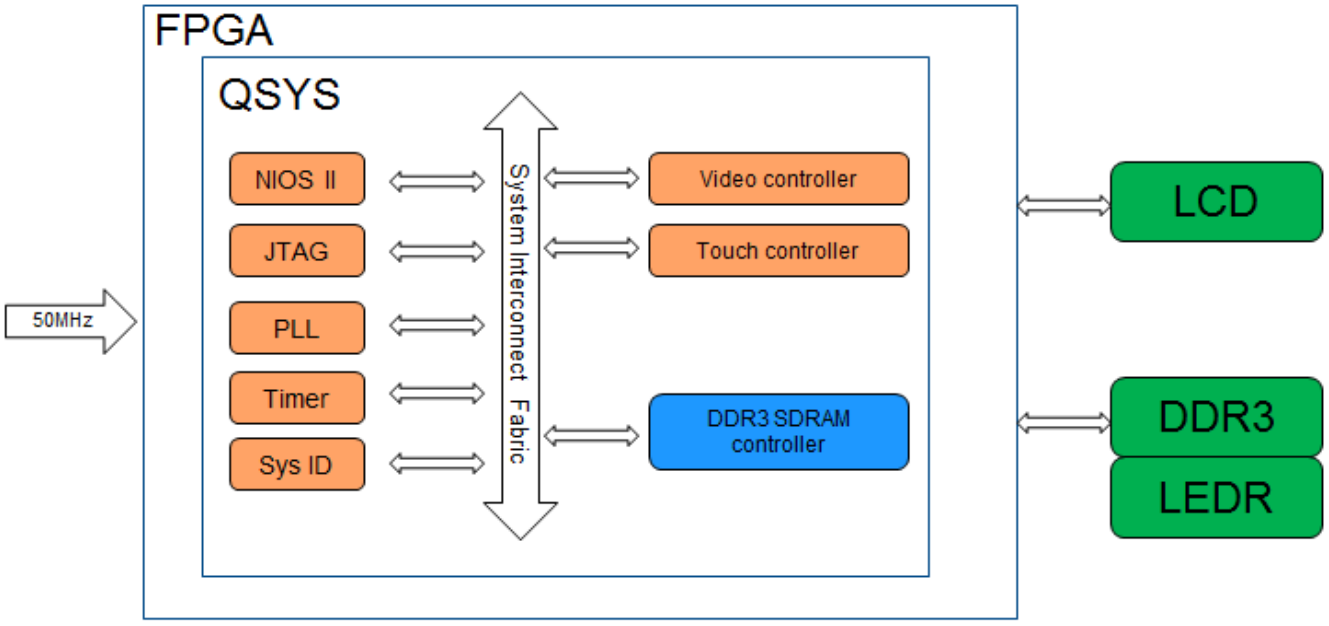


Figure 6-4 Block diagram of DDR3 Test Module

6.2 QSPI Flash Test

When the user selects to test Flash interface, it will enter Flash test sub-screen. Flash test module gives the result by reading and writing to some amount of addresses and verifying the data. After finishing the test, the result is displayed on the LCD shown in Figure 6-5. Once the user touches any area on the screen, it will return the main screen.

The figure 6-6 shows the block diagram of Flash test module.

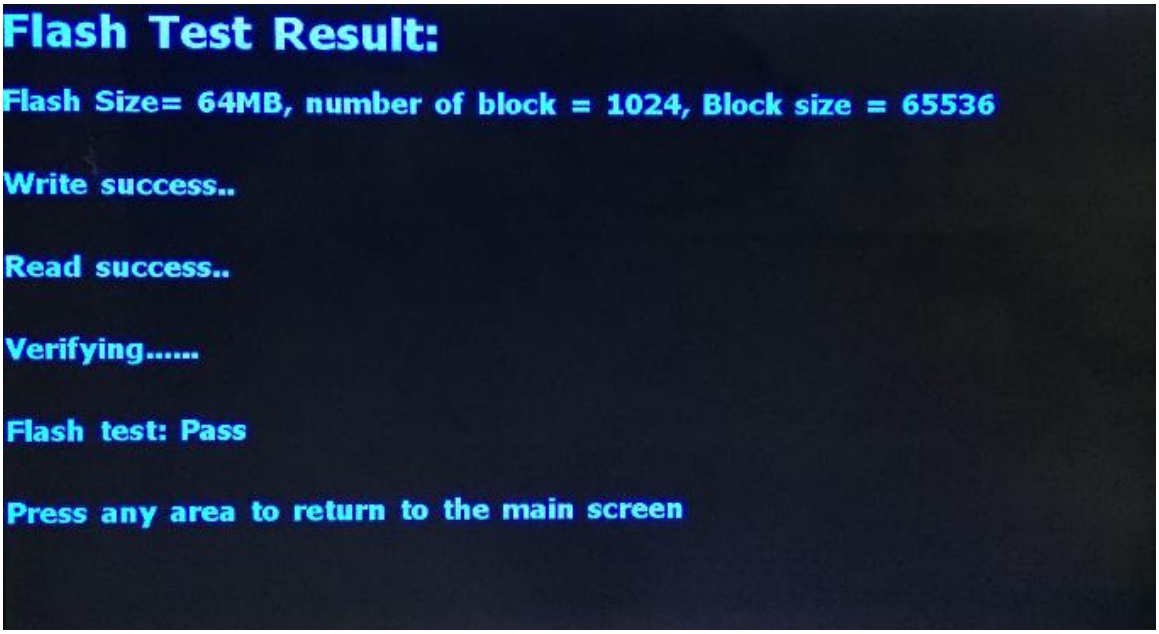


Figure 6-5 Screen Shot of Flash Test Module

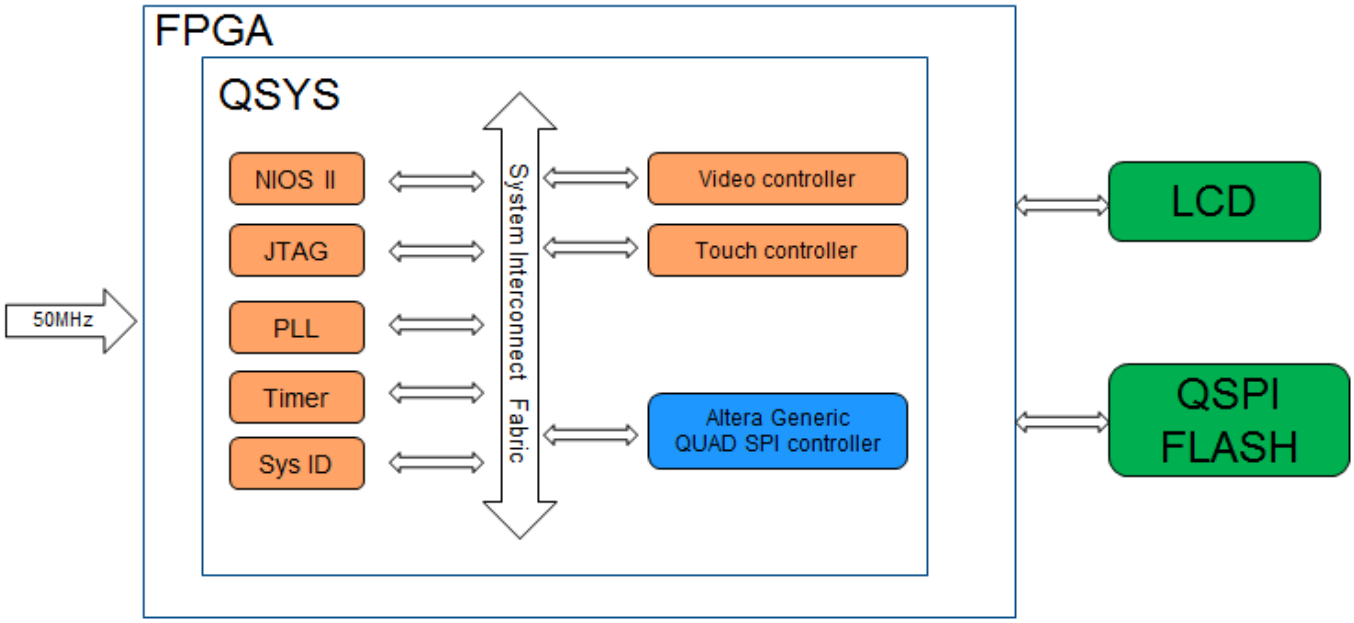


Figure 6-6 Block diagram of Flash Test Module

6.3 SD Card Test

When the user selects to test SD Card interface, it will enter SD Card test sub-screen. SD Card test module gives the result by reading and writing to some amount of addresses and verifying the data. The test is based on SPI bus protocol.

After finishing the test, the result is displayed on the LCD shown in Figure 6-7. If the SD-Card haven't been inserted or been damaged, the result is shown in Figure 6-8. Once the user touches any area on the screen, it will return the main screen.

The figure 6-9 shows the block diagram of Flash test module.

