Preface

This degree project (15 hp) has been performed during the autumn 2011 at the embedded software group at the company CrossControl in Alfta, Sweden. It is considering arrival control of TFT displays, regarding pixel defects.

The project consisted of different types of work: analyze of technical data for different display models, realization of test equipment, programming and investigation of standards. The project has been varying, including contacts with a lot of people.

I want to thank the people at CrossControl for a welcoming attitude and good cooperation, which have made this project easier to perform. Especially I want to thank the following persons for important contribution to this thesis, employees at CrossControl if nothing else mentioned: First of all my supervisor Jens Rubensson for great support, Supervisor at University of Gävle: Niklas Rothpfeffer, Advisory: Olov Hisved, Backlight theory: Roland Andersson (CCFL), Daniel Sjödin (CCFL and LED), Video signals theory: Joakim Bergqvist - consultant JB Elektronik Gävle, Information about display models to test: Lars Olsson, Supply of standard and displays: Stefan Hallgren, Supply of displays: Per Oskar Andersson, Bo Kvick, Supply and modification of test equipment: Alf Luong, Anton Nordenstam, Magnus PE Olsson, Göran Sandström, Information regarding pixel quality at arrival: Hans Formgren, Daniel Lindberg, Support with the article search system at CrossControl: Magnus Olsson, General information about the company: Berith Malmström, Johan Strandberg, IS/IT: Bo Larsson.

This thesis is published with permission from CrossControl.
Abstract

Displays to be used in display computers for vehicles and machines in critical environments shall be possible to inspect regarding pixel defects. The main part of the goal was to realize a test equipment so that different colours can be visualized on displays of different resolutions and interfaces, with a minimum startup time. The inspection of the displays will be visual.

One pixel is consisting of three sub-pixels: red, green and blue, each controlled by a Thin Film Transistor (TFT). If some TFTs are broken, different types of pixel defects occur. There is an ISO standard defining classes for how many defects of each type that can exist on a display before it should be replaced by the supplier. But other limits can be agreed between supplier and customer. To be able to see the different types of pixel faults, 5 different colours should be shown on the display: red, green, blue, black and white.

A list was supplied containing 10 different models of display elements for which tests should be possible. They were thoroughly analyzed regarding their technical data for resolution and interfaces for backlight and video signals. The displays are of 3 different resolutions. 5 displays have backlight of the older technology Cold Cathode Fluorescent Lamps (CCFL) which means neon light from tubes, while 5 have the later technology Light Emitting Diodes (LED). 2 of the displays receive the video signals in parallel, while 8 receive them via Low Voltage Differential Signaling (LVDS). The LVDS connector showed to have a special pin configuration for 2 of the 8 LVDS displays. This was the most important discovery, because if the standard LVDS cable would have been used from the carrier board, the displays would probably have been damaged. Because of these differences different types of boards and cables had to be used, both standard parts and modified, to be able to supply the different display models with backlight and video signals.

To achieve the main part of the goal an existing display computer was modified and used as the base platform for a prototype test equipment. All signaling to the display had to be generated by the FPGA instead of the CPU module. The FPGA project was written in VHDL language. The project included six different modules of which some were written from scratch, some were reused and some were partly reused from an already existing FPGA project. Display resolution is set with a jumper and the 5 colours are visualized in a loop, using a push button.
# Table of contents

Preface ........................................................................................................................................................................ i
Abstract ........................................................................................................................................................................... ii
Table of contents .............................................................................................................................................................. iii

1 Introduction ........................................................................................................................................................................ 1

2 Theory ............................................................................................................................................................................... 3
  2.1 Pixel defects ................................................................................................................................................................. 3
    2.1.1 Basic structure of a TFT-LCD ................................................................................................................................ 3
    2.1.2 Different types of pixel defects ................................................................................................................................ 4
    2.1.3 Standards ................................................................................................................................................................. 5
    2.1.4 Optical inspection ..................................................................................................................................................... 7
  2.2 Backlight ......................................................................................................................................................................... 8
    2.2.1 CCFL ......................................................................................................................................................................... 8
    2.2.2 LED ........................................................................................................................................................................... 8
  2.3 Video signals ................................................................................................................................................................. 9
    2.3.1 RGB data ................................................................................................................................................................. 9
    2.3.2 Timing signals .......................................................................................................................................................... 9
    2.3.3 Pixel clock ............................................................................................................................................................... 9
    2.3.4 Transmission type ................................................................................................................................................... 10

3 Process and results ............................................................................................................................................................ 11
  3.1 Structure of the project work .......................................................................................................................................... 11
  3.2 Collection and analyze of display models ........................................................................................................................... 12
    3.2.1 Analyze of resolution ................................................................................................................................................ 13
    3.2.2 Analyze of backlight supply .................................................................................................................................. 14
    3.2.3 Analyze of video signals supply ................................................................................................................................ 15
  3.3 Programming of FPGA .................................................................................................................................................. 16
    3.3.1 Basic hardware and preparations ............................................................................................................................. 16
1 Introduction

This degree project is performed in Alfta, Sweden, at the company CrossControl, formerly known as CC Systems (Cross Country Systems).

The business area of CrossControl mainly consists of development, production and service of systems for control of vehicles and machines in critical environments (e.g. forestry, railway, mining, cargo handling). The systems can also be used for diagnosis and prognosis. The company was founded in 1991 and has its roots in John Deere’s development and manufacturing of forestry machines. Today, the worldwide customer base are spread over several different industrial branches, thereby the changing of company name.

Some departments located at the head office in Alfta are hardware development, production, sales and administration. There are also offices in Västerås, Uppsala, Finland and Malaysia. The number of employees at Oct. 2011 was 145 in Alfta and appr.240 totally.

The control system products are divided in groups:

- Display computers
- Main controllers
- I/O controllers and devices
- Infrastructure components

This degree project is performed at the embedded software group within hardware development and is concerning TFT displays for the display computers, see Fig. 1.

![Display computers](image)

Fig.1.CrossControl product group Display computers [1]. Published with permission of CrossControl.

The display computers are of 4 different models with different performance, and each model exists in different sizes. Therefore the TFT displays used in them also are of many models and sizes.
From the suppliers of TFT displays CrossControl sometimes get displays which are not fulfilling the quality demands concerning faulty pixels. This is a problem while these faults often are found when the product is already mounted, or even worse, at the customer.

By making an arrival control of the displays these faults can be discovered in a much earlier state. The main goal of the degree project was to develop a test equipment to make it possible to quickly check and verify pixel faults at various types of displays.

At start of the project, the task was divided into following parts:

- Adjustment of VHDL code for a FPGA to generate data for different colours on displays with different resolutions and interfaces. From the different colours it can be evaluated if pixel faults exist.
- Realization of the test equipment needed to test different types of displays. (The construction of a fixture with the test equipment mounted will not be done within the degree project.)
- Investigation of standards regarding pixel faults.
2 Theory

2.1 Pixel defects

2.1.1 Basic structure of a TFT-LCD

A liquid crystal is a fluid substance with some properties similar to a solid crystal [2]. Such a property is an ordered arrangement of its molecules, which because of long shape and electrical properties tend to line up in parallel rows. By putting a light source behind a double glass panel, with a liquid crystal between the glasses, it is possible by adjusting a voltage between the glasses, to change the molecule arrangement and consequently also the light passing through.

In a liquid crystal display (LCD) the panels are divided into small dots called sub-pixels, where the word pixel is an abbreviation of picture element. The sub-pixels have different colours on one glass panel: 1/3 of them each are red, green or blue. The different sub-pixels are evenly spread over the panel in groups of 3, one of each colour. Such a group forms a pixel. The more pixels a display is divided in, the higher resolution which gives sharper image.

The sub-pixel colours red, green and blue are the fundamental colours needed for producing any type of colour. So, by regulating the amount of background light passing through the 3 types of sub-pixels, different colours and images can be viewed on the display.

There are transparent electrodes in the two glass panels, horizontally positioned in one panel and vertically in the other. By applying a voltage between a specific row electrode and a column electrode a specific sub-pixel can be controlled. A thin-film transistor (TFT) holds the voltage over the sub-pixel. This technology is called TFT active matrix because every sub-pixel can be addressed by its position of row and column in a matrix.
2.1.2 Different types of pixel defects

Considering the fact that every sub-pixel is controlled by its own TFT it is easily understandable that the amount of TFTs in a display will be very high. For example, the SVGA resolution consisting of $800 \times 600$ pixels corresponds to $3 \times 800 \times 600 = 1.44$ million TFTs. Therefore defect pixels sometimes occur during the manufacturing process.

There are 3 basic types of pixel defects, defined in the following way [2]:

- **Type 1**: Stuck high pixel means all 3 sub-pixels are constantly lit, which causes white pixel colour.
- **Type 2**: Stuck low pixel means all 3 sub-pixels are constantly turned off, which causes black pixel colour.
- **Type 3**: Defect sub-pixel means 1 sub-pixel (red, green or blue) is constantly stuck high (lit) or stuck low (turned off).

The term cluster is regarding pixels defined as an area of $5 \times 5$ pixels. Cluster pixel defects mean the fault types above within a cluster, and naturally they are easier to see than more spread-out sub-pixel faults.

Stuck high pixels are often more annoying than stuck low. Green stuck high sub-pixels are easier detected by the human eye than red or blue, because they appear brighter. See Fig. 2 for a display with that kind of fault.

![Image](image.png)

*Fig. 2. Pixel defect of type 3: Stuck high green sub-pixel at black colour (looks blue in photo).*
2.1.3 Standards

If every one of the TFTs in all displays delivered from a production line shall work correctly it would be very expensive to produce TFT displays. A lot of the displays would have to be scrapped, because TFTs are not possible to repair.

The International Organization for Standardization, ISO, provides international Standards for Business, Government and Society. To regulate the acceptability of pixel defects and to protect the end user, ISO has created standards that display manufacturers are recommended to follow. ISO recommends how many pixel faults that are acceptable before a display should be replaced.

An older now invalid standard ISO 13406-2 has been replaced by 4 parts within the standard ISO 9241 Ergonomics of human-system interaction. These standards are not available for free and each copy is protected by an End user license. Therefore no contents from the standards can be showed in this thesis, except for the following brief descriptions, provided at each web page where the standards can be ordered. The following descriptions are copied unchanged from [3]-[6], but the markings in bold are made afterwards.

- Part 302 Terminology for electronic visual displays
  This part of ISO 9241 provides a comprehensive terminology for electronic visual displays and explains the terms and definitions used in the other parts of ISO 9241.

- Part 303 Requirements for electronic visual displays
  This part of ISO 9241 establishes image-quality requirements, as well as providing guidelines, for electronic visual displays. These are given in the form of generic — independent of technology, task and environment — performance specifications and recommendations that will ensure effective and comfortable viewing conditions for users with normal or adjusted-to-normal eyesight. This part of ISO 9241 does not address issues of accessibility for people with disabilities. However, it does take into account aspects of the eyesight of older people and could be of value to people dealing with issues of visual impairment in certain cases: the specification of essential characteristics for normal viewing can be used to gauge the severity of different visual abnormalities so that appropriate solutions can be identified. NOTE In addition to the Bibliography, Annex F gives a selected bibliography of documents addressing the needs of people with disabilities, including people with poor, deteriorating or no eyesight.
- Part 305  Optical laboratory test methods for electronic visual displays

This part of ISO 9241 establishes optical test and expert observation methods for use in predicting the performance of a display vis-à-vis the ergonomics requirements given in ISO 9241-303. Note: After contacting Swedish Standards Institute (SIS), the information was given that this standard is going to be released in an updated (concerning LCD and LED) version, probably during 2012.

- Part 307  Analysis and compliance test methods for electronic visual displays

This part of ISO 9241 establishes test methods for the analysis of a variety of visual display technologies, tasks and environments. It uses the measurement procedures of ISO 9241-305 and the generic requirements of ISO 9241-303 to define compliance routes suitable for the different technologies and intended context of use. Information from SIS is that discussions about updating are being held. Eventually, limit values will be put in another standard.

The last part ISO 9241-307:2008 [7] is available at CrossControl. At [6] can be seen if the standard has been updated. It is concerning displays for indoor use. After contacting SIS the information was given that the standard (except for the irrelevant chapter 5.5) includes Industrial and Automotive environments: cars, trains and other vehicles. Automotive environments are highly relevant for CrossControl.

The applicable chapter for TFT displays is 5.2 Emissive flat-panel LCD at [7]:38-89. The important part regarding pixel defects is table 63 - Pixel fault classification, at [7]:64-65 (the same identical table and comments are shown also in the irrelevant chapters 5.3 to 5.5). In table 63 pixel fault classes are defined from different numbers of the 3 pixel fault types described in 2.1.2.

Note: Other limits of pixel defects than in the standard can be agreed between the display supplier and the customer.
2.1.4 Optical inspection

The inspection will be visual by human eye.

To be able to see a faulty sub-pixel, the colour of the surrounding pixels must have a certain contrast compared to the colour emitted from the faulty sub-pixel.

After searching for guidelines for what colours are needed to show on the display during the inspection, some sources say that just black and white is enough [8]:

- Black: Stuck high pixel (Type 1) and stuck high sub-pixel (variant of Type 3) can be seen.
- White: Stuck low pixel (Type 2) and stuck low sub-pixel (variant of Type 3) can be seen.

But most of the found sources also recommend showing red, green and blue [9]:

- Red: Stuck low red sub-pixel (variant of Type 3) can be seen.
- Green: Stuck low green sub-pixel (variant of Type 3) can be seen.
- Blue: Stuck low blue sub-pixel (variant of Type 3) can be seen.

Based on the facts above, this project will include all 5 colours.

The following circumstances must be correct during the inspection:

- Updating frequency and resolution of the video signals
- Ambient light
- Viewing distance
- Viewing angle

In this thesis the last 3 points are not given any special focus. It should be sufficient if the inspections are made at normal ambient light and distance, straight in front of the display.
2.2 Backlight

The background light, called backlight, is spread evenly across the screen by passing through a plastic layer. There is two types of backlight.

