# TABLE OF CONTENTS

INTRODUCTION .................................................................................................................. 3
  1.1 OVERVIEW .................................................................................................................. 3
  1.2 Qsys ........................................................................................................................... 4
  1.3 1.3 Quartus Prime Software ....................................................................................... 4
  1.4 Nios II Software Build Tools for Elipse .................................................................. 5

2 REFERENCE DESIGN HARDWARE .............................................................................. 6
  2.1 Nios II Processor ....................................................................................................... 8
  2.2 DDR3 SDRAM Controller ......................................................................................... 8
  2.3 Video Processing Module ......................................................................................... 8
  2.4 Static Picture/Text Processing Modules ................................................................... 9
  2.5 Alpha Blending Mixer II Processing Module ............................................................. 9
  2.6 Video Output Module .............................................................................................. 10

3 SOFTWARE CONCEPTS .............................................................................................. 11
  3.1 Frame Buffers in DDR3 ............................................................................................ 11
  3.2 Reverse Parking Aid ................................................................................................ 11

4 REFERENCE DESIGN DEMOSTRATION ..................................................................... 12

5 SYSTEM REQUIREMENT ............................................................................................... 13
  5.1 Hardware Requirements ......................................................................................... 13
  5.2 Hardware Requirements ......................................................................................... 13

6 OPENING THE EXAMPLE DESIGN ............................................................................ 14
  6.1 Opening the Quartus II Top-Level Project ................................................................. 14
  6.2 Opening the Qsys System ....................................................................................... 14
  6.3 Viewing the Parameters ......................................................................................... 14
  6.4 Examining and Recompiling the Nios Code ............................................................. 16

7 REVISION HISTORY ...................................................................................................... 17
Introduction

1.1 Overview

The TFT Graphical Controller Example Design demonstrates Intel Max10 FPGAs in a graphics system. The example design runs on the MAX® 10 NEEK board, has the following features:

- SDRAM program store and frame buffer
- 3-layer display design, supporting picture-in-picture, resizing and alpha for all layers
  - Layer 0 – Backdrop picture layer, RGB888
  - Layer 1 – Backdrop text layer, RGB888
  - Layer 2 – Video layer, RGB888.

- Fits on a Max10 10M50 device

The design uses the following design software:

- Qsys for abstract design capture, parametrizable IP cores and switch fabric generation
- The Quartus® Prime software for pin mapping and to build the MAX10 device configuration image
- 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.

Table 1 shows the IP cores used, including IP from the Intel Video and Image processing (VIP) suite

- For more information on the VIP suite see www.altera.com/vip
Table 1. IP cores used in the design

<table>
<thead>
<tr>
<th>VIP suite IP cores:</th>
<th>Nios II processor</th>
</tr>
</thead>
<tbody>
<tr>
<td>CVO (clocked video out)</td>
<td>DDR3 memory controller</td>
</tr>
<tr>
<td>Frame reader</td>
<td>Clock crossing bridge</td>
</tr>
<tr>
<td>Frame buffer</td>
<td>On chip memory</td>
</tr>
<tr>
<td>Mixer II</td>
<td>JTAG UART</td>
</tr>
<tr>
<td>Scaler II</td>
<td>PIO</td>
</tr>
<tr>
<td></td>
<td>System ID</td>
</tr>
<tr>
<td>Terasic camera conduit</td>
<td>Interval timer</td>
</tr>
<tr>
<td>Opencores i2c (touchscreen and camera)</td>
<td>Avalon PLL</td>
</tr>
</tbody>
</table>

1.2 Qsys

The Qsys flow is the primary design flow for the MAX10 NEEK Nios II system development. Specifically, Qsys simplifies the process of system design, 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 interfaces. 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 uses a single memory controller but contains multiple Avalon-MM masters that buffer video data in an external memory.

1.3 Quartus Prime software

The Quartus Prime 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 Prime software includes a wide range of tools to help with timing closure and perform hardware compilation to generate an FPGA programming file.

- For more information about Quartus Prime software, refer to the Quartus Prime help.
1.4 Nios II Software Build Tools for Eclipse

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. (www.altera.com/nios)
2 Reference Design Hardware

Table 2-1 shows the files in the top-level directory.

