

Design of an Electronic Billboard

Complementing R:eFlex' business plan

Delft 2010

Faculty of Electrical Engineering, Mathematics and Computer science

Mekelweg 5

2628 CD Delft

Joost Meerwijk: 1354701

Willem Zwetsloot: 1383612

Professor. R. Ishihara

Preface

This thesis is based upon the business plan of R:eFlex, which is part of the Bachelor Graduation Project of TU Delft. R:eFlex is a start-company that wants to introduce electronic billboards using E-paper, which is currently under development at DIMES; the 'Delft Institute of Microelectronics and Sub-micro-technology, affiliated to the TU Delft. In interviews conducted for the business plan with several companies expressed interest in this idea.

This report describes a design of an electronic billboard system that can be easily created using super E-paper.

Readers who are interested in the background of the electronic billboard using super E-paper technology are directed to chapters 3 and 4. Chapters 5 and 6 are mainly contain technical data. Here the design alternatives are considered and the final design is described. Readers who are interested in how good our design is and to what extend it is easily implementable using super E-paper technology are directed to chapter 7.

We would like to express our sincere gratitude to our supervisor Ryochi Ishihara, who has allowed us to be creative in our design. Further we would like to thank Michael Schumacher and Ton Slats, who have helped us every time we needed a new piece of equipment.

Summary

Two large outdoor advertising companies have expressed the desire to be able to allow their clients to display more interactive and animated advertisements. A natural way to do this is by means of electronic billboards. These billboards could display a slideshow of images or even animations. The advertisement could also be changeable remotely. Billboards that can do all this already exist, but they have serious drawbacks. A more suitable approach is to use super E-paper, which is currently being tested and further developed by DIMES at TU Delft.

The purpose of this report is to show a design of an electronic billboard that can be effectively realized using super E-paper.

Despite advanced research, a prototype of super E-paper is not available. Therefore it is not possible to test the design on the necessary scale with super E-paper. The best way to go forward is to prepare a design that can easily be converted into a real prototype, when the technology catches up. Testing can then be done using a surrogate. This poses some serious design constraints and limits the potential components.

Several decisions have to be made about which components to choose to build up the system. First, a receiver has to be picked to receive the data. Of course a type of storage has to be chosen to store the received images or the animation. The controller has to be chosen in such a way that it can be realized using technology of super E-paper without too much effort. The same applies to the choice of the screen. It has to be driven similar to the screen of super E-paper. For every decision, it has to be kept in mind that later it would have to be implemented using super E-paper technology.

After a couple of multicriteria analyses and a long time of development, the final design was ready.

The receiver that was chosen is a standard 2.4GHz device. Its maximum data rate and the allowed distance are acceptable, while its design is simple.

The controller was designed in VHDL, a hardware description language. The controller consists of three main components:

- *Receiver/ Decoder*. This receives the serial data and converts it to 16-bit packets to be stored in the SRAM.
- *DataStorage*. Takes the data from the *Receiver* and stores it in the memory, while simultaneously retrieving data from the memory to be forwarded to the *ScreenControl*.
- *ScreenControl*. Controls the signals to the screen, making sure a non-moving image is displayed.

SRAM was chosen as the storage technology. It combines speed and robustness with an easy realization using super E-paper technology.

A small SHARP LCD screen, similar to the screen used in a PlayStation Portable, was chosen as a display. Not only was it cheap and simple, driving it was similar to driving the screen of super E-paper.

Due to voltage differences between the surrogate controller and the screen and start-up timing issues, a special circuit had to be designed. It does three things:

- Solves the start-up timing issues. Certain signals have to be asserted in a certain order at initiation. This is solved by having the first signal trigger the second.
- Converts 3.3V logic to 2.5V. The testing controller output is 3.3V, while the screen requires 2.5V. About 40 connections were converted to 2.5V, using the same amount of voltage dividers.
- Amplifies the 5V to 28V to drive the screen's backlight

The entire receiving end of the system works. However, when connected to the transmitting end of the system, that was designed by another team, small problems appeared. Transmission errors, due to interference of wireless systems in the neighbourhood, caused the image to be displayed inaccurately. However, if the interference can be blocked, the entire system works excellent.

The most important issue is how hard it is to implement the design using super E-paper technology. Even though all the parts were chosen primarily on their capabilities to be implemented using super E-paper technology, some work will have to be done to actually achieve this.

The receiver has an analogue and a digital part. The digital part can be made in the same way as the controller will be, which is described later. The analogue part requires a little more work, but the technology behind super E-paper can also be used to realize analogue circuits.

The controller was designed in VHDL and tested to be working correctly on an FPGA, which is basically a reprogrammable piece of hardware. As said before, VHDL is a hardware description language, so the circuit that has been designed can be extracted from the code very simply. An FPGA consists of standard pieces, so-called logic cells. These logic cells can be created using parts from the standard library of parts that have already been made and tested using super E-paper technology. So any design that works on an FPGA, can also work using super E-paper technology.

SRAM is almost just as simple. SRAM cell consists of transistors, which is the basic building block of super E-paper technology, so creating SRAM should not pose a problem.

Content

Preface	i
Summary	ii
Content	iv
List of figures	vii
1 Introduction	1
2 Design Context	2
3 Market position	3
3.1 Interviews with advertising companies	3
3.2 Restrictions on extracting requirements	3
4 Schedule of requirements	4
4.1 Requirements regarding production	4
4.2 Requirements concerning distribution	4
4.3 Requirements for the usage of the product	4
4.3.1 Features of the product extracted from market research	4
4.3.2 Controlling the product	4
4.3.3 Maintenance	4
4.3.4 Ecological Requirements	4
4.4 Requirements for recycling	5
4.5 Requirements regarding the design of the first concept	5
5 Choosing components for concept	6
5.1 Evaluation of alternatives for memory	6
5.1.1 Dynamic Random Access Memory	6
5.1.2 Static Random Access Memory	7
5.1.3 FLASH	8
5.1.4 Multicriteria analysis of memory alternatives	9
5.2 Evaluation of alternatives for a main controller	11
5.2.1 Microcontrollers	11
5.2.2 FPGA	12
5.2. Multicriteria analysis of controller alternatives	13
5.3 Evaluation of alternatives for a screen	15
5.3.1 Video Graphic Array (VGA) monitor	15
	iv

5.3.2	<i>Digital TFT-LCD</i>	15
5.3.3	<i>Multicriteria analysis of screen alternatives</i>	16
6	Final concept for the billboard	18
6.1	<i>Design overview</i>	18
6.2	<i>External receiver</i>	19
6.3	<i>FPGA design</i>	20
6.3.1	<i>Clock Generator</i>	21
6.3.2	<i>Decoder</i>	22
6.3.3	<i>Data storage</i>	23
6.3.4	<i>Screen control</i>	27
6.4	<i>The Screen</i>	28
6.5	<i>Step-up and level converter</i>	29
6.5.1	<i>Step-up converter</i>	29
6.5.2	<i>Level Converter for SHARP LCD</i>	29
7	Testing concept billboard with and without wireless link	30
7.1	<i>Two test set-ups</i>	30
7.2	<i>Results of the first test setup</i>	31
7.3	<i>Results of the second test setup</i>	32
8	Product evaluation	33
8.1	<i>Overall hardware modification</i>	33
8.2	<i>Receiver</i>	34
8.3	<i>Control Hardware</i>	34
8.3.1	<i>Implementation of an FPGA using SG-TFTs</i>	34
8.3.2	<i>Implementation of SRAM using SG-TFTs</i>	36
8.4	<i>Modifications to the screen</i>	37
8.4.1	<i>Size difference between SHARP LCD and super E-paper</i>	37
8.4.2	<i>Resetting the QR-LPD</i>	38
8.4.3	<i>Continues and single shot data</i>	38
8.4.4	<i>Colour control by voltage levels of QR-LPD and LCD</i>	38
9	Conclusion	39
	Sources	41
	List of abbreviations	43
	Appendices	44

List of figures

FIGURE 2-1:	OVERVIEW OF THE PROJECT, DIVIDED INTO DIFFERENT COMPONENTS	2
FIGURE 3-1:	LOGO CBSOUTDOOR	3
FIGURE 3-2:	LOGO JCDECAUX	3
FIGURE 4-1:	CONCEPT PICTURE OF THE BILLBOARD	5
FIGURE 5-1:	AN ARRAY OF DRAM CELLS	6
FIGURE 5-2:	A SINGLE CELL OF SRAM MEMORY CONSTRUCTED WITH 6 TRANSISTORS [3]	7
FIGURE 5-3:	EXAMPLE OF AN SD-CARD. [6]	8
FIGURE 5-4:	A FLASH MEMORY CELL [7]	8
FIGURE 5-5:	A LOGIC-CELL INSIDE AN FPGA [8]	12
FIGURE 6-1:	DESIGN OVERVIEW	18
FIGURE 6-2:	2,4GHZ TRANSCEIVER BASED ON NRF2401A+	19
FIGURE 6-3:	FPGA DESIGN OVERVIEW	20
FIGURE 6-4:	XILINX CLOCKING WIZARD, TO CONFIGURE DCM.	21
FIGURE 6-5:	A TRANSMISSION OF A BYTE VIA RS232	22
FIGURE 6-6:	FLOWCHART OF DECODER	22
FIGURE 6-7:	AN IS61LV25616AL [14]	23
FIGURE 6-8:	OVERVIEW OF A TRI-STATE BUFFER	23
FIGURE 6-9:	FLOWCHART OF SRAM_CTRL.	24
FIGURE 6-10:	THE FLOW DIAGRAM OF SRAM_TOPLAYER	25
FIGURE 6-11:	FLOWCHART OF PIXELBUFFER	26
FIGURE 6-12:	TIMING SIGNALS FOR SCREEN	27
FIGURE 6-13:	SHARP TFT-LCD SCREEN [16]	28
FIGURE 6-14:	STRUCTURE OF DIGITAL DATA DRIVER [17]	28
FIGURE 6-15:	STRUCTURE OF DECODER-BASED 3-BIT DAC [17]	28
FIGURE 6-16:	MC34063A STEP-UP CONVERTER CONFIGURATION [18]	29
FIGURE 6-17:	3,3 TO 2,5V LEVEL CONVERTER	29
FIGURE 7-1:	TEST SETUP INCLUDING ALL PARTS OF THE TOTAL DESIGN	30
FIGURE 7-2:	THE RESULT OF THE FIRST TEST	31
FIGURE 7-3:	NOT YET SUCCESSFUL RESULT USING THE SECOND SETUP	32
FIGURE 8-1:	SIMPLIFIED BLOCK DIAGRAM OF THE CURRENT DESIGN	33
FIGURE 8-2:	BLOCK DIAGRAM OF THE DESIGN BUILD ON SUPER E-PAPER, PARTLY FROM [19]	33
FIGURE 8-3,	A LOGIC CELL [8]	34
FIGURE 8-4:	THE SG STANDARD CELL LIBRARY [18].	34
FIGURE 8-5:	SEPARATE COMPONENTS OF A MULTIPLEXER [22]	35
FIGURE 8-6:	A LOOKUP TABLE	35
FIGURE 8-7:	3-INPUT AND USING SG STANDARD LIBRARY COMPONENTS	35
FIGURE 8-8:	CONSTRUCTION OF AN SRAM CELL [25]	36
FIGURE 8-9:	PIXEL MATRIX OF AN TFT-LCD	37
FIGURE 8-10:	SINGLE TFT IN A PIXEL [27]	37
FIGURE 8-11:	PIXEL MATRIX OF AN QR-LPD [19]	37
FIGURE 8-12:	SINGLE PIXEL OF QR-LPD WITH SG-TFT [19]	37
FIGURE 8-13:	HORIZONTAL TIMING WITH RESET FOR THE QR-LPD	38
FIGURE 8-14:	VOLTAGES APPLIED FOR BLACK AND WHITE [18]	38
FIGURE 8-15:	HYSTERESIS CURVE OF GRAYSCALE SETTINGS [18]	38

1 Introduction

Outdoor advertising is mainly characterised by the use of paper. Every bus stop and billboard has a paper advertisement inside. The reason for this is simple; paper is easy to use and extremely cheap. However, it has some drawbacks as well. Before an image is actually on the street, it first has to be printed, pasted and placed. The journey from design to placement is quite long. Another major disadvantage is the considerable amount of waste generated by the use of paper. Finally, many people just ignore a paper advertisement. Their attention is drawn to the vibrant environment around them and not to a still image.

So not surprisingly, the advertising industry has been searching for years to find an alternative for this kind of advertising. Usually the only alternative that is found is using an LCD screen with a controller, a so-called digital screen. These can show animated images and can be updated instantly and remotely. This reduces the journey from design to placement to only a couple of minutes.

However, digital screens have not been deployed in great numbers. This is because no satisfying products exist thus far. This is due to several problems with the modernisation of outdoor advertising. First, it is still too expensive; it is practically impossible to compete with paper on a financial level. Also, since the objects are outside all the time, they are vulnerable to vandalism. Paper posters are easily replaced, but its digital counterpart is quite a lot more expensive when it gets destroyed and needs to be replaced. Lastly, an electronic billboard requires a lot of energy, which is both a waste of money and harmful to the environment.

A possible alternative is super E-paper. Super E-paper has many of the advantages of conventional paper and digital billboards, without many drawbacks. For instance, once an image has been loaded onto super E-paper, it doesn't need to be refreshed anymore. This decreases its energy consumption significantly. Also, just like a digital screen, it can be updated instantly and remotely. Besides this, super E-paper is relatively cheap and has paper-like visibility and flexibility.

Super E-paper is currently under development in DIMES, the 'Delft Institute for Microsystems and Nanoelectronics', which is affiliated to the TU Delft. It consists of a front and a back plane. The front plane is where the image is displayed. It is a 'Quick Response Liquid Powder Display' (QR-LPD), which is manufactured by Bridgestone and is fully flexible. The back plane is created using 'Single grain thin film transistors' (SG-TFT). In principle, every logic cell can be made using SG-TFTs, while still maintaining its flexibility. Therefore, the back plane is going to be the circuit that drives the screen and manages the receiving and possible storage of multiple images.

The purpose of this report is to show a design of an electronic billboard can be effectively realized using super E-paper.

Several decisions will have to be made. First, to allow remote changing of the image, a receiver has to be integrated. To be able to show a slideshow of images or even animated advertisements, there has to be some kind of data storage. This all has to be controlled by a controller. The controller has to make sure everything works together and drives the screen.

The idea is to do all this using super E-paper. However, despite the advanced research, no prototype of super E-paper exists currently, so testing on actual super E-paper is not yet possible. Therefore, we have to make sure our design can be implemented effortlessly using SG-TFTs, once the technology is ready. This design constraint has to be kept in mind for every design choice. Several alternatives for different components are considered and eventually one full design will be described.

The structure of this report is as follows. First, in chapter 2, the assignment will be outlined after which the current market situation is sketched in chapter 3. From those two chapters a schedule of requirements is distilled in chapter 4. In chapter 5 we compare different solutions as to how we believe these requirements would best be met. In chapter 6, the eventual model will be described. Finally in chapter 7, the changes required to make the design work with SG-TFTs are discussed.

2 Design Context

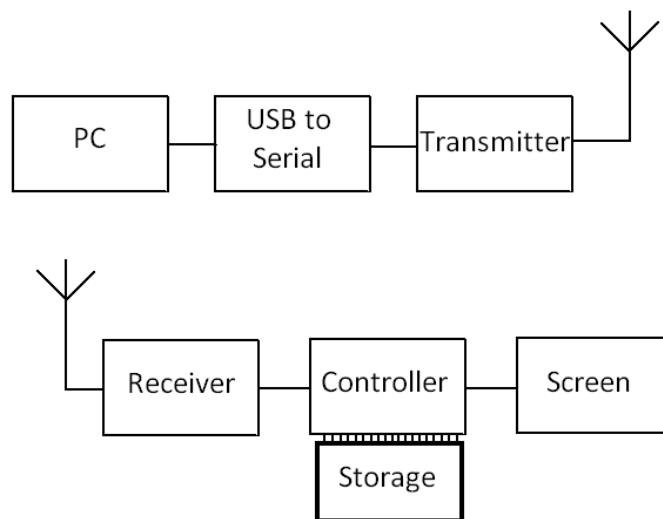
Our goal is to design an electronic billboard that can show images, perhaps even animated ones, and is able to receive new ones remotely. This design constructed using super E-paper has to have several features. First, new images should be able to be received. So of course, a receiver is necessary. In order to be able to receive data remotely, some sort of wireless system has to be set up.

To process the data accordingly, a controller has to be configured. This controller also has to show the image on a screen. A full-blown desktop computer won't work, since it will be impossible to recreate this using SG-TFTs. Some sort of micro processing unit is required to do this task.

In the case of a slideshow of images or even an animation, data has to be stored in memory. The choice of what type of storage to use is an important one as well. It has to be fast enough to both store new data while continuing retrieve data to send to the screen. Of course, best case scenario is if it is implementable using SG-TFTs.

Then there remains the choice of which screen to use. There is not screen that behaves exactly the way the front plane of super E-paper does, so an alternative to show the rest of the design works has to be found. If this alternative is driven in the same way as the screen of super E-paper, it's an added bonus.

Our thesis is part of a greater project. Another team is preparing the software for sending an image and the hardware for the transmission. Also error control is included in their thesis. Since we are on the receiving end, decent agreements on how the data transfer would look like have been made. There is some overlap between our parts on the receiver. The other team has prepared it, but in the final design, it would have to be integrated in the electronic billboard. The total project is illustrated in Figure 2-1: Overview of the project, divided into different components



Therefore, we will show how the receiver can be integrated using SG-TFTs, but not much about how it actually works and why it was chosen. Interested readers are directed to [1].

When everything has been designed and tested to be working correctly, it has to be discussed how it can be implemented using SG-TFTs. Some parts will be easily made with SG-TFTs, while for other parts a new production process may be necessary.

3 Market position

Our product is designed based on certain wishes and needs from the industry. Several interviews with potential customers have given us a clear view of what these are. In this chapter two of these interviews as found in Appendix 3 and 4 of the business plan of R:eFlex are analyzed and an external market analysis is included. From the wishes and needs, various requirements for the final design can be distilled.

3.1 Interviews with advertising companies

The two large outdoor advertising companies that could be reached in the Netherlands were interviewed, aside with some smaller ones. All of these were eager to guide us to create a product that suited their needs most. The interviews with the two large companies, CBS Outdoor (Figure 3-1) and JCDecaux (Figure 3-2), will be discussed in the next section. The interviews can be found in the business plan of R:eflex as 'Interview with CBS Outdoor' and 'Interview with JCDecaux'.

CBS outdoor was very clear about what they believed was the greatest advantage of super E-paper billboards; the ability to send new advertisements using wireless transmission. However, they were concerned about the investments involved. A rough estimate of the break-even point was three years, so to make it interesting, a lifespan of at least four years is required. At the time of the interview, the possible lifespan was not known yet, but we were assured by Prof. Ishihara, our tutor for the project, that it would be at least five years, after a few years of development. So during the interview with JCDecaux, which was about two weeks later, these concerns could be dealt with.

JCDecaux was primarily interest in the animation capabilities of our electronic billboard. As the technical director, Ivo de Graaf, pointed out, it would not be cost-effective to replace all the existing billboards with static images. It is still impossible to compete with paper in that area. Therefore, we should target A-locations where animated and interactive advertising is a large advantage.

Both companies were realistic about the dimensions of the billboards. Current designs are 2m^2 and it wouldn't be worth the investment to change these. So the minimum size of the electronic billboard has to be 2m^2 . JCDecaux was also optimistic about replacing the large billboards, which are 8m^2 .