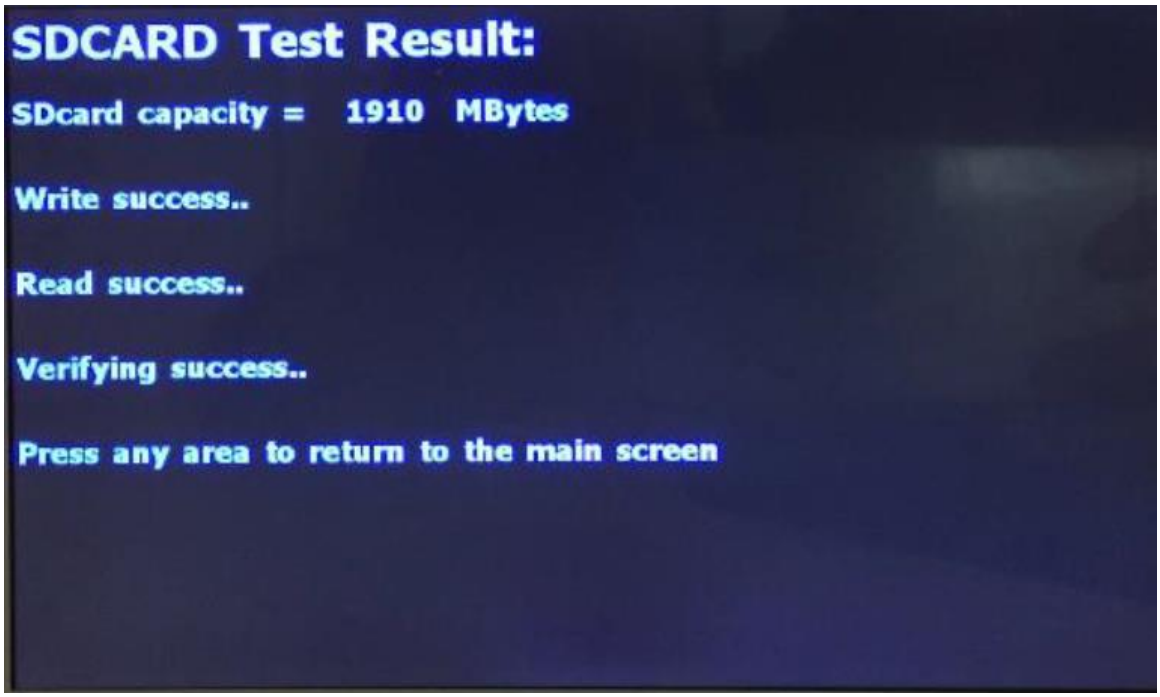


Figure 6-7 Screen Shot of Flash Test Module

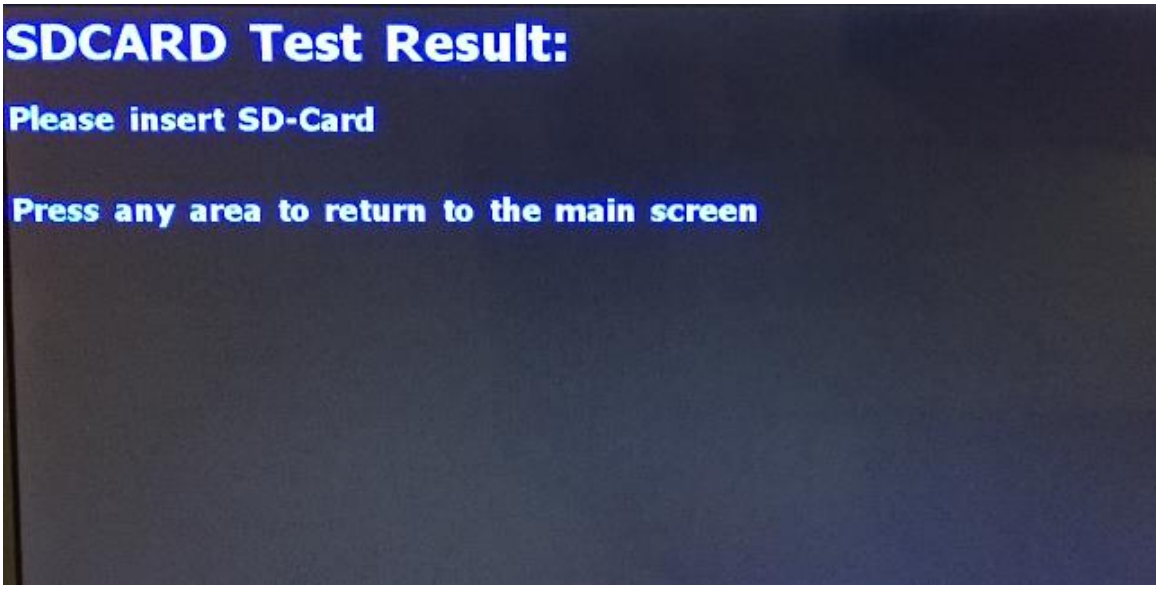


Figure 6-8 Screen Shot of Flash Test Module

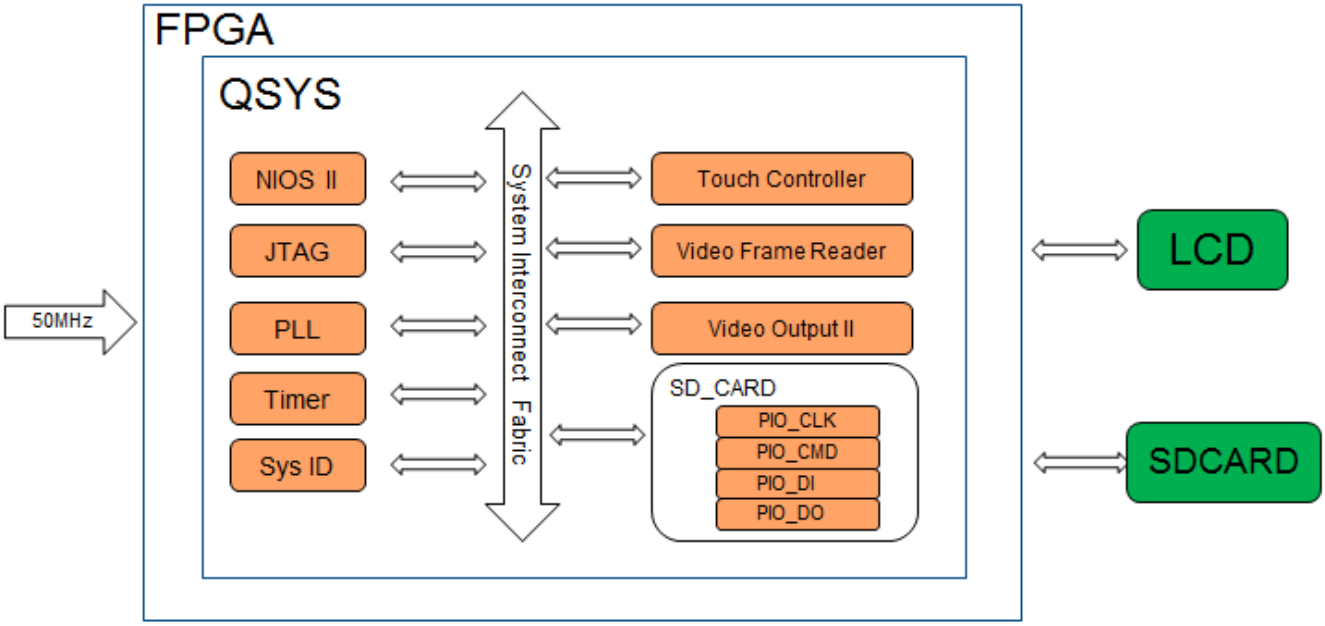


Figure 6-9 Block diagram of Flash Test Module

6.4 Ethernet Test

Before perform Ethernet test, make sure the following things:

1. The enthernet cable has been connected between NEEK10 board and your PC.
2. A DHCP Server runs on your PC. Below is the settings of a DHCP Server .

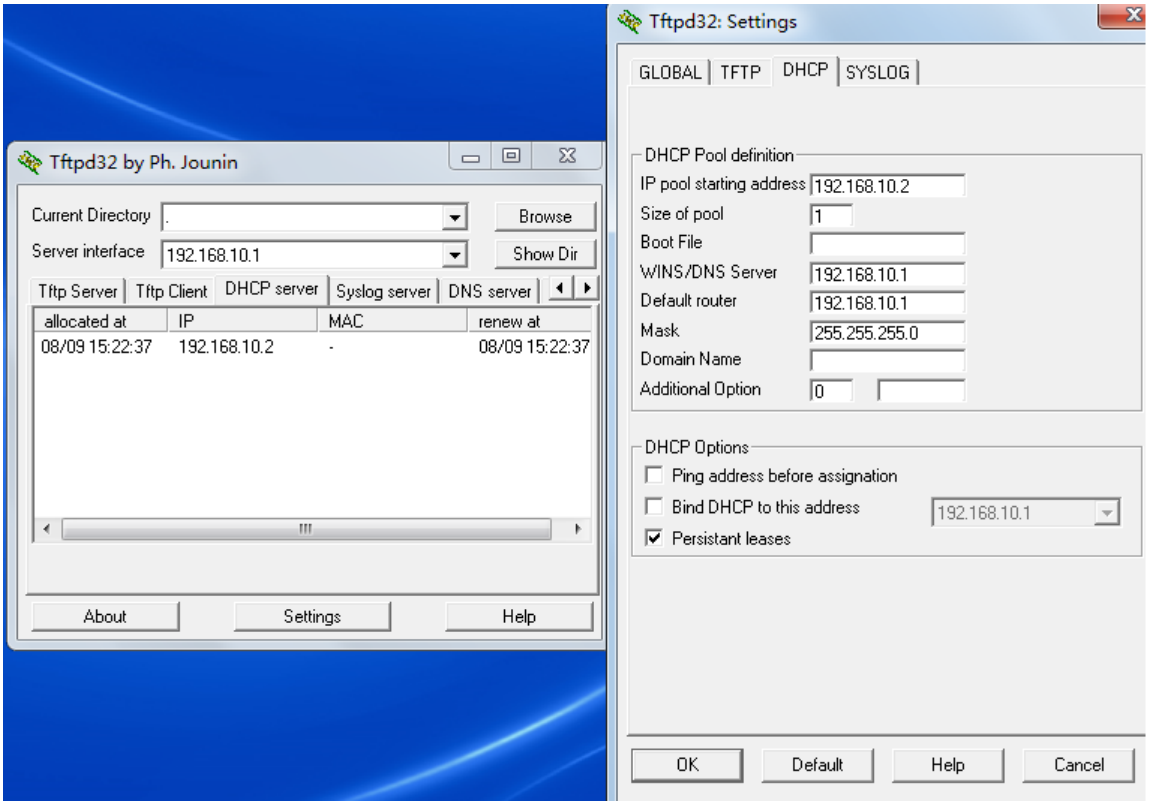


Figure 6-10 Settings of a DHCP Server

When the user selects to test Ethernet interface, it will enter Ethernet test sub-screen. The test result will be displayed on the LCD.