2.2.1 CCFL

Cold Cathode Fluorescent Lamps

This is neon light from fluorescent tubes, which unless the size are similar to those used in office lighting. An inverter is needed to produce the high lamp starting voltage: 12 V DC use to be the input voltage to the inverter, different levels of AC the output. Generally speaking, the longer tubes (larger displays), the higher voltage out from the inverter is needed for start and operation. This is an old technique with a decreasing production.

2.2.2 LED

Light Emitting Diodes

This is a later technique with several advantages compared to CCFL:

- Less energy consuming
- Longer life
- Less complex design
- No high voltage needed at start
- Better control of light direction
- Better control of light colour
- Less or no bad chemicals
- No flickering light (DC instead of AC)

The disadvantage with LED has been the weak light intensity, but it is getting stronger and LED is therefore more and more replacing CCFL.
2.3 Video signals

Standard refresh rate for TFT displays is 60 Hz. It is the number of new images, also called frames, displayed every second. All the displays in this project have this standard value, but they have different resolutions and interfaces. That puts a demand on the video signals to be flexible.

2.3.1 RGB data

The RGB colour model is based on the fact that every pixel in a display is consisting of 3 sub-pixels: R=Red, G=Green, B=Blue. The brightness of every sub-pixel is controlled by a number of bits:

- 6 bits makes \(3 \times 6 = 18\) bits RGB data (for every pixel).
- 8 bits makes \(3 \times 8 = 24\) bits RGB data, with the result of larger colour depth

The RGB data 0,0,0 corresponds to all 3 sub-pixels being switched off, and the pixel will be black. In the case of 24 bits RGB data it is 255, 255, 255 that corresponds to all 3 sub-pixels having maximal brightness, and then the pixel will be white. The RGB data in between corresponds to different colours.

2.3.2 Timing signals

3 types of timing signals control the visibility of the RGB data on the display:

- **hsync**
  Horizontal Synchronization
  A negative transition (falling edge) of the hsync signal will cause a change of line. The hsync signal shall stay low for a predefined number of pixel clock cycles depending on the actual resolution.
- **vsync**
  Vertical Synchronization
  A negative transition of the vsync signal will start the drawing of a new frame (image) from upper left corner of the display. The frame includes all lines. The vsync signal shall stay low for a predefined number of pixel clock cycles depending on the actual resolution.
- **de**
  Data Enable
  When “de” is high the RGB data will be visible on the display. It is low when hsync or vsync is low and during their corresponding so called porch time (seen in Table 4 in subchapter 3.3.2).

2.3.3 Pixel clock

The so called pixel clock controls the transmission rate of both the RGB data and the timing signals. The higher resolution of the display, the faster pixel clock is needed to create the frame rate 60 Hz.
2.3.4 Transmission type

A TFT display receives the RGB data and the 3 timing signals either in parallel or via LVDS.

- **Parallel**
  All bits have their own lines directly connected from transmitter to receiver.

- **LVDS = Low Voltage Differential Signaling**
  2 parallel lines are connected from the transmitter to the receiver, in this case the TFT display [10]. At the receiver, the two wire ends are connected by a resistor, typically 100 Ohm, and they are also connected to the positive respective negative inputs of an operational amplifier. The transmitter generates a current, typically 3.5 mA, which flows towards the receiver in one line and back in the other. Changing the current direction also changes the voltage polarity over the resistor. The polarity represents a logic 0 or 1, see Fig. 3 where $V_+$ and $V_-$ are the voltages at the inputs of the operational amplifier.

\[
\begin{array}{c}
V_+ \\
1 \\
0 \\
1 \\
V_-
\end{array}
\]

*Fig. 3. Differential signaling [10].*

LVDS has several advantages that has made it very popular in computer high-speed networks and buses:

- The facts that the 2 lines are close to each other gives very strong noise immunity.
- Constant current flow and opposite current directions in the 2 lines minimizes switching spikes and electromagnetic induction.
- The signaling can be made at very high speeds over inexpensive twisted-pair copper cables.

For the displays using LVDS, the RGB-bits and the 3 timing signals are transmitted in 3 or 4 parallel data links.

- 18 bits RGB data: The RGB bits and the 3 timing signals will be sent in the first 3 links 0-2, that is 7 bits in each link. Link 3 is not used.
- 24 bits RGB data: As 18 bits with the difference that the extra 6 bits are sent in link 3.

To handle the bit timing, a clock with the frequency $7 \times \text{pixel clock}$ is needed: The LVDS pixel clock is asymmetric; has high level $4/7$ of the period time and low level $3/7$, so that clock state change shall not occur in the middle of a bit.
3 Process and results

3.1 Structure of the project work

The project was divided in five main parts, which generally have been performed in the following order, but sometimes in parallel:

- Studying of theory regarding TFT displays
- Collection and analyze of display models
- Programming of FPGA
- Realizing of special test equipment hardware
- Investigation of pixel quality standards

The programming of the FPGA was made at an early state, because it was considered as the part with the highest risk for problems. Display models of different resolutions were needed to verify the complete VHDL project.

The content of the pixel quality standards was not considered to affect the software and hardware parts. Thereby the investigation of standards was performed in the end of the project.

The results of the first and the last part are already covered in chapter 2.
The other results are exposed here in chapter 3.
3.2 Collection and analyze of display models

The mission of this part was to get the different display models and the corresponding hardware needed for a later verifying of the VHDL project.

A list was supplied, containing 10 display models for which pixel control shall be done at arrival. One unit of every display model was collected. At first hand borrowed, scrapped or by other reasons leftover units (for other qualities than needed here) were used. 4 displays had to be collected as new parts.

All the data sheets for the displays are available at the article search system at CrossControl [11]. They were used for making an overview in App. E of the relevant technical data for the displays. The displays are sorted after article number, which also gives the effect that the 5 displays having CCFL backlight are lined up first and the 5 displays having LED backlight are last.

The important display properties for this project are these:

- Resolution
- Backlight
- Video signals

In the data sheets for 4 of the displays there is also information about optical inspection rules and the promised maximum amount of pixel defects. The data sheet page number where this information can be found is written in the left column of App. E. Some of the information can be seen in Table 1.

<table>
<thead>
<tr>
<th></th>
<th>905-052</th>
<th>905-057, 905-098, 905-112</th>
</tr>
</thead>
<tbody>
<tr>
<td>Colour on display for detection of</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- bright sub-pixel:</td>
<td>Black</td>
<td>Black</td>
</tr>
<tr>
<td>- black sub-pixel:</td>
<td>White</td>
<td>No information</td>
</tr>
<tr>
<td>Ambient light:</td>
<td>40 W white fluorescent</td>
<td>300-500 lx</td>
</tr>
<tr>
<td>Viewing distance:</td>
<td>&gt; 30 cm</td>
<td>Appr. 35 cm</td>
</tr>
<tr>
<td>Viewing angle:</td>
<td>+45° relative normal</td>
<td>+10° relative normal</td>
</tr>
</tbody>
</table>

Table 1. Optical inspection rules from data sheets for 4 displays.
3.2.1 Analyze of resolution

The displays have 3 different resolutions, which correspond to 3 different pixel clocks, which can be seen in Table 2. 6 of the 10 displays are using the highest resolution, XGA.

<table>
<thead>
<tr>
<th>Type</th>
<th>Resolution [pixels]</th>
<th>Pixel clock [MHz]</th>
<th>Number of displays</th>
</tr>
</thead>
<tbody>
<tr>
<td>VGA = Video Graphics Array</td>
<td>$640\times 480$</td>
<td>25.175</td>
<td>2</td>
</tr>
<tr>
<td>SVGA = Super VGA</td>
<td>$800\times 600$</td>
<td>40.0</td>
<td>2</td>
</tr>
<tr>
<td>XGA = Extended Graphics Array</td>
<td>$1024\times 768$</td>
<td>65.0</td>
<td>6</td>
</tr>
</tbody>
</table>

*Table 2. TFT display resolutions [12]-[14].*

The pixel clock frequency in Table 2 is the typical value. In the data sheets for the displays it is possible to see also the acceptable frequency interval for the pixel clock. These results are displayed in Table 3.

<table>
<thead>
<tr>
<th>Display</th>
<th>Page in data sheet</th>
<th>Min. [MHz]</th>
<th>Typ. [MHz]</th>
<th>Max. [MHz]</th>
</tr>
</thead>
<tbody>
<tr>
<td>905-052</td>
<td>5</td>
<td>20</td>
<td>25</td>
<td>30</td>
</tr>
<tr>
<td>905-057</td>
<td>10</td>
<td>35</td>
<td>40</td>
<td>42</td>
</tr>
<tr>
<td>905C065</td>
<td>18</td>
<td>60.0</td>
<td>65.0</td>
<td>68.0</td>
</tr>
<tr>
<td>905C066</td>
<td>21</td>
<td>60.0</td>
<td>65.0</td>
<td>68.0</td>
</tr>
<tr>
<td>905-097</td>
<td>10</td>
<td>50</td>
<td>65</td>
<td>80</td>
</tr>
<tr>
<td>905-098</td>
<td>9</td>
<td>20</td>
<td>25</td>
<td>30</td>
</tr>
<tr>
<td>905-112</td>
<td>10</td>
<td>35</td>
<td>40</td>
<td>42</td>
</tr>
<tr>
<td>905-116</td>
<td>9</td>
<td>50</td>
<td>65</td>
<td>80</td>
</tr>
<tr>
<td>905-117</td>
<td>9</td>
<td>50</td>
<td>65</td>
<td>80</td>
</tr>
<tr>
<td>905-120</td>
<td>10</td>
<td>50</td>
<td>65</td>
<td>80</td>
</tr>
</tbody>
</table>

*Table 3. Acceptable frequency interval for pixel clocks for the displays, available at [11].*

For the 2 displays with resolution VGA, 25 MHz is mentioned instead of 25.175 MHz. The reason should be that no decimals are mentioned.
3.2.2 Analyze of backlight supply

The backlight supply uses its own connectors and cables. As mentioned before, half of the display models are using CCFL backlight and the other half are using LED. A so called carrier board (carries the CPU module) generates video signals and backlight supply. It is produced in different versions regarding backlight supply.

CCFL

Displays using CCFL are no longer used for new constructions of display computers at CrossControl, but they are still used in the production. The 5 CCFL displays all need different start and operation voltages. The reason is, as said in subchapter 2.2.1, mainly that the longer tubes in larger displays need higher voltages. From the start I was supplied with the display 905-057 and a carrier board including backlight supply for that display. Special hardware is needed for the other 4 displays, which are covered in subchapter 3.4.1. 4 displays have 2 tubes and 1 display (905-097) has 4 tubes. The CCFL cables are already mounted on the displays from supplier. They are of 3 different types, each with their own type of connector from the display:

- 1 connector with GND and 2 supply lines (for both tubes)
- 2 connectors, each with GND and 1 supply line (1 each to the 2 tubes)
- 2 connectors, each with GND and 2 supply lines (for 4 tubes).

LED

The 5 LED displays need lower voltages, but higher currents than the CCFL displays. The differences in voltage and current are relatively small compared to CCFL. The carrier board for LED for 905-112 generates the lowest current, which is more critical than voltage. That current showed to be enough for all 5 displays, including the display having 4 LED loops, because of a non-linear relationship between current and light intensity [15]. 4 displays have 2 LED loops and 1 display (905-120) has 4 LED loops. The LED cables must be attached manually to the displays at test. They are of 2 different types with same type of connector towards carrier board, but with different connectors towards the display:

- 4 lines for 2 loops
- 8 lines for 4 loops

As seen in App. E, the cable for 2 loops could be used for backlight supply of displays of all 3 types of resolution. So such a cable was collected.
3.2.3 Analyze of video signals supply

Both transmission types parallel and LVDS are represented:

- Parallel is used only by the 2 VGA-displays, which also are the only of size 6.5”.
- LVDS is used by the other 8 displays.

The length of the RGB data:

- 18 bits are used in 3 displays:
  - The 2 displays using parallel video signals
  - 905C065 where only 3 of the 4 LVDS links are used
- 24 bits are used in the other 7 displays, all LVDS.
  The bit order in the LVDS links of the RGB data and timing signals are the same in all the 8 displays using LVDS.

5 different cables must be used:

- Parallel: Same for both displays.
- LVDS: 4 different cables are represented:
  - 905C065 and 905C066 have their own special pin configuration. See App. E where the most of the pin configuration are reversed compared to the other LVDS displays, note especially + 3.3V! This is the most important result of the analyze.
  - 905-120 has a plastic connector with shorter pins, and the corresponding cable connector shall be of a special type.
  - The other 5 LVDS displays use the standard LVDS cable.

As seen in App. E, a parallel cable and a standard LVDS cable could be used for video signals supply of displays of all 3 types of resolution. So those types of cables were collected.

To be able to verify the VHDL project in next chapter, displays using the 3 resolutions, the 2 video signal transmission types and the 2 lengths of RGB data must be used. (The VHDL project is not depending on backlight type.) Following displays were used:

- 905-057
- 905-098
- 905-117
3.3 Programming of FPGA

3.3.1 Basic hardware and preparations

There is a FPGA (Field Programmable Gate Array) on the carrier board and the reason is mainly security: The board is normally used in a display computer produced for trains, where the FPGA is supervising video data for the pixels via check sums connected to SIL (Safety Integrity Level) classification. (The possibility to show analog video from a camera on the display is another reason for the FPGA.) All video data to the display passes the FPGA, which is an essential condition to make this project possible. The FPGA is of type Xilinx Spartan 3 (type number XC3S1400A).

Changing the resolution using the CPU module on the carrier board is much more complex than for an office PC. So, to make it possible to easily use displays of different resolutions, all the video signals shall be generated from the FPGA, without support from the CPU. Selection of the correct resolution, i.e. the correct pixel clock, was solved by adding 2 parallel pin lists on the carrier board. The pins on one list were connected to ground and on the other list 3 pins were connected to specific test points in to FPGA. Pixel clock is selected by connecting a jumper between the pin lists at 1 of the 3 positions.

It also results in a much quicker startup, because no operating system has to be loaded. Therefore, the compact flash containing Linux operating system on the back side of the carrier board is not needed.