Figure 3-1: Logo CBSoutdoor



Figure 3-2: Logo JCDecaux

3.2 Restrictions on extracting requirements

Obviously, the industry had a specific idea of what the final solution should look like. However, besides their ideas, the law placed some restrictions on our design. For instance, not every material can be used, no moving images near the highway and certain environmental issues. So extracting requirements is more complicated than it may appear at first sight. This should be kept in mind while reading the next chapter.

4 Schedule of requirements

Following from a combination of the assignment delivered by our tutor and the results extracted from R:eFlex' business plan, described in chapter 2.2, a schedule of requirements is made to clarify what the product should be. The schedule has been organized, according to the product's lifecycle. This means that for each state of its life (e.g. Production, Distribution, Usage and Recycling) the requirements are ordered.

4.1 Requirements regarding production

- [1.1] The billboard will be made with de μ -Czochralski process and a QR-LPD screen.
- [1.2] No external components ought to be attached, all electronics are included on super E-paper.
- [1.2] Billboard should be fully flexible.
- [1.3] Production of Super E-paper should meet international ecological requirements.

4.2 Requirements concerning distribution

- [2.1] Distributing the posters should be doable by a single person.
- [2.2] Product should be packable in a roll.
- [2.3] Installation should require nothing more than hanging the billboard on its place.

4.3 Requirements for the usage of the product

4.3.1 Features of the product extracted from market research

- [3.1.1] The product should be able to show images, if multiple in a slide show.
- [3.1.2] Next to images, also short animations should be displayed.
- [3.1.3] Display format should meet standard 'abri' format (120 by 160 cm).
- [3.1.4] Whole screen should be full-colour.
- [3.1.5] Product should be sold for approximately 2500 euro.

4.3.2 Controlling the product

- [3.2.1] Display information is received through an RF link.
- [3.2.2] During normal use, the billboard works autonomic. It only acts on the RF link.
- [3.2.3] Controlling the billboard shouldn't require technical information of the product.

4.3.3 Maintenance

- [3.3.1] The billboard should be water proof.
- [3.3.2] Removing the billboard should be easy, as a new one has to be replaced quickly.
- [3.3.3] If no displayable content is yet available, a 'default' image should be displayed.

4.3.4 Ecological Requirements

- [3.4.1] Standby current should be minimal.
- [3.4.2] Product should meet international requirements on EM-radiation.
- [3.4.3] Product should meet international requirement on hazardous materials.

4.4 Requirements for recycling

- [4.1] Every screen should be replaceable for a new one, or a be used as a spare part for another billboard.
- [4.2] Removing the E-paper screens should be easy to do without damaging the screen itself.
- [4.3] Product should meet standards for recycling. So the product is treated in such way that is minimizes damage to the environment.

4.5 Requirements regarding the design of the first concept

Because Super E-paper is not yet implementable for our design, therefore a leading product of the final design will be made. Taking in consideration what adjustments will eventually have to be made, to live up to schedule described above. Next to this, there are some other issues we have to keep in mind during the design of this leading product, these are all mentioned below.

- [5.1] The design has been spitted into two parts, our responsibility is to design everything from the receiver up to the screen.
- [5.2] Signal between receiver and decoding hardware meets RS232 standard.
- [5.3] All parts needed for the concept should not exceed 150 euro's.
- [5.4] Design should be finished within 7 weeks.

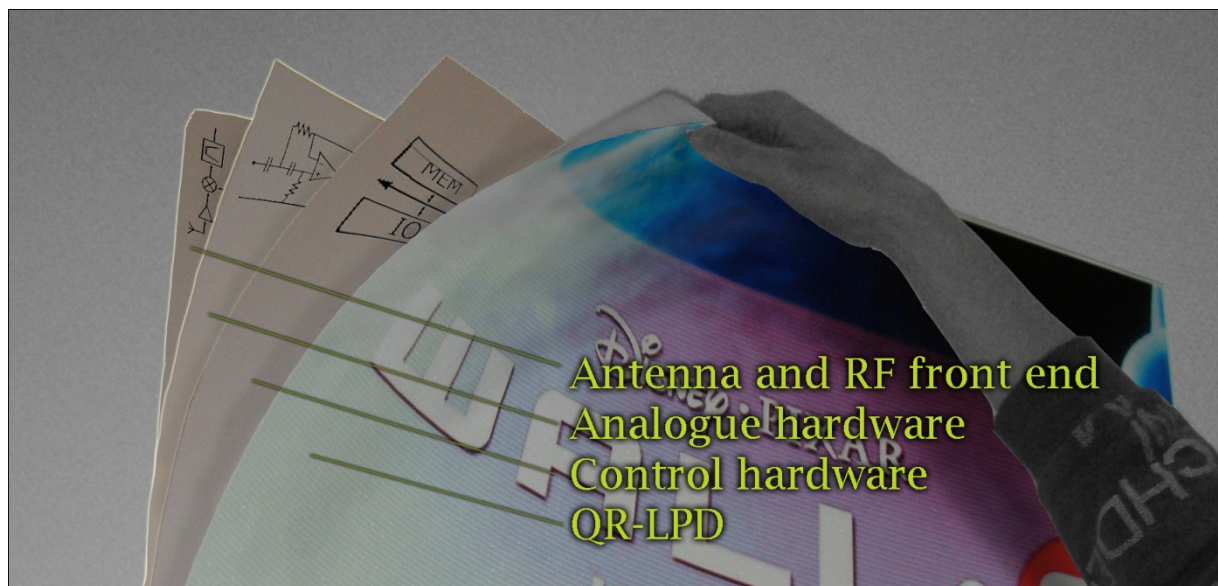


Figure 4-1: Concept picture of the billboard constructed with super E-paper

5 Choosing components for concept

There are several ways in which the final design can be implemented. For example, there are numerous alternatives for storing data. Also the choice of which controller to use isn't yet definite. Finally, a substitute for the front plane, which is essentially the screen of super E-paper, has to be picked.

5.1 Evaluation of alternatives for memory

Three alternatives for storing data have been chosen to be assessed. Dynamic Random Access Memory (DRAM), Static Random Access Memory (SRAM) and Flash are the most common and easily available types of storage. Therefore, these are discussed in the next section.

5.1.1 Dynamic Random Access Memory

Dynamic Random Access Memory (DRAM) is probably the most widespread kind of memory available. Nowadays, it is used in some form in basically all personal computers. 'Dynamic' represents the fact that the data has to be refreshed continuously. Otherwise the data would simply 'leak away'.

Access Time

Because DRAM needs to refresh its data all the time, an external controller is required. Also, due to its design, the access time of DRAM itself is relatively high as well.

Cost

DRAM is easy to manufacture and its design allows for high density. A single bit of storage is constructed using only two components, see Figure 5-1 [2]. Therefore, it is quite cheap, which is why it is implemented on such a great scale.

Interface

A single chip could be available in standard DIP packaging, which can easily be connected to the central controller. However, as mentioned before, these connections would lead to increased access time. Also, the central controller needs a bunch of outputs to drive the DRAM.

Yet another controller would be needed to continue to refresh the memory. So unless this is already included in the package, another chip would have to be programmed.

Technical Details

As illustrated in Figure 5-1, one bit of storage of DRAM is constructed using only one transistor and one capacitor. When the capacitor is charged, it represents a '1', otherwise it represents a '0'. The capacitor loses its charge over time. To keep the data, it has to be recharged a couple times each second. Therefore, once no more power is supplied, the data on the DRAM is lost.

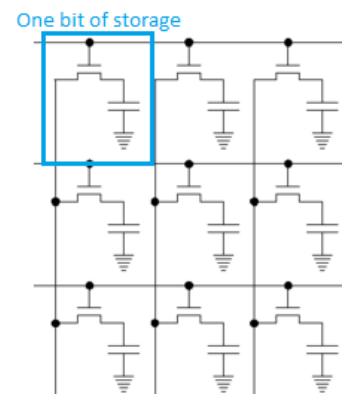


Figure 5-1: An array of DRAM cells. One bit of storage is constructed using a transistor and a capacitor. Voltage level on capacitor indicates '1' or '0'. Image based on [2]

5.1.2 Static Random Access Memory

Static Random Access Memory (SRAM) is often used in systems where speed is of the essence. The 'Static' is explained by the fact that it retains its data without a refresh, as long as power is supplied. It is most commonly used to temporarily store data in modern computers, known as cache.

Access Time

Since SRAM doesn't need to constantly refresh the memory like DRAM has to, it can access data very fast. Typically, SRAM can operate at processor speeds of 250 MHz and faster, while accessing times are still below one clock period.

Cost

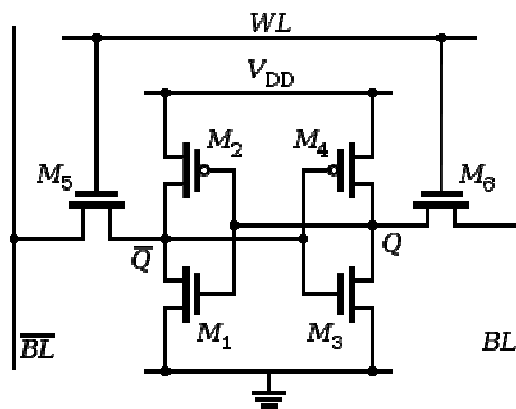
SRAM is constructed using six transistors. Because of that, the storage capability of standard SRAM chips is severely limited due to size. So to acquire more storage capability, more surface area and therefore more money is required for a reasonable capacity.

Interface

The SRAM chip that is under consideration is already integrated on an FPGA board we were also considering. Interfacing simply requires correct routing, which can be done using Xilinx. The normal number of wires needed to access SRAM depends on the storage capability.

Technical Details

SRAM is usually constructed using six transistors. Four of those form two cross-coupled inverters that actually hold the data. The other two are used to control access to the memory cell in case of a read or a write operation, see Figure 5-2 [3].



5-2: A single cell of SRAM memory constructed with 6 transistors [3]

5.1.3 FLASH

Flash is most likely the type of memory most used in consumer electronics. It is omnipresent in digital cameras, USB storage devices and mobile phones. It's called Flash because the entire sections of the memory can be 'flushed'; meaning it can be erased all at once in a split second [4]. The type of Flash memory under consideration is in the form of an SD-card, which is available in virtually every electronics store.

Access Time

When accessing an SD-card directly, the access times are pretty fast. Clock speeds up to 25MHz are theoretically possible, which translates to access times of 40ns[5]. This does, however, require strict timing. Therefore, access times of 60ns are more realistic.

Cost

SD-cards are not very expensive. The cheapest are for sale for about €5 for a 2GB card, while the faster ones that would be required for our purpose are about €30.

Interface

SD-cards usually have access strips at the end of the card as illustrated in Figure 5-3 [6]. These can simply be accessed by soldering a wire to it and connecting it to the central controller. However, these wires would have some propagation delay, which would add to the access times. Therefore, the connections should be as short as possible.

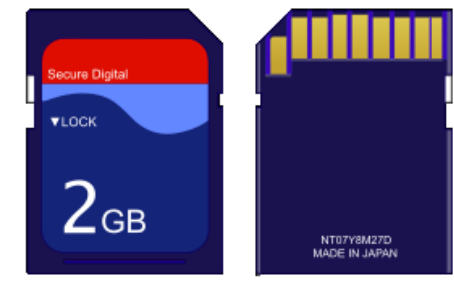


Figure 5-3: Example of an SD-card. [6]

Technical Details

One bit of storage of a Flash memory device is constructed using two transistors divided by a thin oxide layer. If a charge is applied to the floating gate, it starts to act as an electron gun. This traps electrons in the oxide layer. These electrons act as a barrier between the control gate and the floating gate. This can be measured by a device called a cell sensor, which measures the amount of flow of charge. Above 50% is a '1', while a '0' is depicted by a flow a charge of less than 50% Figure 5-4 [7].

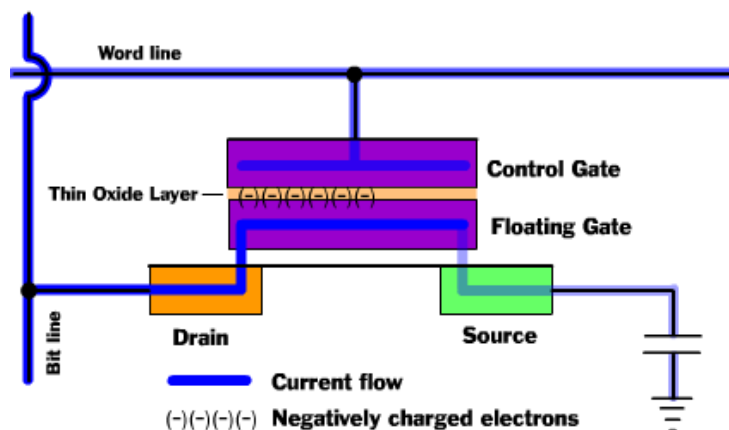


Figure 5-4: A FLASH memory cell [7]

5.1.4 Multicriteria analysis of memory alternatives

To make a choice between the three alternatives for storing data, four criteria have been used. In the next section, these are described and awarded a weight, based on the significance to the final product.

Convertibility: 4

The first criterion is how it would be implemented in the final product. If the alternative can be constructed using SG-TFTs, this is of course a huge advantage. Since the actual purpose of this project is to create something that is effortlessly converted to an actual electronic billboard, convertibility is an important factor in the final decision. Therefore the weight is 4.

Simplicity: 2

Due to limited available time, the chosen alternative has to work without too much hassle, since there is not enough time to figure everything out. More work can be done if the storage device works without a lot of configuring. Because this is an important issue, its weight is 2.

Speed: 1

If the chosen alternative isn't fast enough, it doesn't do any good. The faster it works, the faster data can be accessed and the easier it becomes to build a controller around it, since the timing constraints won't be as strict. However, above a certain threshold, it doesn't matter how fast it is, so its weight is limited to 1.

Storage Capability: 1

Of course, the chosen alternative has to have enough storage capability to store a number of pictures. However, once again, above a certain threshold, adding more adds no more value. Using the same reasoning as above, the weight is only 1.

Evaluation

The multicriteria analysis results are listed in Table 5-1. Each criterion is discussed separately and the best alternative is chosen every time.

Criterion	Weight	SRAM		DRAM		Flash	
		Score	Weighted	Score	Weighted	Score	Weighted
Convertibility	4	10	40	9	36	4	16
Simplicity	2	9	18	7	14	9	18
Speed	1	8	8	7	7	8	8
Storage Capability	1	4	4	7	7	9	9
Total			70		64		51

Table 5-1: Evaluation of storage alternatives

Convertibility: SRAM

SRAM is constructed using just six transistors. Experiments have already been done and within a year, actual SRAM can be made using SG-TFTs. Because of its simplicity, SRAM gets awarded a 10. DRAM consists of one transistor, but also a capacitor. Even though this is still very easy to create, it isn't as easy as SRAM's six transistors. Flash is created in an entirely different way, for which new production methods have to be thought up for. Therefore, it gets awarded a low score.

Simplicity: SRAM/Flash

This time, it's a tie between SRAM and Flash. The controller only needs to set certain signals for a read or a write, supply the correct address and the right data is accessed or stored respectively. DRAM needs an additional controller to ensure no data leaks from the capacitors, which adds to the complexity of the required storage controller. Flash integrated in an SD-card usually already has some kind of controller integrated [5], so it isn't very difficult to access or store the data, just supply the correct signals to the SD-card and you're done.

Speed: SRAM/Flash

Once again, SRAM and Flash prove to be equal. While SRAM is theoretically faster, to practically build a controller around it usually limits the maximum processor speed to 50MHz. Since Flash isn't far behind, it gets awarded the same score. DRAM comes in last once again, it's just slower than both of its competitors.

Storage Capability: Flash

Flash has by far the largest storage capability for on a small chip. The more expensive ones can hold 32 GB of data on an ordinary SD-card or USB device. DRAM takes a decent second place, while a large SRAM is just very expensive.

Conclusion: SRAM

Even though Flash took first place quite often, it isn't a very good alternative. The sole reason is the difficult creation of Flash using SG-TFTs. That exact criterion is the reason why SRAM took the lead in the overall score. Even though DRAM is easily convertible and can store more data for a lower price, in every other respect it is inferior to SRAM.

5.2 Evaluation of alternatives for a main controller

There are two alternatives for the main controller. All the different types of microcontrollers are discussed together, because the difference between microcontrollers is usually a matter of taste. In principle, all types of microcontrollers can perform all functions. The competitor is a Field Programmable Gate Array, a reprogrammable piece of hardware.

5.2.1 Microcontrollers

Microcontrollers are available in a wide range of sorts and flavours. And should therefore be evaluated properly to be able to choose the right one. Their main purpose, as to functionality, is to execute a certain list of instructions specified by the designer. And can therefore offer solutions to many problems.

Programming language

The language in which a controller can be programmed depends on the sort of controller. Although there are a couple of standard well-known languages, these do require a compiler converting code to machine language; which are specific for the architecture of a controller. Some of these well known languages include the basic and C structure.

Suppliers

There are many companies, all offering their own range of microcontrollers. A few common architectures are: ARM, Atmel AVR and Microchip's PIC.

Implementation

Because microcontrollers can be designed as a single system on a chip, implementing them is pretty easy. Apart from a supply and sometimes an oscillator, they often do not require other external hardware to function on a basic level.

Peripherals

As mentioned in the section before, microcontrollers can be designed as a single system on a chip. This is because of the peripherals built into the microcontroller, depending on the specifications of the chip; they offer peripherals like serial data buses, timers, analogue to digital converters and many others. And as they are often designed integrated on a development board, others functionalities are added. Because of these aspects microcontroller development board are easy to implement and design systems with

Programming software

Low level programming is dependent on the instruction set for the specific controller. But almost all controllers are programmable with a basic or C compiler. In many cases, using a compiler with full compatibility and a good programmer can cost something.

5.2.2 FPGA

A Field Programmable Gate Array (FPGA), consists of hundreds of thousands of logic cells, see Figure 5-5 [8], that can be combined to form a circuit. By rearranging the connections between individual logic cells, a new circuit is created. The fact that this rearranging of connections can be done indefinitely is utilized by many hardware engineers to test their designs.

Programming language

Usually, a Hardware Description Language (HDL) is used to create the models. The two HDLs accepted by the IEEE are VHDL and Verilog. Both languages are capable of designing all hardware, so a choice between either of these two is up to personal taste. However, VHDL is used more often than Verilog, because of a couple of features, such as code reuse and easy managing of large designs [9].

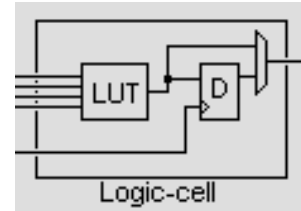


Figure 5-5: A logic-cell inside an FPGA [8]

Suppliers

Two large companies dominate the FPGA market; Xilinx and Altera.

Xilinx invented the FPGA and has traditionally been the silicon technology leader. Its general philosophy is 'to provide all the features possible, at the cost of extra complexity.' Altera dominates the rest of the market with a philosophy 'to provide features most people want, while keeping the devices easy to use'. [10]

Implementation

A FPGA alone is not very useful. Usually, they are incorporated on a development board. These boards usually have, besides an FPGA, a number of peripherals integrated to allow for better access of the FPGA. A development board available in large quantities at the TUDelft is the Digilent S3 board, which has a Xilinx Spartan-3 FPGA integrated.

Peripherals

These boards come with a number of useful peripherals. First of all, outputting a signal can be done using one of 120 I/O pins present on the board. These can also be used to connect other peripherals that are not present on the board. Secondly, there are eight LEDs and four seven-segment displays available. Another feature is the integrated SRAM module, which is described in more detail in chapter 5.3. [11] There are still other peripherals, but these are not applicable for this project, so do not require further explanation.

