Niklas Blomqvist, Philip Johansson, Matteus Laurent, Johan Levinsson,
Oscar Petersson, Erik Peyronson

Group 41 - Project Report

Högskoleingenjörsutbildning i datateknik, 180 hp

Contents

1 Introduction

Har du inte läst README.txt (efter 23/10), gör det omgående!

This is the final report treating the final project in the course TSIU03 System design. The project was to design and implement a device used for audio signal processing.

Using a DE2 board an external audio source can be connected via a 3.5mm port, the signal is then processed by the application and an image is rendered on a vga screen displaying four signal level bars representing the left and right channels both before and after manipulation. There is also two bars displaying the current volume and balance setting.

The application also accepts input via a ps/2 keyboard used for setting volume

2 Achievements

2.1 User Noticable Achievements

- Remarkable astonishing graphics.

- Power level indicators for both the input and output sound signal.

- The indicators are moving at a smooth rate with absolutely no flickering.

- Displaying the current volume and balance adjustments made in a user friendly way.

- Both the minimum and maximum level of adjustments available to the volume and balance is visible to the user.

- A mute button is shown if the volume is set to be muted.

- Output sound is sent to a Class-D amplifier.

- Each bar has a peak level indicator.

- The peak level indicator falls slowly when the power bar is lower than the peak level indicator.

2.2 Design Challenges

- Use the SRAM to store information about a background picture.

- Implement a low-pass filter for our signal level indicators.

- Bar graph rendering (which pixels to write/blank).

- The logic of adjusting the volume and balance correctly.

- Detect each key press only once to prevent drastic changes.

3 System Level Description
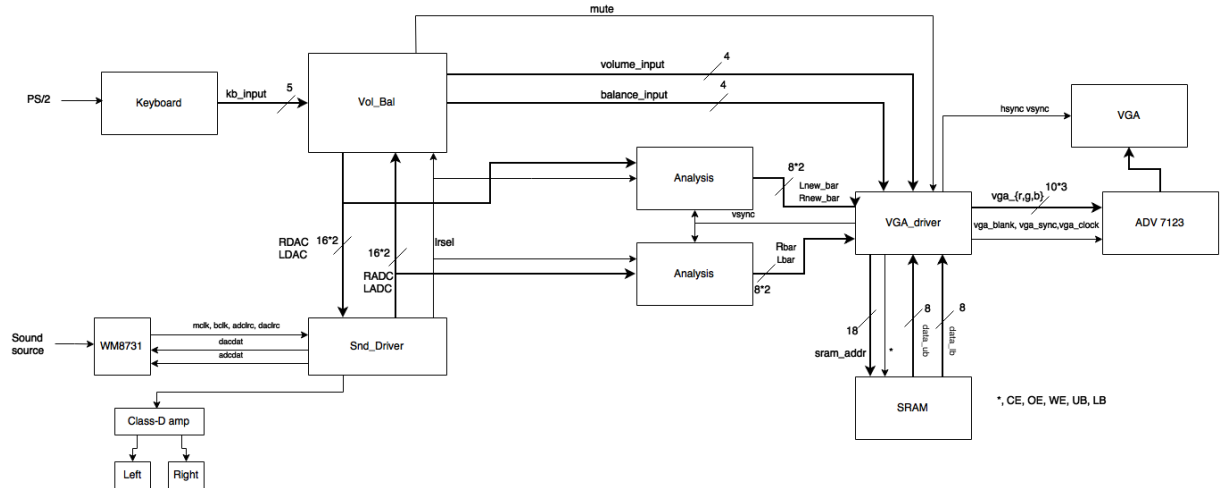
Figure 1: G41 Project System Overview

The project consists of five major modules – Keyboard, Vol_Bal, Snd_Driver, Analysis, and VGA_Driver. Presented here is an overview of the modules, and more detailed functionality is described in the *Design Specification [1]*.

The system takes, with help of Snd_Driver, digitally encoded sound from the WM8731 chip. The sound data is passed on to the first instance of Analysis which provides (in order to smoothen the bar movements) low-pass filtered control signals to VGA_Driver:bartender to assist the rendering of the pre-processing bar graphs done by VGA_Driver as a whole. The audio data from Snd_Driver is also passed on to Vol_Bal which in combination with the control signals from Keyboard adjusts the audio balance and volume and sends it back to Snd_Driver which sends signals for the class-D amplifier through the GPIO:s. The adjusted audio is also forked off to a second Analysis instance, which assists the post-processing bar graph rendering.