The System Supervisor (SS) on the carrier board is a processor that normally supervises a lot of signals from the FPGA. In this project all these signals does not exist and the SS would therefore shut down the system. For that reason a so called Console version of SS was loaded. It does not have the supervising functions, so the lack of signals is no problem. Both sides of the carrier board for LED backlight are seen in Fig. 4.

![Fig. 4. Carrier board LED backlight (modified with pin lists and patch lines).](image)
The colour loop for the display is controlled by a push button from a display computer front overlay, connected to the carrier board. When used in a display computer, every push of the button increases the backlight one step. When used in this project, every push changes the colour in the loop.

The following connections were made, as seen in Fig. 5:

- Display to carrier board with cables for backlight and video signals.
- Power supply +24 V to carrier board.
- Push button at an overlay from a display computer to carrier board, used to change colour.
- To load the VHDL project to FPGA:
  
  Laptop PC – Xilinx DLC10 Platform Cable USB II – Update card – Carrier board.

![Image](image.jpg)

Fig. 5. Development setup using display 905-112.

At start the VHDL project was verified for the display 905-057, corresponding to a pixel clock of 40 MHz. When this worked, the 2 displays representing the other 2 resolutions were used for verifying the development with the corresponding pixel clocks.
3.3.2 VHDL software project

The FPGA is programmed in VHDL (VHSIC (Very High Speed Integrated Circuit) Hardware Description Language). The VHDL code is written in Xilinx ISE Project Navigator (release version 13.2). The simulation function in the Project Navigator was often used to check that different signal levels were changed at the correct moment.

The original VHDL project for the display computer has a lot of modules handling security. They are not needed for just showing different colours on the displays. If all those modules would be included it would take approximately 40 min – 1 hour to build the so called bit file, which is needed to test code changes. Without them it takes some minute, so starting a new project was an easy choice.

In the original VHDL project, the CPU module generates a pixel clock and a LVDS video signal. They go in to the FPGA, where a VHDL module called “lvds_deserialiser” deserializes the LVDS signals (3 timing signals and 24 RGB data signals). At every rising clock edge the 24 RGB bits are sent out in a vector. The meaning of this module is to make the safety supervising and the adding of analog video possible. Out from the same module also come the 3 timing signals, the passed-through pixel clock and a 7 × pixel clock. The signals from the VHDL module “lvds_deserialiser” in the FPGA must be generated without support from the pixel clock and the LVDS video signal coming from the CPU module. That is best solved by replacing “lvds_deserialiser” with 2 new VHDL modules with different functions:

- Generation of clocks
- Generation of video signals

In the new VHDL project there are 6 VHDL modules. 2 of them were written especially for this project, 2 were copied from the original project and then modified and 2 were copied without change.

Loading of the VHDL project to the FPGA was done using the programming tool Xilinx ISE iMPACT and could be done in two ways:

- The code was mainly loaded as a bit-file, “.bit”, directly to the FPGA, a quick method but the bit-file disappears as soon as the voltage is turned off.
- When the project was finished the code was loaded as a .mcs-file to the configuration flash memory for FPGA. The .mcs-file is stored in the flash memory even if the voltage is turned off.
A flow chart describing the main function of the VHDL project after power on is shown in App. B.

The code with comments for each module is shown in App. D. The properties of each module are described briefly in this subchapter. The user constraints file is also described. All inputs and outputs from the program are there connected to specific pins on the FPGA.

**top_level.vhd**

App. D, page D1-D5. Some parts from the top level module in the original project were reused, mainly inputs and outputs. All VHDL modules in the project, called “PixelTest”, are connected to each other by creating signals between them. No process is needed. The relationships between the modules are shown in the block schedule in App. C. Other changes compared to the original project:

- Test-points 3, 6, 8 were added as inputs for selection of pixel clock.
- Test-point 9 was added as output for measuring of pixel clock.
- The backlight was activated by setting “bl_pwm” to 1.

**button_debouncer.vhd**

App. D, page D6-D7. This VHDL module was copied from the original project and then modified. It handles the problem of contact bounces, coming from the push button for changing of colour. The output port “debouncedButton” changes state to pressed or released when the push button has been free from bounces for a time defined by “DebounceCounter”. Changes of input ports:

- “fp_btn_inc” is normally used for increasing backlight intensity, but is here used for colour change.
- “LPCClock25MHz” was exchanged to “pixel_clk”.
- “pulse1kHz” is instead generated internally in a new process, adapted for 40 MHz pixel clock. This worked fine also for the other pixel clocks.
- The active low reset “reset_n” from “clock_gen.vhd” was added.

Other changes:

- The variable "ChangeButtonState" has been removed and "DebounceCounter(7)" is used directly instead.
**clock_gen.vhd (Clock generator)**

App. D, page D8-D10. This VHDL module was written specially for the project and was the first that was written. The reset signal from the carrier board goes into this module. At power on, the reset is released after a while. It generates the pixel clock and two clocks of frequency $3.5 \times \text{pixel clock}$, 180 deg. phase shifted in between.

The Spartan 3 FPGA used have 4 Digital Clock Managers (DCM:s), which are Xilinx IP Intellectually Property) logic cores. They can be used for generating clocks needed by the application. [16] was used in the process of generating the clocks in the Xilinx Core Generator within the ISE Project Navigator. 2 DCM:s were used to generate the pixel clock and the two clocks of frequency $3.5 \times \text{pixel clock}$:

- dcm_pixel_test_1.vhd
- dcm_pixel_test_2.vhd

The in- and outputs used in the project for a DCM are seen in Fig. 6. The generation process for the clocks is based on the setting of these in- and outputs.

```
<table>
<thead>
<tr>
<th>Inputs</th>
<th>Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLIN</td>
<td>CLK0</td>
</tr>
<tr>
<td>RST</td>
<td>CLKFX</td>
</tr>
<tr>
<td></td>
<td>CLKFX180</td>
</tr>
<tr>
<td></td>
<td>LOCKED</td>
</tr>
</tbody>
</table>
```

*Fig. 6. Used inputs and outputs of Digital Clock Manager.*

- **DCM 1** is used to generate a 40 MHz clock which is the pixel clock frequency for SVGA, as seen in Table 2 in subchapter 3.2.1.

Inputs, both from the carrier board:
- CLIN: 25 MHz clock
- RST: Main reset

Outputs:
- CLK0: 25 MHz clock passing through
- CLKFX: 40 MHz clock generated by multiplying and dividing the 25 MHz input
- CLKFX180: Not used
- LOCKED: High when all clocks in and out from DCM 1 are stable
DCM 2 is used to generate the $7 \times \text{pixel clock}$ for LVDS described in subchapter 2.3.4.

Inputs:
- CLKin: A clock selected from the following 3 alternatives, where the selected alternative is detected at power on:
  * 25 MHz from the carrier board, within the allowable limits for VGA (typ. freq. 25.175 MHz). This can be seen in Table 3 in subchapter 3.2.1.
  * 40 MHz generated in DCM 1, the typical pixel clock frequency for SVGA.
  * 66.5 MHz already generated by dividing a 133 MHz clock from the carrier board. Within the allowable limits for XGA (typ. freq. 65 MHz).
- RST: DCM 2 is reset if DCM 1 is unlocked.

Outputs:
- CLK0: The passed-through pixel clock is used as input clock to the other 4 VHDL modules, see App. C.
- CLKFX, CLKFX180: The two clocks of frequency $3.5 \times \text{pixel clock}$ are used for generation of the data bit stream for LVDS. Spartan 3 FPGA is too slow to fulfill the timing demands using a $7 \times \text{pixel clock}$, so instead two clocks of frequency $3.5 \times \text{pixel clock}$ are delivered, one of them 180 degrees phase shifted compared to the other. The number of rising edges will in this way be $2 \times 3.5 = 7$.
- LOCKED: “reset_n” is high when all clocks are stable. It goes to all other modules except “parallel_video_out”.

The two DCM:s used together are seen in Fig. 7.

![Fig. 7. Generation and selection of pixel clock (reset and locked signals for DCM 1 not displayed).](image-url)
video_sig_gen.vhd (Video signal generator)

App. D, page D11-D15. This module was written specially for the project. The video signal generator module generates 27 video signals:

- The colour values red, green and blue, each made up of 8 bits, in a vector
- The 3 timing signals de, hsync, and vsync in parallel

Values for 5 different colours are sent to the display, one colour at a time. Changing of colour is controlled by “debouncedButton”, coming from the module “button_debouncer”. The colours are changed in a loop: red – green – blue – black – white – red and so on.

In one process the timing values for the selected pixel clock is set from 3 alternatives, one for each pixel clock. For example, the index “10” represents the 40 MHz clock for SVGA 800 × 600. The timing values are set from the standard values in Table 4.

**General timing**

- Screen refresh rate: 60 Hz
- Pixel clock: 40 MHz

**Horizontal timing (line), polarity of horizontal sync pulse is positive.**

<table>
<thead>
<tr>
<th>Scanline part</th>
<th>Pixels</th>
<th>Time [us]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Visible area:</td>
<td>800</td>
<td>20</td>
</tr>
<tr>
<td>Front porch:</td>
<td>40</td>
<td>1</td>
</tr>
<tr>
<td>Sync pulse:</td>
<td>128</td>
<td>3.2</td>
</tr>
<tr>
<td>Back porch:</td>
<td>88</td>
<td>2.2</td>
</tr>
<tr>
<td>Whole line:</td>
<td>1056</td>
<td>26.4</td>
</tr>
</tbody>
</table>

**Vertical timing (frame), polarity of vertical sync pulse is positive.**

<table>
<thead>
<tr>
<th>Frame part</th>
<th>Lines</th>
<th>Time [ms]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Visible area:</td>
<td>600</td>
<td>15.84</td>
</tr>
<tr>
<td>Front porch:</td>
<td>1</td>
<td>0.0264</td>
</tr>
<tr>
<td>Sync pulse:</td>
<td>4</td>
<td>0.1056</td>
</tr>
<tr>
<td>Back porch:</td>
<td>23</td>
<td>0.6072</td>
</tr>
<tr>
<td>Whole line:</td>
<td>628</td>
<td>16.5792</td>
</tr>
</tbody>
</table>

*Table 4. Timing values for 40 MHz pixel clock [13].*

Front porch and back porch, before and after the sync pulse, are extra time included to get correct screen refresh rate. The extra time was originally needed for CRT displays (Cathode Ray Tube).
lvds_serialiser.vhd
App. D, page D16-D18. This VHDL module was copied unchanged from the original project. The 27 video signal bits coming from the video signal generator are serialized to a 4 link data stream ("dataout"), where the bit-timing is generated by the two clocks of frequency $3.5 \times pixel\ clock$; “pixel_clk35” and the 180 degrees phase shifted “pixel_clk35_n”. These clocks are internal in the FPGA and the data stream is not converted to a LVDS signal until in the module top_level.vhd. The pixel clock coming in (“pixel_clk”) has 50 % duty cycle and the LVDS pixel clock going out (“clkout”) is asymmetric (with unchanged frequency): It has high level $4/7$ of the period time and low level $3/7$. The reason is that the number of bits transferred in each data link is 7, and the clock shall not change state in the middle of a bit.

parallel_video_out.vhd
App. D, page D19-D20. This VHDL module was copied unchanged from the original project. As seen in the name this module is used by displays using parallel video data. They also have only 18 bits RGB data.

top_level.ucf
App. D, page D21-D22. “ucf” means user constraints file. Only some parts from the corresponding ucf-file for the original project were needed, and those were copied unchanged. Nothing had to be added. All inputs and outputs in “top_level.vhd” are connected to specific pins on the FPGA:

Inputs:
- “cb_reset_n” is the main reset, active low from carrier board
- 2 clocks, 25 MHz and 133 MHz from carrier board
- Signal from push button for colour change
- Test points tp3, tp6 and tp 8 are used for pixel clock selection. “PULLUP” means the function of a pull-up resistor, that is they are active low.

Outputs:
- Regarding the video signal transmission the pins are always active for each transmission type:
  - LVDS: The clock and the 4 links use 2 pins each: positive “_p” and negative “_n”, which together make the LVDS signal.
  - Parallel: The 3 timing signals and the 18 rgb-bits all have their own pin.
- Backlight
- Testpoint tp 9 is used for measuring the pixel clock with oscilloscope.
3.4 Realizing of special test equipment hardware

The mission of this part was to get the special hardware necessary to test the rest of the display models.

The work was divided in:
- Defining the special hardware and eventual modification needed, for every display model
- Collecting the hardware
- Modifying the hardware if needed
- Verifying the function of the hardware

The carrier boards and some standard cables were already available since the VHDL project. To define some of the additional hardware and the eventual modifications needed, some other persons were consulted, see Preface.

The special test equipment hardware were in the same way as the other hardware at first hand collected by getting scrapped or by other reasons left-over units (for other qualities than needed here), or by borrowing units. Some parts had to be collected as new.

Modifying were made by other persons, see Preface.

For every display model its corresponding test equipment hardware were verified.

An overview of some specifications for all the test equipment hardware is seen in App. F.

Hardware and settings needed for all display models are covered in App. G.

An instruction of how to test a display is seen in App. H.
3.4.1 Realizing of special hardware for backlight

CCFL

The special CCFL backlight supplies are of 3 types, seen in Fig. H4 in App. H, and marked as mentioned first in the following parts:

- “052”: A special base board used only for the backlight for 905-052, which shall have lower voltages. (The corresponding power cable marked “24V” is of same type as for the carrier boards.) Note: There is a risk to connect the backlight supply of the carrier board instead for the board “052”, because the backlight connectors are of the same type.
- “XL”: A separate small inverter module for 905C065 and 905C066. The reason for the marking “XL” is the fact that these 2 displays, of different sizes, are used in the display computer model XL. As seen in App. E both display sizes have the same current and relatively small difference in voltage, which makes it possible to use the same module for both. The inputs GND, 5 V and 12 V might be possible to use from the carrier board, but to save time and to avoid damage on the carrier board this will wait until the construction of a fixture, which is not included in this project. Instead two separate power supplies are used, connected at grounding point. The corresponding cable, also marked “XL”, is special made for that purpose.
- “097”: A display board for 905-097, the only display with 4 tubes. When used in a display computer, the main processor sends commands to a processor on the display board, controlling the backlight via 2 CCFL controller circuits. Each of the controller circuits controls 2 tubes. For this project, the processor controlling the backlight is disconnected by lifting its 2 output pins for backlight enable respective brightness. The corresponding input pins at the controller circuits are set high for maximum brightness, by strapping 5 V there. (The intensity is controlled in the range 0.5 V – 2.0 V. Voltages below results in min intensity and voltages above in max intensity.) The board is, as for “XL”, supplied with GND, 5 V and 12 V from the two power supplies. A modified ribbon cable is used, also marked “097”. Note: The cable should always be attached to the board, connected as in Fig. H4. There is a risk of turning the connector in wrong way.