Programming software

To efficiently use board like this, software from Xilinx is very useful. The Xilinx IDE helps managing projects, has a Core Generator that can automatically generate components, can check the syntax of the code and, finally, programs the entire thing onto FPGA. However, if desired, other programs can be used for this purpose as well, such as Synplify.

5.2. Multicriteria analysis of controller alternatives

To decide between which type of controller to use, a microcontroller or an FPGA, five criteria have been used. In the next section, each of these will be described and then the weight is awarded, based on the significance to the final product.

Convertibility: 4

Once again, the first criterion is the effort required to convert the design to a model that is easily implemented using SG-TFTs. This is the actual purpose of this project and therefore its weight is quite high; 4.

Integrated Development Environment: 2

The second criterion is how user friendly the integrated development environment (IDE) with which the software can be programmed on to the selected controller is. This is important because of the limited time available for this project. Work can be done more efficiently if an IDE is easier to use, therefore its weight is 2.

Plug 'n Play: 1

Another criterion that originates from having limited time is the Plug 'n Play capability. When a lot of peripherals are necessary, it can be a fuss connecting everything correctly, which is also prone to error. However, it is not something that can't be overcome, so its weight is 1.

Availability/Price: 1

Of course, availability is an issue as well. If there is a gap of six weeks between an order and delivery, there is not enough time for testing. Also price can be an issue here, since our budget is limited to €150. Availability and price are in this case related, so can be assessed together. The weight is 1.

Outputs: 0.5

Lastly, the controller we choose to use has to have enough output connections. There are already 24 pins needed for the colours alone, so driving the screen requires about 40 output pins. SRAM needs about 35 data lines. However, this problem could be solved using multiplexers, so its weight is only 0.5.

Evaluation

The multicriteria analysis results are listed in Table 5-2. Each criterion is discussed separately and the best alternative is chosen every time.

Criterion	Weight	FPGA		Microcontroller	
		Score	Weighted	Score	Weighted
Convertibility	4	9	36	6	24
IDE	2	9	18	8	16
Plug 'n Play	1	8	8	8	8
Availability/Price	1	4	4	10	10
Outputs	0.5	8	4	6	3
Total			70		61

Table 5-2: Evaluating of controller alternatives

Convertibility: FPGA

The most important criterion was the convertibility from design to implementation using SG-TFTs. The programming language for an FPGA is VHDL. This is a hardware description language. Many programs exist that can convert VHDL to a transistor-level design, so it is very easy to convert a design for an FPGA to a final design. In case of a microcontroller it is possible to emulate it on the back plane of a super E-paper using SG-TFTs. Then the design has to be loaded onto the emulated microcontroller. This is a lot more tedious than implementing the design using VHDL, though.

IDE: FPGA

A couple of different development environments exist for microcontrollers. For PICs, the best known is probably MPLAB. This large IDE can compile both C and assembly and has a large number of options. Other microcontroller families have decent IDEs as well. However, these often cost a lot of money. The IDE for the FPGA, Xilinx, can be downloaded at no cost by students affiliated with the TU Delft. Xilinx comes with programs to increase efficiency and is very user friendly. Therefore, it is slightly better than the IDEs available for microcontrollers.

Plug 'n Play: FPGA/Microcontroller

The Plug 'n Play capability is where the FPGA board also has a significant advantage over a microcontroller. A development board like the S3 board from Digilent has everything onboard that is necessary for testing. Even SRAM is integrated and easily accessible. However, microcontroller development boards can also come with a number of different options. To achieve a clock speed high enough, a separate crystal can be supplied. For testing, additional LEDs have already been connected. Only SRAM would have to be connected using outputs of the microcontroller, since SRAM on a development board is not very common. Both an FPGA and microcontrollers are easy to start with, so it's a tie.

Availability/Price: Microcontrollers

The availability and price is where microcontrollers are superior. The FPGA board is at least €100, without shipping and added taxes. It has to be ordered in the United States, from where it will take at least two weeks before it arrives. Microcontroller development boards are in large supply in many Dutch electronics stores and can be delivered within a day and a decent one costs about €40. Therefore, microcontrollers have an easy victory for this criterion.

Outputs: FPGA

To connect the screen, a lot of outputs are necessary. Microcontrollers need even more, because the SRAM is not wired internally as is the case on the FPGA board. Altogether, a microcontroller with at least 75 outputs is required, so unless a multiplexer is used to increase the number of outputs, the choice of which microcontroller to use is limited to only a couple of alternatives. The FPGA board has 120 outputs and the SRAM is already connected internally. It is clear that an FPGA board is the best choice when considering the output capabilities.

Conclusion: FPGA

All things considered, for this project, an FPGA is superior to a microcontroller. Only its availability is an issue, but since a couple of these boards are lying around TU Delft anyway, this is not a significant problem. Its price is a sacrifice that is justified by the superior capabilities.

5.3 Evaluation of alternatives for a screen

There are two real options to use as a screen; a VGA monitor or a 'loose' LCD screen. Although there are many different options in each sector, they will be discussed in general and evaluated based on this information.

5.3.1 Video Graphic Array (VGA) monitor

VGA monitors will be evaluated in this section, many different types exist. Aspects that include most sorts will be evaluated.

Control

Controlling these sorts of monitors requires a VGA signal. As this signal has been specified in detail, no further attention has to be paid to other factors that influence the screen. Complying with the VGA standard ensures a reliable operation. VGA includes a vertical and horizontal synchronization signal, and three analogue signals representing the intensity of red, green and blue, in order to control each pixel.

Quality

The VGA standard was introduced by IBM in 1987 and is therefore a pretty old standard. The standard resolution for VGA is 640x480, but this can be altered; only then the official name isn't VGA anymore. As the signals are analogue, they are easily influenced by noise. Nowadays, because of digital signals, VGA is often converted from digital in computers to analogue and then back to digital in monitors and is therefore a redundant step between data transfer.

Costs

As the industry for monitors is big and still developing rapidly, it is possible to buy these in a range of prices. Although new models often contain new standards for imaging, VGA is still common. Buying a monitor with VGA can be very cheap.

5.3.2 Digital TFT-LCD

The second type of screen is a 'loose' LCD screen, often used to build into a product and not use as a product.

Control

Controlling a digital LCD screen that hasn't got a driver already connected requires a bunch of signals to address each pixel with a specific amount of bits to represent a colour. These signals are often brought out by a thin or small connector, as these LCD are designed to be implemented in another product.

Quality

Assuming a fairly modern digital screen, the full quality of the screen can be addressed as the signals are all digital. This quality depends on the amount of pixels per inch, number of bits per colour, reflectivity, clarity and other parameters.

Costs

As described above, there are many sorts of screen, depending on the resolution, size and other quality aspects the price can also differ. Nevertheless as these screens are in their 'raw' format as no

controller or housing are integrated, they are often sold in batches but can be offered for a smaller price.

5.3.3 Multicriteria analysis of screen alternatives

The evaluation of the two options that have been discussed above results in a solution that is most suitable for us, a digital LCD screen.

Convertibility: 3

The screen is only implemented to generate a general idea of how the product will look like in the future, as Super E-paper will be implemented once it is fully developed. Nevertheless this criterion is important, as the design of the screen controller should be converted to hardware for the E-paper. The smaller the modifications to be made; the better. The weight for this criterion is 3.

Availability and price: 3

Just as for the controller; the availability and price are criteria especially for this project, as the product has to be developed in a few weeks time, and at a maximum of 150 euro's; it is important that the screen can be bought. Weight is 3.

Quality: 2

This aspect of the screen is a reflection of how the final screen should look. The specifications of the screen of super E-paper aren't definite yet, but a timing diagram is available for comparison. The screen of choice is then compared to the super E-paper screen in chapter 7. A screen that is driven like the final screen gets the best score. Weight 2.

Implementation: 1

Although the screen will eventually be replaced by Super E-paper, the time and fuss it costs to implement the screen in our design is important as the project is on a tight schedule. Weight is 1.

Evaluation

The multicriteria analysis results are listed in Table 5-3. Each criterion is discussed separately and the best alternative is chosen every time.

Criterion	Weight	VGA monitor		LCD	
		Score	Weighted	Score	Weighted
Convertibility	3	6	18	8	24
Availability/Price	3	8	24	7	21
Quality	2	6	12	7	14
Implementation	1	9	9	6	6
Total			63		65

Table 5-3: Evaluation of possible screen solutions

Convertibility: LCD

Making a design that is closest to the final product when de Super E-paper can be used is worth much, as this is best for the LCD screen for their many connections.

Quality: LCD

Loose LCD screen can be controlled digitally and are therefore better than VGA monitors; also the pixel format is comparable to super E-paper.

Implementation: VGA monitor

As said before: meeting the specifications of the VGA signal insures a good operation and therefore these sorts of monitors win.

Availability and price: VGA monitor

For a “proof of concept” a recycled monitor could be sufficient, as monitors are widely available their costs are also very neat.

6 Final concept for the billboard

After evaluating the possible solutions, a design was extracted from the possibilities. This chapter will first give a brief overview of the total solution and then give detailed information on each individual part.

6.1 Design overview

The total design is able to read a full image via the data input and display it on its screen. This is done by a series of separate entities. During a data transmission via the wireless receiver; the data comes to the FPGA board where it is decoded and transferred into the SRAM. After transmission, the SRAM holds all the data for a total image, and the data is lead to the 'screen control entity' which generates timing and data signals for the screen. The screen itself operates via a series of level converters directly from the FPGA, while the backlight of the screen is fed via a 28v step-up converter. For a visual explanation; see Figure 6-1.

A. Total design

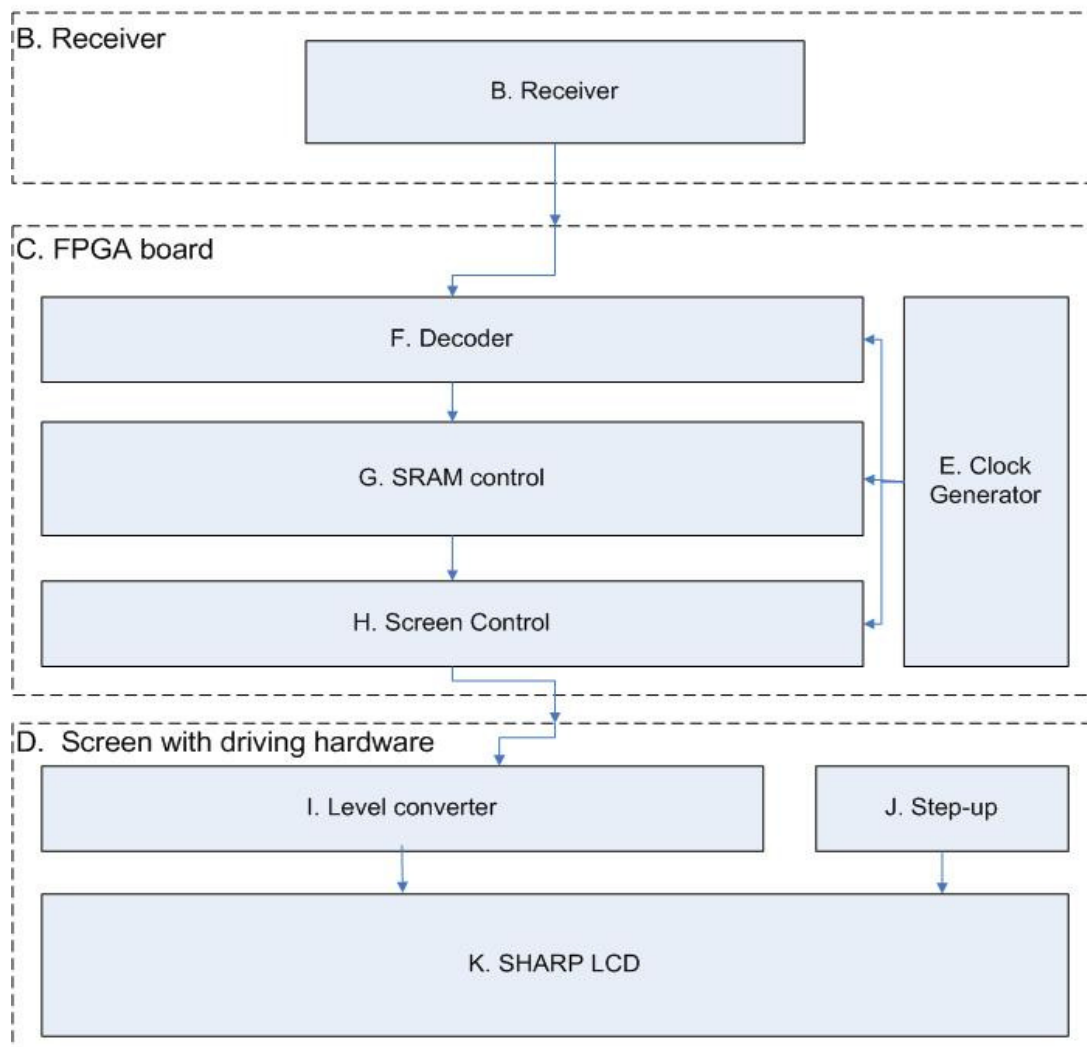


Figure 6-1: Design overview (a). Containing the receiver (b), the FPGA board (c) with its components; clock generator(e) decoder(f), SRAM control(g) and screen control(h). And the Screen with driving hardware (d) consists of a level converter(i) a step up converter(j) and the SHARP LCD(k)

6.2 External receiver

The external receiver is a WRL- 000691 breakout board from SparkFun [12] based on the nRF2401A+ [13] single chip 2,4GHz transceiver, see Figure 6-2. The breakout board makes it possible to connect the following pins externally: VCC, CE, CSN, SCK, MOSI, MISO, IRQ, GND. And is therefore easily implemented in the total system. Some key features:

- 250kbps to 2Mbit Data Rate
- On-board 3.3V LDO Regulator (3.3 to 7V supply allowed)

The design and implementation of this part has been done outside this project and is only included to give understanding of the functionality. Information regarding implementation and signal handling can be found in [1].

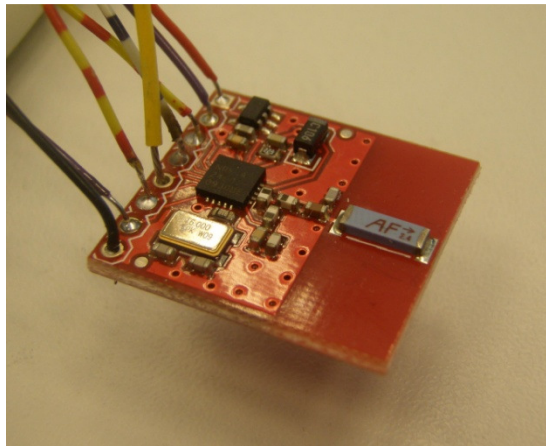


Figure 6-2: 2,4GHz transceiver based on nRF2401A+

6.3 FPGA design

The model is designed to perform a number of tasks. First, it needs to be able to coherently receive the data that is send into the device by a receiver. Next, it has to store the received data in the onboard SRAM. After all the data has been collected, an image has to be displayed on the screen. For all these functions a different controller is present to guide the data in the correct way. The complete circuit is shown in Appendix C-3: Total.

The inputs:

- **Clock**, this is the 50 MHz clock which is present on all FPGA boards. It is used to synchronize the entire circuit.
- **Reset**, the system reset is used to reset every controller to initial conditions.
- **Data_in**, the actual data that is received from the computer
- **Screen_off**, a separate button to enable manual disabling of the screen

The outputs:

- **lb_a_n, ub_a_n, we_n, oe_n, ce_a_n, dio_a, ad**, these are the signal used to control the SRAM
- **AN, sseg**, signals used to drive the 7-segment display which is available on the FPGA board
- **Ledstate**, also present on the FPGA board are 8 LEDS, which show additional data
- **H_sync, V_sync, CK, DISP, BL, R, G ,B**, the signals used to get a picture on the screen

As shown in Appendix C-3: Total 6-3, there are four main blocks that describe the circuit. These will be discussed in more detail in the rest of this chapter.

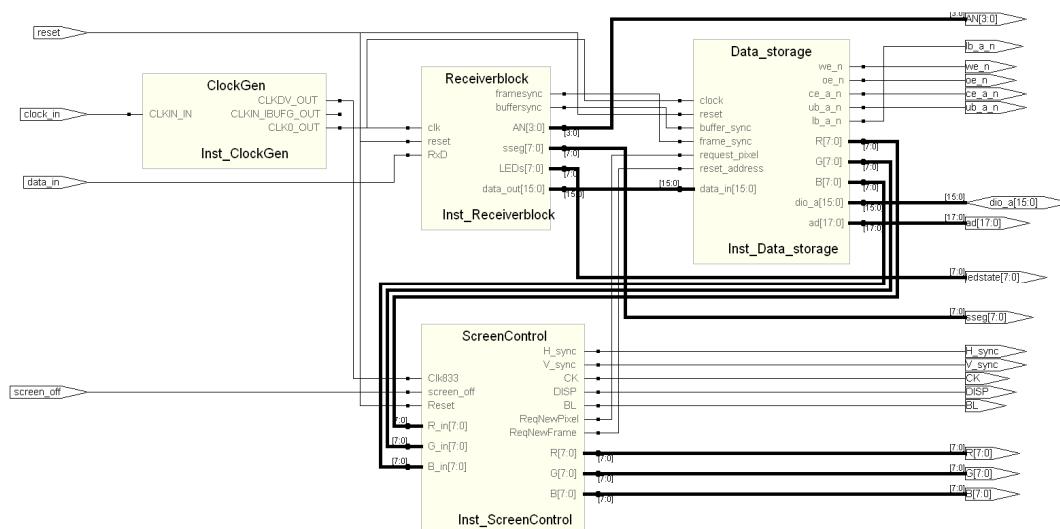


Figure 6-3: FPGA design overview

6.3.1 Clock Generator

The clock generator, *ClockGen*, divides an incoming 50 MHz signal to create a clock signal of 8.33 MHz, which is used to drive the screen, and a separate 50 MHz signal to synchronize the rest of the system. The 8.33MHz clock is created by dividing the 50MHz clock by six, as shown in Figure 6-4.

A clock generator works using a Digital Clock Manager (DCM), with some additional buffers to ensure a valid clock signal. This is shown in Appendix C-3: Clockgen.

A DCM uses a Delay-locked loop (DLL), which is basically the digital equivalent of the Phase-locked loop that is used in many telecommunication areas. The main difference, however, is the absence of an internal oscillator in a DLL.

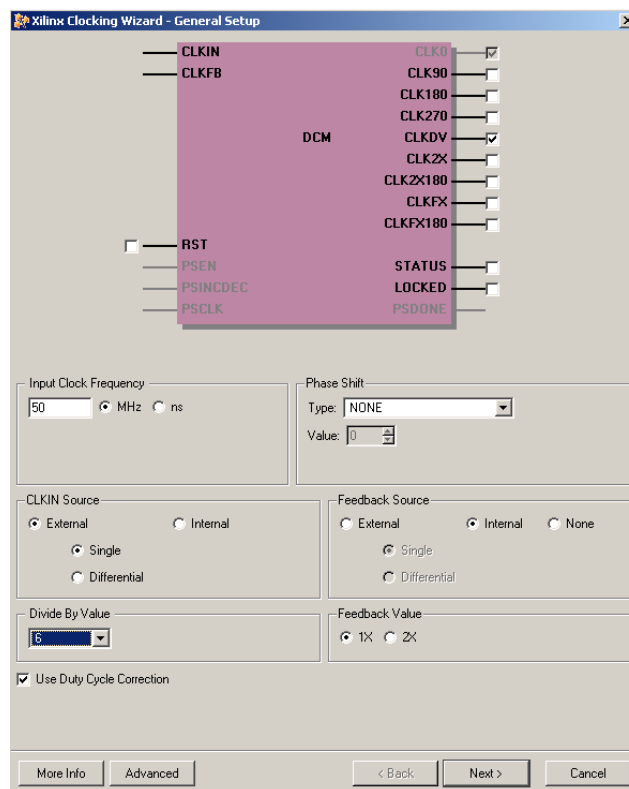


Figure 6-4: Xilinx clocking wizard, to configure DCM.