If the MAX10 NEEK could get IP, then the result is shown in Figure 6-11.a.

If the link has been established, but couldn't get the IP address, then the result is shown in Figure 6-11.b, it happens when the cable has been connected, but the DHCP Server doesn't run.

If the link couldn't be established, then the result is shown in Figure 6-11.c, it happens there is not an Ethernet cable between MAX10 NEEK and your PC.

Once the user touches any area on the screen, it will return the main screen.

The figure 6-12 shows the block diagram of Ethernet test module.

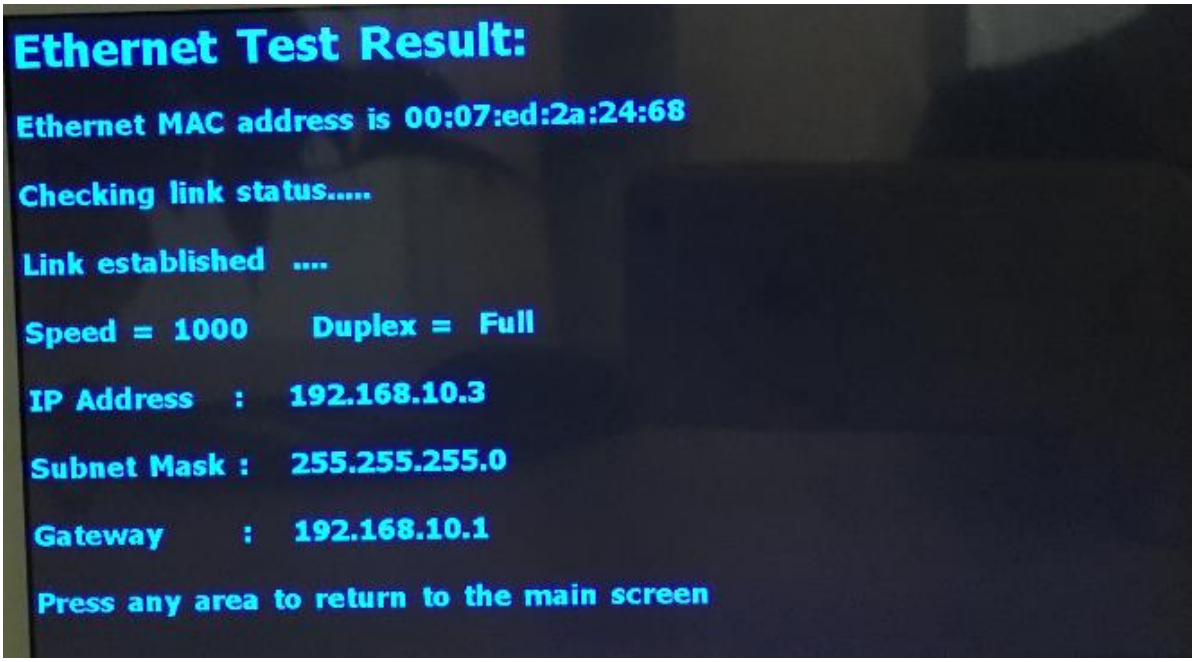


Figure 6-11.a Screen Shot of Triple Ethernet Module

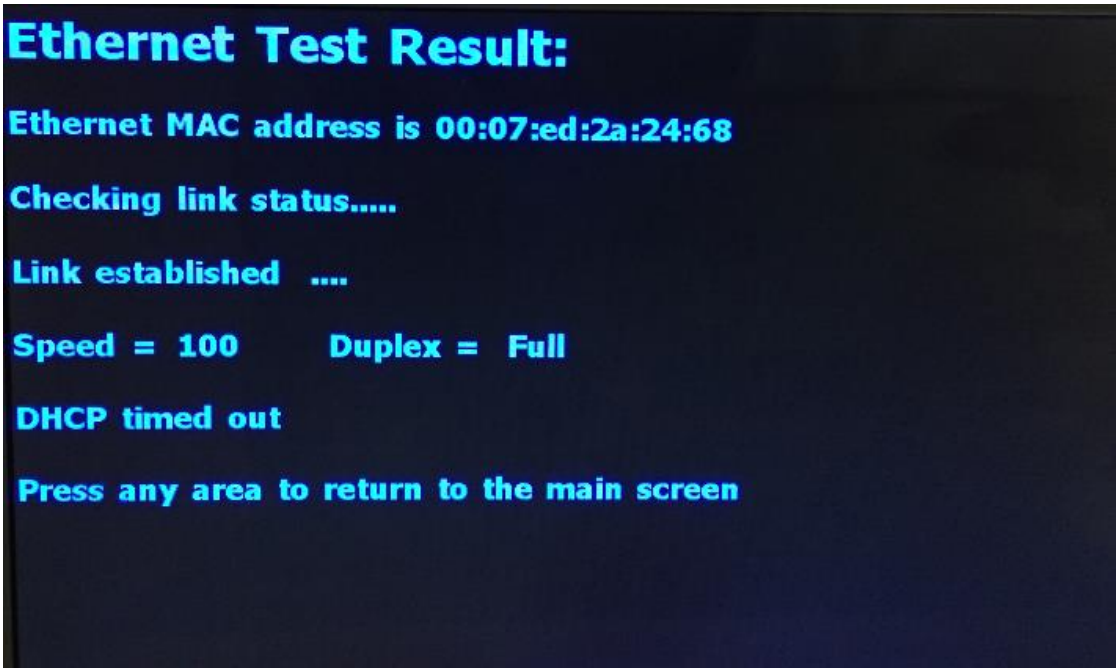


Figure 6-11.b Screen Shot of Triple Ethernet Module

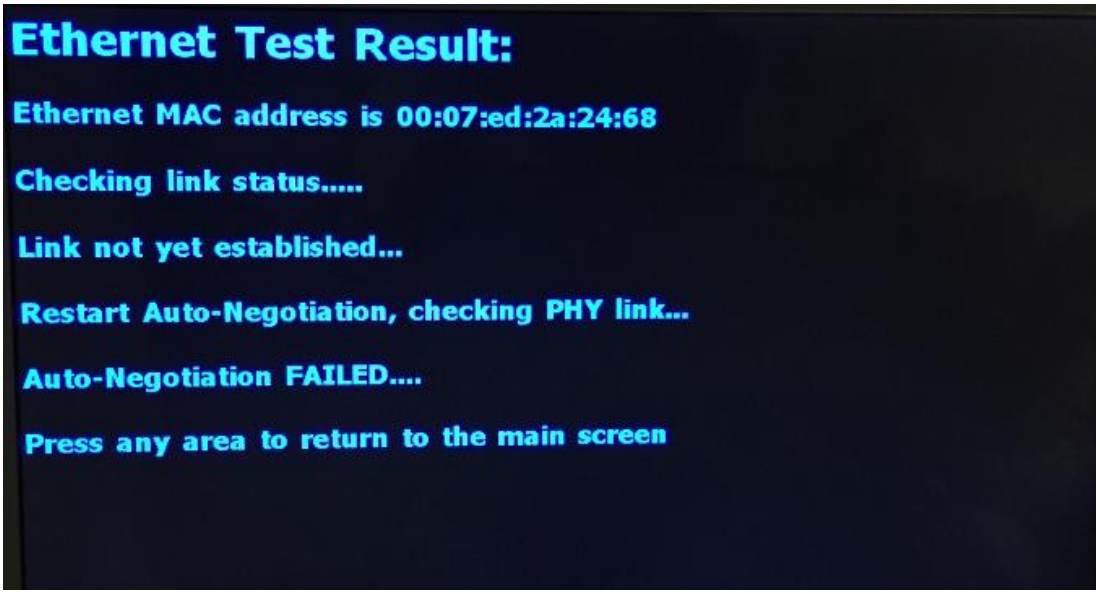


Figure 6-11.c Screen Shot of Triple Ethernet Module

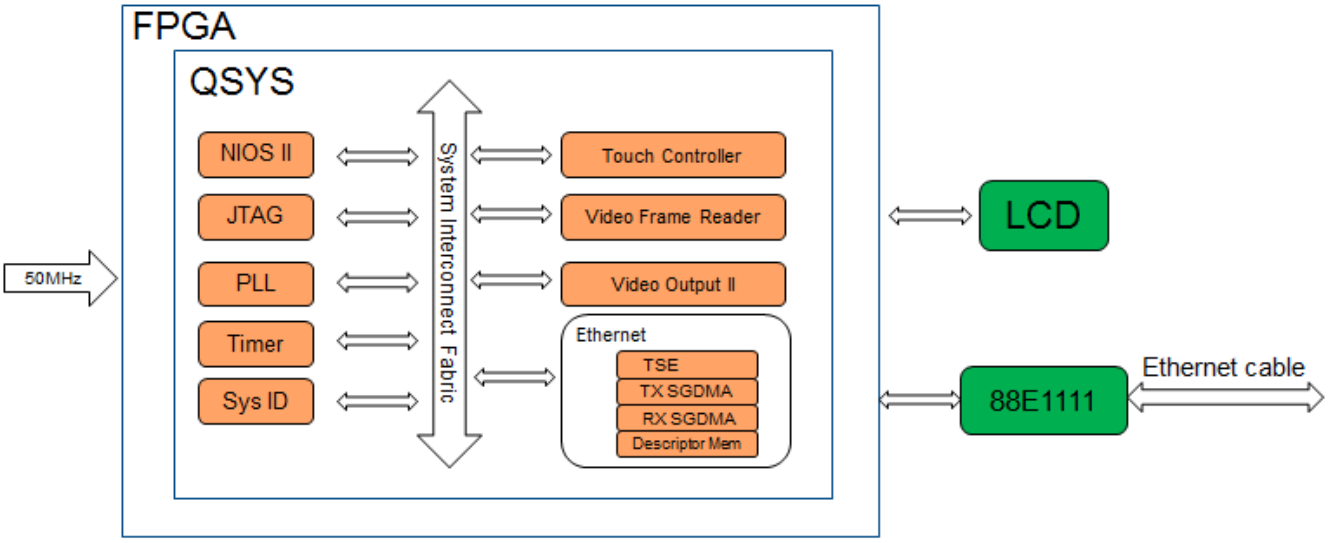


Figure 6-12 Block diagram of Triple Ethernet Module

6.5 ADC & DAC Test

ADC & DAC module consists of three modules, wave_generation, wave_detection and wave_display.

The module wave_genetation drives one external 16 bit DAC device to generate kinds of waveform. When the switch 'SW9' is ON, it means the sine wave is sent, when 'SW9' is OFF, it means the triangular wave is sent. It is implemented by Verilog.

