Lab 3.0
Camera, LCD & VGA conceptual design

Introduction

In the previous labs, you focused on the design of custom *slave* interfaces which are targeted towards simple use cases where *small* amounts of data needs to be moved between a peripheral and a processor. In this lab we will instead focus on the design of custom *master* interfaces which are targeted towards complex use cases where *large* amounts of data need to be moved:

- From a peripheral to memory (e.g. camera)
- From memory to a peripheral (e.g. display)

Goal

The goal of this lab is for you to *propose a detailed design* for an FPGA-based system which can interface with one of the following peripherals on the DE0-Nano-SoC:

- TRDB-D5M camera
- LT24 LCD display
- CDK3404 Triple Video DAC

FIGURE 1. TRDB-D5M, LT24 & VGA INTERFACES

René Beuchat, Sahand Kashani

You will *not code anything* during this lab and will instead spend time to come up with a *rock-solid* system design that achieves the required task. This project is much larger than the ones you have performed until now, and having an agreed-upon detailed design is crucial to avoid getting lost when implementing the design at a later stage.

It is only once you have your full design ready that you will be authorized to implement your proposed design in lab 4.0 next time.

Theory

Due to the large amount of data that needs to be moved to/from these peripherals, your custom IP component will have a *master* interface to automatically handle the transfer of image frames from the camera to memory and from memory to the display.

Since all master interfaces need to be configured by the processor, master interfaces also forcefully have a *slave* control interface. Your IP component will therefore have both a slave interface for configuration purposes, and a master interface for efficient high-speed data transfers to/from memory.

Finally, as for the custom slave interfaces developed in the previous labs, your component will have a peripheral-specific *conduit* interface to connect to the specific pins of your chosen peripheral.

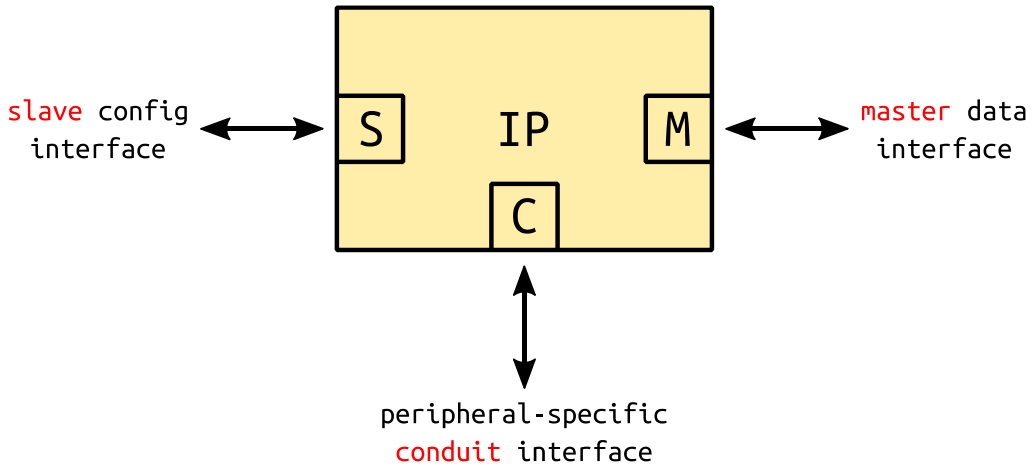

FIGURE 2. GENERAL BLOCK DIAGRAM OF TARGET IP COMPONENT

You will then package your custom component as an IP component in Qsys and add it to a base Qsys system where the processor will then access and program your IP component as needed to control the external peripheral and transfer an image from the camera to memory, or from memory to the display.

TRDB-D5M

The TRDB-D5M is a 5 megapixel digital camera available as an extension card for Terasic FPGAs. It connects to a development board through a 2x20 pin GPIO connector.

The camera uses an I^2C control interface and provides frame data on dedicated pins. It supports resolutions up to 2592x1944, but since the final goal of the project is for you to transfer an image taken with the camera onto the LT24 display, your task will be to output an image with a resolution of 320x240 (the same as the LT24). However, the minimum resolution supported by the camera is 640x480, so your design be able to *down sample* the images to 320x240 during acquisition *without* involving the processor in any way.

Your task will be to concentrate on the data acquisition part of the design (i.e. transferring a frame from the camera to memory). We will provide you with an I^2C controller that you can use to communicate with the camera for configuration purposes. The only requirement we impose for this interface is that you plug it into the `GPIO_1` connector on the development board (DE0-Nano-SoC).

You can find the user guide and datasheet for the device in the SystemCD provided by Terasic. Note that Terasic provides "example designs" in the SystemCD, but we ask that you do not waste any time by look at them. The reason is simple: no documentation is provided, and a demo with no documentation is a useless demo.

LT24

The LT24 is a 2.4-inch LCD touch module with a resolution of up to 320x240. It connects to a development board through a 2x20 pin GPIO connector.

The display contains a small controller and uses a full-custom protocol for communication through its pins, so we redirect you to its datasheet for more information on how to interface with it. Your task will be to transfer a frame from memory to the LCD. The only requirement we impose for this interface is that you plug it into the `GPIO_0` connector on the development board (DE0-Nano-SoC).