The actual design has not been done manually; the IDE from Xilinx can create one automatically using the integrated Core Generator. A user just has to configure the target using the Xilinx Clocking Wizard. Once the correct settings have been applied, one simply clicks generate to finish the setup.

6.3.2 Decoder

The receiver's task is to read data and format it in such form that *DataStorage* can put it directly into the SRAM memory. To do this the incoming serial data has to be decoded from its start and stop bits and aligned with two bytes at a time in order to comply with the SRAM's architecture.

Incoming data

As shown in Figure 6-5, the incoming signal is sent according to RS232 timing standards, only the voltage levels differ; they have to meet the specification for the FPGA board. A total transmission consists of 130560 red, green and blue bytes followed up by each other, therefore the receiver should continuously convert the data.

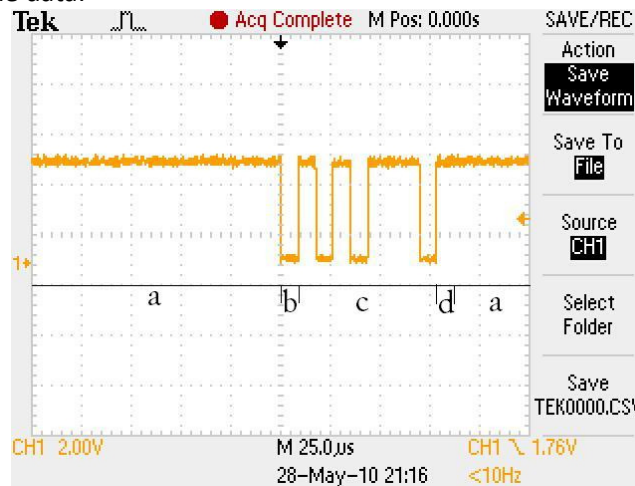


Figure 6-5: A transmission of a byte via RS232: the line is idle (a), the transmission starts with a start bit (b) followed up by 8 data bits (c) and the transmission ends with a stop bit (d)

Total package

A total transmission holds all the information for an image of 480 by 272 pixels of 3 colour bytes. So when a transmission starts the receiver should wait for 391680 bytes to be received.

Data to DataStorage

After decoding the RS232 signal, every two bytes that follow each other up should be set in a 16 bit vector and send to the *DataStorage*. This is synchronized by setting the *buffer_sync* bit when data is valid, and clearing it while filling this buffer with two new bytes of data.

Next to *buffer_sync*, *frame_sync* is used to synchronize a full frame. Receiver sets this signal to low while still receiving 391680 bytes. Pixelbuffer can decide with this signal whether to show a stored image or a pending image. The flow diagram of decoder is shown below, Figure 6-6.

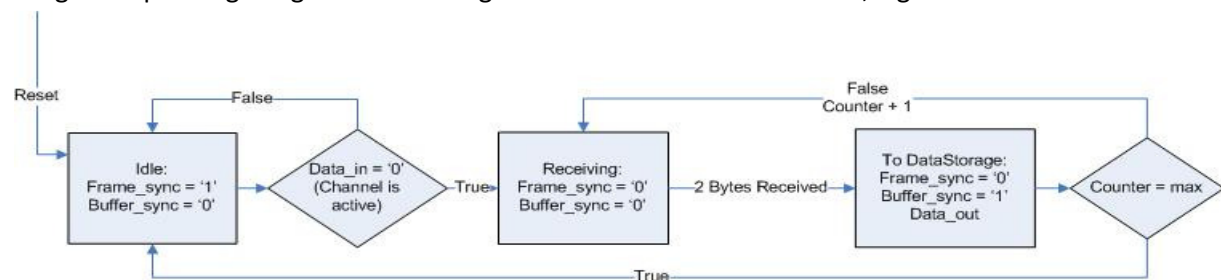


Figure 6-6: Flowchart of decoder

6.3.3 Data storage

DataStorage is by far the largest component in the design. It not only interfaces with the onboard SRAM, but it's also alert to new data coming from the *Receiver* and pixel requests from the *ScreenControl*. *DataStorage* consists of two main blocks; *Pixelbuffer* and *SRAM_control* (Appendix C-3: Data storage). However, it is first important to understand how the SRAM has to be driven.

Static Random Access Memory

The specific SRAM module (IS61LV25616AL), see Figure 6-7 [14], on the FPGA board has 256.000 addresses available, each able to store 16 bits of data. In total, that allows for the storage of 8Mbits. This is large enough for our purpose, since one image for the screen is 3.1Mbits.



Figure 6-7: An IS61LV25616AL [14]

There are a number of signals to the SRAM needed to get it to work, but there are only four that are important. The first one is the *address* line, consisting of 18 bits. This determines where an action has to take place. Then there are write enable, *we_a_n*, and output enable, *oe_a_n*. The '*_n*' signifies that both signals are active low. These signals enable a write and a read operation respectively. Also, they are mutually exclusive, meaning they shouldn't be enabled at the same time as this could have unpredictable consequences.

The last important signal is *dio_a*, which is essentially the data bus. Since it is of type inout, it has to be driven by a tri-state buffer. In case of a read operation, the FPGA should not put any kind of data on the data lines. This is done by putting "Z", or high impedance, on the data bus instead. Otherwise, in case of a write operation, the data from the FPGA should be forwarded. Figure 6-8 shows how a tri-state buffer is implemented in the design.

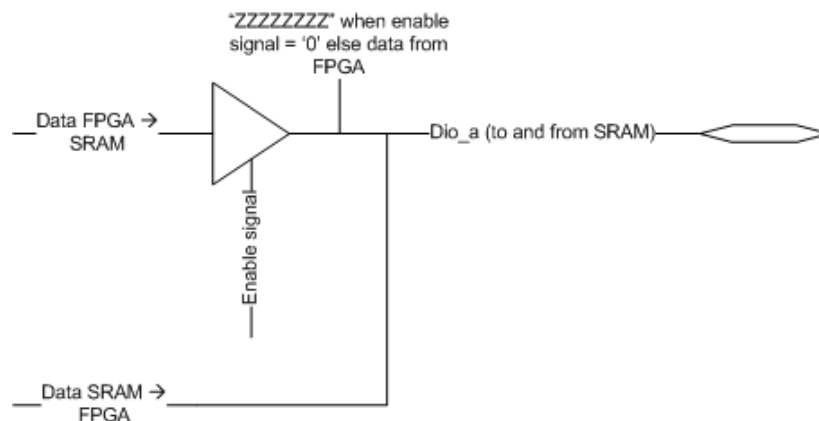


Figure 6-8: Overview of a tri-state buffer

SRAM Control

As shown in Appendix C-3: SRAM control, *SRAM_control* itself is also made up from two blocks; *SRAM_ctrl* and *SRAM_toplayer*.

SRAM_ctrl

SRAM_ctrl is directly connected to the SRAM. It is basically used as a synchronized shell around the asynchronous SRAM. This block can perform two functions; a write and a read operation on the SRAM. Both read and write operations take 60 nanoseconds and can only be invoked by the other block, *SRAM_toplayer*. It is theoretically possible to speed this up to 20 nanoseconds. However, trying to achieve this kind of performance has not been accomplished as of yet.

This part is the only block that interfaces with the SRAM. How the SRAM works has already been described in 5.1.2 Static Random Access Memory.

The *SRAM_ctrl* has to be triggered to perform an operation. The flowchart of Figure 6-9 explains this in more detail.

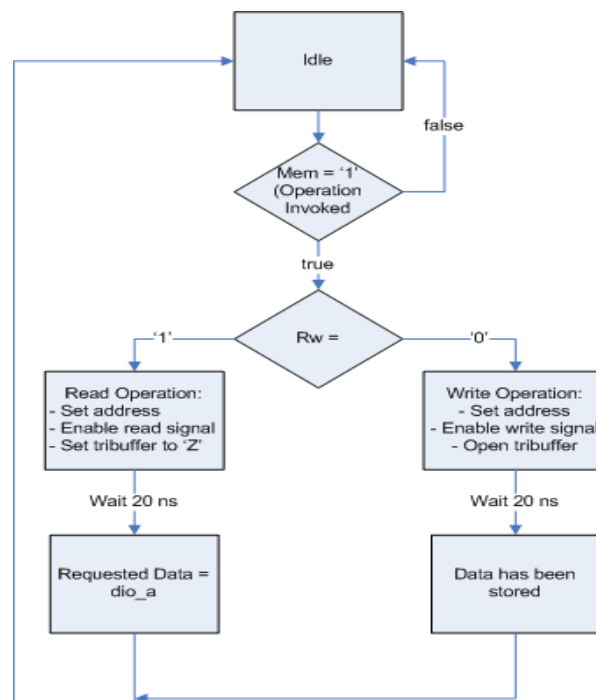


Figure 6-9: Flowchart of *SRAM_ctrl*.

Explanation of the signals:

- *Mem*: Starts a memory operation.
- *Rw*: Determines whether it is a write or a read operation. A '1' is for a read, a '0' for a write.
- *Dio_a*: The data bus to the SRAM. This is of type inout and therefore has to be driven by a tristate buffer as described in 5.1.2 Static Random Access Memory.

SRAM_toplayer

Besides invoking read and write operations, the main functions of *SRAM_toplayer* is to keep track of the SRAM address, where new data has to be written to or read from and to prioritize between a request from the *Pixelbuffer* to read the memory and a request from the *Receiver* to write to the memory. Attempts to process these request simultaneously have proven to be unsuccessful; the SRAM simply can't keep up. The solution is to award receiving data the highest priority and to ignore new data requests from the *Pixelbuffer* once new data is being received. The fact that this would cut the *ScreenControl* off from new pixels is solved inside the *Pixelbuffer*. This ignoring of data requests is also clearly visible in the flowchart of Figure 6-10.

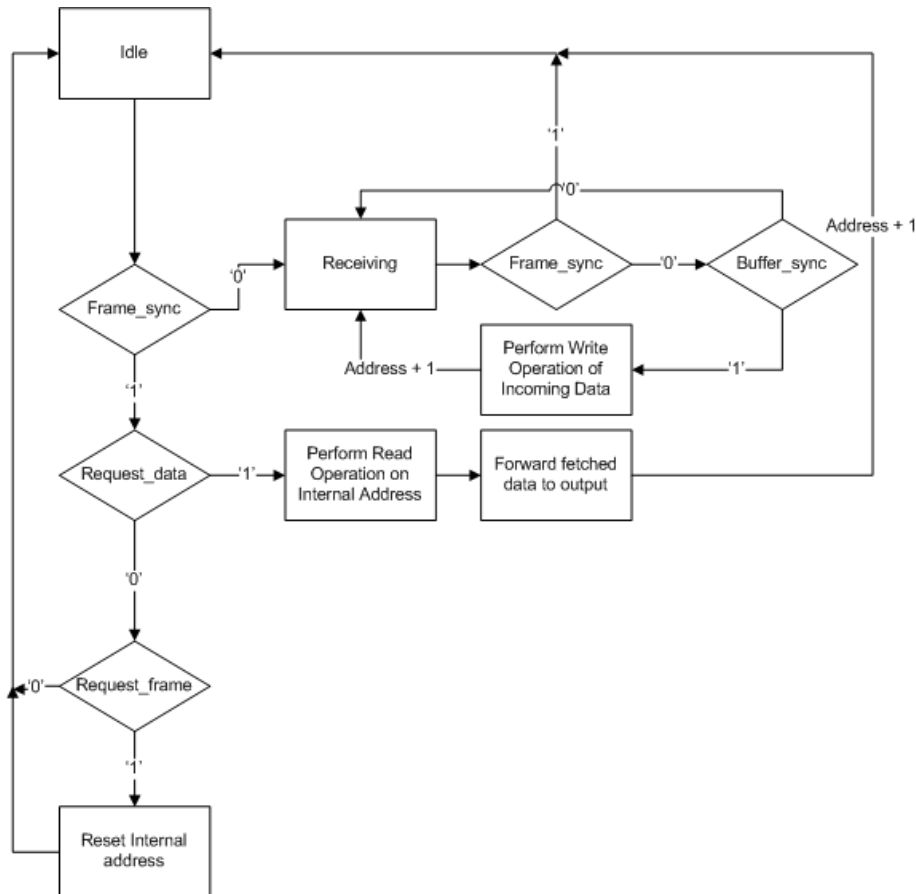


Figure 6-10: The flow diagram of *SRAM_toplayer*

Explanation of signals:

- **Frame_sync**: Signal coming from the *Receiver*. A '0' means new data is being received.
- **Buffer_sync**: Another signal from the *Receiver*. Activated once a new set of 16 bits is ready to be stored.
- **Request_data**: This request is from *Pixelbuffer* to make the next data in the SRAM available on the output.
- **Request_frame**: Another signal from *Pixelbuffer*. Once the end of the screen has been reached by the *ScreenControl*, the fetching of data has to be restarted to the first address, so it is reset.

Pixelbuffer

The reason why *Pixelbuffer* exists might not be obvious, but it is very necessary to ensure that correct data reaches the screen. One pixel is 24 bits, while one SRAM address only holds 16 bits of data. So when *ScreenControl* requests a new pixel, every other time two addresses have to be read. Since one read operation takes about 60 nanoseconds, the maximum time between a pixel request and when the corresponding data is available on the output is 120 nanoseconds. The maximum allowed difference is about 60 nanoseconds, so connecting *ScreenControl* directly to the *SRAM_control* would pose problems half of the time. *Pixelbuffer* solves this problem by prefetching pixels, so that when *ScreenControl* requires a new pixel, it is available within 20 nanoseconds. Another issue solved by *Pixelbuffer* is when data is being received, the SRAM won't respond to requests to read the SRAM.

Using Xilinx Core Generator once again, block ram available on the FPGA can be utilized to create a small memory unit of about 170 Kbit. Even though this is very small compared to the SRAM, it is possible to store a small, low-quality picture of our logo. So while a new image is being received, instead of a black screen, our logo is displayed. Once the new picture is completely ready, our logo disappears and the freshly received image is shown on the screen. Once again, this process has been visualized. Figure 6-11 shows the flowchart for *Pixelbuffer*.



Figure 6-11: Flowchart of *Pixelbuffer*

Explanation of signals:

- **Frame_sync:** As described before, this is a signal from *Receiver* to alert this component that a new image is being received.
- **Request_pixel:** A signal from the *ScreenControl*. Once it has been activated, the next pixel has to be available on the output in less than 60 nanoseconds. *Request_pixel* self is basically an 8.33MHz clock with some gaps.
- **Buffer1 and Buffer2:** These two 48-bit buffers can hold two pixels each. Once Buffer2 is empty, it gets Buffer1's contents and Buffer1 get filled with new pixels.
- **Request_frame:** Another signal from *ScreenControl*, once the end of the screen has been reached, the fetching of data has to restart at the first pixel. So the buffers have to be refilled, since they still have the old data.

6.3.4 Screen control

This block is the last one in line of the total system; it is connected to the screen and applies the correct signals to insure normal operation of the SHARP LCD screen. The *Screen Control* has two main task; insuring a steady operation of the screen by generating the correct timing signals like the horizontal synchronization , the vertical synchronization and other specific delay's. Next to that *Screen Control* has to request its data from *Pixel Buffer* and synchronize this with the clock signal for the screen in order to fulfil its task of displaying an image.

Start-up and standard operation

In Appendix B-3: SHARP LCD screen, the timing signals for start-up and power down are specified, therefore *screen control* should also take in account for these signals. This is done by initializing a start-up sequence after a reset. And when powering-down, a switch should be set before switching off the supply. The main considerations to be taken are:

- Do not apply AVDD (5V), before VCC (2,5V).
- Wait with applying timing and data signals for at least 50ms after power is applied.
- Display signal should be set to high 500ns after data signals.

Standard operation signals

After the start-up, *screen control* should generate vertical and horizontal synchronization signals in order for the screen to operate normally. In short; it comes to a couple of signals, also in Figure 6-12:

- Vsync: used for the indication of the start of a new screen.
- Hsync: used to generate a synchronization for every horizontal line of 480 pixels.
- CK: used to synchronize the write to each individual pixel.
- Data: the red, green and blue signals (each 8 bits) should be synchronized with CK.

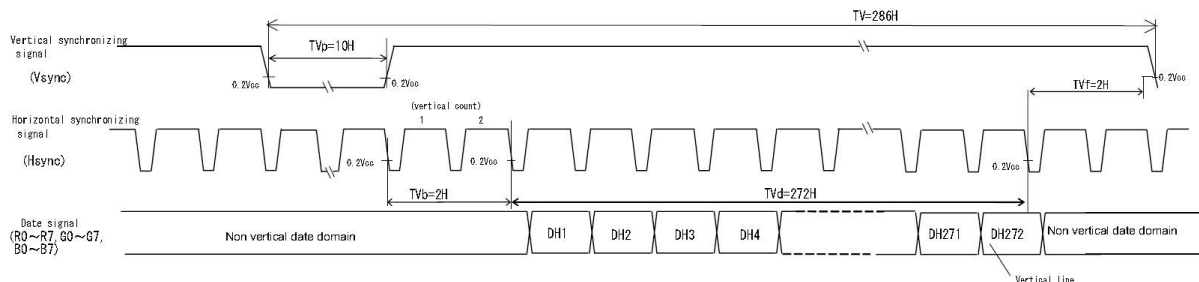


Fig 2 Input signal timing

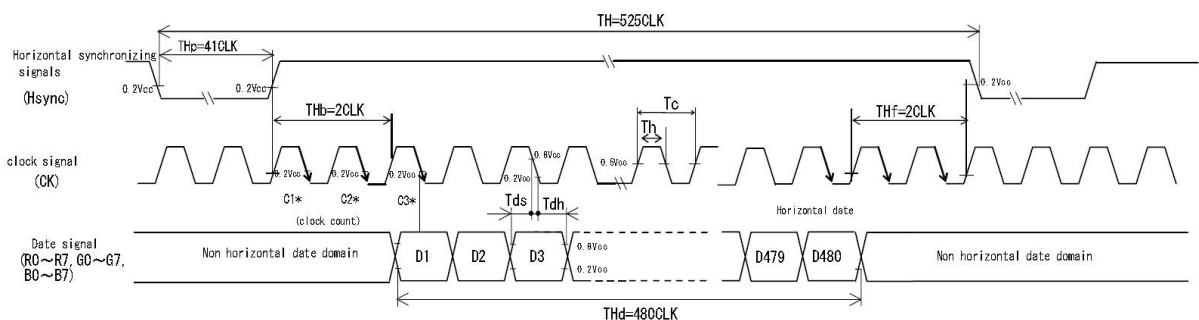


Figure 6-12: Timing signals of Vertical and Horizontal synchronization, clock and data[15]

6.4 The Screen

To display images our product makes use of a SHARP LQ043T3DX02 TFT-LCD screen [15], Figure 6-13,[16]. The screen consists of a backlight, control hardware and a LCD panel. Some key features from the datasheet:

- 480 horizontal by 272 vertical pixels with 24 bit colour.
- Requires supply voltages of 2,5 and 5 volt.
- Timing characteristics enables a refresh rate of 50Hz.



Figure 6-13: SHARP TFT-LCD screen [16]

Further data from the datasheet learns us that the screen “is a color active LCD module incorporating amorphous silicon TFT” [15]. Further information about the specific construction of the drivers is not given. Figure 6-14 and Figure 6-15 show how a LCD has been build up [17], the data driver and a DAC has been give an understanding how pixels are addressed. In our case the 3-bit DAC should be and 8-bit.