The module wave_detection get the waveform came from ANAIN1 connector through the Altera ADC IP core, calculates the its period and check its max&min value and sends the waveform to CPU through dual-ram. It is implemented by Verilog.

The module wave_display in NIOS is responsible for display waveform on the LCD.

Before selecting ADC&DAC test, make sure to connect the SMA cable between the ANAIN1 connector(ANAIN1 dedicated ADC channel) on the board and the DAC SMA OUT.

When the user selects to test ADC&DAC interface, it will enter ADC&DAC test sub-screen. The waveform will be displayed on the LCD shown in Figure 5-11. Once the user touches any area on the screen, it will return the main screen.

The figure 6-13 and figure 6-14 shows the block diagram of ADC & DAC module.

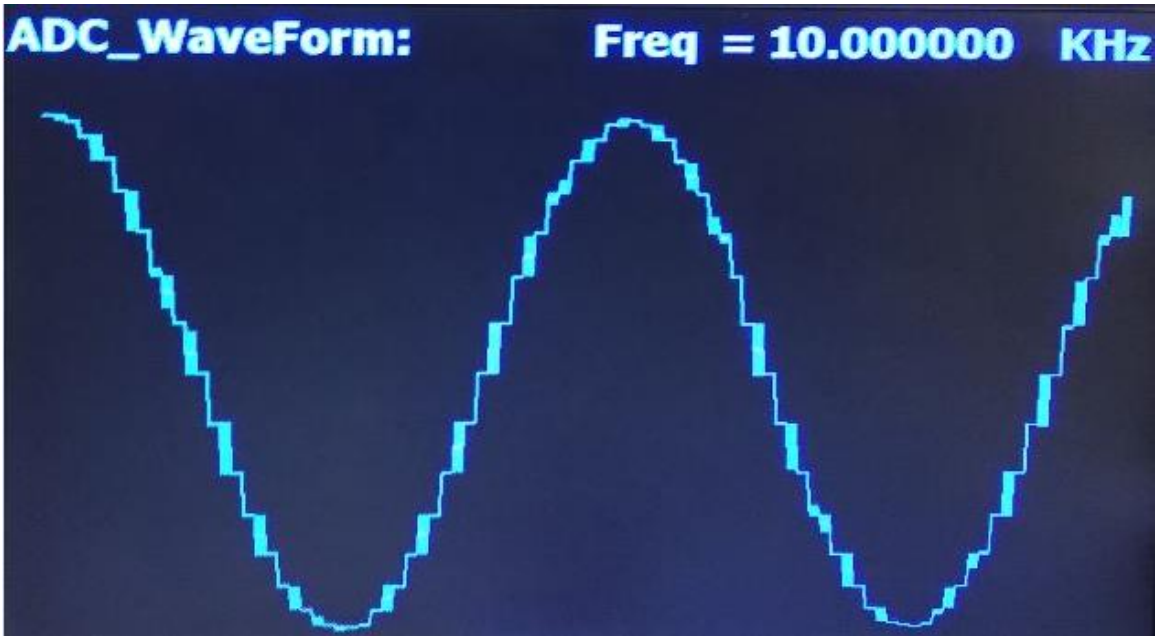


Figure 6-13 Screen Shot of ADC&DAC Test Module

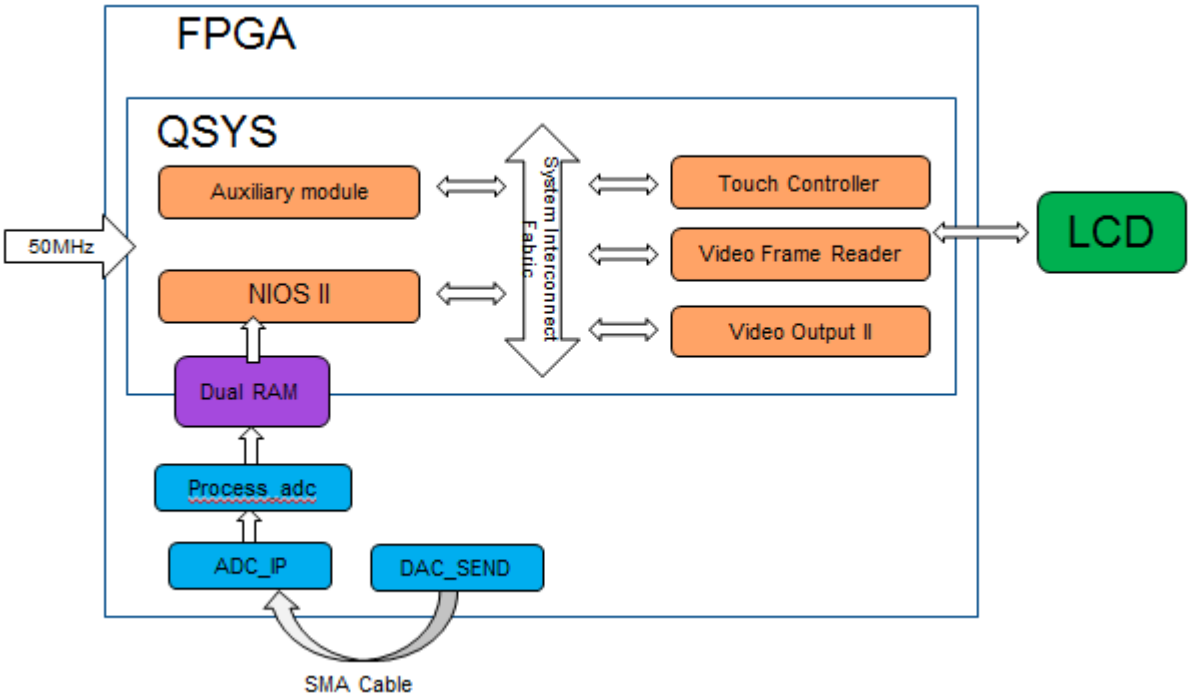


Figure 6-14 Block diagram of ADC&DAC Test Module

6.6 On-die temperature measurement

MAX10 has a on-die temperature sensor to perform on-chip temperature measurement. When the user selects to display the on-die temperature, it will enter the on-die temperature display sub-screen. The test result is displayed on the LCD shown in Figure 6-15. Once the user touches any area on the screen, it will return the main screen. The figure 6-16 shows the block diagram of Flash test module.

