Final project proposal

Digital photo frame

Project group name: I deer you, it'll work (IDY)
Project group members: Martin Olsson (e03mo), Samuel Skånberg (dt05ss5), Thomas Eriksson (dt05te2)
Mentor: Flavius Gruian (Department of computer science, LTH)
Course: EDA385 - Design of Embedded Systems Advanced Course, LP1 Ht09

Project proposal

Brief description

The project we have set out to do is a interactive photo frame based on a MicroBlaze FPGA. As a dream scenario the photo frame loads bitmap images (BMP) from and SD card inserted into the SD card reader module. It then displays the first image on the SD card on a VGA display connected to the built in interface on the MicroBlaze. In order to view the next image it should be possible to use the buttons on the MicroBlaze. If time is given a slideshow function, nice transitions, displaying of JPEG images and displaying EXIF information are possible improvements to the design.

Backup plan: If the SD card reader turns out to bee too great of a challenge we would like to use the (already) implemented VGA and button controllers for a game instead. If so the project plan well be revised with the appropriate details.

Components

During the planning of the project we have identified the following hardware components involved in the project. The Department of computer science at LTH has let us know that they can provide us with all of these hardware components.

- **FPGA** (MicroBlaze Spartan 3e)
- **Buttons** (built in on the MicroBlaze FPGA)
- **VGA display**
- **SD card reader** (and SD card)

I/O processes

The I/O processes we have identified are the following:

- **VGA display** - used for displaying the image in the image buffer
- **SD card reader** - used for retrieving the images from the SD card
- **Buttons** - used to get interaction from the user

The data exchange we have found would be for the following:

For the VGA display

- Resolution - Number of distinct pixels that will be displayed, in width and height
- Vertical and horizontal frequency - How often the pixels will be drawn horizontally and how often each line will be drawn vertically

For the buttons

There's not a lot of IO going to and from the buttons. It's just a matter of checking if a button is pressed. There is no protocol to speak of.

For the SD card reader

There exists three modes and therefore three protocols for the SD card reader. It looks like the SPI mode protocol is the mode we will use. That protocol is a simple command-response protocol where all commands are initiated by the master (that is, the FPGA in our case).

The data exchange for the SPI mode would be:

- Command frame sent by the FPGA
- Response frame sent by the SD card module
- Data sent by the SD card module

Then there is also the handling of the filesystem which probably will be FAT, this will likely be written in C rather than in VHDL and doesn't really have to do anything with data exchange since it will just call the respective functions to read and to write data. The data exchange will be handled separately from the filesystem.

Memory and CPU considerations

We currently have limited experience of the involved hardware and their need of memory buffers etc. The list below is a preliminary estimate of the memory the system will need.

- Graphic routines and button handling will need to exist on BRAM.
- File information exists on the SD card, from where images are loaded on demand. A BMP image sized 640x480 pixels with 8 bit color depth has the approximate size of 2.5 MB. We want to show at least 5 different images which will then require an SD card of at least 13 MB memory.
- On the FPGA we will need memory to put two (decoded) images i.e. 5 MB. There will be one image in a read buffer and one image in a display buffer. We will also need a maximum of 1 MB memory to put the C code in that will handle the reading of the filesystem, the reading and reacting of the buttons and the changing of the image buffer for the VGA display. Since we have 16 MB of SDRAM to utilize the memory requirements won't be a problem to fulfill.
- Since some parts of the system will most likely be written in VHDL we will also need a (tiny) memory to put that in. That won't affect the SDRAM though.

Two CPUs will probably be the most efficient way of building the system. One processor can be used to prepare the next image for being displayed, and the other can handle the displaying (hence input from the buttons and output to VGA display).

Improvements

Buttons

An improvement would be to use specific keys to move forward or backward amongst the images. Yet another improvement would be to start an automatic slideshow by pressing another specified button.

Transition
An improvement to the slideshow could be animated transitions.

SD card
Initially we would read fixed size BMP images from the SD card. Next step would be to use a FAT filesystem for easy computer exchange. Initially we would read a fixed number of images, an improvement would be to read all the images in the root directory of the fat filesystem.

Image display
Initially we would like to display a simple in-ram bitmap on screen and then try to load images from an SD card. Further improvements would be scaling, JPEG decoding, displaying EXIF information and a simple filesystem listing.

Timeline

1. VGA controller
    1. Connect VGA display to FPGA and access it from Xilinx. Learn about pins, protocol, sync, timing, etc.
    2. Draw one pixel on the VGA display
    3. Paint whole screen blue, red, green, etc.
    4. Take image buffer and print to VGA display
2. Button controller
    1. Learn how to use the buttons for controlling a VHDL/C program.
    2. Fix an "interrupt" solution for reading from the buttons
3. Write C program! "Do it all program"
    1. Let button actions be shown on the monitor (ie. light up the screen blue if you press akey)
    2. Read buttons and take the next image (in BRAM memory) and put in the image buffer when some button has been pressed.
4. SD module
    1. Connect SD card module to FPGA and access it from Xilinx. Learn about pins, protocol, etc.
    2. Read raw data from card
    3. Learn about fat filesystem. List files in the root directory
    4. Read a whole file from SD card
    5. Modify the C program so that images will be read from the SD card rather than from BRAM
5. Improvements (if time is given)
    1. Slideshow function
    2. Image transitions
    3. Display of JPEG images
    4. Display of EXIF information
    5. Simple file system listing
6. Project details
    1. Final report
    2. Presentation
    3. Demo

Estimated Time Plan

Week 1

Thomas, Martin, Samuel: Project proposal and background studies

Week 2

Thomas & Martin: 1.1, 1.2, 1.3, 1.4
Samuel: 2.1, 2.2

Week 3

Thomas & Martin: 1.1, 1.2, 1.3, 1.4
Samuel: 2.1, 2.2

Week 4

Thomas & Martin: 1.1, 1.2, 1.3, 1.4
Samuel: 3.1

Week 5

Samuel & Martin: 4.1, 4.2
Thomas: 3.2

Week 6

Samuel & Martin: 4.3, 4.4, 4.5
Thomas: 3.2

Week 7

Thomas, Martin, Samuel: 6.1, 6.2, 6.3