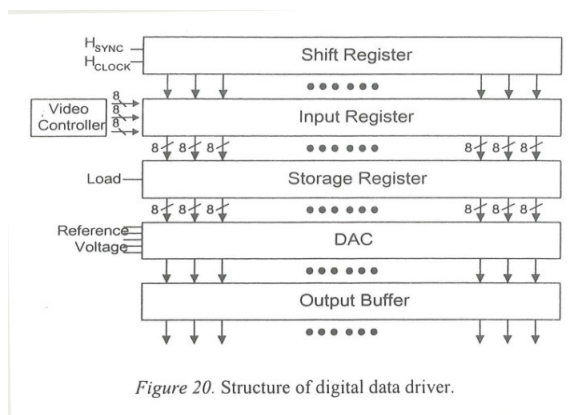


Figure 6-14: Structure of digital data driver [17]

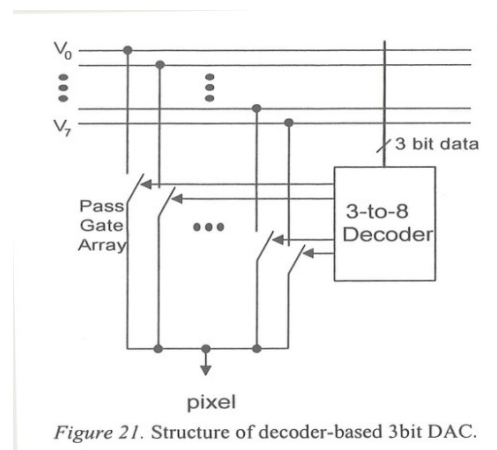


Figure 6-15: Structure of decoder-based 3-bit DAC [17]

With the timing signals Figure 6-12, it becomes clear how each individual pixel is written. After a vertical synchronization the first row of pixels is selected from the TFT-LCD matrix. At each clock signal the shift register from Figure 6-14 ‘points’ to the next row, for our screen there would be 480 lines. This pointer enables the data from the controller, for each colour (red green and blue) 8 bits of information, to be passed on so that a voltage can be generated for that specific pixel. This is done by a digital to analogue converter, with 8 bits selecting a specific voltage, which is shown simplified in Figure 6-15.

6.5 Step-up and level converter

Next to the hardware described above there are two other components important for the system to work. They provide secondary functions and are therefore described in this section

6.5.1 Step-up converter

For the backlight to work properly, a 28 V supply is needed which can provide up to 18 mA of current. As we already have a 5 V supply accessible on the FPGA board it is easiest to create it from this source. Therefore we need a step-up converter, the one used is a MC34063A [18]. Figure 6 in the datasheet, also displayed below as Figure 6-16 shows how the circuitry is build.

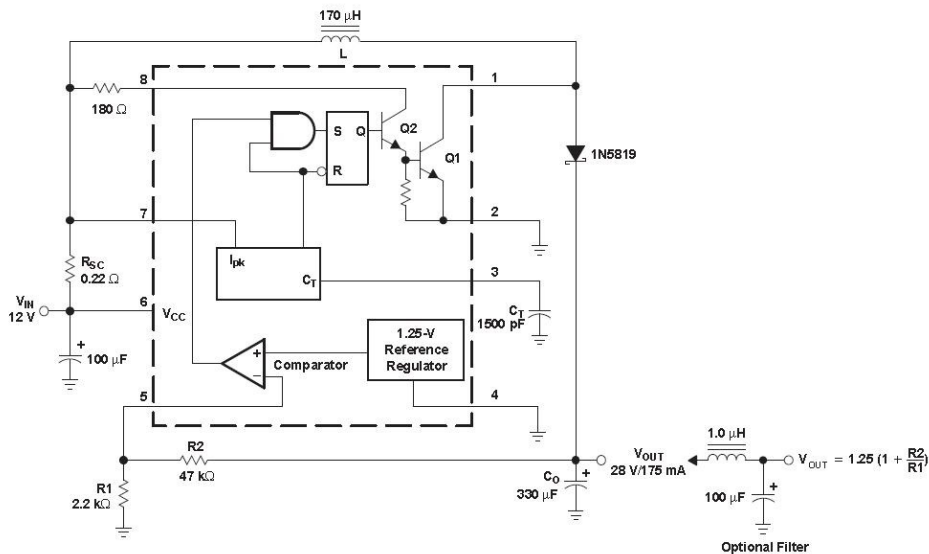


Figure 6-16: MC34063A step-up converter configuration [18]

6.5.2 Level Converter for SHARP LCD

As described in the previous chapter; the SHARP LCD requires 2,5V on its inputs for the data and an additional 5V as analogue supply. To meet these specifications an array of resistor pairs is used, creating 3,3V to 2,5V level converters, see Figure 6-17.

Apart from the data signals, the screen also needs two power supplies, in [15] we find that the 2,5V supply has to be applied before the 5V. As the 2,5V is generated from the 5V, a transistor switch is used turning on the 5V only when the 2,5V is available.

The full level converter design can be found in Appendix C : Hardware design.

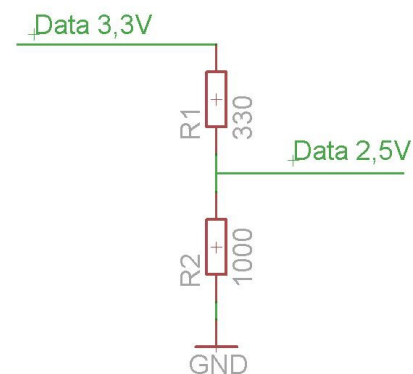


Figure 6-17: 3,3 to 2,5V level converter

7 Testing concept billboard with and without wireless link

Two separate test setups have been used. The first one has been carried out without the other team. The receiving end was simply directly connected to a computer via USB, so there was no transmitter-receiver combination necessary. This eliminated the chance that there could be errors in the transmission. So if an error occurred, it would have to be in the decoding, storing or screen control.

Once we were satisfied with the results, the transmitter-receiver modules could be hooked up. Also the software changed. Instead of working directly from MATLAB files, a graphical user interface was used to send the picture.

7.1 Two test set-ups

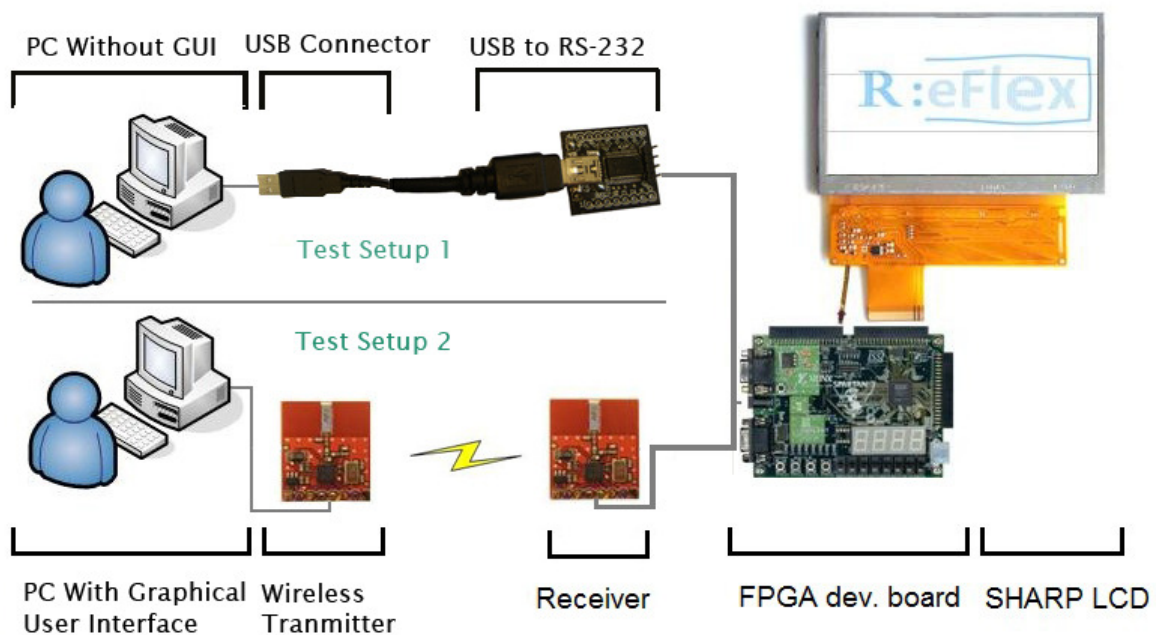


Figure 7-1: Test setup including all parts of the total design

As described before and illustrated in Figure 7-1, two different tests were set up. In the first test, a computer with MATLAB, but without a graphical user interface was used, a USB connector and a USB-to-RS232 converter. This setup was also used for debugging. The second test also used a computer with MATLAB, but this time a graphical user interface was programmed. A wireless transmitter was connected to the computer and a wireless receiver was connected to the FPGA board.

7.2 Results of the first test setup

The first test was carried out as follows. The user had to change a MATLAB file to send the desired picture. The data was then transmitted via USB to a USB-to-RS232 converter. This converter then forwarded the data to the FPGA, which in turn stored the image in the memory. Once the image has been fully received, the FPGA starts reading the SRAM and outputting the signals that are required to drive the screen. These signals were then lowered by a voltage divider, which was necessary due to voltage differences between the FPGA and the screen. Finally, the image is displayed on the LCD screen of which the result is shown in Figure 7-2.

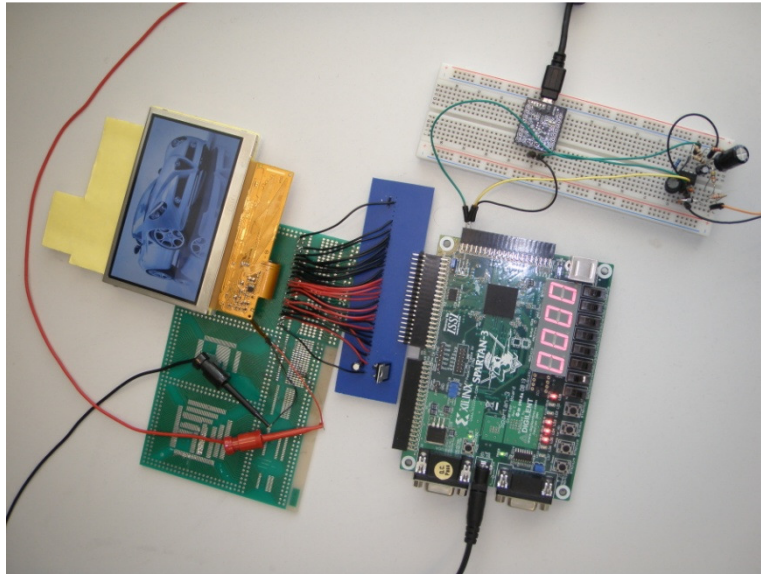


Figure 7-2: The result of the first test

During the test, the FPGA is keeping track of how many bytes it has received and displays that number on the four 7-segment displays. Also, while receiving, the LCD screen displays the R:eFlex logo, which is preloaded in the blockram of the FPGA.

As expected for a debugging setup, the first tests carried out with this setup weren't always successful. However, thanks to a handful of extra debugging outputs, we eventually got rid of all the problems and created a working system, as can be concluded from the correct image that is shown on the LCD screen, see Figure 7-2.

7.3 Results of the second test setup

The second test is a slightly different. First the user sits behind the computer to which the wireless transmitter is hooked up. After choosing a suitable picture in the graphical user interface, it was sent to the wireless transmitter, which in turn transmits it to the wireless receiver that was placed three meters away. Then the receiver forwards the received image data to the main controller, the FPGA. The receiver also does error control, to ensure a faultless transmission. Once again, the FPGA stores the image in the onboard SRAM and finally drives the screen to show the image.

The results weren't quite as successful as those of the first test setup. Due to transmission errors, the error checking wireless receive filtered out the pixels that weren't received correct. Therefore, most of the tests performed with the second test setup looked more like Figure 7-3. Sometimes the correct image was received, but that was rare.



Figure 7-3: Not yet successful result using the second setup

These errors were due to transmission interference. The test was carried out in an environment that was loaded with electrical equipment, such as computers, screens and other electronics. Also the transmitter-receiver modules operated at a frequency that is widely used by numerous wireless devices. Different tests carried out by the other team showed successful tests in the hallway, where the walls blocked most interference.

Despite the unsuccessful last tests, we were able to confirm that our part of the design basically works as it should. There are two flaws, both having to do with the SRAM. First of all, there wasn't enough of it on the FPGA board to allow enough pictures to be stored to display a slideshow of images, let alone animations. The second problem is the storage controller. Since it couldn't handle storing new data while continuing to read, an image that wasn't stored on the SRAM was displayed during the receipt of a new picture. Both these problems can be solved with a larger block of SRAM, which, as paragraph 8.3.2 [Implementation of SRAM using SG-TFTs](#) explains, is possible.

8 Product evaluation

In the chapter the performance of the system is analyzed and to what extent it has met the criteria. The most important criterion that will be examined is the effort required to convert the design to a working prototype electronic billboard, created with super E-paper technology, especially SG-TFTs. Since our system consists of three main components, each of these will be assessed independently.

8.1 Overall hardware modification

The whole system is still in a test stage, all components are separately implemented from each other. For future purpose, these have to be integrated on a single 'super E-paper'; a flexible backplane with analogue front ends, RF circuit, digital data computing and the QR-LPD on top. With figure BLABLA the steps to be made can be explained.

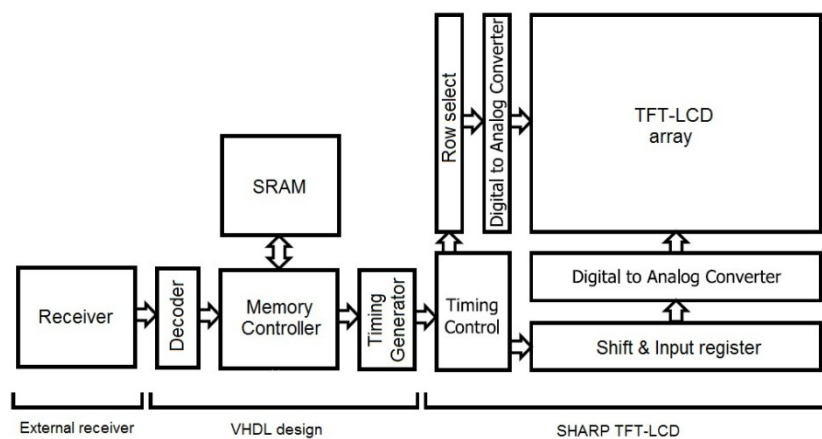


Figure 8-1: Simplified block diagram of the current design

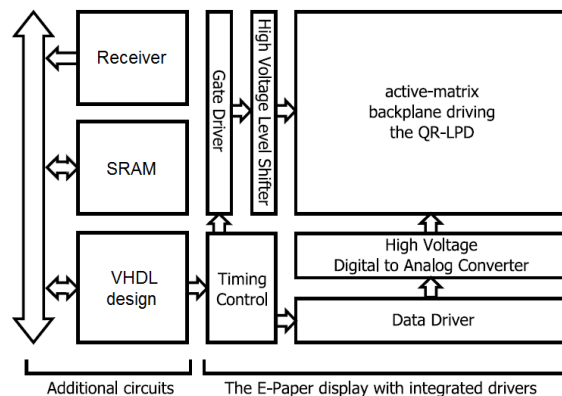


Figure 8-2: Block diagram of the design build on super E-paper, partly from [19]

Comparing Figure 8-1 with Figure 8-2 shows that the biggest adjustments to be made for the system involve the transfer of the design to SG-TFT's. This then again leads to other questions involving the differences in both figures. These problems will be discussed in the following chapters, categorized by the different pieces of the current design.

8.2 Receiver

For data input a receiver is needed, currently the design for this receiver has been used from [1] and is fully external, as it relies on different technologies it is hard to copy this design onto SG-TFT hardware. And as the antenna is an on chip ceramic antenna, it needs many changes before it can be implemented on flexible hardware.

A design of a 433 MHz ISM band amplifier has been modeled, simulated and build [20], this paper shows that it is feasible to embed certain amplifiers on SG-TFT and it should therefore be possible to have a 433MHz receiver on the backplane of our super E-paper.

Nevertheless would it be better if a higher frequency could be used. As many modern devices already have certain 2,4GHz abilities implemented and these kind of receivers also offer more data rate, which is almost necessary as modern pictures and other data files require more bandwidth. Simulation of SG-TFTs shows that amplifiers in the range of 2,4GHz can be made, although not with large gains, 2.4.3 from [21]. So designing a RF receiver that works at 2,4GHz requires more effort, but is not unthinkable.

8.3 Control Hardware

8.3.1 Implementation of an FPGA using SG-TFTs

The design has been tested to be working correctly on an FPGA. Such an FPGA consists of many logic cells, as described chapter 5.2.2. According to the datasheet of Appendix B-1, our FPGA has 4320 logic cells. If a logic cell can be constructed using SG-TFT technology, it is also possible to implement the entire design with that technology. In this paragraph it is discussed if that is possible and how.

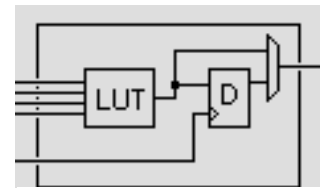


Figure 8-3, A logic cell [8]

Type of cell	Cell name	Cell description
Inverters	inv	Inverters with different sizes 1, 4 and 16.
	inv4	
	inv16	
NAND Gate	nand2	Two ports NAND Gate
NOR Gate	nor2	Two ports NOR Gate
Flip-Flop	dff	D Type Flip-Flop
Buffers	buf4	Buffer with sizes 4 and 16.
	buf16	

Figure 8-4:the SG Standard Cell Library [18].

According to paragraph 2.1.3 of Wei Man Chin's Master Thesis [19], it is possible to construct all the cells that are in the SG Standard Cell Library [22]. An overview of this library is shown in Figure 8-4. As can be seen in Figure 8-3, a logic cell is made up of three main components; a lookup table, a d-flip

flop and a 2-input multiplexer, or mux, to bypass the flip-flop if desired. Both the lookup table and the multiplexer aren't standard cells.

A lookup table is essentially a multiplexer with constant inputs. The inputs of the lookup table are the select bits of the multiplexer. This is illustrated in Figure 8-6. So the question is now, how is a multiplexer constructed using cells from the standard library.

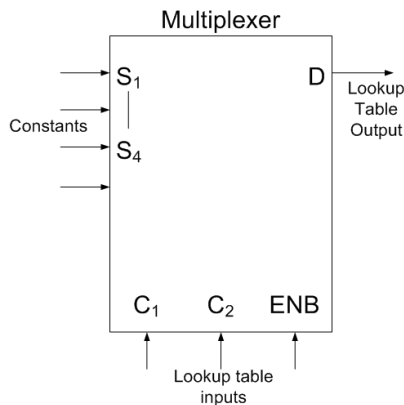


Figure 8-6: A lookup table created using a 4-input multiplexer with constant inputs and controlled select bits

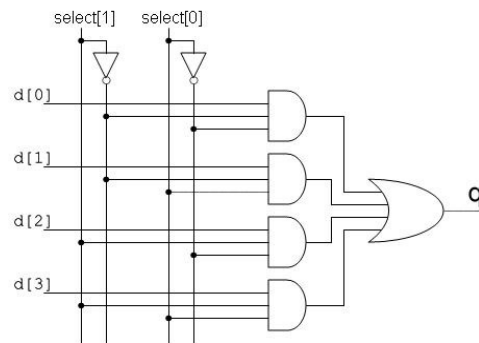


Figure 8-5: Separate components of a multiplexer [22]

As illustrated in Figure 8-5[23], a multiplexer can be constructed using basic components. A 4-input mux that can be used for the lookup table consists of two inverters, four 3-input ANDs and a 4 input OR. Even though ANDs and ORs aren't the standard components, they can easily be implemented using a simple combination. A 3-input logic AND can be made with a NOR, a NAND and an inverter as illustrated in Figure 8-7. An OR is simply a combination of a NOR and an inverter.