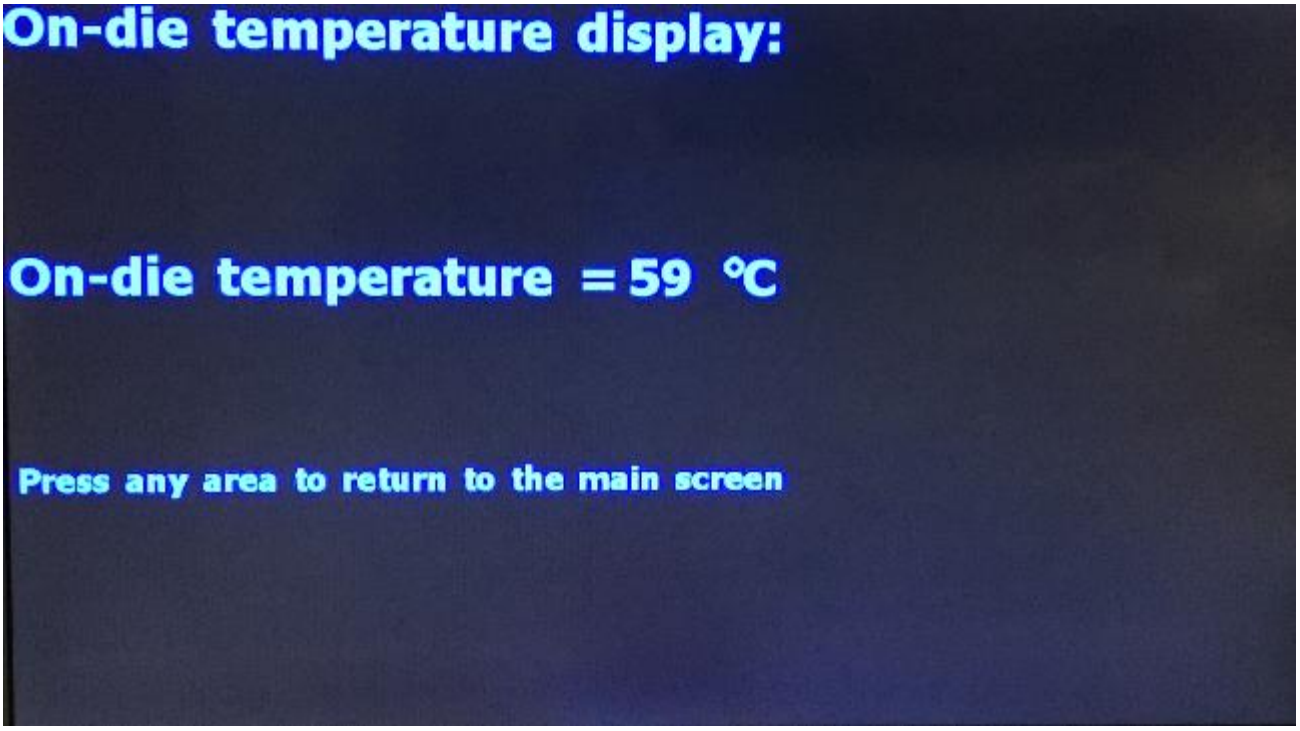


Figure 6-15 Screen Shot of on-die temperature Test Module

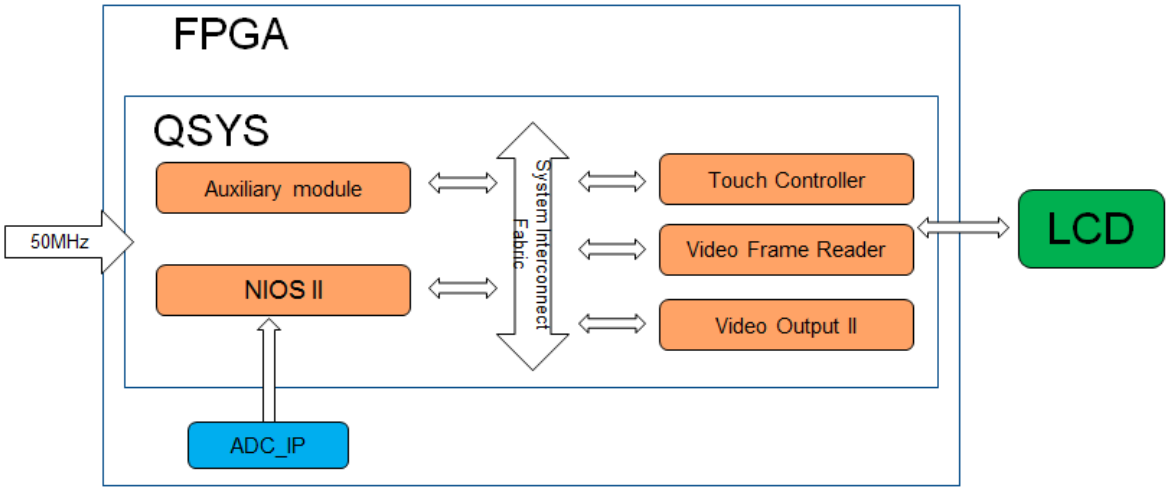


Figure 6-16 Block diagram of on-die temperature Test Module

6.7 External sensor measurement

The design uses I2C protocol to control the ADXL345 digital accelerometer and the APDS-9300 Miniature Ambient Light Photo Sensor and HDC1000 Accuracy Digital Humidity Sensor with Temperature Sensor. When the user selects

to test the external sensors, it will enter external sensor sub-screen. The test result is displayed on the LCD shown in Figure 6-17.

The value of light sensor will change as the brightness changes around the light-sensor.

The value of digital accelerometer will change as the user tilts the NEEK10 board.

The value of the temperatue and humidity will change as the environment changes around the HDC1000.

Once the user touches any area on the screen, it will return the main screen.

The figure 6-18 shows the block diagram of the external sensor measure module.