LED

The standard cable for 4 LED loops could be used also from the carrier board made for 2 loops. Pins 1 and 2 on carrier board normally power only anode 1, but power here anode 1 and 3 and so on.
3.4.2 Realizing of special hardware for video signals

As mentioned in subchapter 3.2.3, all pins in the LVDS connector for the displays 905C065 and 905C066 have different configuration compared to the other displays and to each other, 3 pins differ between the 2 displays. It is seen in App. E. Pins 1 to 16 from carrier board corresponds to pin 20 down to 5 on the display, as seen in App. F.

- 905C065 uses only 18 bit RGB, therefore link 3 is not used. The data sheet for the display, available at [11], says at part 4.5.4 that both pins for link 3, pins 1 and 2, are grounded internally in the display and that they also shall be connected to ground. It is also seen that pin 4 shall be open.
- For 905C066 pin 4 is used for FRC, which means Frame Rate Control. It corresponds to MODE on carrier board, where high level means 24 bit RGB. In the circuit diagram for the carrier board is seen that MODE is always set high. This means link 3 is always connected through the cable.

The result of the facts above is that a unique LVDS video cable had to be made for each of the both displays. All the LVDS cables for this test were made longer by joining two standard LVDS cables. These 2 cables were made by connecting different lines in the joining point. The cables are marked:

- “065 XL10” for 905C065
- “066 XL12” for 905C066

“XL” is mentioned because the cables are used for the 2 displays in the display computer model XL, the same marking as on the cable for the inverter module for backlight, used by both displays. “10” and “12” are the sizes of the displays: 10.4” and 12.1”. Note: There is a double risk for wrong connection: The correct cable must be used and it must be turned the right way. If the connection is not made the correct way, the +3.3 V from the carrier board might be connected to ground (065) or to link 3 (066) at the display!

- “120”: For 905-120, with the shorter connector pins, the regular cable was not available and the cable type is seldom ordered. Until a regular cable is available, both sides of one connector on a standard LVDS cable were cut off, to be able to push the connector longer into the display connector. Note: There might be a risk for turning the connector upside down. The metal pins of the cable connector shall be upwards at connection!

After realizing all special hardware, it was verified successfully by testing the corresponding display models. The rest of the display models were also tested successfully. So, all 10 display models can be tested.
3.5 Inspection of different pixel defects at different colours

Some displays with defect pixels had been discovered in the production. The colour showed from the faulty pixel had been written on a note attached to each display. One display each with different pixel defect colour was collected. Tests were then performed, using the test equipment. The faulty pixel was observed while the different colours in the loop were displayed, to check the change of the visibility of the pixel defects. The results can be seen in table 5.

The visibility of the displays has been graded in 1-5, with 5 as the best visibility. The grades have been set from a subjective point of view. The sign “-” mean that the faulty pixel is not seen at all.

<table>
<thead>
<tr>
<th>Colour from defect pixel</th>
<th>Display showing:</th>
<th>Red</th>
<th>Green</th>
<th>Blue</th>
<th>Black</th>
<th>White</th>
</tr>
</thead>
<tbody>
<tr>
<td>Red</td>
<td>-</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>Green</td>
<td>2</td>
<td>-</td>
<td>2</td>
<td>5</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>Blue</td>
<td>2</td>
<td>1</td>
<td>-</td>
<td>4</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>Purple</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>White</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>5</td>
<td>-</td>
<td></td>
</tr>
</tbody>
</table>

Table 5. Subjective visibility of different pixel defects at different colours on the display (grade 5 best).

All these pixel defects should be stuck high sub-pixels, because none of them are seen at white colour and all appear brighter than the colour shown on the display. They are naturally not seen when the same colour is shown on the display and they are best seen at black colour on the display, because of the largest contrast. In subchapter 2.1.2 is said that a green stuck high sub-pixel are easier detected by the human eye than red or blue, because it appear brighter. That is confirmed at black colour. It is also logic that green colour on the display is the colour where stuck high sub-pixels are most difficult to see; the brightest colour gives the smallest contrast to them.

The purple pixel is probably the result of 2 sub-pixels stuck high: red and blue. Therefore it is also seen for every one of the colours red, green and blue on the display; some other sub-pixel than the actual colour is always lit. The white pixel is the result of all 3 sub-pixels stuck high.

If a stuck low sub-pixel had existed it should have been seen as a darker dot at white colour on the display, where all the sub-pixels shall be fully lit. It should also have been seen at the same colour as the sub-pixel would have shown if it was OK.
4 Discussion

The VHDL project for the FPGA and the realization of test equipment hardware resulted in that all 10 display models can be tested without problems, so the main goal of the project was achieved.

When testing with deliberately wrong resolution for 905-112, that is wrong pixel clock from carrier board, it was shown that the image on the display is disturbed in both cases. When the selected resolution is lower, that is lower pixel clock frequency, than the resolution of the display the colour does not cover the whole display. When it is higher, vertical stripes are seen. See Fig. 8.

![Fig. 8. 905-112 SVGA: To the left selected pixelclock for VGA, to the right XGA.](image)

The module “button_debouncer.vhd” was as all other VHDL modules first made only for 40 MHz pixel clock, as seen in line 53. It turned out to work without problems also for the other two pixel clocks: 25 MHz and 66.5 MHz: The colours are changed distinctly when the push button is pressed. Eventual problems would not have affected reliable test conditions. Therefore no adaption to different pixel clocks was made.

The chosen inspection method, visual by human eye, has naturally a big disadvantage in the risk of missing pixel defects, but that was a given condition. An eventual automatic detection of pixel defects is not contained in this project.

Based on recommendations, it seems a stuck low sub-pixel should be easier to detect if red, green and blue are shown in addition to white (black is for stuck high). At least it should be easier to find out the colour it should have shown, even though it is not needed for comparing to the pixel defect standard. But a display with a stuck low sub-pixel was not available for verifying. All the collected displays with pixel defects showed to have stuck high sub-pixels.
5 Conclusions

In a TFT display, every pixel is consisting of 3 sub-pixels (red, green and blue). Every sub-pixel is controlled by its own thin-film transistor (TFT). The large amount of TFTs in a display causes the risk of pixel faults of 3 basic types: Type 1: Stuck high pixel means all 3 sub-pixels are always lit, Type 2: Stuck low pixel means all 3 sub-pixels are always turned off, Type 3: 1 sub-pixel is always stuck high (lit) or stuck low (turned off). Cluster pixel faults are faults of type 1 to 3 within a cluster of $5 \times 5$ pixels.

Based on these definitions the International Organization for Standardization, ISO, has created a Pixel fault classification, which is included in the standard ISO 9241Ergonomics of human-system interaction – Part 307:2008: Analysis and compliance test methods for electronic visual displays [7]. The classification is a table containing specific allowable numbers of the different types of pixel faults. If the numbers are exceeded the display supplier are recommended to replace the display. But other limits of pixel defects than in the standard can be agreed between the display supplier and the customer.

At a visual control of TFT displays regarding faulty pixels, the following colours shall, each at a time, be shown on the display: Red, green, blue, black and white.

10 different display models were selected by CrossControl for coming arrival inspection. Their properties in brief:

- **Resolution:**
  2 have $VGA = 640 \times 480$ pixels, 2 have $SVGA = 800 \times 600$ and 6 have $XGA = 1024 \times 768$. Therefore 3 different pixel clocks had to be generated.

- **Backlight:**
  5 have the older technology Cold Cathode Fluorescent Lamps (CCFL), which means neon light from tubes, and 5 have Light Emitting Diodes (LED).
  - For the CCFL displays, special backlight supply of 3 types for 4 displays had to be constructed because of the different supply voltages needed by the displays. For one display there is a risk of connecting the wrong type of supply, with risk of damage. The cables are of 3 different types and are already mounted on the displays from supplier.
  - The backlight voltage from the LED carrier board worked for all the 5 LED displays. The LED cables must be connected manually to the displays and they are of 2 different types: for 2 or 4 LED loops, with different types of connectors to the display. In summary, the backlight supply for LED was much easier to realize than for CCFL.
Video signals:
2 have parallel transmission and 8 have LVDS (Low Voltage Differential Signaling). Special cables had to be made for 3 of the LVDS displays. 2 of those cables have a completely different pin configuration, also compared to each other. This was the most important discovery, because if the standard LVDS cable would have been used from the carrier board, the displays would probably have been damaged. Those cables should be kept in a special place, to avoid connection to wrong type of display.

2 display computer carrier boards, one each for the backlight type CCFL respective LED, were used from the start. They are including a FPGA of type Xilinx Spartan 3. The programming of the FPGA was best solved by opening a new project and using some modules and code parts from the original project. In the original project several VHDL modules are handling security. They are unnecessary for this project and would have caused a very long building time for the project. 4 modules were copied from the original project, but 2 of them were modified.

The VHDL module “lvds_serialiser.vhd” in the original project gets a pixel clock and a LVDS video signal containing the colour code from the CPU. To be able to easily change the colours on the display, the CPU had to be disconnected from the colour generation. Instead, the FPGA was programmed to generate the different colour codes. Therefore 2 completely new VHDL modules had to be written to replace the module “lvds_deserialiser.vhd”: One that generates the pixel clock and the LVDS pixel clock and one that generates the video signals.

2 clocks from the carrier board were used: 25 MHz and 133 MHz, divided to 66.5 MHz. Together with 2 cascade coupled Digital Clock Managers they generate the correct frequencies for the pixel clock and the two clocks of frequency $3.5 \times \text{pixel clock}$, phase shifted 180 deg. inbetween. The last 2 clocks form together the LVDS pixel clock of frequency $7 \times \text{pixel clock}$. The correct pixel clock is selected with the position of a jumper on two parallel pin lists.

All 10 display models can be tested without problems.

Suggestions for developments of the project:
- Checking of the backlight, by testing one CCFL tube or LED loop at a time.
- Mounting of the test equipment in a mechanical fixture, including:
  - Cooling of CPU.
  - A 3-state switch for the 3 different display resolutions. The pin list where the jumper is used can instead be connected to the switch.
- Another type of push button for changing of colour.
- If 5 V and 12 V from the carrier board can be used to supply the backlight supply equipment for 905C065, 905C066 and 905-097 it will reduce the number of external power supplies needed from 3 to just 1.
- The carrier board for LED backlight could be used for video signal generation for all displays if special backlight supply is attached also for 905-057. (The carrier board for CCFL backlight can supply only 905-057 with backlight voltage and the rest of the 4 CCFL displays need special backlight supply.)
- A camera might be used for documentation, either manually or automatic.
- Automatic detection of pixel defects might be implemented, maybe by measuring the emitted light strength from the display or by measuring the current consumption.
References


[9] ”Dead pixel tester with display aids and illusions.” Internet: http://www.dataproductservices.com/dpt


[12] SECONS Ltd. “VGA Signal 640 x 480 @ 60 Hz Industry standard timing.“, Internet: http://tinyvga.com/vga-timing/640x480@60Hz

[13] SECONS Ltd. “SVGA Signal 800 x 600 @ 60 Hz timing.“, Internet: http://tinyvga.com/vga-timing/800x600@60Hz
[14] SECONS Ltd. "XGA Signal 1024 x 768 @ 60 Hz timing", Internet: 
http://tinyvga.com/vga-timing/1024x768@60Hz


[On-line]. Available: 
### Appendix A – Table of acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCFL:</td>
<td>Cold Cathode Fluorescent Lamps</td>
</tr>
<tr>
<td>CPU:</td>
<td>Central Processing Unit</td>
</tr>
<tr>
<td>DCM:</td>
<td>Digital Clock Manager</td>
</tr>
<tr>
<td>DE:</td>
<td>Data Enable</td>
</tr>
<tr>
<td>FPGA:</td>
<td>Field Programmable Gate Array</td>
</tr>
<tr>
<td>HSYNC:</td>
<td>Horizontal Synchronization</td>
</tr>
<tr>
<td>ISO:</td>
<td>International Organization for Standardization</td>
</tr>
<tr>
<td>LCD:</td>
<td>Liquid Crystal display</td>
</tr>
<tr>
<td>LED:</td>
<td>Light Emitting Diodes</td>
</tr>
<tr>
<td>LVDS:</td>
<td>Low Voltage Differential Signaling</td>
</tr>
<tr>
<td>MUX:</td>
<td>Multiplexer</td>
</tr>
<tr>
<td>RGB:</td>
<td>Red Green Blue</td>
</tr>
<tr>
<td>RTL:</td>
<td>Register Transfer Level</td>
</tr>
<tr>
<td>SIL:</td>
<td>Safety Integrity Level</td>
</tr>
<tr>
<td>SIS:</td>
<td>Swedish Standards Institute</td>
</tr>
<tr>
<td>SS:</td>
<td>System Supervisor</td>
</tr>
<tr>
<td>SVGA:</td>
<td>Super Video Graphics Array, 800x600 pixels</td>
</tr>
<tr>
<td>TFT:</td>
<td>Thin Film Transistor</td>
</tr>
<tr>
<td>USB:</td>
<td>Universal Serial Bus</td>
</tr>
<tr>
<td>VGA:</td>
<td>Video Graphics Array, 640x480 pixels</td>
</tr>
<tr>
<td>VHDL:</td>
<td>VHSIC (Very High Speed Integrated Circuit) Hardware Description Language</td>
</tr>
<tr>
<td>VSYNC:</td>
<td>Vertical Synchronization</td>
</tr>
<tr>
<td>XGA:</td>
<td>Extended Graphics Array, 1024x768 pixels</td>
</tr>
</tbody>
</table>
Appendix B – VHDL flow chart