How these components from the standard library are constructed in practice is beyond the scope of this thesis. More information on this can be found in the Bachelor thesis of Miki Trifunovic and Amit Gupta [24].

But recreating an FPGA isn't enough; it also has to be able to operate at high enough frequencies. In [20] it is reported that a feasible operating frequency of a single transistor is 5 GHz. Since a multiplexer, the most important part of a logic cell, as depicted in Figure 8-5 consists of only a couple components, the maximum operation frequency is bounded by the

number of transistors in the longest path. All the paths are equally long; first an inverter, then a 3-input AND and finally a 4-input OR. The total number of transistors in the longest path is about ten; two in the inverter, four in the 3-input AND and four in the 4-input OR. Therefore, the maximum operating frequency is 500 MHz, which is still high enough for our purposes, since the clock frequency of the FPGA board is only 50 MHz.

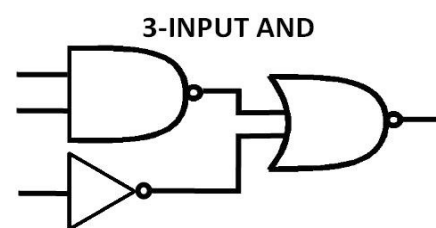


Figure 8-7, 3-input AND using SG standard library components

8.3.2 Implementation of SRAM using SG-TFTs

In the design, SRAM was successfully implemented to store images. To be able to use it in an electronic billboard made of super E-paper, it has to be flexible. Of course, this means it has to be made using SG-TFTs. Successful tests have already been done in DIMES [25].

The SRAM was created using six transistors and is therefore called 6T-SRAM as illustrated in Figure 8-8 [26]. This standard design was then fabricated in the low temperature process called μ -Czochralski process.

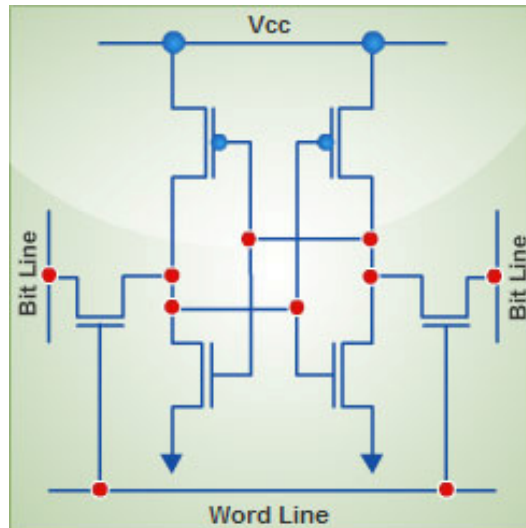


Figure 8-8: Construction of an SRAM cell [25]

According to the measurements done in the test, the SRAM performs very well. The static noise margin (SNM) of small transistors for read operations wasn't good enough. Larger transistors had a decent SNM.

The reported maximum operating frequency is high enough for our purposes. At 87MHz wordline frequency the measured read access time was equal to 13 ns. The write access time was measured at 50MHz, which measured 8 ns. These access times are fast enough to be implemented in an electronic billboard such as ours.

The SRAM can be implemented on the back of the billboard, so the maximum physical size of the SRAM is not an issue. Therefore, there will not be a problem with not enough storage for images or animations, since the maximum storage size is almost limitless.

8.4 Modifications to the screen

When comparing the actual implementation of the screen and the future design, some modifications have to be made. To give an understanding of both technologies, The four images below show an overview of two pixel matrices, together with the belonging single transistor.

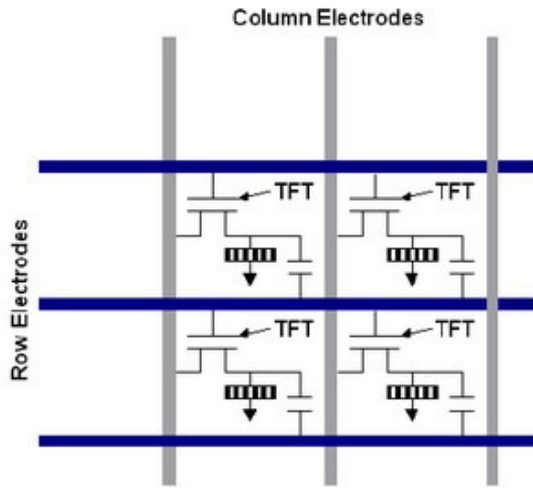


Figure 8-9: Pixel matrix of an TFT-LCD

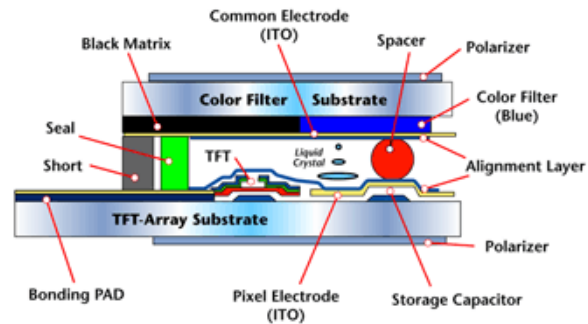


Figure 8-10: Single TFT in a pixel [27]

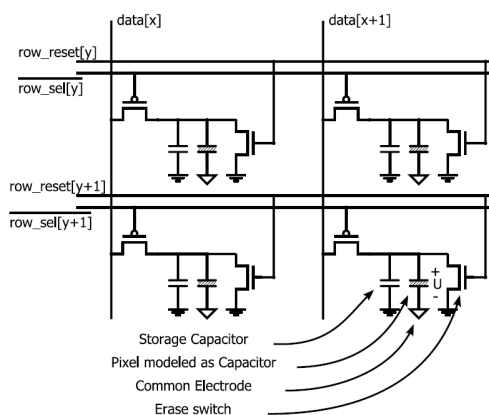


Figure 8-11: Pixel matrix of an QR-LPD [19]

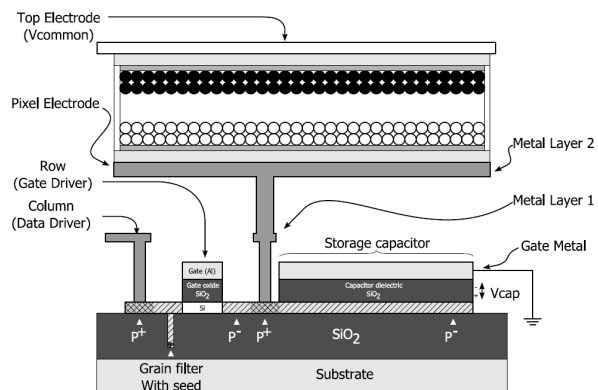


Figure 8-12: Single pixel of QR-LPD with SG-TFT [19]

8.4.1 Size difference between SHARP LCD and super E-paper

Not the most radical modification, but as the current screen only contains a total of 480 by 272 pixels changes for the timing generation are mandatory. For a complete billboard (keeping future implementation in mind), pixel amount would be approximately 3000 by 4000. By choosing a high enough pixel clock, a total still image can be still written in an acceptable time.

The total amount of pixels of a poster is quite large. To be able to display a short animation; the pixel clock should be much faster than it is now. To generate a fluent motions around 25 frames per second is preferable. This would require a pixel clock of approximately 300 MHz. For future animation implementation, cascading multiple QR-LPD displays could also be an option.

8.4.2 Resetting the QR-LPD

Due to charge on the liquid pixel material an QR-LPD requires a reset before setting the new pixel colour while a TFT LCD screen can be rewritten directly. This adjustment has influence on the design of the control hardware; before writing a line of pixels during a horizontal synchronizations, as described in 6.3.4 Screen control, the whole line has to be reset. Figure 8-13 shows the modifications to the timing signals. During a new horizontal synchronization a pixel line will reset, before each pixel is written.

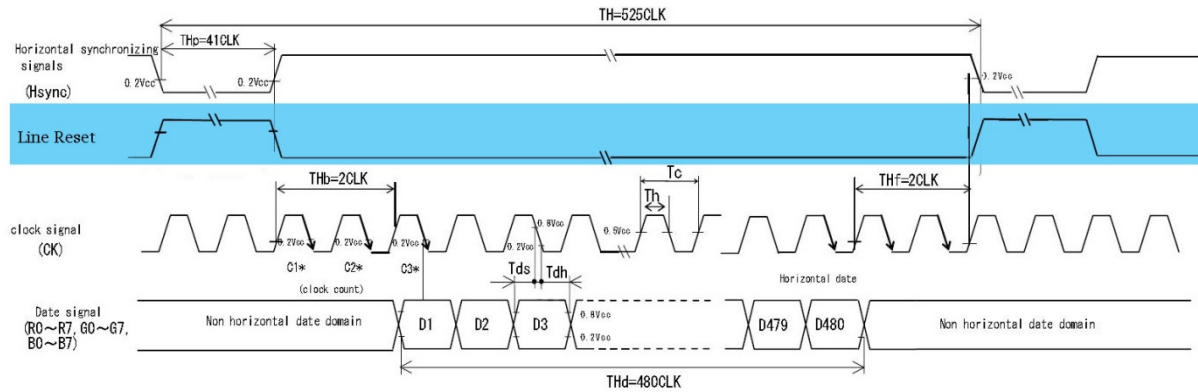


Figure 8-13: Horizontal timing with reset for the QR-LPD

8.4.3 Continues and single shot data

Another difference between TFT-LCD and QR-LPD is the fact that a liquid powder display is 'non-volatile' and can therefore keep its state of charge per pixel, and 'hold' an image. While the TFT-LCD is volatile and requires a continuous stream of data signals, even when a still image is displayed. While redesigning the timing generator, the screen should not be written with 50hz but only once a new image is uploaded. For our design, the screen is primarily included for multiple still images, and therefore also needs SRAM. The SHARP LCD requires a pixel clock of approximately 9 MHz, while this speed is not necessary for still images on a QR-LPD, this can be maintained to allow the possibility of future animation implementation.

8.4.4 Colour control by voltage levels of QR-LPD and LCD

The voltage swing needed to control an SG-TFT element is much larger than needed for a TFT. Documentation of the SHARP LCD screen is not sufficient to give detailed information on voltage levels, but a standard TFT requires a voltage swing between -5V and +20V to have full control of the pixel colour. As the SG-TFT requires a swing of 70V between the top plane (Vcommon) and the pixel electrode, the level shifters needed to have full control over pixel colour have to be redesigned. The level shifters should be able to set 0V at one electrode and 35 to 70V at the other to control the greyscale. Figure 8-15 and Figure 8-14 gives an explanation on these matters.

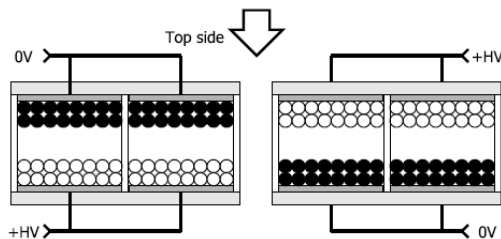


Figure 8-14: Voltages applied for black and white [18]

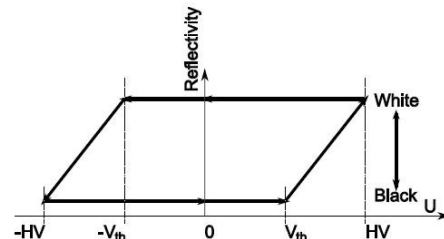


Figure 8-15: Hysteresis curve of grayscale settings [18]

9 Conclusion

The purpose of this report was to show a design of an electronic billboard can be effectively realized using super E-paper. The conclusion is split in two. The first part describes what design choices have been made, what the design looks like and how it performs. What is needed to implement the design using SG-TFTs is discussed in the second part.

As a receiver, a standard 2.4GHz device, found in many other applications, suffices. Other research showed that both the distance at which the data could be received and the transmission rates were acceptable.

The controller that will be constructed using SG-TFT technology consists of three components. First a receiver/decoder is required. This buffers the serial data from the receiver device and forwards it to the storage management component in 16-bit packets.

The central component, which acts as the main controller, stores the received data in the memory while at the same time sending data to the screen.

The screen is driven by the last component of the controller, the screen controller. This drives the screen we have chosen.

SRAM was chosen to store the image data. It is fast enough to satisfy the minimum required data rate to fill the screen and was easiest to implement using SG-TFTs.

Testing the design with a super E-paper front plane was not possible, so an alternative screen was chosen. The screen that was chosen is an LCD screen which is similar to the one used in a PlayStation Portable. Driving this simple screen is not much unlike how the front plane of super E-paper has to be driven, so it allowed for decent testing of the rest of the design.

In order to test the design it was loaded onto an FPGA, which is basically reprogrammable hardware. Together with a standard 2.4GHz receiver and the simple screen, a small scale prototype was created. With this setup, it is possible to send an image from a computer via USB, which is then sent to a transmitter. The receiver picks up the signal, transfers it to the FPGA, which in turn stores in the SRAM it and sends it to the screen.

To drive this screen correctly, some additional hardware had to be designed. First of all, the FPGA output pins operate at 3.3V, while the screen requires 2.5V. Therefore, a circuit that acts as a large voltage divider had to be made. This circuit also manages the power up of the screen. To light the backlight of the screen, 28V are required. This is generated by a separate circuit.

The prototype performed the basic functions without problems. It was possible to send data via USB to the FPGA at high data rates, so it wouldn't take too long to load the image. Successful tests have been done at 115.2 Kb/s, but much faster are also possible. However, at this speed the transfer takes only 33 seconds, so going faster isn't really necessary, it will just increase the probability of an error.

The image is shown correctly, which means that the SRAM, SRAM controller and the controller of the screen work fine.

However, there is one minor flaw. It isn't possible to continue to display an image while another is being received, due to limitations of the memory controller. Instead, a preloaded image is displayed, the logo of R:eFlex.

The most important aspect of the entire design is how much effort is required to implement it using SG-TFTs.

The design that is loaded onto an FPGA is programmed in VHDL. An FPGA consists of basic building blocks called logic cells. These logic cells can be made using only cells that are already in the SG Standard Library. From the VHDL code, a transistor level design can be extracted. So the entire design can be constructed by imitating the design as it is working on the FPGA.

SRAM is almost just as simple. One SRAM cell consists of six transistors, so creating SRAM using only SG-TFTs should not pose a problem.

Using the technology behind SG-TFTs, liquid silicon, it is also possible to make analogue hardware. This is how the analogue part of the receiver can be created. The digital part can be made in the same way as the part of the design that was programmed in VHDL.

The largest change to the design is to the controller of the screen. Even though the screen is driven comparable to the front plane display of super E-paper, some signals will have to be added and others will have to be omitted. This will have to be changed in the VHDL design

Sources

- [1] J Kruit M. de Jong, "Design of a wireless data transmission system," TU Delft - faculty of Electrical Engineering, Mathematics and Computer Science, Delft, Bachelor Thesis 2010.
- [2] Chrihern. (2009, Feb.) Wikipedia.org - DRAM. [Online].
http://en.wikipedia.org/wiki/File:Square_array_of_mosfet_cells_read.png
- [3] Christophe Desager. (2009, Oct.) http://nl.wikipedia.org/wiki/Bestand:Schema_SRAM.png. [Online].
http://upload.wikimedia.org/wikipedia/commons/thumb/4/4a/Schema_SRAM.png/120px-Schema_SRAM.png
- [4] Steve Collins. (2008, July) TechTarget. [Online].
http://searchstorage.techtarget.com/sDefinition/0,sid5_gci212130,00.html
- [5] Transcend. Transcend USA. [Online].
http://www.transcendusa.com/support/dlcenter/Datasheet/TSXGUSD-S3%20_DS.pdf
- [6] Tgd2007. www.wikipedia.org.
- [7] Jeff Tyson. (2000, August) HowStuffWorks. [Online]. <http://electronics.howstuffworks.com/flash-memory.htm>
- [8] Logic Cell. [Online]. [onbekend](#)
- [9] Douglas J. Smith. (2003) Angelfire. [Online].
<http://www.angelfire.com/in/rajesh52/verilogvhdl.html>
- [10] (2010, February) FPGA4fun. [Online]. <http://www.fpga4fun.com/FPGAinfo1.html>
- [11] Digilent Inc. (2005, May) Digilent Inc. [Online].
http://www.digilentinc.com/Data/Products/S3BOARD/S3BOARD_RM.pdf
- [12] SparkFun Electronics. [Online].
http://www.sparkfun.com/commerce/product_info.php?products_id=691
- [13] SparkFun Electronics - nRF24L01 Datasheet. [Online].
http://www.sparkfun.com/datasheets/Wireless/Nordic/nRF24L01P_Product_Specification_1_0.pdf
- [14] Octopart.com. [Online]. <http://octopart.com/is61lv6416-10tl-issi-714748>
- [15] SparkFun Electronics. [Online]. <http://www.sparkfun.com/datasheets/LCD/LCD-08335-1.pdf>
- [16] <http://www.4coolday.com>. [Online].

http://www.4coolday.com/images/Sony_PSP_1000/psp_lcd_screen.jpg

- [17] Yue Kuo, *Thin Film Transistors - Materials and Processes*, 2nd ed., Yeu Kuo, Ed. Norwell, United States: Kluwer Academic Publishers, 2004.
- [18] (2009, Dec.) <http://www.ti.com>. [Online]. <http://focus.ti.com/lit/ds/symlink/mc34063a.pdf>
- [19] Wei Man Chim, "A Flexible Electronic Paper with Integrated Display Driver using Single Grain TFT Technology," TU Delft - faculty of Electrical Engineering Mathematics and Computer Science, Delft, MSc Thesis 2009.
- [20] Nitz Saputra et al, "Analog and RF Design Using the u-Czochralski Single Grain TFT Technology," Electronic Research Laboratory Faculty of EEMCS Delft University of Technology, Delft, Abstract.
- [21] Alessandro Baiano, "Single Grain TFTs for High Speed Flexible Electronics," University of Technology Delft, Delft, MSc Thesis 978-90-8570-430-0, 2009.
- [22] W. Fang et al, "Automated Digital Circuits Design Based on Single-Grain Si TFTs Fabricated Through u-Czochralski (Grain Filter) Process," *AM-FPD Digest of Technical papers*, pp. 47-50, 2006.
- [23] Unknown. Electrosofts.com. [Online]. http://electrosofts.com/verilog/mux_circuit.jpg
- [24] A. Gupta M. Trifunovic, TU Delft - Faculty of Electrical Engineering, Mathematics and Computer Science, Delft, BSc Thesis 2010.
- [25] N. Golshani et al, "High Speed 6-Transistor Static Random Access Memory Cells Using Single Grain Thin Film Transistors Fabricated at Low Temperature Process," *Japanese Journal of Applied Physics*, Mar. 2010.
- [26] mosys.com. [Online]. http://www.mosys.com/userfiles/image/1T-SRAM_01.jpg
- [27] Unknown. www.auo.com. [Online]. http://www.auo.com/auoDEV/gfx/tft-intro_lcd-framework.gif
- [28] M. Trifunovic et al, "R:eflex - Business Plan," TU Delft - faculty of Electrical Engineering, Mathematics and Computer Science, Delft, Business Plan 2010.
- [29] (2009, Apr.) <http://www.xilinx.com/>. [Online]. http://www.xilinx.com/support/documentation/data_sheets/ds099.pdf
- [30] (2006, Feb.) www.issi.com. [Online]. <http://www.issi.com/pdf/61LV25616AL.pdf>
- [31] Pong P. Chu, *FPGA prototyping by VHDL examples*, 1st ed. Hoboken, New Jersey, United States: John Wiley & Sons, Inc., 2008.