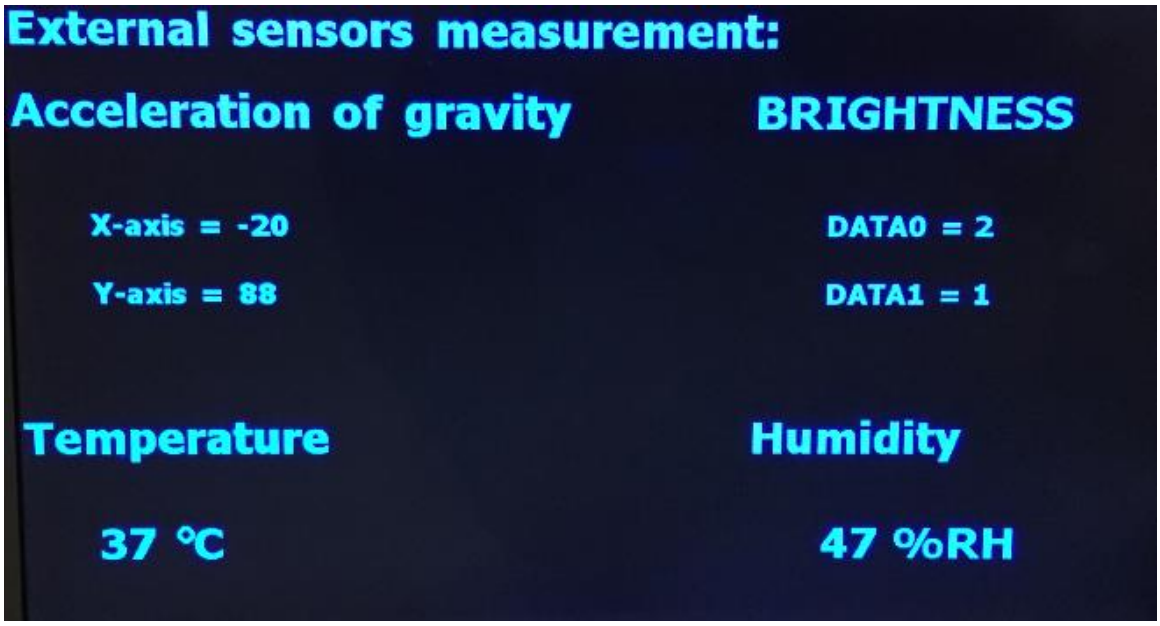


Figure 6-17 Screen Shot of external sensor measure Module

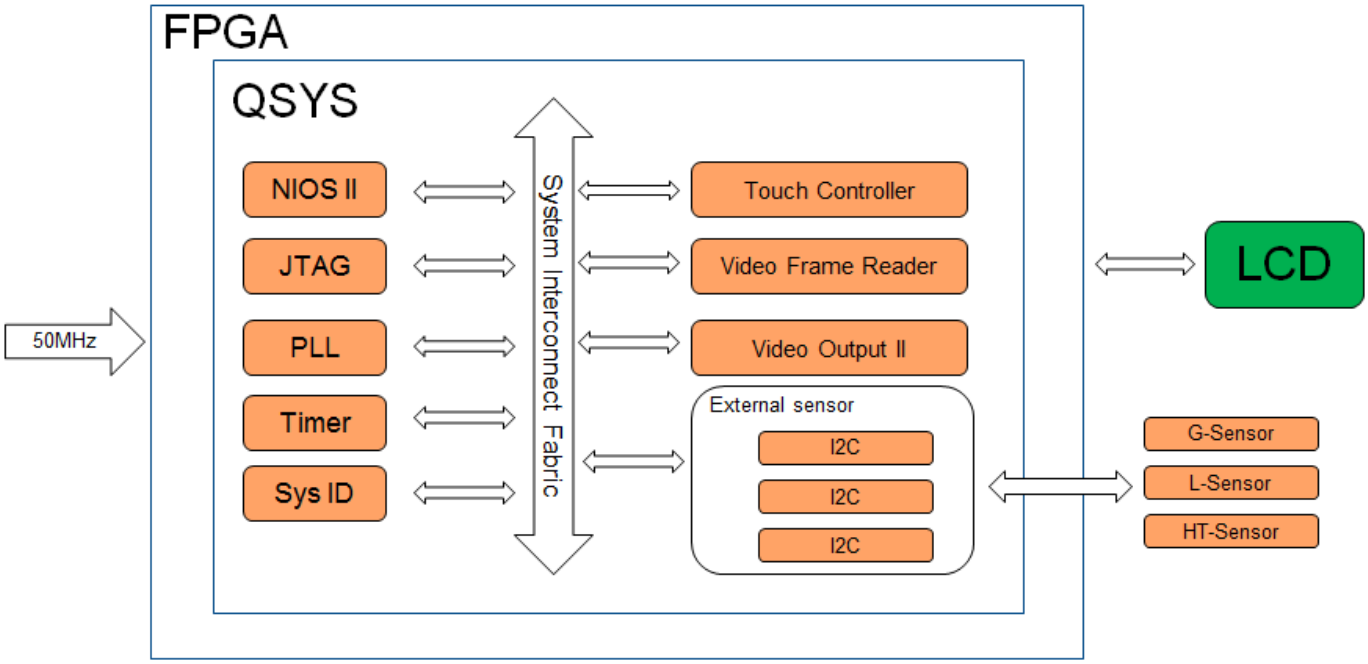


Figure 6-18 Block diagram of external sensor measure Module

6.8 Camera Test

Camera test module consists of four modules, I2C manage interface, mipi interface, Altera Video Frame Buffer IP and Altera Video Switch IP. The driver for these modules is implemented in NIOS.

When the user selects to test the Camera, it will enter camere sub-screen. The image came from Camera is displayed on the LCD shown in Figure 6-19. Once the user touches any area on the screen, it will return the main screen.

The figure 6-20 shows the block diagram of the Camera test module.

Figure 6-19 Screen Shot of Camera Test Module

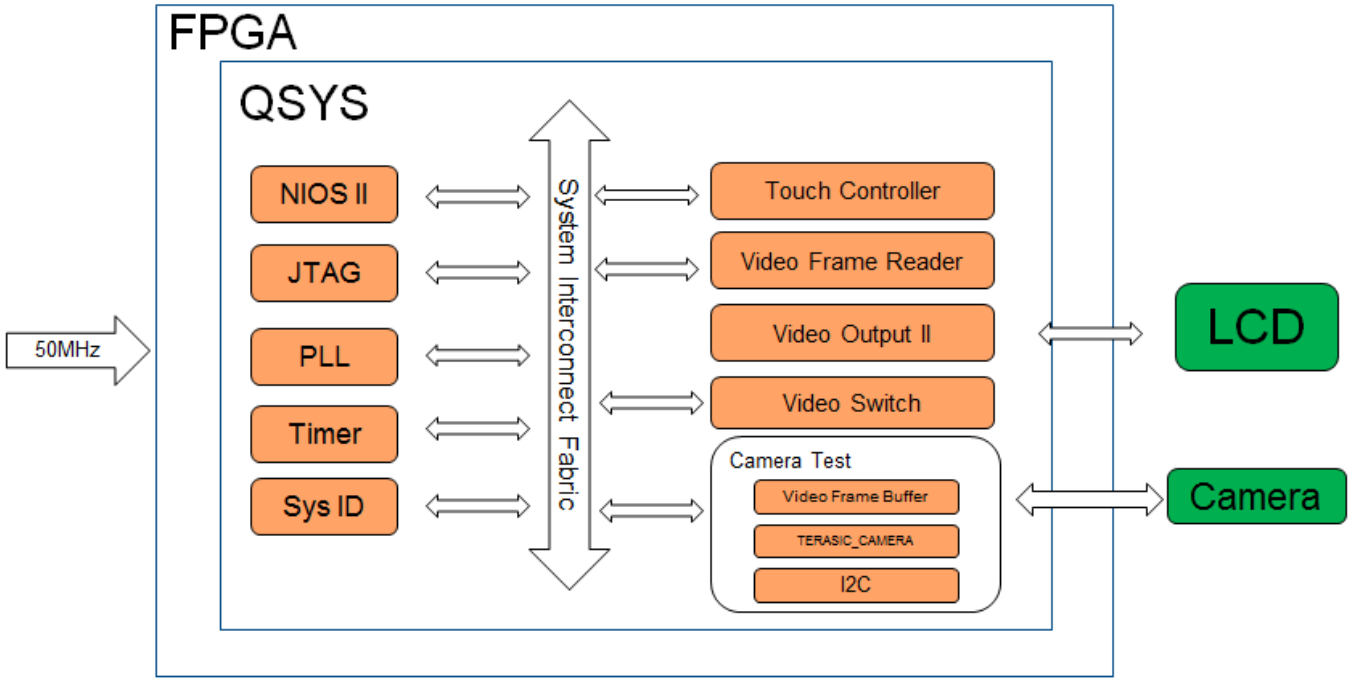


Figure 6-20 Block diagram of Camera Test Module

7 Interface

The reference design uses some of interfaces that connects kinds of peripherals.

7.1 DDR3 Interface

DDR3 SDRAM interface is between MAX10M50DAF484 and IS43TR16640A and IS43TR81280A.

DDR3 interface block diagram is shown as Figure 7-1.

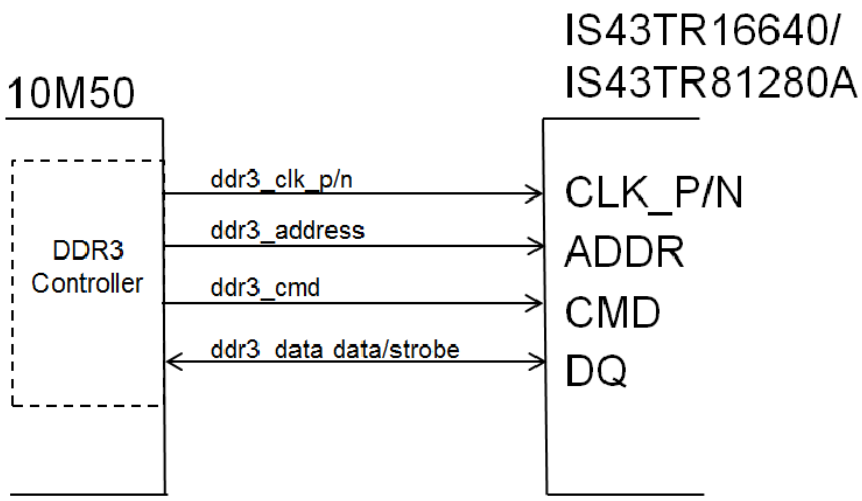


Figure 7-1 Block diagram of DDR3 interface

7.1.1 Pin List Table:

The Table 7-1 shows the interface pin list of DDR3_interface block in NEEK10 FPGA.

Table 7-1 DDR3_Interface Pin List

Signal Name	Width	IO standard	Direction	Description
DDR3_A	15	SSTL-15	Output	Address bus
DDR3_BA	3	SSTL-15	Output	Bank address bus
DDR3_CAS_n	1	SSTL-15	Output	Row address bus
DDR3_CKE	1	SSTL-15	Output	Clock enable

DDR3_CK_n	1	Differential 1.5-V SSTL	inout	Differential output clock
DDR3_CK_p	1	Differential 1.5-V SSTL	inout	Differential output clock
DDR3_CS_n	1	SSTL-15	Output	Chip select
DDR3_DM	3	SSTL-15	Output	Write mask byte lane 0/1/2/3
DDR3_DQ	24	SSTL-15	inout	Data bus byte lane 0/1/2/3
DDR3_DQS_n	3	SSTL-15	inout	Data strobe N byte lane 0/1/2/3
DDR3_DQS_p	3	SSTL-15	inout	Data strobe P byte lane 0/1/2/3
DDR3_ODT	1	SSTL-15	Output	On-die termination enable
DDR3_RAS_n	1	SSTL-15	Output	Row address select
DDR3_RESET_n	1	SSTL-15	Output	Reset
DDR3_WE_n	1	SSTL-15	Output	Write enable

7.2 QSPI Flash Interface

QSPI flash interface is between MAX10M50DAF484 and SPANSION (S25FL512SAGMFIG11) N25Q512A83GSF40F.

DDR3 interface block diagram is shown as Figure 7-2.

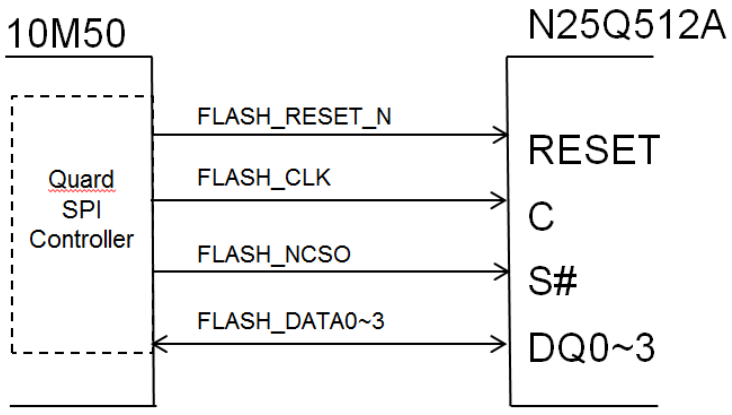


Figure 7-2 Block diagram of QSPI Flash interface

7.2.1 Pin List Table:

The Table 7-2 shows the interface pin list of QSPI Flash interface block in NEEK10 FPGA.