- Detect selected pixel clock: clk_select = tp3 & tp6 & tp8
  - Set clk_index to “01”, “10” or “11”
- Is DCM1 locked: dcm1_locked_out=1?
  - Is button pushed and debounced: fp_btn_inc = debouncedButton = 0?
    - Yes
      - Set default RGB data to red: r=0xFF, g=0x00, b = 0x00
      - Set rgb in video interfaces for parallel and LVDS to red: tft_r=0x3F, tft_g=0x00, tft_b=0x00, dataout r-bits=1, g-bits=0, b-bits=0
    - No
      - Between 2 frames, set RGB data to green:
        - When vsync_n=0, set r=0x00, g=0xFF, b =0x00
        - Between 2 frames, set RGB data to blue:
          - When vsync_n=0, set r=0x00 g=0x00, b =0xFF
          - Between 2 frames, set RGB data to black:
            - When vsync_n=0, set r=0x00, g=0x00, b =0x00
            - Between 2 frames, set RGB data to white:
              - When vsync_n=0, set r=0xFF, g=0xFF, b =0xFF
              - Set rgb in video interfaces for parallel and LVDS to white: tft_r=0x3F, tft_g=0x3F, tft_b=0x3F, dataout r-bits=1, g-bits=1, b-bits=1
          - Set rgb in video interfaces for parallel and LVDS to black:
            - tft_r=0x00, tft_g=0x00, tft_b=0x3F, dataout r-bits=0, g-bits=0, b-bits=1
  - No
    - Is DCM2 locked: reset_n = 1?
      - Yes
        - Start generating video timing signals:
          - Set 0 or 1 for de, hsync_n, vsync_n
      - No
        - Set rgb in video interfaces for parallel and LVDS to green:
          - tft_r=0x00, tft_g=0x3F, tft_b=0x00, dataout r-bits=0, g-bits=1, b-bits=0
Appendix C – VHDL block schedule

- pixel_clk
- fp_btn_inc
- reset_n
- clk_25MHz
- clk_133MHz
- clk_select
- rgb
- reset
- debounced
- clk_index
- parallel_video_out
- video_sig_gen
- button_debouncer
- debounced
- Button
- pixel_clk_x3_x5
- pixel_clk_x3_5n
- pixel_clk35
- pixel_clk35_lckd
- pixel_clk
- dataout
- clkout
- hsync_n
- vsync_n
- tft_clk
- tft_r
- tft_g
- tft_b
- tft_ena
- tft_hsync_n
- tft_vsync_n
Appendix D – VHDL code modules

top_level.vhd

```vhdl
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
use IEEE.NUMERIC_STD.ALL;

-- Uncomment the following library declaration if instantiating
library UNISIM;
use UNISIM.VComponents.all;

-- All inputs and outputs copied from hmi410_top.vhd, some comments added:

entity top_level is
port(
  -- Active low reset from carrier board:
ob_reset_n : in std_logic;

  -- Clock inputs from carrier board:
clk_25MHz : in std_logic;
clk_133MHz : in std_logic;

  -- Button input for colour change:
fp_btn_inc: in std_logic;

  -- Test points:
tp3 : in std_logic; -- 1 tp of 3 shall be
tp6 : in std_logic; -- set low to select
tp8 : in std_logic; -- pixel clock.
tp9 : out std_logic; -- Possibility to measure pixel clock
);
```

D1
Erik Unosson

ARRIVAL CONTROL OF TFT DISPLAYS

```
-- LVDS video out (to display):

tft_txclk_p : out std_logic;
tft_txclk_n : out std_logic;
tft_tx0_p   : out std_logic;
tft_tx0_n   : out std_logic;
tft_tx1_p   : out std_logic;
tft_tx1_n   : out std_logic;
tft_tx2_p   : out std_logic;
tft_tx2_n   : out std_logic;
tft_tx3_p   : out std_logic;
tft_tx3_n   : out std_logic;

-- Parallel video out (to display)
tft_clk    : out std_logic;
tft_vsync_n: out std_logic;
tft_hsync_n: out std_logic;
tft_ena    : out std_logic;
tft_r      : out std_logic_vector(5 downto 0);
tft_g      : out std_logic_vector(5 downto 0);
tft_b      : out std_logic_vector(5 downto 0);

-- Backlight:
b1_pwm    : out std_logic;

end top_level;

architecture rtl of top_level is

signal cb_reset : std_logic;
signal clk_index_i : std_logic_vector(1 downto 0);
signal clk_select : std_logic_vector(2 downto 0) := "111";
signal de_i      : std_logic;
signal debouncedButton_i : std_logic;
signal hsync_n_i : std_logic;
signal pixel_clk_i : std_logic;
signal pixel_clk_x1_5_i : std_logic;
signal pixel_clk_x3_5_n_i : std_logic;
signal reset_n_i : std_logic;
signal rgb_i      : std_logic_vector(23 downto 0);
signal tft_txclock : std_logic;
signal vsync_n_i : std_logic;

begin

-- Copy internal signals to outputs

-------------------------------------------------------------

b1_pwm <= '1'; -- Backlight activated
cb_reset <= not cb_reset_n ; -- Active high reset is needed to DCM
clk_select <= tp3 & tp6 & tp8; -- Test operator sets one testpoint low
tp9 <= pixel_clk_i; -- Test point for pixel clock from DCM2
```

D2
-- Buffer instantiations for differential I/Os

-- LVDS clock:

i_OBUFD3_tft_clk : OBUFD3
  GENERIC MAP (  
    IOSTANDARD => "DEFAULT")
  PORT MAP (  
    0 => tft_clk_p,  -- Output, positive
    OB => tft_clk_n,  -- Output B, negative
    I => tft_clk);  -- Input

-- LVDS Link 0 of video data:

i_OBUFD3_tft_tx0 : OBUFD3
  GENERIC MAP (  
    IOSTANDARD => "DEFAULT")
  PORT MAP (  
    0 => tft_tx0_p,  
    0B => tft_tx0_n,  
    I => tft_tx(0));

-- LVDS Link 1 of video data:

i_OBUFD3_tft_tx1 : OBUFD3
  GENERIC MAP (  
    IOSTANDARD => "DEFAULT")
  PORT MAP (  
    0 => tft_tx1_p,  
    0B => tft_tx1_n,  
    I => tft_tx(1));

-- LVDS Link 2 of video data:

i_OBUFD3_tft_tx2 : OBUFD3
  GENERIC MAP (  
    IOSTANDARD => "DEFAULT")
  PORT MAP (  
    0 => tft_tx2_p,  
    0B => tft_tx2_n,  
    I => tft_tx(2));

-- LVDS Link 3 of video data:

i_OBUFD3_tft_tx3 : OBUFD3
  GENERIC MAP (  
    IOSTANDARD => "DEFAULT")
  PORT MAP (  
    0 => tft_tx3_p,  
    0B => tft_tx3_n,  
    I => tft_tx(3));
-- Component instantiation

-- 3 clocks and 1 reset are linked:
i_clock_gen : entity work.clock_gen rtl port map(
  -- Inputs:
  clk_25MHz => clk_25MHz, -- From carrier board to DCM
  clk_13MHz => clk_13MHz, -- From carrier board to DCM
  clk_select => clk_select, -- 1 of 3 pixel clocks selected
  reset => cb_reset, -- Reset from carrier board to DCM
  -- Outputs:
  clk_index => clk_index_i, -- Index for selected pixel clock
  pixel_clk => pixel_clk_i, -- Pixel clock
  pixel_clk_x3_5 => pixel_clk_x3_5_i, -- 2 pcs. of 3.5x clock with 180
  pixel_clk_x3_6_n => pixel_clk_x3_6_n_i, -- deg. phase shift inbetween
  reset_n => reset_n_i); -- High when DCM is locked

-- 27 parallel bits for RGB-data and timing are linked:
i_video_sig_gen : entity work.video_sig_gen rtl port map(
  -- Inputs:
  clk_index => clk_index_i,
  debouncedButton => debouncedButton_i,
  pixel_clk => pixel_clk_i,
  reset_n => reset_n_i,
  -- Outputs:
  rgb => rgb_i, -- 3x8=24 bits for Red, Green, Blue
  de => de_i, -- data enable: Read RGB-data
  hsync_n => hsync_n_i, -- Change to new line
  vsync_n => vsync_n_i); -- Start displaying new frame

-- The 27 parallel bits for video data (24 RGB, de, hsync_n, vsync_n)
-- are serialized and sorted in correct order into 4 parallel LVDS
-- serial data stream links:
i_lvds_serialiser : entity work.lvds_serialiser rtl port map(
  -- Inputs:
  pixel_clk => pixel_clk_i,
  pixel_clk35 => pixel_clk_x3_5_i, -- 2x3.5=7x pixel clock used
  pixel_clk36_n => pixel_clk_x3_6_n_i, -- for LVDS bit timing
  pixel_clk36_clkd => reset_n_i, -- High when DCM is locked
  rgb => rgb_i,
  dc => de_i,
  vsync_n => vsync_n_i,
  hsync_n => hsync_n_i,
  -- outputs:
  dataout => tft_tx, -- Vector for 4 parallel video data
  clkout => tft_txclk); -- LVDS clock to display;
  -- high 4 bits, low 3 bits
Some displays have 18 bits parallel RGB-data:

d5

-- Inputs:

-- Outputs:

-- Button bounce filtering:

end rtl;
-- File: button_debouncer.vhd
---
-- Copyright: Copyright (C) 2009 CC Systems AB. All rights reserved.
-- Created: 2009-02-19
-- Modified and supplemented by Erik Unosson
-- Description:
-- "With 20 MHz this module does 50ms debounce filtering for the input signal."
--
-- The copyright to the source code and computer program(s) herein is
-- the property of CC Systems AB. This source code may not be used,
-- in full or in part, without written permission from CC Systems AB.
---
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

-- Uncomment the following library declaration if instantiating
-- any Xilinx primitives in this code.
-- library UNISIM;
-- use UNISIM.VComponents.all;

entity button_debouncer is
  Port (    
    pbtn_n : in std_logic;
    pixel_clk : in std_logic;
    reset_n : in std_logic; -- Active low reset from clock_gen.vhd
    -- 0 from start, 1 when DCM is stable
    debouncedButton : out std_logic
  );
end button_debouncer;

architecture rtl of button_debouncer is
  
  signal counter : std_logic_vector (15 downto 0);
  signal debounceCounter : std_logic_vector(7 downto 0);
  signal pulse1kHz : std_logic := '0';
  signal stateOfButton : std_logic := '0';

begin

  debouncedButton <= stateOfButton;
  
  -- Generation of pulse1kHz:
  process (pixel_clk, reset_n)
  begin
    if reset_n = '0' then
      pulse1kHz <= '0';
      counter <= (others=>'0');
    elsif rising_edge(pixel_clk) then
      if counter > 40000 then -- 40 MHz/(40000) = 1 kHz
        pulse1kHz <= '1';
        counter <= (others=>'0'); -- Reset timer
      else
        pulse1kHz <= '0';
        counter <= counter + 1; -- Increase timer
      end if;
    end if;
  end process;

end rtl;
-- Some debounce filtering for the input signal:

process (pixel_clk, reset_n)
begin
  if reset_n = '0' then
    debounceCounter <= "11001101"; -- Start value = 128+77
    stateOfButton <= '0';
  elsif rising_edge(pixel_clk) then
    -- If state of button is the same as button input, which is the fact
    -- if button is unpressed or if bouncing occurs, counter will be
    -- initialized to start value again:
    if stateOfButton = fp_btn_inc then
      debounceCounter <= "11001101";-- Divisor changed 19.2.2009 because
      -- the LPC bus frequency changed.
    -- If state of button is not the same as button input then count down:
    else
      if pulse1kHz = '1' then
        debounceCounter <= debounceCounter-1;
      end if;
    end if;
    -- When bits 0-6 all are 0, next decrease will change bit 7 from 1 to 0.
  end if;
  if debounceCounter(7) = '0' then
    stateOfButton <= fp_btn_inc;
  end if;
end process;
end rtl;}
clock_gen.vhd

LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_ARITH.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;

-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
-- use IEEE.NUMERIC_STD.ALL;

-- Uncomment the following library declaration if instantiating
-- any Xilinx primitives in this code.
library UNISIM;
use UNISIM.VComponents.all;

entity clock_gen is
port (  
  clk_25MHz : in std_logic;  -- From carrier board
  clk_133MHz : in std_logic;  -- From carrier board
  clk_select : in std_logic_vector(2 downto 0);  -- Vector 2 to 0,
  -- index for selected clock from test operator
  reset     : in std_logic;  -- From carrier board to DCX
  clk_index : out std_logic_vector(1 downto 0);  -- Index for
  -- selected pixel clock
  pixel_clk : out std_logic;  -- Selected pixel clock
  pixel_clk_k3_5 : out std_logic;  -- Together with _n it produces a
  -- 5 bit timing clock for LVDS
  pixel_clk_k3_5_n : out std_logic;  -- Serializer
  reset_n : out std_logic;  -- Active low, when low all
  -- modules are in reset state
end clock_gen;
architecture rtl of clock_gen is

signal clk : std_logic; -- Buffer for pixel clock
signal clk_25MHz_i : std_logic; -- From carrier board, passed through DCM1
signal clk_40MHz : std_logic <= '0'; -- Generated in DCM1
signal clk_66_5MHz : std_logic := '0'; -- 133 MHz from carrier board / 2
signal clk_change : std_logic := '0'; -- Index for selected clock,
signal clk_index_i : std_logic_vector(1 downto 0); -- Index for selected clock,
signal cnt : std_logic_vector(2 downto 0) := (others=>'0'); -- Counter in delay
signal dcmLocked_out : std_logic;
signal reset_in_dcm2 : std_logic := '1'; -- Indicates that DCM2 is reset
signal selected clk : std_logic := '0'; -- Selected pixel clock
signal switch_change : std_logic := '0'; -- Indicates if switch for pixel

begin

clk_index <= clk_index_i;
pixel_clk <= clk; -- "pixel_clk" can not be read without using "clk" as a buffer
-- 2 pcs. of DCM's (Digital Clock Manager) creates selected pixel clock
-- and 7x pixel clock to LVDS Serializer.
-- DCM 1:
i_dcm_pixel_test_1: entity work.dcm_pixel_test(BEHAVIORAL) port map(
  CLKIN_IN => clk_25MHz, -- From carrier board
  RST_IN => reset, -- Main reset from carrier board
  CLKFX_OUT => clk_40MHz, -- Generated in DCM1
  CLKIN_IBUF_OUT => open,
  CLK_OUT => clk_25MHz_i, -- Passed through
  LOCKED_OUT => dcmLocked_out); -- 0 from start, 1 when DCM 1 is stable
-- DCM 2:
i_dcm_pixel_test_2: entity work.dcm_pixel_test_2(BEHAVIORAL) port map(
  CLKIN_IN => selected_clk, -- Selected pixel clock
  RST_IN => reset_in_dcm2, -- Set in process
  CLKFX_OUT => pixel_clk_x8_5, -- 7x pixel clock to LVDS serializer
  CLKFX80_OUT => pixel_clk_x8_5_n, -- is generated by two 3.5x clocks
  -- 180 deg. phase shifted inbetween
  -- (one 7x is too fast for FPGA).
  CLK0_OUT => clk, -- Pixel clock via buffer "clk"
  LOCKED_OUT => reset_n); -- 0 from start, 1 when DCM 2 is stable