<table>
<thead>
<tr>
<th>File Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NEEK10_TFT_controller.qpf</td>
<td>Quartus Prime project file.</td>
</tr>
<tr>
<td>NEEK10_TFT_controller.qsf</td>
<td>Quartus Prime settings file.</td>
</tr>
<tr>
<td>NEEK10_TFT_controller.v</td>
<td>Top-level file.</td>
</tr>
<tr>
<td>NEEK10_TFT_controller.sof</td>
<td>Programming file.</td>
</tr>
<tr>
<td>NEEK10_TFT_controller.sdc</td>
<td>Contains top-level clock requirements</td>
</tr>
<tr>
<td>Max10_neek10.sopcinfo</td>
<td>System design file</td>
</tr>
</tbody>
</table>

Figure 2-2 shows the hardware block diagram of the reference design.
The design comprises the following blocks:

- A Terasic Camera component for receiving pixel data from a camera
- A NiosII processor for real-time control
- A DDR3 controller for data storage in external memory
- FrameReader/FrameBuffer for reading storing video frames in external memory and transporting them using the Avalon-ST Video protocol.
- A ScalerII IP for changing the input resolution
- A three-layer Blend MixerII IP with the following layers:
  - Layer 0 – Backdrop picture layer, RGB888
  - Layer 1 – Backdrop text layer, RGB888
  - Layer 2 – video layer, RGB888.
- A Clocked Video Output (CVO) IP

The design also includes the following components that are not on the block diagram:

- Avalon ALTP PLL
- I2C bus Master
- System ID/PIO Peripheral
- JTAG UART

All layers require access to the DDR3 SDRAM. Because of the bandwidth limitation, the design limits the camera resolution to 400x240. The scalerII is responsible for changing the CAMERA image to 800x480 resolution.

All VIP cores use the Avalon-ST Video protocol to transmit data, which increases productivity through design flexibility and reuse.

2.1 Nios II Processor

The Nios II processor is generated using Intel’s Qsys tool. The Nios II processor initializes and configures multiple video system components, including the I2C external bus masters and the Video and Image Processing Suite IP cores.

The design connects the Nios II Avalon-MM data master to each Avalon-MM slave port it needs access to. Qsys automatically constructs the bus architecture. The system stores the Nios II program code in external memory.

2.2 DDR3 SDRAM Controller

The reference design software is stored in the SDRAM, where is it executed. The Intel DDR3 SDRAM Controller controls the data flow to and from the external DDR3 SDRAM. The design uses the controller in half-rate mode, operating at 300 MHz, with a local data interface width of 24 bits. This configuration provides a theoretical maximum available bandwidth of 150×24 = 3.6 Gbps. The design uses the 50-MHz system clock as the PLL reference clock of DDR3 controller.

2.3 Video Processing Module

The Avalon video module provides a flexible video capture solution, it has the following features:

- Terasic camera IP connected to camera module
  
  - Because of the bandwidth limitation of DDR3, the output video data of Camera Component is a 400x240 resolution.

- Scaling of input image
  
  - For normal display, scaler II IP expands the input video with 400x240 pixels to 800x480.

- Avalon master to write/read image(s) to/from frame buffer memory

- Avalon register slave for control and status
2.4 Static Picture/Text processing modules

Nios II processes generates the pre-loaded backdrop picture/text and stores them to DDR3. The FrameReader IP core transfers them to the Mixer II IP for further processing.

2.5 Alpha Blending Mixer II processing module

The Alpha Blending Mixer II mixes three layers with alpha blending to provide a picture-in-picture effect. The design uses the Avalon-MM Slave interface at run-time to specify the location of all the foreground layers, relative to the background layer.

To simplify the design, the MixerII IP uses a static alpha value for all input layers. The Nios II processor controls the Mixer II via the Avalon-MM slave interface.
Figure 2-5 shows the mixer II processing module.

2.6 Video Output Module

The design streams the mixed image to the Clocked Video Output II IP core, which forms the video stream output boundary to the Qsys system.

The CVO II IP core converts from the Avalon-ST Video protocol to clocked video formats. It uses the Avalon-ST Video control and active picture packets to format Avalon-ST Video into clocked video by inserting horizontal and vertical blanking and generating horizontal and vertical sync information. The active picture data is not converted; the color plane information remains the same as in the Avalon-ST Video format.

This design configures the Clocked Video Output IP core to accept video data with three color planes in parallel, namely, RGB. The IP core also outputs clocked video data with the sync signals on separate wires.

The clocked video output also provides clock crossing capabilities to provide greater design flexibility by enabling the video processing system clock to be decoupled from the video output pixel clock. The output pixel clock is set to 33 MHz and the system clock is 150 MHz.

2.7 Terasic IP modules

The MAX 10 NEEK development kit is designed and sold by Terasic. In the design example the MIPI camera and touch LCD functions have been ported directly from the MAX 10 NEEK kit example designs.

➢ For more information about the MAX 10 NEEK kit, refer to the following link.
  
3 Software Concepts

3.1 Frame Buffers in DDR3

The example design requires that the start address of each frame buffer is word aligned. All Frame Buffers use 24 bits per pixel to represent 16,777,216 colors using RGB888 encoding. All frame buffer memory must be allocated with 3 bytes per pixel, e.g., a 800x480 display requires a frame buffer of 1,152,000 bytes.

The example design uses the following buffers located in DDR3 for frame buffer:

- One 800x480x3 Bytes buffer for framereader, stores background picture
- One 800x480x3 Bytes buffer for framereader, stores background text
- Three 400x240x3 Bytes buffer for framebuffer, stores camera images for performing simple frame rate conversion

3.2 Reverse Parking Aid

The example design shows a simple reverse parking aid function. The function is mixed image for a pre-drawn static picture and camera image. The camera video is overlaid on the static picture with a trapezoid clear region in the center of the screen surrounded by other opaque region. The alpha-value of camera video is 50%.

