Altera DE2:
ISP1362 USB Controller
Application Notes

By: Tarek Kaddoura
Jigar Nahar

Table of Contents

Introduction

The Altera DE2 Board contains a ISP1362 USB OTG Controller. The controller supports a wide range of functions as described in the reference manual. It can act as a USB host or act as a USB device. This document describes the basic procedure to set up the ISP1362 in Quartus, as well as the software drivers necessary to get it working.

Hardware Configuration

Terasic has provided the HDL and TCL files necessary to add the component to a system through SOPC builder. These files can be downloaded from
https://www.ualberta.ca/~delliott/local/ece492/appnotes/2013w/USB ISP1362/ISP1362.tar

SOPC Builder

Note: The following steps only show what needs to be added to get the ISP1362 working. They assume that a NIOSII processor and other components (such as RAM) have already been added to the system.

In the downloaded file, there is a folder named hardware. Copy this folder to the root of your Quartus project.

1. In Quartus, go to Tools → SOPC Builder.
2. In the component list in SOPC Builder, there should be a new category: Terasic Technologies. Under this category is the ISP1362 component. Add this component to the system.
3. Generate the system.
4. Add the generated files to your project.

Top Level Modifications

Now that the component has been added, it needs to be connected through the top level HDL file.

Note: The following instructions assume that the ISP1362 component was named “isp1362” in SOPC Builder. Also, all necessary code is shown in VHDL.

1. Add the pins necessary in your top level entity.

```
-- USB Ports
OTG_CS_N        :   out     std_logic;
OTG_RD_N        :   out     std_logic;
OTG_WR_N        :   out     std_logic;
OTG_RST_N       :   out     std_logic;
OTG_ADDR        :   out     std_logic_vector(1 downto 0);
OTG_DATA        :   inout   std_logic_vector(15 downto 0);
OTG_INT0        :   in      std_logic;
OTG_INT1        :   in      std_logic;
OTG_FSPEED      :   out     std_logic;
OTG_LSPEED      :   out     std_logic;
OTG_DACK0_N     :   out     std_logic;
OTG_DACK1_N     :   out     std_logic;
```

```
OTG_DREQ0       :       out         std_logic;
OTG_DREQ1       :       out         std_logic;
```

2. Add the necessary signals into your system component.

```
-- the_ISP1362
avs_hc_export_OTG_ADDR_from_the_ISP1362 : OUT STD_LOGIC_VECTOR (1 DOWNTO 0);
avs_hc_export_OTG_CS_N_from_the_ISP1362 : OUT STD_LOGIC;
avs_hc_export_OTG_DATA_to_and_from_the_ISP1362 : INOUT STD_LOGIC_VECTOR (15 DOWNTO 0);
avs_hc_export_OTG_INT0_to_the_ISP1362 : IN STD_LOGIC;
avs_dc_export_OTG_INT1_to_the_ISP1362 : IN STD_LOGIC;
avs_hc_export_OTG_RD_N_from_the_ISP1362 : OUT STD_LOGIC;
avs_hc_export_OTG_RST_N_from_the_ISP1362 : OUT STD_LOGIC;
avs_hc_export_OTG_WR_N_from_the_ISP1362 : OUT STD_LOGIC;
```

3. Connect the pins as shown below in the VHDL code.

```
avs_hc_export_OTG_ADDR_from_the_ISP1362 => OTG_ADDR,
avs_hc_export_OTG_CS_N_from_the_ISP1362 => OTG_CS_N,
avs_hc_export_OTG_DATA_to_and_from_the_ISP1362 => OTG_DATA,
avs_hc_export_OTG_RD_N_from_the_ISP1362 => OTG_RD_N,
avs_hc_export_OTG_RST_N_from_the_ISP1362 => OTG_RST_N,
avs_hc_export_OTG_WR_N_from_the_ISP1362 => OTG_WR_N,
avs_hc_export_OTG_INT0_to_the_ISP1362 => OTG_INT0,
avs_dc_export_OTG_INT1_to_the_ISP1362 => OTG_INT1,
```

The ISP1362 component should now be connected. The final step is to program the board with the new configuration.

Software Configuration

After adding the component to the system, the software drivers are not added automatically in the NIOS II IDE. The following additions and initializations need to be done in software.

2. Add all the files under the folder software from the tar file into your root project directory.
3. If you named your component something other than “ISP1362” in SOPC Builder, then open the ISP1362_HAL.H file and add the line:

```
#define ISP1362_BASE SOPCNAME_BASE
```

After the software modifications are complete, the ISP1362 should be working. The following section will test the chip and confirm the interface is working.

Hardware Test

In order to test if the interface between the NIOS II and the ISP1362 is working, the chip ID of the USB Chip will simply be read in software. In the reference manual of the ISP1362, the chip ID register is supposed to return 3630 in hex. Hence, if this shows up then the interface is working.

Start by creating a simple Hello World project in the NIOS II IDE.

1. Perform the software modifications mentioned above.
2. Call the following function wherever possible to watch the output on stdout

   Hal4D13_RegAccess();
3. Compile and run the program on the board

The function Hal4D13_RegAccess() performs several functions. It will reset the USB controller, print out the chip id, check the chip RAM, and print which configuration modes the chip is in.

Once the program runs, if the chip ID is readable and equal to 3630, then the interface to the ISP1362 is set up and working.

Further Reading

This document only outlines how to get the ISP1362 connected and working. It does not explain the different USB modes, nor how to actually use the USB with a device.

Altera provides two different examples on USB integration. The projects are the DE2_NIOS_HOST_MOUSE_VGA and DE2_NIOS_DEVICE_LED.

The first project uses USB Host in order to interface with a mouse. It performs different actions on left or right mouse click. The second project uses USB in order to communicate with a computer.

Both projects provide a multitude of example code on how to communicate through the USB. They also contain implementations of Chapter 9 of the USB Device Specifications in Chap_9.c.

Note: The demonstration projects provided on the University labs are quite old, and will likely lead to hardware or other compilation errors. A newer version of the demonstrations updated to support Quatus II 10.0 are the DE2_70 demonstrations. These demonstrations are not for the Cylone II, but they contain excellent examples of updated code and drivers for the newer Quartus. These demonstrations can be downloaded from here: http://www.terasic.com/downloads/cd-rom/de2_70/DE2_70_demonstrations_V10.rar

References

Datasheet for the ISP1362:
http: //www.cs.columbia.edu/~sedwards/classes/2013/4840/Philips-ISP1362-USB-controller.pdf

Initial drivers and HDL files: http://www.terasic.com/downloads/cd-rom/de2_70/DE2_70_demonstrations_V10.rar