You can find the user guide and datasheets for the device in the SystemCD provided by Terasic. As for the TRDB-D5M, note that Terasic provides "example designs" in the SystemCD, but they again come with no documentation and are therefore useless. Please don't waste your time looking at them.

CDK3404

The CDK3404 is a triple video DAC which supports digital to analog video signal conversions for RGB, YC_bC_r, and "Composite, Y, C" video formats.

The goal is to use the CDK3404 to display an image on a standard desktop monitor through a VGA cable. Your VGA controller will need to use the CDK3404 in RGB mode to successfully output an image. The VGA standard supports many target resolutions, but you will implement the most standard one (*640x480@60 Hz*). You can find a collection of industry-standard timings on this website. The VGA standard is fairly simple and you can find numerous websites which describe its signals. Figure 3 gives an example of the HSYNC and VSYNC signals that are needed to synchronize the display output.

You can find the VGA extension board's schematic and the CDK3404's datasheet on Moodle.

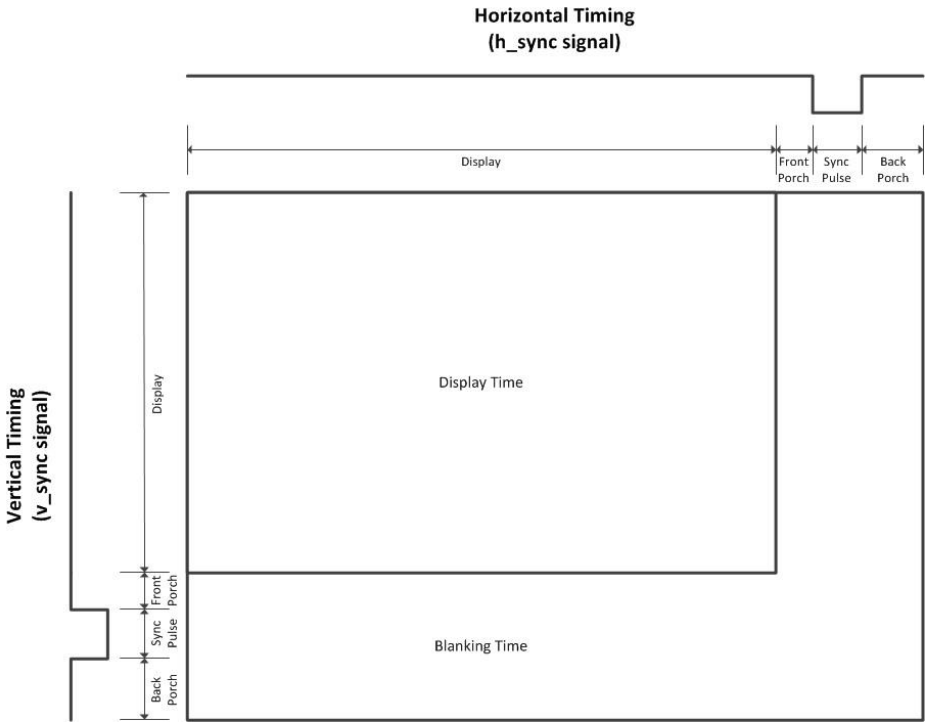

FIGURE 3. VGA VSYNC & HSYNC SIGNALS

(REFERENCE: HTTPS://EEWIKI.NET/PAGES/VIEWPAGE.ACTION?PAGEID=15925278)

Practice

The goal is to propose a *detailed design* of a complete FPGA-based system that can interface with one of the supplied peripherals. This means that you have to perform the following tasks:

1. Select which peripheral you are targeting. You will need to read their datasheets to get a sense of what each component's interface is, and how it functions internally.
2. Provide the mandatory documents that all designs must have (described in lab 2.2):
    a. *Full system block diagram* detailing the general architecture of the FPGA-based system that is designed (with processor, memory, interconnect, peripherals, …)
    b. *Custom IP block diagram* detailing how your interface is it modularized into different sub-blocks and the detailed design of each sub-block.
    c. *Custom IP finite state machine (FSM) diagram* detailing how your interface sequences it's various subcomponents to achieve its function.
    d. *Custom IP register map* detailing how a software programmer would use your custom IP.
    e. *Top-level block diagram* detailing connections between your custom IP and the development board's pins.
3. Describe the memory organization you will use to store your images (display buffer), as well as the pixel representation and pixel ordering within the buffer.

4. Describe the sequence of commands that must be supplied to the TRDB-D5M (I^2C commands) and the LT24 (custom commands) for actions to take place. These commands include everything that is needed to set up the corresponding device resolutions, oscillators, …

Finally, you will *write a report* where you put together all this information for reference when you start implementing your design during the next lab.

The report will be *extremely* detailed and will specify everything that is needed to implement the design. As a general goal, the report should be written with the idea that, if you were to hand it over to *another engineer* who knows *nothing* about the device, that he/she would be able to successfully implement the full design.