-- If DCM 1 is unlocked or clock is being changed, then DCM 2 will be reset.
-- If DCM 1 is locked or clock have been changed, reset shall be released
-- after a delay:
process(clk_40MHz, clk_change, dcmLocked_out) is
begin
  if dcmLocked_out = '0' or clk_change = '1' then
    reset_in_dcm2 <= '1';
    cnt <= (others=>'0');
  elsif rising_edge(clk_40MHz) then
    if cnt = "111" then -- Delay completed after 7 clock pulses
      reset_in_dcm2 <= '0'; -- Release reset of DCM2
    end if;
  end if;
end process;
-- If DCM 1 is locked, set index for selected pixel clock
-- and release reset of DCM2:
process(clk_40MHz, dcm1_locked_out, reset_in_dcm2) is
begin
  -- If DCM 1 is not locked, pixel clock can not be used:
  if dcm1_locked_out = '0' then
    clk_change <= '0';
    clk_index_i <= "00";
    switch_change <= '0';
  -- If DCM 1 is locked:
  elsif rising_edge(clk_40MHz) then
    if switch_change = '0' then
      switch_change <= '1';
      -- Make it impossible to change
      -- pixel clock again without
      -- switching off and on power
      clk_change <= '1';
      -- DCM2 will reset in process above
      if clk_select = "011" then -- TP2 set low
        clk_index_i <= "01";
      elsif clk_select = "101" then -- TP6 set low
        clk_index_i <= "10";
      elsif clk_select = "110" then -- TP8 set low
        clk_index_i <= "11";
      else
        clk_index_i <= "10";
        -- If none of the 3 pins are grounded,
        -- then the middle pixel clock
        -- frequency will be selected.
      end if;
    else
      clk_change <= '0';
      -- Release reset of DCM2
    end if;
  end if;
end process;

-- Generate 66.5 MHz clock from 133 MHz from carrier board:
process(clk_133MHz)
begin
  if rising_edge(clk_133MHz) then
    clk_66_5MHz <= not clk_66_5MHz; -- Changing state at rising edge of
    -- 133 MHz gives half that frequency
  end if;
end process;

-- Set selected frequency for CLkin to DCM 2:
process(clk_25MHz_i, clk_40MHz, clk_66_5MHz, clk_index_i, dcm1_locked_out) is
begin
  if dcm1_locked_out = '0' then
    selected_clk <= '0';
    elsif clk_index_i = "01" then
      selected_clk <= clk_25MHz_i; -- For VGA= 640x480 (25.175 MHz)
    elsif clk_index_i = "10" then
      selected_clk <= clk_40MHz;
    elsif clk_index_i = "11" then
      selected_clk <= clk_66_5MHz; -- For XGA= 1024x768 (65 MHz)
    else
      selected_clk <= clk_40MHz;
    end if;
end process;
end rtl;
video_sig_gen.vhd

-- Company: CrossControl
-- Engineer: Erik Unosson - Degree project
-- "Arrival control of TFT-displays"
-- Create Date: 10:00:30 08/19/2011
-- Design Name:
-- Module Name: video_sig_gen - rtl
-- Project Name:
-- Target Devices:
-- Tool versions:
-- Description: Generation of 27 parallel video bits from FPGA:
-- * 3x8-24 bits RGB-data: Red, Green, Blue
-- * de: Data enable = Read RGB-data
-- * hsync_n: Change to new line
-- * vsync_n: Start displaying new frame
-- Dependencies:
-- Revision:
-- Revision 0.01 - File Created
-- Additional Comments:
--
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_ARITH.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;

-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
-- use IEEE.NUMERIC_STD.ALL;

-- Uncomment the following library declaration if instantiating
-- any Xilinx primitives in this code.
-- library UNISIM;
-- use UNISIM.VComponents.all;

entity video_sig_gen is

port ( clk_index : in std_logic_vector(1 downto 0); -- Index for selected clock
debouncedButton : in std_logic; -- Debounced push button status
pixel_clk : in std_logic; -- Pixel clock
reset_n : in std_logic; -- Active low reset, 0 from start,
-- 1 when DCM is stable
de : out std_logic := '0'; -- Data enable: 1=Read RGB-data
hsync_n : out std_logic := '1'; -- Horizontal sync: 0=Change to new line
vsync_n : out std_logic := '1'; -- Vertical sync: 0=Start displaying
-- new frame
rgb : out std_logic_vector (23 downto 0)); -- 3x8-24 bits
-- RedGreenBlue-data
end video_sig_gen;
architecture rtl of video_sig_gen is

signal btn_state_d0 : std_logic := '0'; -- Present push button state, d=delay
signal btn_state_d1 : std_logic := '0'; -- Previous push button state
signal change_colour : std_logic := '0'; -- Not time for colour change

signal h_count : std_logic_vector (10 downto 0); -- Actual pos. within whole line
signal h_max : std_logic_vector (10 downto 0); -- End pos. within whole line
signal hsync_min : std_logic_vector (10 downto 0); -- First pos. within hsync pulse
signal hsync_max : std_logic_vector (10 downto 0); -- End pos. within hsync pulse
signal h_video : std_logic; -- 1 means inside video area horizontally
signal h_video_max : std_logic_vector (10 downto 0); -- End pos. within horizontal video area

signal v_count : std_logic_vector (9 downto 0); -- Actual pos. within whole column
signal v_max : std_logic_vector (9 downto 0); -- End pos. within whole column
signal vsync_min : std_logic_vector (9 downto 0); -- First pos. within vsync pulse
signal vsync_max : std_logic_vector (9 downto 0); -- End pos. within vsync pulse
signal v_video : std_logic; -- 1 means inside video area vertically
signal v_video_max : std_logic_vector (9 downto 0); -- End pos. within vertical video area

signal vsync_n_sig : std_logic := '1'; -- Same value as the output vsync_n

type colour is (RED, GREEN, BLUE, BLACK, WHITE);
n signal presentColour : colour;

begin

-- Assign outputs
rgb <= r & g & b;
n
-- Update present and previous push button status:
process(pixclk, reset_n)
begin

if reset_n = '0' then
btt_state_d0 <= '0'; -- Present state, d=delay
btt_state_d1 <= '0'; -- Previous state

when rising_edge(pixclk) then

if vsync_n_sig = '0' then
btt_state_d0 <= debouncedButton;
-- d0 is updated with new button status
btt_state_d1 <= btt_state_d0;
-- d1 is updated with old button status

end if;
end if;
end process;
-- Handle push button status change and colour change indicator:
process (pixel_clk, reset_n)
begin
  if reset_n = '0' then
    change_colour <= '0';
  elsif rising_edge(pixel_clk) then
    if btn_state_d1 = '1' and btn_state_d0 = '0' then
      if wayo_nSig = '0' then -- Between two frames
        change_colour <= '1'; -- Change of colour shall be done
      end if;
    else
      change_colour <= '0';
    end if;
  end if;
end process;

-- Prepare for colour change by deciding the colour that shall be set.
-- Handle output signals and next state definitions:
process (pixel_clk, reset_n)
begin
  if reset_n = '0' then
    r <= (others => '1'); g <= (others => '0'); b <= (others => '0');
    present_colour <= RED; -- Start colour
  elsif rising_edge(pixel_clk) then
    if change_colour = '1'
    then
      case present_colour is
      when RED =>
        r <= (others => '0'); g <= (others => '1'); b <= (others => '0');
        present_colour <= GREEN;
      when GREEN =>
        r <= (others => '0'); g <= (others => '0'); b <= (others => '1');
        present_colour <= BLUE;
      when BLUE =>
        r <= (others => '0'); g <= (others => '0'); b <= (others => '0');
        present_colour <= BLACK;
      when BLACK =>
        r <= (others => '1'); g <= (others => '0'); b <= (others => '0');
        present_colour <= WHITE;
      when WHITE =>
        r <= (others => '1'); g <= (others => '0'); b <= (others => '0');
        present_colour <= RED;
      when others =>
        present_colour <= null;
      end case;
    end if;
  end if;
end process;
-- Set timing values for selected clock:
process (pixel_clk, reset_n)
begin
  if reset_n = '0' then
    h_max <= (others=>'0');
    h_video_max <= (others=>'0');
    hsync_min <= (others=>'0');
    hsync_max <= (others=>'0');
    v_max <= (others=>'0');
    v_video_max <= (others=>'0');
    vsync_min <= (others=>'0');
    vsync_max <= (others=>'0');
  elseif rising_edge(pixel_clk) then
    case clk_index is
    when "01" => -- 25 MHz clock for VGA 640x480
      h_max <= conv_std_logic_vector(799,11);
      h_video_max <= conv_std_logic_vector(839,11);
      hsync_min <= conv_std_logic_vector(655,11); -- 639+16
      hsync_max <= conv_std_logic_vector(751,11); -- 639+16+96
      v_max <= conv_std_logic_vector(524,10);
      v_video_max <= conv_std_logic_vector(479,10);
      vsync_min <= conv_std_logic_vector(485,10); -- 479+10
      vsync_max <= conv_std_logic_vector(491,10); -- 479+10+2
    when "10" => -- 40 MHz clock for SVGA 800x600
      h_max <= conv_std_logic_vector(1055,11);
      h_video_max <= conv_std_logic_vector(799,11);
      hsync_min <= conv_std_logic_vector(839,11); -- 799+40
      hsync_max <= conv_std_logic_vector(967,11); -- 799+40+126
      v_max <= conv_std_logic_vector(627,10);
      v_video_max <= conv_std_logic_vector(599,10);
      vsync_min <= conv_std_logic_vector(600,10); -- 599+1
      vsync_max <= conv_std_logic_vector(604,10); -- 599+1+4
    when "11" => -- 66.5 MHz clock for XGA 1024x768
      h_max <= conv_std_logic_vector(1345,11);
      h_video_max <= conv_std_logic_vector(1023,11);
      hsync_min <= conv_std_logic_vector(1047,11); -- 1023+24
      hsync_max <= conv_std_logic_vector(1163,11); -- 1023+24+136
      v_max <= conv_std_logic_vector(805,10);
      v_video_max <= conv_std_logic_vector(767,10);
      vsync_min <= conv_std_logic_vector(770,10); -- 767+3
      vsync_max <= conv_std_logic_vector(776,10); -- 767+3+6
    when others =>
      null;
    end case;
end if;
end process;

-- Generation of horizontal counter:
process (pixel_clk, reset_n)
begin
  if reset_n = '0' then -- DCM unstable
    h_video <= '0';
    h_count <= (others=>'0'); -- Reset the counter
  elsif rising_edge(pixel_clk) then
    h_video <= '1';
    if h_count = h_max then -- End of whole line reached
      h_count <= (others=>'0'); -- Reset the counter
    else
      h_count <= h_count + 1; -- Increase counter within whole line
    end if;
  end if;
  if h_count > h_video_max then -- End of horizontal video area exceeded
    h_video <= '0'; -- Outside horizontal video area
  end if;
end process;
-- Generation of vertical counter:
process (pixel_clk, reset_n)
begin
    if reset_n = '0' then -- DCM unstable
        v_video <= '0'; -- Outside vertical video area
    elsif rising_edge(pixel_clk) then
        if h_count = h_max then -- End of whole line reached
            v_count <= (others=>'0'); -- Reset the counter
        else
            v_count <= v_count + 1; -- Increase counter in whole frame
        end if;
    end if;
if v_count > v_video_max then -- End of vertical video area exceeded
    v_video <= '0'; -- Outside vertical video area
end process;

-- Generation of hsync_n and vsync_n:
process (pixel_clk, reset_n)
begin
    if reset_n = '0' then
        hsync_n <= '0';
        vsync_n <= '0';
    elsif rising_edge(pixel_clk) then
        -- Horizontal:
        if (h_count)\=hsync_min and h_count\=hsync_max then -- During hsync pulse
            hsync_n <= '0'; -- Signal to change line
        else
            hsync_n <= '1';
        end if;
        -- Vertical:
        if (v_count)\=vsync_min and v_count\=vsync_max then -- During vsync pulse
            vsync_n <= '0'; -- Signal to change frame
        else
            vsync_n <= '1';
        end if;
    end if;
end process;

-- Generation of de (data enable):
process (pixel_clk, reset_n)
begin
    if reset_n = '0' then
        oe <= '0';
    elsif rising_edge(pixel_clk) then
        oe <= h_video and v_video; -- Active when inside video frame
    end if;
end process;
end rtl;
lvds_serialiser.vhd