List of abbreviations

ARM	Advanced RISC Machine
AVR	Advanced Virtual RISC
DCM	Digital Clock Manager
DIMES	Delft Institute of Microsystems and Nanoelectronics
DLL/PLL	Delay Locked Loop / Phase Locked Loop
DRAM	Dynamic Random Access Memory
FPGA	Field Programmable Gate Array
IDE	Integrated Development Environment
LCD	Liquid Crystal Display
LED	Light Emitting Diode
Mux	Multiplexer
PIC	Peripheral Interface Controller
QR-LPD	Quick Respons – Liquid Powder Display
RGB	Red Green Blue
RISC	Reduced Instruction Set Computing
SD card	Secure Digital Card
SG-TFT	Single Grain – Thin Film Transistor
SRAM	Static Random Access Memory
TFT	Thin Film Transistor
VGA	Video Graphics Array
VHDL	Very High Speed Integrated Circuit Hardware Description Language

Appendices

Content

<i>List of Figures</i>	44
<i>Appendix A : Business plan R:eFlex</i>	45
<i>Appendix B : Datasheets</i>	46
<i>Appendix B-1: Field Programmable Gate Array</i>	46
<i>Appendix B-2: Static Random Access Memory</i>	46
<i>Appendix B-3: SHARP LCD screen</i>	49
<i>Appendix B-4: MC34063A</i>	50
<i>Appendix C : Hardware design</i>	51
<i>Appendix C-1: SHARP TFT-LCD level converter</i>	51
<i>Appendix C-2: Step-up converter</i>	51
<i>Appendix C-3: RTL schematic from VHDL</i>	53

List of figures

<i>Figure B1-1: Xilinx Spartan 3-family datasheet page 1</i>	46
<i>FigureB2-1: ISSI RAM memory datasheet page 1</i>	47
<i>FigureB2-2: Write cycle no.2</i>	48
<i>Figure B2-3: Read cycle no. 2</i>	48
<i>Figure B3-1: Timing information for startup and power down.</i>	49
<i>Figure B3-2: Timing information regarding horizontal synchronization</i>	49
<i>Figure B3-3: Timing information regarding vertical synchronization</i>	49
<i>Figure B4-1: Texas Instruments MC34063A data sheet page 1</i>	50
<i>Figure C1-1: Schematics of the SHARP LCD level converter</i>	51
<i>Figure C2-1: Test setup of step-up converter</i>	52
<i>Figure C3-5: Synthesized VHLD schematic of "SRAM control"</i>	53
<i>Figure C3-4: Synthesized VHLD schematic of "Data Storage"</i>	53
<i>Figure C3-3: Synthesized VHLD schematic of "Decoder"</i>	54
<i>Figure C3-2: Synthesized VHLD schematic of "Clockgen"</i>	54
<i>Figure C3-1: Synthesized VHLD schematic of "total"</i>	54

Appendix A : Business plan R:eFlex

This section has been copied from [28]. It explains why we are making a digital billboard using super E-paper.

E-poster

“The E-poster can be used to replace paper posters in bus shelters and billboards. The E-poster is a large E-paper, optimized for advertisement. It can display a very clear image in full colour and will be available in multiple sizes. The smallest standard size will be A3, the largest will be 2m². Because the dimensions of these posters are the same as the current paper posters, it will be easy to replace these with e-posters. If a customer wants a poster with larger dimensions it will also be possible to overlap smaller poster in order to get a larger one. This way, we can manufacture posters the size of large 8m² billboards.

Furthermore, the E-poster can display a slide-show, showing multiple images one after another. This way, the customer can show a different image every few seconds, just like the boarding in soccer stadiums. High-end E-posters will even be able to show moving images, so animations are also possible.

The E-poster is a solid product and can endure punches and kicks because of its flexible technology. Due to this property, the E-poster is resistant to vandals who try to damage the poster. These properties make the E-poster ideal for advertisement companies.”

Appendix B : Datasheets

Appendix B-1: Field Programmable Gate Array

Information in this appendix has been summarized from the datasheet of the xc3s200 FPGA [29], in order to highlight the most important information. It is an overview of the FPGA we have used as our main controller in the test setup.



DS099-1 (v1.1) April 24, 2003

Spartan-3 1.2V FPGA Family: Introduction and Ordering Information

Advance Product Specification

Introduction

The 1.2V Spartan™-3 family of Field-Programmable Gate Arrays is specifically designed to meet the needs of high volume, cost-sensitive consumer electronic applications. The eight-member family offers densities ranging from 50,000 to five million system gates, as shown in Table 1.

The Spartan-3 family builds on the success of the earlier Spartan-IIE family by increasing the amount of logic resources, the capacity of internal RAM, the total number of I/Os, and the overall level of performance as well as by improving clock management functions. Numerous enhancements derive from state-of-the-art Virtex™-II technology. These Spartan-3 enhancements, combined with advanced process technology, deliver more functionality and bandwidth per dollar than was previously possible, setting new standards in the programmable logic industry.

Because of their exceptionally low cost, Spartan-3 FPGAs are ideally suited to a wide range of consumer electronics applications, including broadband access, home networking, display/projection and digital television equipment.

The Spartan-3 family is a superior alternative to mask programmed ASICs. FPGAs avoid the high initial cost, the lengthy development cycles, and the inherent inflexibility of conventional ASICs. Also, FPGA programmability permits design upgrades in the field with no hardware replacement necessary, an impossibility with ASICs.

- Densities as high as 74,880 logic cells
- 326 MHz system clock rate
- Three separate power supplies for the core (1.2V), I/Os (1.2V to 3.3V), and special functions (2.5V)
- ✦ SelectIO™ signaling
 - Up to 784 I/O pins
 - 622 Mb/s data transfer rate per I/O
 - Seventeen single-ended signal standards
 - Six differential signal standards including LVDS
 - Termination by Digitally Controlled Impedance
 - Signal swing ranging from 1.14V to 3.45V
 - Double Data Rate (DDR) support
- ✦ Logic resources
 - Abundant, flexible logic cells with registers
 - Wide multiplexers
 - Fast look-ahead carry logic
 - Dedicated 18 x 18 multipliers
 - JTAG logic compatible with IEEE 1149.1/1532 standards
- ✦ SelectRAM™ hierarchical memory
 - Up to 1,872 Kbits of total block RAM
 - Up to 520 Kbits of total distributed RAM
- ✦ Digital Clock Manager (up to four DCMs)
 - Clock skew elimination
 - Frequency synthesis
 - High resolution phase shifting
- ✦ Eight global clock lines and abundant routing
- ✦ Fully supported by Xilinx ISE development system
 - Synthesis, mapping, placement and routing

Features

- ✦ Revolutionary 90-nanometer process technology
- ✦ Very low cost, high-performance logic solution for high-volume, consumer-oriented applications

Table 1: Summary of Spartan-3 FPGA Attributes

Device	System Gates	Logic Cells	CLB Array (One CLB = Four Slices)			Distributed RAM (bits ¹)	Block RAM (bits ¹)	Dedicated Multipliers	DCMs	Maximum User I/O	Maximum Differential I/O Pairs
			Rows	Columns	Total CLBs						
XC3S50	50K	1,728	16	12	192	12K	72K	4	2	124	56
XC3S200	200K	4,320	24	20	480	30K	216K	12	4	173	76
XC3S400	400K	8,064	32	28	896	56K	288K	16	4	264	116
XC3S1000	1M	17,280	48	40	1,920	120K	432K	24	4	391	175
XC3S1500	1.5M	29,952	64	52	3,328	208K	576K	32	4	487	221
XC3S2000	2M	46,080	80	64	5,120	320K	720K	40	4	565	270
XC3S4000	4M	82,208	96	72	6,912	432K	1,728K	96	4	712	312
XC3S5000	5M	74,880	104	80	8,320	520K	1,872K	104	4	784	344

Notes:
1. By convention, one Kb is equivalent to 1,024 bits.

Figure B1-1: Xilinx Spartan 3-family datasheet page 1

Appendix B-2: Static Random Access Memory

Important information on the SRAM can be found in this section, for further details on this device please consult the datasheet [30].

IS61LV25616AL

256K x 16 HIGH SPEED ASYNCHRONOUS CMOS STATIC RAM WITH 3.3V SUPPLY

ISSI®

FEBRUARY 2006

FEATURES

- High-speed access time:
 - 10, 12 ns
- CMOS low power operation
- Low stand-by power:
 - Less than 5 mA (typ.) CMOS stand-by
- TTL compatible interface levels
- Single 3.3V power supply
- Fully static operation: no clock or refresh required
- Three state outputs
- Data control for upper and lower bytes
- Industrial temperature available
- Lead-free available

DESCRIPTION

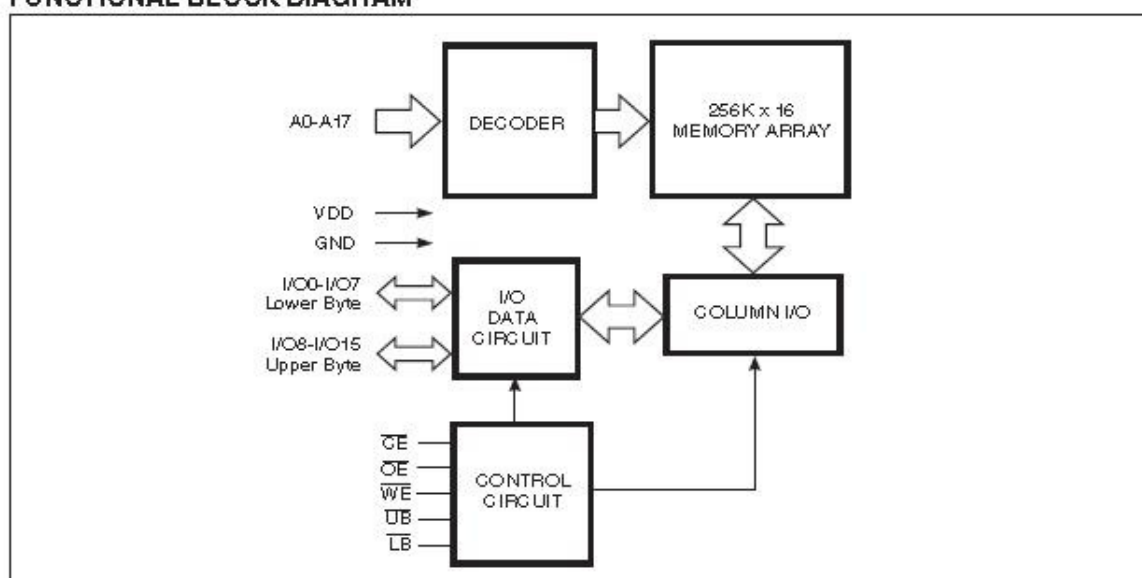
The ISSI IS61LV25616AL is a high-speed, 4,194,304-bit static RAM organized as 262,144 words by 16 bits. It is fabricated using ISSI's high-performance CMOS technology. This highly reliable process coupled with innovative circuit design techniques, yields high-performance and low power consumption devices.

When $\overline{CE}$ is HIGH (deselected), the device assumes a standby mode at which the power dissipation can be reduced down with CMOS input levels.

Easy memory expansion is provided by using Chip Enable and Output Enable inputs, $\overline{CE}$ and $\overline{OE}$. The active LOW Write Enable ($\overline{WE}$) controls both writing and reading of the memory. A data byte allows Upper Byte ($\overline{UB}$) and Lower Byte ($\overline{LB}$) access.

The IS61LV25616AL is packaged in the JEDEC standard 44-pin 400-mil SOJ, 44-pin TSOP Type II, 44-pin LQFP and 48-pin Mini BGA (8mm x 10mm).

FUNCTIONAL BLOCK DIAGRAM

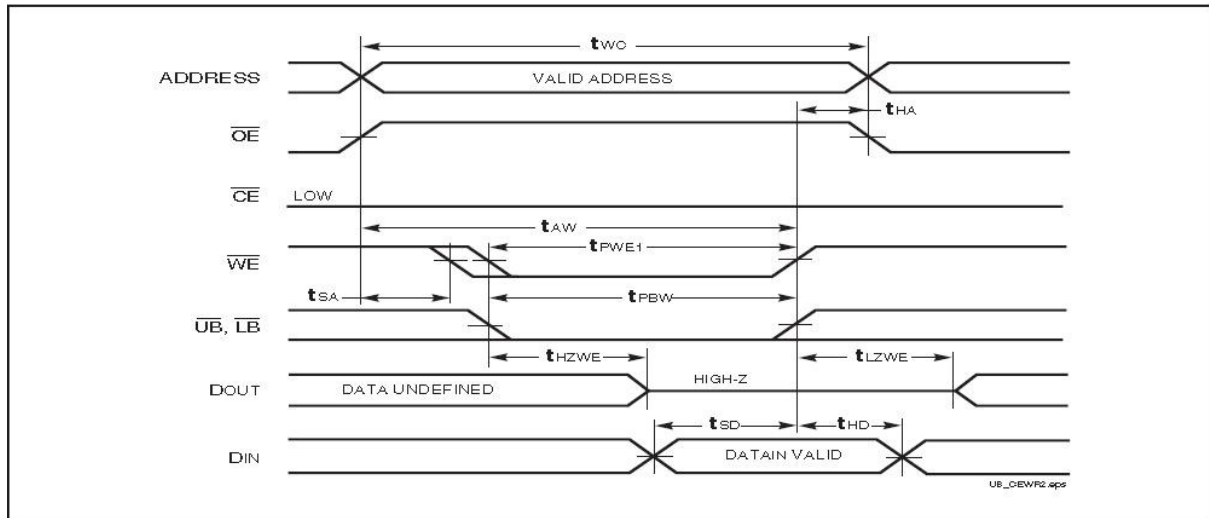


Copyright © 2006 Integrated Silicon Solution, Inc. All rights reserved. ISSI reserves the right to make changes to this specification and its products at any time without notice. ISSI assumes no liability arising out of the application or use of any information, products or services described herein. Customers are advised to obtain the latest version of this device specification before relying on any published information and before placing orders for products.

FigureB2-1: ISSI RAM memory datasheet page 1

Timing details for a write cycle to the SRAM

This is the timing diagram we have used to control a write operation on the SRAM. As can be seen, the chip is always active.



FigureB2-2: Write cycle no.2

Timing details for a read cycle from the SRAM

This is the timing diagram we have used to control a read operation on the SRAM, except we allowed the chip to be active all the time. Also we always wanted the entire 16-bit packet, so lb and ub were also low all the time.

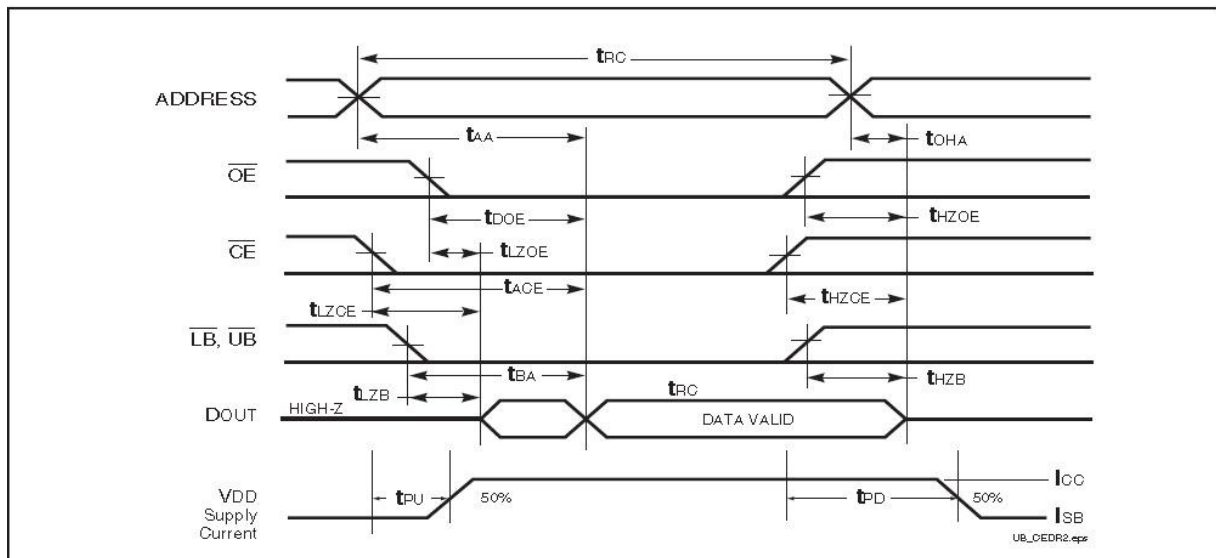


Figure B2-3: Read cycle no. 2

Appendix B-3: SHARP LCD screen

Most important information relevant for this thesis has been put in this appendix, other specifications can be found in the complete datasheet [15]. At startup, certain signal have to be asserted in a certain order. First VCC, then AVDD and finally the other signals. At power down, this order has been switched around.

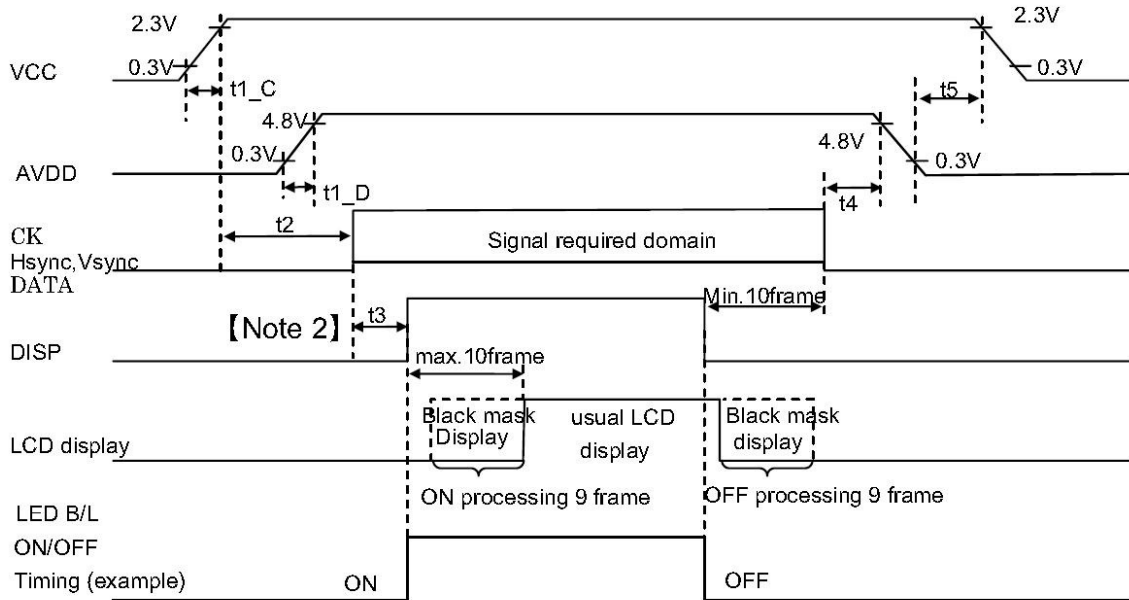


Figure B3-1: Timing information for startup and power down.