Whereas `Keyboard` and `Analysis` already are fairly small and monolithic modules, and `Snd_Driver` and `Vol_Bal` consists of three submodules each, `VGA_Driver` consists of no less than twelve submodules.

`Keyboard` handles the PS/2 keyboard user input. The module filters break scan codes ($F0_{16}$, XX_{16}) bytewise (11 bit/byte) and compares the XX_{16} byte (iff directly preceeded by $F0_{16}$) to the control key values. This approach will allow the system to respond to the regular keys (ARROW keys and END) as well as corresponding numeric keypad keys as initial $E0_{16}$ bytes are discarded. Upon a registered valid key release, a 5-bit control signal (`kb_input`)is sent for a single clock cycle to `Vol_Bal`.

`Analysis` reads the `ADC` signals and lowpass filters these with a saturation time of approximately 100 ms (4096 clock cycles). These signals are converted to a graph height (in pixels) and passed on to `bar_tender`.

`Snd_Driver` have the three submodules `Snd_Driver:{Ctrl,Channel_Mod}` if we count the two instances of `Channel_Mod`. These are verbatim copies of *Laboration 4*. The only difference is that the outputs are sent to GPIO pins (and from there via an i2c adapter to the class-D amp) as well as back to the WM8731 codec.

`Vol_Bal` consists of the three submodules `Vol_Bal:{Current_Vol_Bal,{Volume,Balance}_Adjustment}`. `Current_Vol_Bal` reads `kb_input` and updates registers representing system levels of volume, balance and mute. `Volume_Adjustment` adjusts volume according to the volume_level input from `Current_Vol_Bal`, decreasing amplitude by up to -30 dB in 3 dB decrements, and `Balance_Adjustment` adjusts the audio according to system balance level, resulting in a linear scaling the incoming amplitude. The stronger a channel bias, the higher the signal reduction of the opposing channel. The affected channel loses 1/8 of the amplitude per level of bias.

`VGA_Driver` with its submodules handles the rendering of the UI. The module is, in essence, a modified version of the module used in *Laboration 3*. The most significant difference is found in the two new modules `VGA_Driver:Bar_Tender` and `VGA_Driver:Bar_Mixer`. `Bar_Tender` uses the output from `Vol_Bal` and each of the `Analysis` modules along with `{h,v}cnt` to calculate which pixel is to be rendered and if it should be rendered from the background image or if to draw it black. The result of the calculations is the control signal `render_bar`. This signal is then passed to `Bar_Mixer`, which is basically a multiplexer, either forwarding the loaded background colors or a blacked out pixel depending on `render_bar` from `Bar_Tender`.

4 Justification of the Achievements

4 to 6 pages

In this subsection, explain how you have solved the challenges described in Subsection 2. It is important that you show how you got from the challenge to a specific solution, and do not miss steps. From the challenge, to the algorithm (mathematical explanation or model), maybe doing some tests in Matlab to take design decisions and validate that the algorithm works. Include the mathematical equations if the algorithm is not obvious. From the algorithm to the hardware implementation, making the numbers of sizes for memories, timing for the signals, design alternatives, decision that you took and why, etc. Include relevant pictures of the circuits or sub-modules of the system to support the explanations. IMPORTANT: You can structure this subsection as a description of the sub-modules of your system, but you have to be aware

Skriv i separata filer, inkludera i justification.tex

5   User Interface

1 page How to control the system + image visualized on the screen

The system is controlled by 5 keys on a PS/2-connected keyboard. Down in the table you can see which key that changes what in the system.

KEY	Function
U ARROW	Volume Increase
L ARROW	Balance Bias Left
D ARROW	Volume Decrease
R ARROW	Balance Bias Right
END	Mute Volume

Figure 2: PS/2 keys and how the input affects the system.

The volume level has eleven diffrent stages, exclusive mute. When the volume is at max, the whole bar will be red, and when the volume is decreased by one the bar will decrease to show the current volume.

The balance level has eighteen diffrent stages. Eight for left balance, eight for right balance and one stage when the signal is equal. When the balance is equal for left and right the only thing you will notice is the bar and divider. But when you for example press the left arrow, the left side will be filled and the outgoing volume on the right will of course decrease.