Table 7-2 QSPI Flash Interface Pin List

Signal Name	Width	IO standard	Direction	Description
FLASH_DATA	4	3.3-V LVTTL	inout	data bus
FLASH_DCLK	1	3.3-V LVTTL	Output	clock
FLASH_NCSO	1	3.3-V LVTTL	Output	Chip select
FLASH_RESET_n	1	3.3-V LVTTL	Output	reset

7.3 SD Card Socket Interface

Micro SD Card interface is between MAX10M50DAF484 and Micro SD Card Socket (J23). The interface block diagram is shown as Figure 7-3.

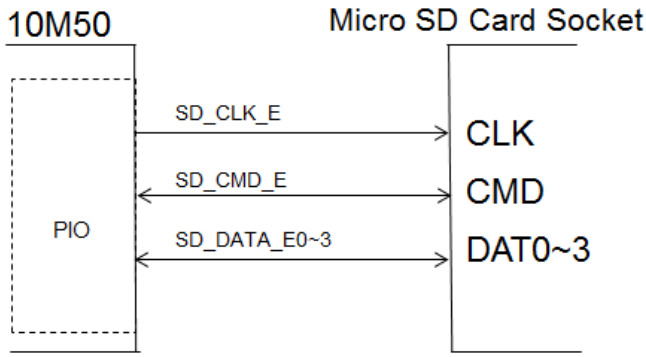


Figure 7-3 Block diagram of SD CARD interface

7.3.1 Pin List Table:

The Table 7-3 shows the interface pin list of SD CARD_interface block in NEEK10 FPGA.

Table 7-3 SD CARD Interface Pin List

Signal Name	Width	IO standard	Direction	Description
SD_CLK	1	2.5 V	output	clock
SD_CMD	1	2.5 V	inout	Data bus output
SD_DATA0	1	2.5 V	input	Data bus input
SD_DATA3	1	2.5 V	Output	Chip select

7.4 LCD Interface

The MAX10 NEEK features a7-inch capacitive amorphous TFT-LCD panel. The LCD touch screen offers resolution of (800x480) to provide users the best display quality for developing applications. The LCD panel supports 24-bit parallel RGB data interface.

The interface block diagram is shown as Figure 7-4.

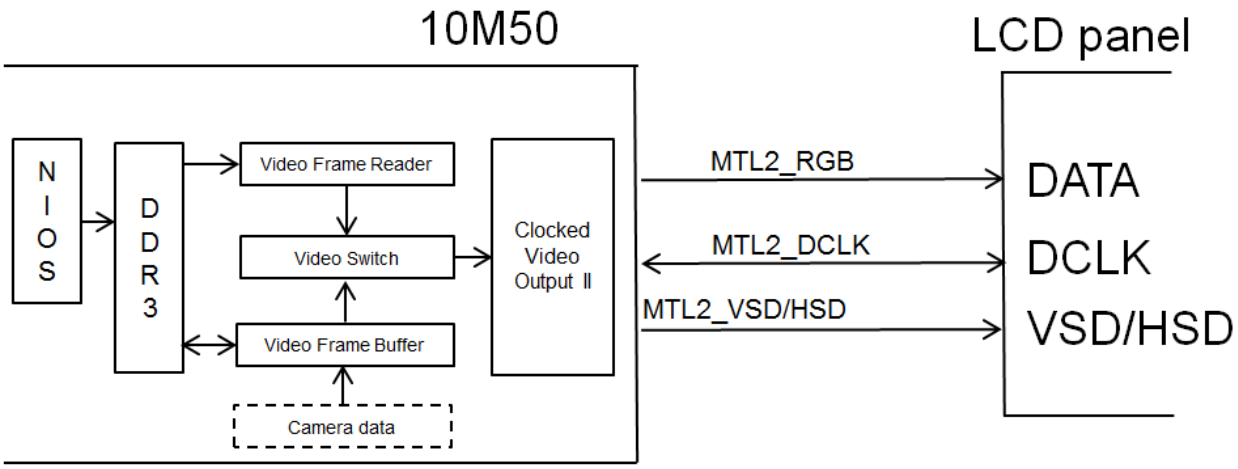


Figure 7-4 Block diagram of LCD interface

7.4.1 Pin List Table:

Table7-4 LCD Interface Pin List

Signal Name	Width	IO standard	Direction	Description
MTL2_B	8	3.3-V LVTTL	output	LCD blue data bus
MTL2_BL_ON_n	1	3.3-V LVTTL	inout	lcd backlight enable
MTL2_DCLK	1	3.3-V LVTTL	input	LCD clock
MTL2_G	8	3.3-V LVTTL	Output	LCD green data bus
MTL2_HSD	1	3.3-V LVTTL	Output	Horizontal sync input
MTL2_R	8	3.3-V LVTTL	Output	LCD red data bus
MTL2_VSD	1	3.3-V LVTTL	Output	Vertical sync input

7.5 Mulit Touch Interface

Multi Touch Screen is a standard I2C interface between MAX10M50DAF484 and Touch Screen. The interface block diagram is shown as Figure 7-5.

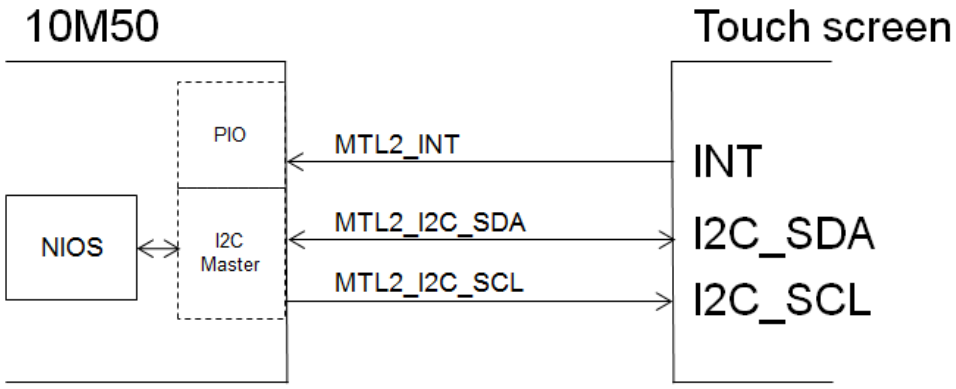


Figure 7-5 Block diagram of Multi Touch interface

7.5.1 Pin List Table:

Table 7-5 Multi Touch Interface Pin List

Signal Name	Width	IO standard	Direction	Description
MTL2_INT	1	3.3-V LVTTL	input	Touch interrupt
MTL2_I2C_SCL	1	3.3-V LVTTL	output	Touch I2C clock
MTL2_I2C_SDA	1	3.3-V LVTTL	inout	Touch I2C Data

7.6 Ethernet Interface

The Ethernet interface is between MAX10M50DAF484 and an external Marvell 88E1111 PHY. It supports 10/100/1000 base-T Ethernet using Alter Triple-Speed Ethernet MegaCore MAC function. The interface block diagram is shown as Figure 7-6.

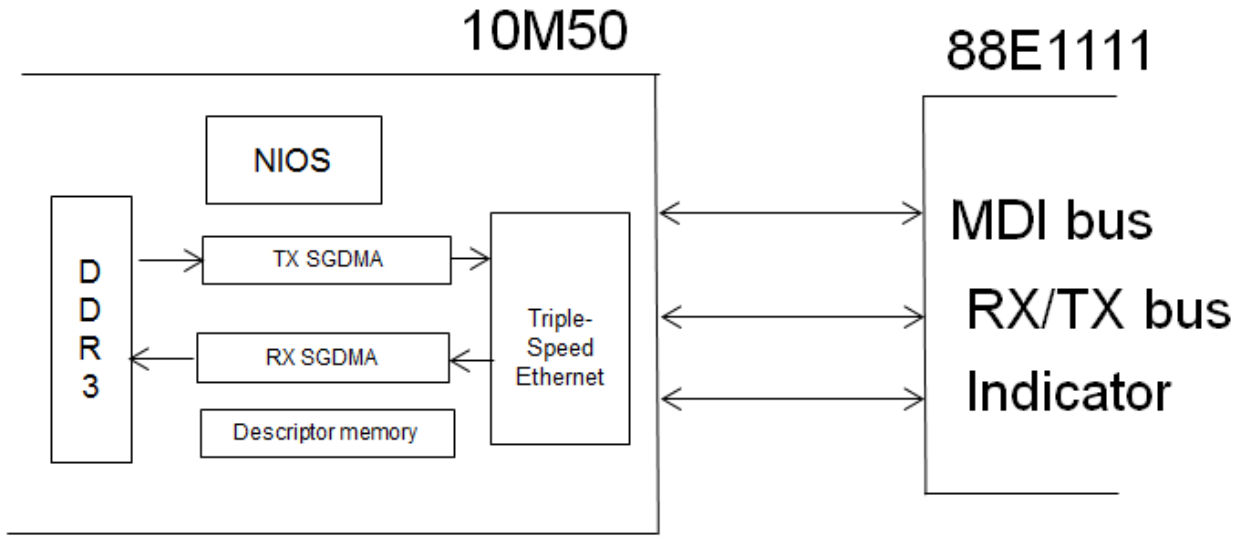


Figure 7-6 Block diagram of Ethernet interface

7.6.1 Pin List Table:

Table 7-6 Ethernet Interface Pin List

Signal Name	Width	IO standard	Direction	Description
NET_GTX_CLK	1	2.5 V	output	125 MHz RGMII TX clock
NET_INT_n	1	2.5 V	input	Management bus interrupt
NET_LINK100	1	2.5 V	input	100 Mb link LED
NET_MDC	1	2.5 V	Output	MDI clock
NET_MDIO	1	2.5 V	inout	MDI data
NET_RST_N	1	2.5 V	Output	reset
NET_RX_CLK	1	2.5 V	input	RGMII RX clock
NET_RX_COL	1	2.5 V	input	MII Collision
NET_RX_CRS	1	2.5 V	input	MII Carrier Sense
NET_RX_D	4	2.5 V	input	RGMII TX data
NET_RX_DV	1	2.5 V	input	RGMII RX valid
NET_RX_ER	1	2.5 V	input	MII RX error
NET_TX_CLK	1	2.5 V	input	25/2.5 MHz MII TX clock

NET_TX_D	4	2.5 V	output	RGMII TX data
NET_TX_EN	1	2.5 V	output	RGMII TX enable
NET_TX_ER	1	2.5 V	output	MII TX error

7.7 DAC Interface

The DAC interface is between MAX10M50DAF484 and DAC8551 based on SPI protocol. The interface block diagram is shown as Figure 7-7.