-- File: lvds_serialiser.vhd
--
-- Copyright: Copyright (C) 2009 CC Systems AB. All rights reserved.
-- Created: 2009-03-11
--
-- Description:
-- Serialises parallel video data (rgb, de, vsync_n and hsync_n) to a
-- LVDS serial data stream.
--
-- The copyright to the source code and computer program(s) herein is
-- the property of CC Systems AB. This source code may not be used
-- in full or in part, without written permission from CC Systems AB.
--
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL ;
LIBRARY UNISIM ;
USE UNISIM.VCOMPONENTS.ALL ;

ENTITY lvds_serialiser IS PORT (  
pixel_clk: IN STD_LOGIC;
pixel_clk35: IN STD_LOGIC;
pixel_clk35_n: IN STD_LOGIC;
pixel_clk35_locked: IN STD_LOGIC;
rgb: IN STD_LOGIC_VECTOR(23 DOWNTO 0);
dc: IN STD_LOGIC;
vsync_n: IN STD_LOGIC;
hsync_n: IN STD_LOGIC;
dataput: OUT STD_LOGIC_VECTOR(3 DOWNTO 0);-- serial data outputs
clkout: OUT STD_LOGIC; -- clock output
END lvds_serialiser;

ARCHITECTURE rtl OF lvds_serialiser IS
 SIGNAL datasin : STD_LOGIC_VECTOR(27 DOWNTO 0);  
 SIGNAL reset_s : STD_LOGIC;
 SIGNAL outdata : STD_LOGIC_VECTOR(7 DOWNTO 0);  -- output data lines
 SIGNAL cclkinta : STD_LOGIC_VECTOR(1 DOWNTO 0);  
 SIGNAL clkoutaint : STD_LOGIC;  -- forwarded output clocks
 SIGNAL txdata : STD_LOGIC_VECTOR(27 DOWNTO 0);  -- data for transmission
 SIGNAL tx_output_reg : STD_LOGIC_VECTOR(3 DOWNTO 0);  
 SIGNAL tx_output_fix : STD_LOGIC_VECTOR(7 DOWNTO 0);  
 SIGNAL pixel_clk35_locked : std_logic;
 CONSTANT TX_SWAP_MASK : STD_LOGIC_VECTOR(3 DOWNTO 0) := "0000" ;
BEGIN

pixel_clk35_lockd_n <- NOT pixel_clk35_lockd;

-- Map RGB, vsync_n, hsync_n and DE to LVDS bit stream
datain(0) <- rgb(8); -- G0
datain(1) <- rgb(1);  -- B1
datain(2) <- d_e;    -- DE
datain(3) <- '0';    -- unused
datain(4) <- rgb(21);-- R5
datain(5) <- rgb(0); -- B0

datain(6) <- vsync_n; -- vsync_n
datain(7) <- rgb(7);  -- B7
datain(8) <- rgb(20);-- R4
datain(9) <- rgb(13);-- G5

datain(10) <- hsync_n;-- hsync_n
datain(11) <- rgb(6);  -- R6
datain(12) <- rgb(19);-- R3
datain(13) <- rgb(12);-- G4

datain(14) <- rgb(5);  -- B5
datain(15) <- rgb(15);-- G7

datain(16) <- rgb(18);-- R2
datain(17) <- rgb(11);-- G3

datain(18) <- rgb(4);  -- B4


datain(19) <- rgb(14);-- G6


datain(20) <- rgb(17);-- R1


datain(21) <- rgb(10);-- G2


datain(22) <- rgb(3);  -- B3


datain(23) <- rgb(23);-- R7


datain(24) <- rgb(16);-- R0


datain(25) <- rgb(9);  -- G1


datain(26) <- rgb(2);  -- B2

datain(27) <- rgb(22);-- R6


dataout <- tx_output_reg;
clkout <- clkoutaint;

loop1 : FOR i IN 0 to 3 GENERATE
  ld_d : ODDR2
  GENERIC MAP(
    DBR_ALIGNMENT => "NONE",
    SR_TYPE => "ASYNC"
  )
  PORT MAP(
    c0 => pixel_clk35,
    c1 => pixel_clk35_n,
    d0 => tx_output_fix(i+4),
    d1 => tx_output_fix(i),
    oe => '1',
    r => pixel_clk35_lockd_n,
    s => '0',
    q => tx_output_reg
  );

  tx_output_fix(i) <= outdata(i) XOR TX_SWAP_MASK(i);
  tx_output_fix(i+4) <= outdata(i+4) XOR TX_SWAP_MASK(i);
END GENERATE;
process(pixel_clk, reset_s)
begin
  if reset_s = '1' then
    txdata <= (others => '0');
  else if pixel_clk'event and pixel_clk = '1' then
    txdata <= datin;
  end if;
end process;

-- Generate a registered reset signal for the tx clock

end rtl;
parallel_video_out.vhd

-- File: parallel_video_out.vhd

-- Copyright: Copyright (C) 2009 CC Systems AB. All rights reserved.
-- Created: 2009-03-09
-- Description:
-- Output parallel video data (rgb, de, vsync_n and hsync_n)
-- The copyright to the source code and computer program(s) herein is
-- the property of CC Systems AB. This source code may not be used,
-- in full or in part, without written permission from CC Systems AB.

LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;

LIBRARY UNISIM;
USE UNISIM.VCOMPONENTS.ALL;

ENTITY parallel_video_out IS PORT (
  pixel_clk : IN std_logic;
  r          : IN std_logic_vector($DOWNTO 0);
  g          : IN std_logic_vector($DOWNTO 0);
  b          : IN std_logic_vector($DOWNTO 0);
  de         : IN std_logic;
  vsync_n    : IN std_logic;
  hsync_n    : IN std_logic;
  tft_clk    : OUT std_logic;
  tft_r      : OUT std_logic_vector($DOWNTO 0);
  tft_g      : OUT std_logic_vector($DOWNTO 0);
  tft_b      : OUT std_logic_vector($DOWNTO 0);
  tft_ena    : OUT std_logic;
  tft_vsync_n: OUT std_logic;
  tft_hsync_n: OUT std_logic);
END parallel_video_out;

ARCHITECTURE rtl OF parallel_video_out IS
  SIGNAL pixel_clk_n : std_logic;
BEGIN
  pixel_clk_n <= NOT pixel_clk;
  i_clkout : ODDR2
    GENERIC MAP (    
      d0 => '1',
      d1 => '0',
      c0 => pixel_clk,
      cl => pixel_clk_n,
      ce => '1',
      r  => '0',
      s  => '0',
      q  => tft_clk
    );

END rtl;
PROCESS(pixel_clk)
BEGIN
  IF pixel_clk'event AND pixel_clk = '1' THEN
    tft_r <= r;
    tft_g <= g;
    tft_b <= b;
    tft_ena <= de;
    tft_vsync_n <= vsync_n;
    tft_hsync_n <= hsync_n;
  END IF;
END PROCESS;
END rtl:
top_level.ucf

1 #************************************************************************************
2 ## File: top_level.ucf
3 ##
4 ## Copyright: Copyright (C) 2009 CC Systems AB. All rights reserved.
5 ## Created: 2009-03-16
6 ## Modified by Erik Unosson
7 ## Description:
8 ## All inputs and outputs from top_level.vhd is connected to specific pins on
9 ## the FPGA.
10 ## The copyright to the source code and computer program(s) herein is
11 ## the property of CC Systems AB. This source code may not be used,
12 ## in full or in part, without written permission from CC Systems AB.
13 #************************************************************************************
14
15 # React input:
16 NET "cb_reset_n" LOC = V19 | IOSTANDARD = "LVCMOS33"; # Active low reset from
17 # carrier board to FPGA
18
19 # Clock input:
20 NET "clk_25Mhz" LOC = L21 | IOSTANDARD = "LVCMOS33";
21 NET "c1k_133Mhz" LOC = AB12 | IOSTANDARD = "LVCMOS33";
22
23 # LVDS video out (to display):
24 NET "tft_txcclkl_p" LOC = C5 | IOSTANDARD = "LVDS_33";
25 NET "tft_txcclkl_n" LOC = D5 | IOSTANDARD = "LVDS_33";
26 NET "tft_txc0l_p" LOC = B6 | IOSTANDARD = "LVDS_33";
27 NET "tft_txc0l_n" LOC = A5 | IOSTANDARD = "LVDS_33";
28 NET "tft_tx1l_p" LOC = C6 | IOSTANDARD = "LVDS_33";
29 NET "tft_tx1l_n" LOC = D6 | IOSTANDARD = "LVDS_33";
30 NET "tft_tx2l_p" LOC = A4 | IOSTANDARD = "LVDS_33";
31 NET "tft_tx2l_n" LOC = B4 | IOSTANDARD = "LVDS_33";
32 NET "tft_tx3l_p" LOC = A3 | IOSTANDARD = "LVDS_33";
33 NET "tft_tx3l_n" LOC = B3 | IOSTANDARD = "LVDS_33";
34
35 # Parallel video out (to display):
36 NET "tft_clk" LOC = A17 | IOSTANDARD = "LVCMOS33" | DRIVE = 12 | SLEW = SLOW;
37 NET "tft_vsync_n" LOC = C15 | IOSTANDARD = "LVCMOS33" | DRIVE = 12 | SLEW = SLOW;
38 NET "tft_hsync_n" LOC = B17 | IOSTANDARD = "LVCMOS33" | DRIVE = 12 | SLEW = SLOW;
39 NET "tft_ena" LOC = D15 | IOSTANDARD = "LVCMOS33" | DRIVE = 12 | SLEW = SLOW;
40 NET "tft_r<0>" LOC = B16 | IOSTANDARD = "LVCMOS33" | DRIVE = 12 | SLEW = SLOW;
41 NET "tft_r<1>" LOC = F16 | IOSTANDARD = "LVCMOS33" | DRIVE = 12 | SLEW = SLOW;
42 NET "tft_r<2>" LOC = C17 | IOSTANDARD = "LVCMOS33" | DRIVE = 12 | SLEW = SLOW;
43 NET "tft_r<3>" LOC = D19 | IOSTANDARD = "LVCMOS33" | DRIVE = 12 | SLEW = SLOW;
44 NET "tft_r<4>" LOC = A20 | IOSTANDARD = "LVCMOS33" | DRIVE = 12 | SLEW = SLOW;
45 NET "tft_r<5>" LOC = B20 | IOSTANDARD = "LVCMOS33" | DRIVE = 12 | SLEW = SLOW;
46 NET "tft_g<0>" LOC = F15 | IOSTANDARD = "LVCMOS33" | DRIVE = 12 | SLEW = SLOW;
47 NET "tft_g<1>" LOC = C16 | IOSTANDARD = "LVCMOS33" | DRIVE = 12 | SLEW = SLOW;
48 NET "tft_g<2>" LOC = A18 | IOSTANDARD = "LVCMOS33" | DRIVE = 12 | SLEW = SLOW;
49 NET "tft_g<3>" LOC = A16 | IOSTANDARD = "LVCMOS33" | DRIVE = 12 | SLEW = SLOW;
50 NET "tft_g<4>" LOC = B19 | IOSTANDARD = "LVCMOS33" | DRIVE = 12 | SLEW = SLOW;
51 NET "tft_b<0>" LOC = C17 | IOSTANDARD = "LVCMOS33" | DRIVE = 12 | SLEW = SLOW;
52 NET "tft_b<1>" LOC = D17 | IOSTANDARD = "LVCMOS33" | DRIVE = 12 | SLEW = SLOW;
53 NET "tft_b<2>" LOC = C16 | IOSTANDARD = "LVCMOS33" | DRIVE = 12 | SLEW = SLOW;
54 NET "tft_b<3>" LOC = D16 | IOSTANDARD = "LVCMOS33" | DRIVE = 12 | SLEW = SLOW;
55 NET "tft_b<4>" LOC = E14 | IOSTANDARD = "LVCMOS33" | DRIVE = 12 | SLEW = SLOW;
56 NET "tft_b<5>" LOC = C14 | IOSTANDARD = "LVCMOS33" | DRIVE = 12 | SLEW = SLOW;
# Backlight:

NET "b1_pwm" LOC = AA4 | IOSTANDARD = "LVCMOS33" | DRIVE = 4 | SLEW = SLOW;

# Button for changing to next colour:

NET "fp_btn_inc" LOC = W22 | IOSTANDARD = "LVCMOS33";

# Test points:

NET "tp3" LOC = AB9 | IOSTANDARD = "LVCMOS33" | PULLUP;
NET "tp4" LOC = AB11 | IOSTANDARD = "LVCMOS33" | PULLUP;
NET "tp5" LOC = AB10 | IOSTANDARD = "LVCMOS33" | PULLUP;
NET "tp6" LOC = U11 | IOSTANDARD = "LVCMOS33" | DRIVE = 4 | SLEW = SLOW;
Appendix E – Technical data for the displays

10 different types  

The displays are sorted in groups in priority as follows: Backlight type (CCFL or LED), Common interfaces, Article number.