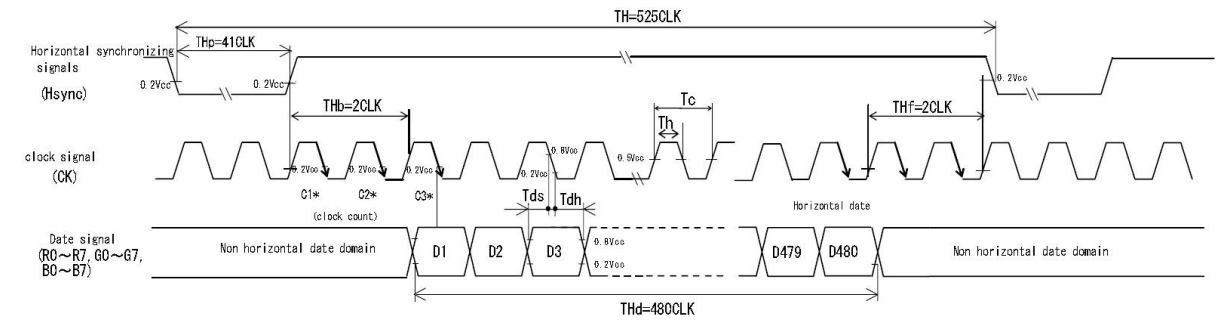


Figure B3-2: Timing information regarding horizontal synchronization

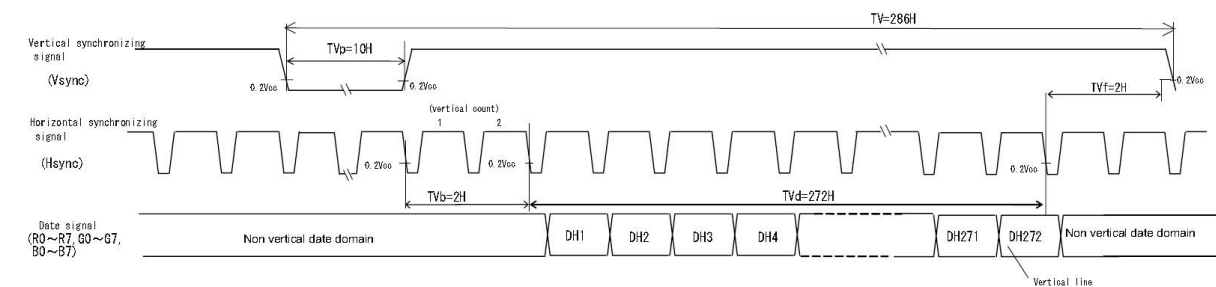


Fig 2 Input signal timing

Figure B3-3: Timing information regarding vertical synchronization

Appendix B-4: MC34063A

The general description, important for this thesis can be found in this appendix, further details on this device can be found in the datasheet [18]. This chip has been used to boost a 5V signal to a 28V signal, to power the backlight.



www.ti.com

SLLS636L – DECEMBER 2004 – REVISED DECEMBER 2009

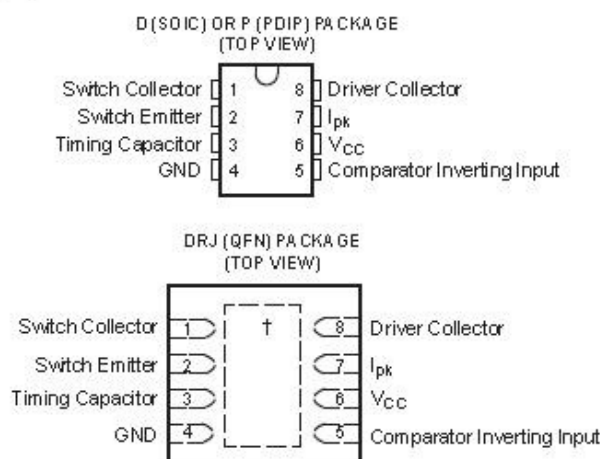
MC33063A
MC34063A

1.5-A PEAK BOOST/BUCK/INVERTING SWITCHING REGULATORS

Check for Samples: MC33063A MC34063A

FEATURES

- Wide Input Voltage Range: 3 V to 40 V
- High Output Switch Current: Up to 1.5 A
- Adjustable Output Voltage
- Oscillator Frequency Up to 100 kHz
- Precision Internal Reference: 2%
- Short-Circuit Current Limiting
- Low Standby Current



† The exposed thermal pad is electrically bonded internally to pin 4 (GND).

DESCRIPTION/ORDERING INFORMATION

The MC33063A and MC34063A are easy-to-use ICs containing all the primary circuitry needed for building simple dc-dc converters. These devices primarily consist of an internal temperature-compensated reference, a comparator, an oscillator, a PWM controller with active current limiting, a driver, and a high-current output switch. Thus, the devices require minimal external components to build converters in the boost, buck, and inverting topologies.

The MC33063A is characterized for operation from -40°C to 85°C , while the MC34063A is characterized for operation from 0°C to 70°C .

Figure B4-1: Texas Instruments MC34063A data sheet page 1

Appendix C : Hardware design

Appendix C-1: SHARP TFT-LCD level converter

Design of the level converter has been done in eagle; a layout and schematic editor. Schematic is shown beneath.

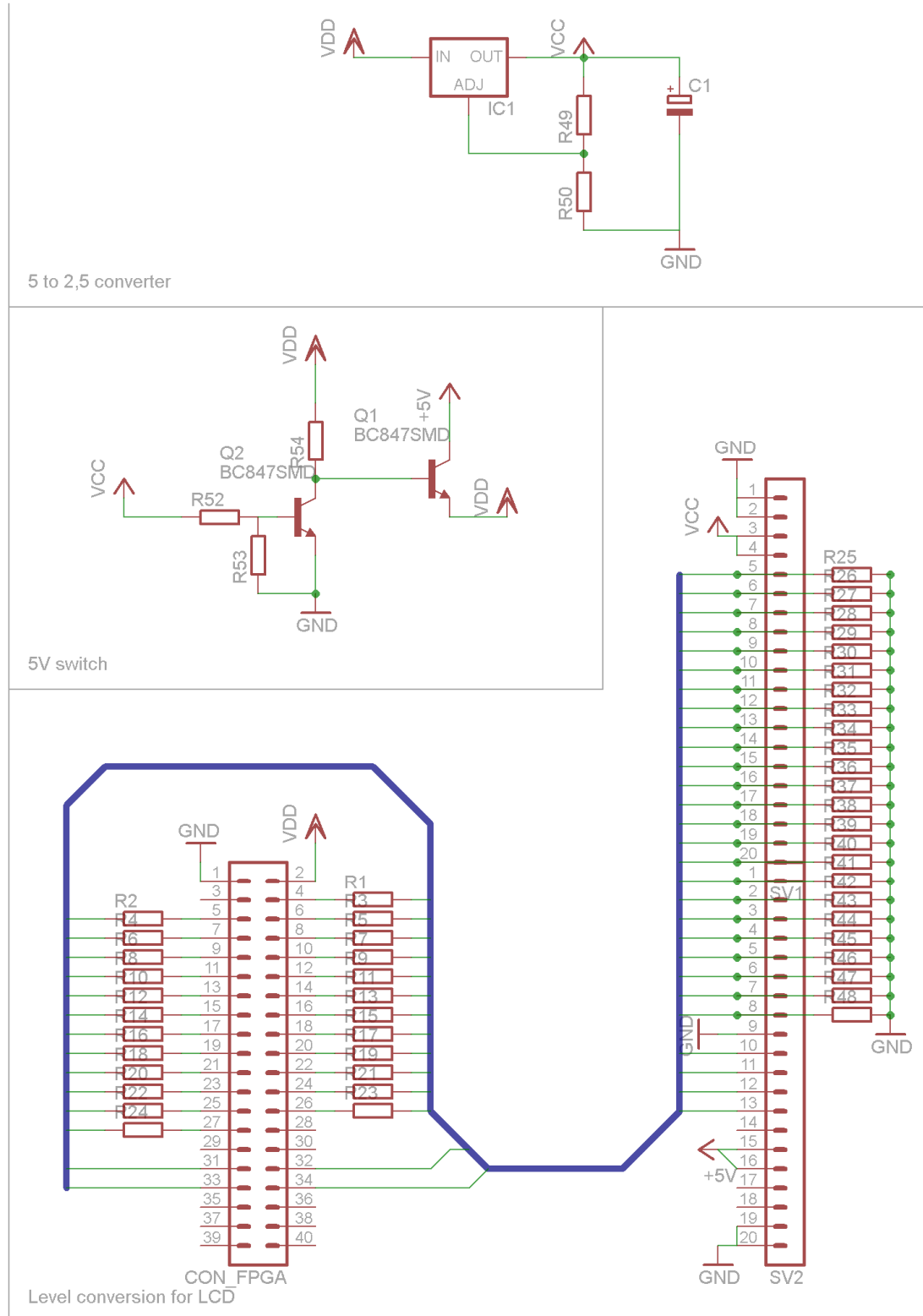


Figure C1-1: Schematics of the SHARP LCD level converter

Appendix C-2: Step-up converter

Design of the step-up converter has been explained in chapter 6.6.1, a figure of the test configuration is shown below. The chip described in Appendix B-4 is in the middle.

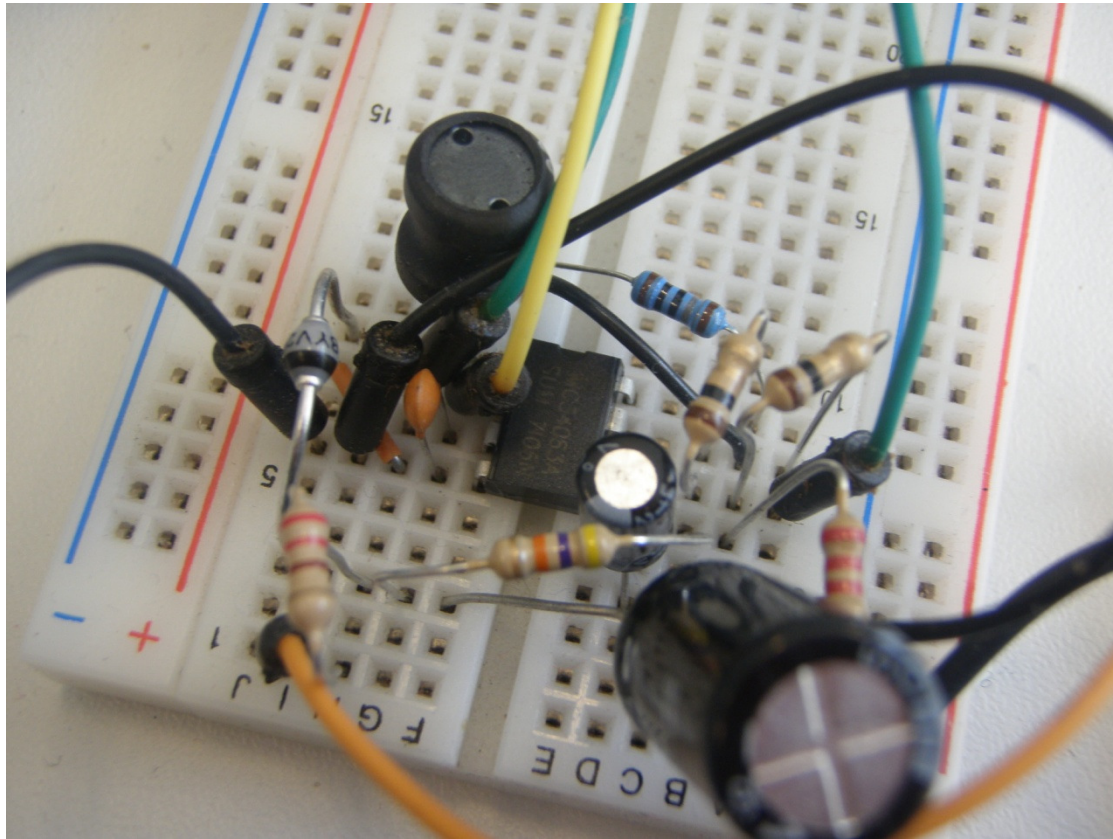


Figure C2-1: Test setup of step-up converter

Appendix C-3: RTL schematic from VHDL

Appendix C-3: Total

An overview of all the major components of the design. Clockgen, Receiverblock (decoder), DataStorage and ScreenControl are all present.

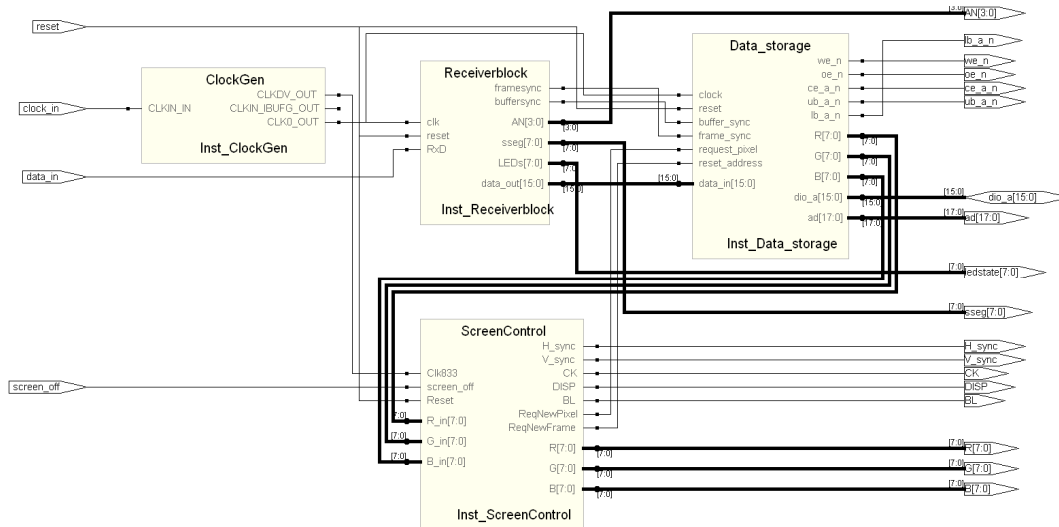


Figure C3-1: Synthesized VHDL schematic of "total"

Appendix C-3: Clockgen

Overview of Clockgen. It's basically a DCM with a few input and output buffers.

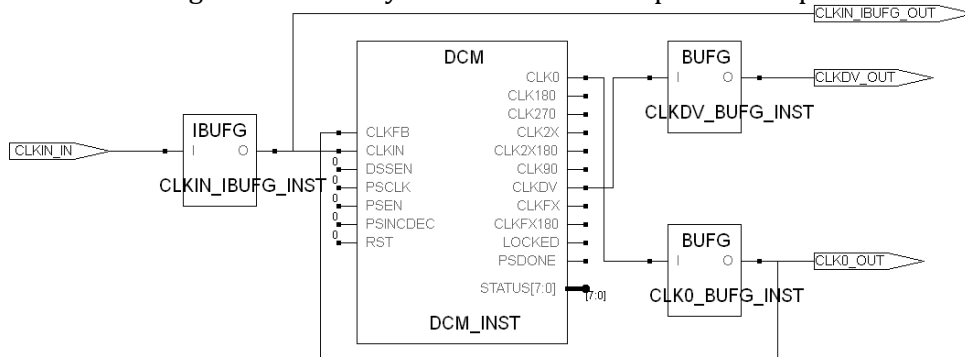


Figure C3-2: Synthesized VHDL schematic of "Clockgen"

Appendix C-3: Receiverblock (decoder)

Renamed *Decoder*, it decodes the data received by the receiver. Disp_hex_mux drives the seven-segment display.

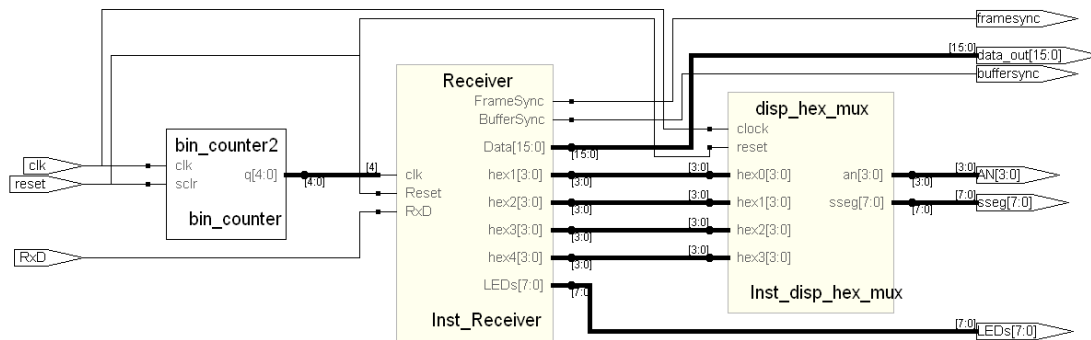


Figure C3-3: Synthesized VHDL schematic of "Decoder"

Appendix C-3: Data storage

Data Storage consists of two large components. SRAM control interfaces with the SRAM, while Pixelbuffer preloads pixels to send to the screencontrol.

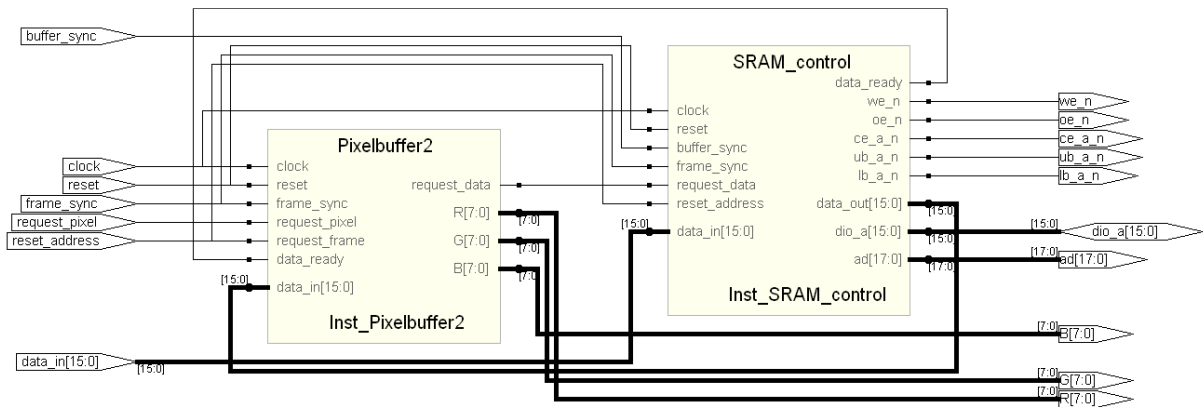


Figure C3-4: Synthesized VHDL schematic of "Data Storage"

Appendix C-3: SRAM control

SRAM control itself consists of two individual parts. SRAM_ctrl is directly connected to the SRAM, while SRAM_toplayer interfaces with the rest of the circuit.

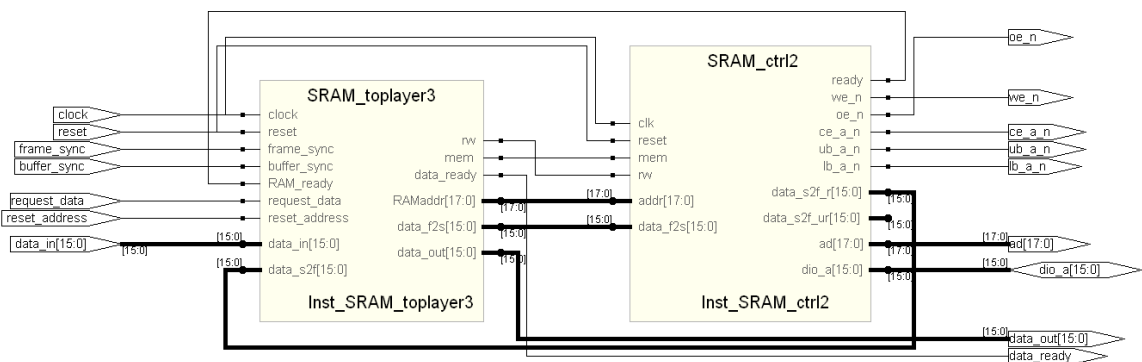


Figure C3-5: Synthesized VHDL schematic of "SRAM control"