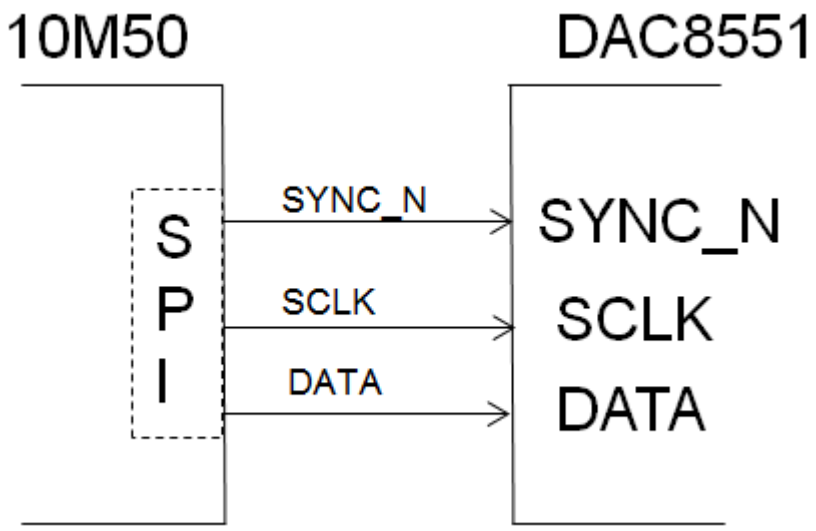


Figure7-7 Block diagram of DAC interface

7.7.1 Pin List Table:

Table 7-7 DAC Interface Pin List

Signal Name	Width	IO standard	Direction	Description
DAC_DATA	1	3.3-V LVTTL	inout	Serial data
DAC_SCLK	1	3.3-V LVTTL	output	Serial clock
DAC_SYNC_n	1	3.3-V LVTTL	output	Level-triggered control input

7.8 ADC Interface

The reference design implements two functions through ADC interface, wave_detection came from ADC0_channel7 and on-die temperature measurement.

The ADC pins are dedicated, so the user need not to specify these pins.

7.9 External Sensor Interface

NEEK10 has three external sensors, including digital accelerometer, miniature ambient light photo Sensor and Accuracy Digital Humidity Sensor with Temperature Sensor. The design is implemented based on I2C protocol. The interface block diagram is shown as Figure 7-8.

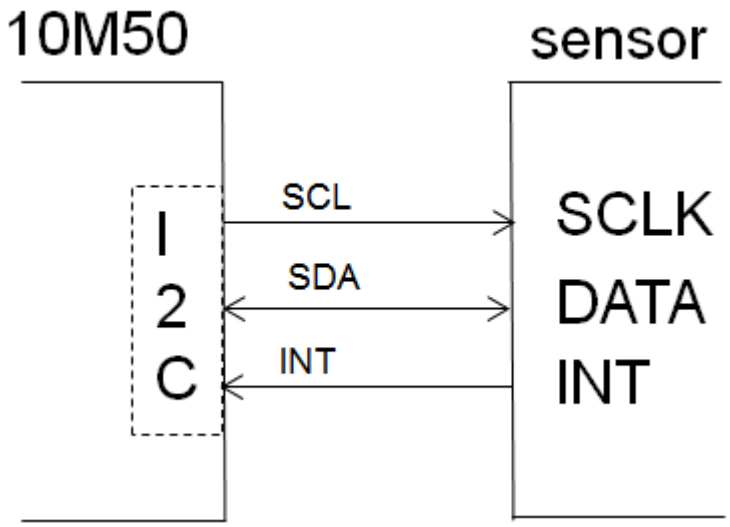


Figure 7-8 Block diagram of external sensor interface

7.9.1 Pin List Table:

Table 7-8 External sensor Interface Pin List

Signal Name	Width	IO standard	Direction	Description
GSENSOR_CS_n	1	2.5 V	output	Chip select
GSENSOR_INT	2	2.5 V	input	Interrupt
GSENSOR_SCLK	1	2.5 V	inout	Serial clock
GSENSOR_SDI	1	2.5 V	inout	Serial data
GSENSOR_SDO	1	2.5 V	inout	Serial data

LSENSOR_INT	1	3.3-V LVTTL	inout	Interrupt
LSENSOR_SCL	1	3.3-V LVTTL	output	Serial clock
LSENSOR_SDA	1	3.3-V LVTTL	inout	Serial data
RH_TEMP_DRDY_n	1	3.3-V LVTTL	input	Data ready
RH_TEMP_I2C_SCL	1	3.3-V LVTTL	Output	Serial clock
RH_TEMP_I2C_SDA	1	3.3-V LVTTL	inout	Serial data

7.10 Camera Interface

The Camere interface is between MAX10M50DAF484 and connector J27. The interface block diagram is shown as Figure 7-9.

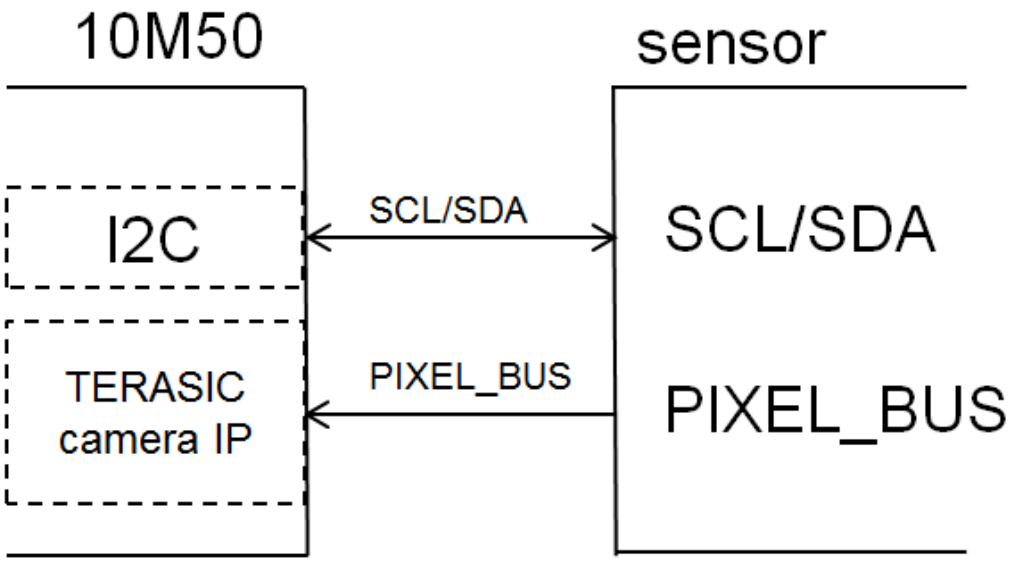


Figure 7-9 Block diagram of Camera interface

7.10.1 Pin List Table:

Table 7-9 Camera Interface Pin List

Signal Name	Width	IO standard	Direction	Description
MIPI_CORE_EN	1	3.3-V LVTTL	output	MIPI enable
MIPI_CS_n	2	3.3-V LVTTL	output	Chip select
MIPI_I2C_SCL	1	3.3-V LVTTL	output	I2C Serial clock
MIPI_I2C_SDA	1	3.3-V LVTTL	inout	I2C Serial data

MIPI_PIXEL_CLK	1	3.3-V LVTTL	input	Camera input clock
MIPI_PIXEL_D	24	3.3-V LVTTL	input	Camera input data
MIPI_PIXEL_HS	1	3.3-V LVTTL	input	Horizontal sync input
MIPI_PIXEL_VS	1	3.3-V LVTTL	input	Vertical sync input
MIPI_RESET_n	1	3.3-V LVTTL	output	Camera reset
MIPI_REFCLK	1	3.3-V LVTTL	Output	Camera ref clock
MIPI_WP	1	3.3-V LVTTL	Output	

8 Revision History

shows the revision history for the Reference Design Document*: NEEK10 NIOS Design Example.*

Table 8-1 Reference Design Document Revision History

Version	Date	Change Summary
1.0	August 2015	First release of this Reference Design Document.