Figure 3-1 shows the reverse-parking aid overlay.

![Figure 3-1 reverse-parking aid overlay](image-url)
4 Reference Design Demonstration

The example design demonstrates the following actions:

- Fade in background image
- Fade in/out introductory text image
- Reverse parking Aid
- Fade in/out full frame video
- Fade in/out half frame video
- Dynamic video resizing
- Move picture-in-picture video around the screen
- Fade in/out end text image
- Fade out background image
5 System Requirement

This section describes the hardware and software requirements to run the MAX 10 NEEK TFT Graphical Controller design example.

5.1 Hardware Requirements

The MAX 10 NEEK TFT Graphical Controller Design Example requires the following hardware:

- MAX 10 NEEK Development Kit
- Mini-USB cable for programming the device
- 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.

5.2 Design Software Requirements

Ensure that you download and install the design max10_neek_tft_controller.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 MAX10 NEEK Development Kit.

You must install the Quartus Prime software, version 16.0, which includes the MegaCore IP Library and the Nios II SBT, on your PC.

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

Ensure that you can extract max10_neek_tft_controller.qar and open max10_neek_tft_controller.qpf in your Quartus Prime software.
6 Opening the Example Design

This section describes how to open the MAX 10 NEEK TFT Graphical Controller Design Example. This section includes the following sections:

- Opening the Quartus Prime 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

6.1 Opening the Quartus Prime Top-Level Project

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

1. Launch the Quartus Prime software.
2. On the File menu, click Open Project, browse to <design example install directory>, and select the neek10_tft_controller.qpf Quartus Prime project file.
3. On the File menu, click Open, browse to <design example install directory>	op, and select the neek10_tft_controller.v top-level Verilog HDL design file.

6.2 Opening the Qsys System

To open the design example Qsys system, with the neek10_tft_controller.qpf project open in the Quartus Prime 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 max10_neek.qsys file for this design.

6.3 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).
Table 6-1 shows the main component names to click to view the parameters of each IP core.

![Figure 6-1 Parameter Settings for the DDR3 SDRAM controller with UniPHY](image)

Table 6-1 IP Cores in Design Example IP Cores

<table>
<thead>
<tr>
<th>Name</th>
<th>IP Core</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mem_if_ddr3_emif</td>
<td>DDR3 SDRAM Controller with UniPHY</td>
</tr>
<tr>
<td>Alt_vip_vfr</td>
<td>Video Frame Reader</td>
</tr>
<tr>
<td>Alt_vip_vfb</td>
<td>Video Frame Buffer</td>
</tr>
<tr>
<td>Alt_vip_cl_cvo</td>
<td>Clocked Video Output II(4K Ready)</td>
</tr>
<tr>
<td>TERASIC_CAMERA</td>
<td>TERASIC CAMERA Component</td>
</tr>
<tr>
<td>Alt_vip_cl_mixer</td>
<td>Mixer II (4k Ready)</td>
</tr>
<tr>
<td>Alt_vip_cl_scl</td>
<td>Scaler II (4k Ready)</td>
</tr>
</tbody>
</table>
6.4 Examining and Recompiling the Nios II 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 Launcher 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 max10_neek.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 max10_neek_tft_controller.
7. Select Blank Project from the Templates list and then click Next.
8. Select Create a new BSP project based on the application project template.
9. Accept the default project name, max10_neek_tft_controller_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 max10_neek_tft_controller and max10_neek_tft_controller_bsp projects appear in the Project Explorer view.
13. In windows Explorer, navigate to design’s software\source directory from max10_neek_tft_controller project.
14. Select all files in source\ folder and drag them onto max10_neek_tft_controller in the Nios II SBT for Eclipse Project Explorer view to include them in the project. Select all files in source\folder and drag them onto max10_neek_tft_controller in the Nios II SBT for Eclipse Project Explorer view to include them in the project.
15. Right-click max10_neek_tft_controller_bsp in the Project Explorer view, point to NiosII, click Generate BSP.
16. Right-click max10_neek_tft_controller in the Project Explorer view, click Build Project. The Nios II SBT for Eclipse compiles the max10_neek_tft_controller_bsp and max10_neek_tft_controller software projects and builds the max10_neek_tft_controller.elf software executable.
7 Revision History

This section shows the revision history for the Reference Design Document:

Nios II / MAX10 NEEK TFT Graphical Controller Design Example

Table 7-1 Reference Design Document Revision History

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Change Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>June 2017</td>
<td>Add Chapter 2.7 Teraasic IP modules</td>
</tr>
<tr>
<td>1.0</td>
<td>Dec. 2016</td>
<td>First release of this Reference Design Document.</td>
</tr>
</tbody>
</table>