To indicate that mute is enable you can see one green speaker with a cross to the left side of the picture, and when mute is disable there will only be black. You will also be able to see that the after bars will not show anything, due that nothing is beeing sent out.

You will be able to see one peak level indicator for each bar, the peak level indicator falls slowly when the power bar is lower than the peak level indicator. With our peak level indicator our user expreience is taken to a whole new level, as you will notice when you use our system you will be overwhelming by the effect.

6   Improvements

1/2 page

In this subsection, discuss what could be improved in the system and which additional features could be added.

7   Evaluation of the Project Execution

1/2 page

Figure 3: User interface

In this subsection, discuss how the execution of the project was, if it was according to the original plan, how was the coordination between the members, the time spent of each member in his/her tasks (you have the table below for that), difficulties that you faced, what took more time than expected, why, what took less time than expected, what was easier/more difficult than expected.

Time spent for each member, and respective member's tasks:

Group Member	Main Tasks	Total Time
Niklas Blomqvist	Analysis code	~65
Philip Johansson	Analysis code	~70
Matteus Laurent	Vol/Bal code, project coordinating	~75
Johan Levinsson	Top TB code	~55
Oscar Petersson	Keyboard code, documentation	~70
Erik Peyronsson	VGA code modifications	~70
		405 h

The overarching theme of the project execution would have to be the difficulties faced when one or more members' participation was prevented by illness or similar obstacles, ensuring a less than ideal attendance. Our project group was set back early on because of this very reason, leading to us trailing the schedule by approximately a week. However, the effort invested to catch up was admirable and sufficient, allowing for the eventual completion of the project. Aside from this, the project went mostly according to plan, with a few notable exceptions.

Cooperation and coordination within the group was satisfactory. The project manager was given sufficient authority to lead the distribution and coordination of tasks. A certain level of trust combined with good design of the module overview allowed for more individual workloads, without sacrificing shared communication and system understanding.

All things considered, the design specification was the task that exceeded its set aside time the most. This was due to the document outlining a lot more module details than originally planned,

ultimately giving an indepth view of the system. The document was not only expanded upon on recommendation from our project supervisor, but also because of our own natural inquiry when discussing the design overview. Laying the groundwork this way gave voice to design uncertainties early on, and thus helped a lot with subsequent tasks.

Thanks to a thorough design specificiation, the foundation for our first prototype essentially wrote itself. The initial stages of coding were completed swiftly, making up for any time lost (and more) on writing the design document.

8 Personal Experiences

1/2 to 1 pages

In this subsection, include your experiences. What did you liked most in the project? What you have learned? Did you learn from some mistakes/failures? Did you have a good time? Is there something in the course that should be improved? IMPORTANT: The personal experiences are individual and each student in the group has to write a personal paragraph in this subsection. If you want to add more information, or the information is sensitive, please, talk to the Course Responsible.

8.1 Niklas

With a well written design the conversion from text to VHDL went super smooth and was therefore very enjoyable. One thing I will take as a lesson is the fact that you should consider which different ways of implementing the same problem there is, consider the pros and cons of each way and from there on pick the most suitable one. I definitely had a good time working on this project.

8.2 Philip

We spent a lot of time on design. We discussed a lot in the group and calculated on much, the whole design was very well thought out and I an had very good control throughout the system. This led to the implementation going very well, and there not being much trouble in assembling the system. This is something I will take with me in life.

8.3 Matteus

8.4 Johan

8.5 Oscar

Foremost, the project has significantly increased my comprehension of digital circuit design. It has also shown that (which I am sure all of us already knew) that preparation is indeed key to success. Considering my fetish of documentation, I can not truthfully say that the sparse form of documentation has been a favourite of mine, but considering the given time frame, the intention with the project, and the fact that the course runs in parallel with TDDI02, I have to give it a pass. Over all, it has been a greatly enjoyable course.

8.6 Erik

9 References to the Project Files

[1] Group 41 - Design Specification, *Linköping*, 2015-10-19.

[2] Group 41 - Requirement Specification, *Linköping*, 2015-09-22.

[3] Group 41 - First Presentation, *Linköping*, 2015-10-15.

[4] Group 41 - Project Plan, *Linköping* 2015-09-24.

All documents available at https://github.com/oscpe262/TSIU03.Project. The references in the PDF version of this document are hyperlinks to the document listed. The PDF can be downloaded at: https://github.com/oscpe262/TSIU03.Project/blob/master/Projektrapport/report.pdf.