<table>
<thead>
<tr>
<th>Display properties</th>
<th>3 resolutions</th>
<th>Explanations</th>
<th>Corresponding connector always on cable.</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>CC Article no.</td>
<td>Size</td>
<td>VGA: 640x480 pixels</td>
<td>D = Display connector</td>
<td></td>
</tr>
<tr>
<td>(- = Lead free, C = Indefinite)</td>
<td>Pixel clk 25.175 MHz</td>
<td>CCFL: Inverter (cables mounted)</td>
<td>C = Corresponding connector:</td>
<td></td>
</tr>
<tr>
<td>Max. pixel defects in data sheet at page, N.I. = Not Incl.</td>
<td>SVGA: 800x600 pixels</td>
<td>Explanations</td>
<td>R = Red, G = Green, B = Blue</td>
<td></td>
</tr>
<tr>
<td>Manufacturer,</td>
<td>Pixel clk 40 MHz</td>
<td>LED: Cable</td>
<td>HD = HSYNC, VD = VSYNC, DE = Data Enable</td>
<td></td>
</tr>
<tr>
<td>Model no. on label.</td>
<td>XGA: 1024x768 pixels</td>
<td>Pixel clk 65 MHz</td>
<td>Link 0-3: RGB data via LVDS (at 18 bit only 0-2)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>JAE, JST, HIROSE = Manufacturers</td>
<td>CLKIN-/+ = Pixel clock</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>FRC=MODE: low = 18 bit RGB, high = 24 bit</td>
<td>SC = reverse Scan Control (Low:Normal, High:Reverse)</td>
<td></td>
</tr>
</tbody>
</table>

Group 1: Backlight type CCFL = Cold Cathode Fluorescent Lamps: 5 displays

<table>
<thead>
<tr>
<th>TFT display</th>
<th>Resolution</th>
<th>Backlight</th>
<th>Video</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>905-052</td>
<td>6.5”</td>
<td>VGA</td>
<td>2 CCFL: GND + supply for 2 tubes</td>
<td>Parallel</td>
</tr>
<tr>
<td>Optrex, p. 21</td>
<td></td>
<td>D: BHR-02(8.0)VS-1N (JST)</td>
<td>D: DF9B-31P-1V (HIROSE)</td>
<td></td>
</tr>
<tr>
<td>T-51750GD065J-FW-ADN</td>
<td></td>
<td>C: SM02(8.0)B-BHS (JST)</td>
<td>C: DF9B-31S-1V (HIROSE)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Start volt. max 520 V</td>
<td>Operation: typ. 320 V, 6.0 mA</td>
<td></td>
</tr>
</tbody>
</table>

Both VGA displays, this and 905-098, have parallel video signals and identical pin configuration on 31 pins:

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
<th>19</th>
<th>20</th>
</tr>
</thead>
<tbody>
<tr>
<td>GND</td>
<td>DCLK</td>
<td>HD</td>
<td>VD</td>
<td>GND</td>
<td>R0</td>
<td>R1</td>
<td>R2</td>
<td>R3</td>
<td>R4</td>
<td>R5</td>
<td>GND</td>
<td>G0</td>
<td>G1</td>
<td>G2</td>
<td>G3</td>
<td>G4</td>
<td>G5</td>
<td>GND</td>
<td>B0</td>
</tr>
<tr>
<td>21</td>
<td>22</td>
<td>23</td>
<td>24</td>
<td>25</td>
<td>26</td>
<td>27</td>
<td>28</td>
<td>29</td>
<td>30</td>
<td>31</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B1</td>
<td>B2</td>
<td>B3</td>
<td>B4</td>
<td>B5</td>
<td>GND</td>
<td>DE</td>
<td>+3.3V</td>
<td>+3.3V</td>
<td>OPEN</td>
<td>SC</td>
<td>(DCLK=Clock signal for sampling catch data signal)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Continuation of Group 1: Backlight type **CCFL** = Cold Cathode Fluorescent Lamps: 5 displays

<table>
<thead>
<tr>
<th>TFT display</th>
<th>Resolution</th>
<th>Backlight</th>
<th>Video</th>
</tr>
</thead>
<tbody>
<tr>
<td>905-057</td>
<td>10.4&quot;</td>
<td>2 CCFL as 905-052 above.</td>
<td>LVDS</td>
</tr>
<tr>
<td>Optrex,</td>
<td></td>
<td>D: FI-SEB20P-HFE (JAE)</td>
<td></td>
</tr>
<tr>
<td>T-51944D104J-FW-A-AA</td>
<td></td>
<td>(B means sunk-into-the board and is only used on this display but the cable connector is the same.)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>C: FI-S205</td>
<td></td>
</tr>
<tr>
<td>905-095</td>
<td>10.4&quot;+antireflex</td>
<td>LVDS video data <strong>pin configuration</strong> on 20 pins:</td>
<td></td>
</tr>
<tr>
<td>905C065</td>
<td>10.4&quot;</td>
<td>1 CCFL: GND + 2 supply for 2 tubes</td>
<td>LVDS connector as 905-057 above, but <strong>different pin configuration</strong>.</td>
</tr>
<tr>
<td>NEC,</td>
<td></td>
<td>D: BHR-03VS-1</td>
<td></td>
</tr>
<tr>
<td>NL10276BC20-04</td>
<td></td>
<td>C: SM03 (4.0) B-BHS-1-TB</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Start volt. min. 850-1100 V</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Operation: typ. 520 V, 5.0 mA</td>
<td></td>
</tr>
<tr>
<td>905C066</td>
<td>12.1&quot;</td>
<td>1 CCFL as 905C065 above.</td>
<td>LVDS connector as 905-057 above, but <strong>different pin configuration</strong>.</td>
</tr>
<tr>
<td>NEC,</td>
<td></td>
<td>D: BHR-04VS-1</td>
<td></td>
</tr>
<tr>
<td>NL10276BC24-13</td>
<td></td>
<td>C: SM04(4.0)B-BHS(LF)(SN) (JST)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Start volt. min. 960 V</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Operation: typ. 600 V, 5.0 mA</td>
<td></td>
</tr>
<tr>
<td>905-097</td>
<td>12.1&quot;</td>
<td>2 CCFL, each with GND and 2 supply lines, for 4 tubes</td>
<td>LVDS connector and pin config. as 905-057 above.</td>
</tr>
<tr>
<td>Mitsubishi, AA121XJ01</td>
<td></td>
<td>D: BHR-04VS-1 (JST)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>C: SM04(4.0)B-BHS(LF)(SN) (JST)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Start volt. min. 1000-1290 V</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Operation: typ. 540 V, 13.0 mA</td>
<td></td>
</tr>
</tbody>
</table>
### Group 2: Backlight type LED = Light Emitting Diodes: 5 displays

<table>
<thead>
<tr>
<th>TFT display</th>
<th>Resolution</th>
<th>Backlight</th>
<th>Video</th>
</tr>
</thead>
<tbody>
<tr>
<td>905-098</td>
<td>6.5”</td>
<td>VGA</td>
<td>1 LED with cathode and anode for 2 loops.</td>
</tr>
<tr>
<td>Optrex,</td>
<td></td>
<td></td>
<td>Parallel as 905-052.</td>
</tr>
<tr>
<td>T-55465D065J-LW-A-AAN</td>
<td>p.20</td>
<td></td>
<td>D: SM06B-SHLS-TF (JST)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>C: SHLP-06V-S-B-(JST)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Typ. 2x90 mA, Max. 2x150 mA</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Typ. 2x20.4 V, Max. 2x23,7 V</td>
</tr>
<tr>
<td>905-112</td>
<td>10.4”</td>
<td>SVGA</td>
<td>1 LED as 905-098 above.</td>
</tr>
<tr>
<td>Optrex,</td>
<td></td>
<td></td>
<td>LVDS connector and pin config. as 905-057.</td>
</tr>
<tr>
<td>T-55563D104J-LW-A-AAN</td>
<td>p.22</td>
<td></td>
<td>Typ. 2x70 mA, Max. 2x80 mA</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Typ. 2x27 V, Max. 2x36,0 V</td>
</tr>
<tr>
<td>905-116</td>
<td>12.1”</td>
<td>XGA</td>
<td>1 LED as 905-098 above.</td>
</tr>
<tr>
<td>Mitsubishi,</td>
<td>N.I.</td>
<td></td>
<td>LVDS connector and pin config. as 905-057.</td>
</tr>
<tr>
<td>AA121XK01</td>
<td></td>
<td></td>
<td>Typ. 2x120 mA, Max. 2x130 mA</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Typ. 2x24 V, Max. 2x33.6 V</td>
</tr>
<tr>
<td>905-117</td>
<td>10.4”</td>
<td>XGA</td>
<td>1 LED as 905-098 above.</td>
</tr>
<tr>
<td>Mitsubishi,</td>
<td>N.I.</td>
<td></td>
<td>LVDS connector and pin config. as 905-057.</td>
</tr>
<tr>
<td>AA104XD02</td>
<td></td>
<td></td>
<td>Typ. 2x70 mA, Max. 2x80 mA</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Typ. 2x27 V, Max. 2x36.0 V</td>
</tr>
<tr>
<td>905-120</td>
<td>15”</td>
<td>XGA</td>
<td>1 LED with cathode and anode for 4 loops.</td>
</tr>
<tr>
<td>Mitsubishi,</td>
<td>N.I.</td>
<td></td>
<td>LVDS</td>
</tr>
<tr>
<td>AA150XT01</td>
<td></td>
<td></td>
<td>D: DF14H-20P-1.25H(56) (HIROSE)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>C: DF14-205-1.25C (HIROSE)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Plastic with shorter pins causes the need of a special cable connector.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Pin config. as 905-057.</td>
</tr>
</tbody>
</table>
## Appendix F – Test equipment

The test equipment is sorted in following groups: Power supplies, CCFL boards, Carrier boards, Cables. 1 piece of every article is needed, unless anything else is mentioned.

<table>
<thead>
<tr>
<th>Marking</th>
<th>CC Article nbr.</th>
<th>Type</th>
<th>Applicable for display model</th>
<th>Connect</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
<td>C000105-17</td>
<td>Overlay 10,4”</td>
<td>All</td>
<td>To J3 on CCFL, LED</td>
<td>Pin 3 Backlight increase button for switching colours</td>
</tr>
</tbody>
</table>

### Power supplies

<table>
<thead>
<tr>
<th>Marking</th>
<th>CC Article nbr.</th>
<th>Type</th>
<th>Applicable for display model</th>
<th>Connect</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
<td>-</td>
<td>Power supply +24 V</td>
<td>All</td>
<td>Cable +24 V</td>
<td></td>
</tr>
<tr>
<td>-</td>
<td>-</td>
<td>Power supply +12 V</td>
<td>905C065,905C066,905-097</td>
<td>Cable XL or -097</td>
<td>Common ground for</td>
</tr>
<tr>
<td>-</td>
<td>-</td>
<td>Power supply +5 V</td>
<td>905C065,905C066,905-097</td>
<td>Cable XL or -097</td>
<td>+5 V and +12 V</td>
</tr>
</tbody>
</table>

### CCFL boards

3 types

<table>
<thead>
<tr>
<th>Marking</th>
<th>CC Article nbr.</th>
<th>Type</th>
<th>Applicable for display model</th>
<th>Connect</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>052</td>
<td>C000048-147</td>
<td>Base board</td>
<td>905-052</td>
<td>Cable +24V to X1</td>
<td></td>
</tr>
<tr>
<td>XL</td>
<td>905C013</td>
<td>DC/AC INVERTER</td>
<td>905C065, 905C066</td>
<td>Cable XL to CN1 Display to CN2 or CN3</td>
<td>470-640 V, 5 mA</td>
</tr>
</tbody>
</table>
Marking | CC Article nbr. | Type | Applicable for display model | Connect | Comments
--- | --- | --- | --- | --- | ---
097 | C000107-17 | Display board 12" | 905-097 | Cable -097 to J1 | Note! Risk of turning the connector in wrong way.

Modified this way:
- Pins
C4:15-16
- Pins lifted to disconnect the processor which normally controls the backlight by commands from the main processor.
- These two pins enable backlight and control brightness.

<table>
<thead>
<tr>
<th>Connect</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cable -097 to J1</td>
<td>Note! Risk of turning the connector in wrong way.</td>
</tr>
<tr>
<td>Cable -097 to J1</td>
<td>Generates volt. for 4 tubes.</td>
</tr>
</tbody>
</table>

Carrier boards - backlight and video signals

- Modifications of both carrier boards:
  - Compact flash card containing Linux not needed.
  - Console version of System Supervisor loaded.
  - .mcs-file loaded to configuration flash for FPGA.
  - Test points 3, 6, 8 connected to pins on a pin list and GND to corresponding pins in a parallel pin list.
  - A jumper connects the pin lists when choosing pixel clock.
  - Pixel clock can be measured at tp9.

CCFL | C000105-06 | Carrier board, CCFL | All 5 CCFL displays, but backlight supply only applicable for 905-057. | Both have FPGA Xilinx Spartan 3, type no XC3S1400A.

LED | C000105-100 | Carrier board, LED | All 5 LED displays |
## ARRIVAL CONTROL OF TFT DISPLAYS

<table>
<thead>
<tr>
<th>Marking</th>
<th>CC Article nbr.</th>
<th>Type</th>
<th>Applicable for display model</th>
<th>Connect</th>
<th>Comments</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Cables</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>24V</td>
<td>Similar to</td>
<td>Straight M12 to Laboratory plugs</td>
<td>All</td>
<td>+24 V to X1 on boards CCFL, LED, -097</td>
<td>90 cm</td>
<td></td>
</tr>
<tr>
<td></td>
<td>C000002-31</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>24V</td>
<td>Power</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>3 types</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>The two cables for +24V works the same way. Both at same time are needed only for 905-052.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Connector towards carrier board plugged in the middle for eliminating risk of wrong connection.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>24V</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>200 cm</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>24V</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>XL</td>
<td>Special made of:</td>
<td>905C065, 905C066</td>
<td></td>
<td>+5 V, +12 V to CN1 on board XL</td>
<td>38 cm</td>
<td></td>
</tr>
<tr>
<td>810C151</td>
<td>Yellow banana plug 4 mm</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>810C152</td>
<td>Red banana plug 4 mm</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>810C153</td>
<td>Black banana plug 4 mm</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>810-359</td>
<td>Connector 8 pol. to inverter board</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>852C134</td>
<td>8 cable lines 0.15 mm</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pins</td>
<td>Voltage</td>
<td>Comments</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1-2</td>
<td>DC in</td>
<td>+12 V</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5-6</td>
<td>SLEEP#</td>
<td>+5 V</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3-4</td>
<td>GND</td>
<td>0 V</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7-8</td>
<td>VSYNC</td>
<td>0 V</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Length of cables are without connectors.
<table>
<thead>
<tr>
<th>Marking</th>
<th>CC Article nbr.</th>
<th>Type</th>
<th>Applicable for display model</th>
<th>Connect</th>
<th>Comments</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>097</td>
<td>Special made of:</td>
<td>905-097</td>
<td>+5 V, +12 V to J1 on board -097</td>
<td>67 cm</td>
<td></td>
<td></td>
</tr>
<tr>
<td>810C151</td>
<td></td>
<td>Yellow banana plug 4 mm</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>810C152</td>
<td></td>
<td>Red banana plug 4 mm</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>810C153</td>
<td></td>
<td>Black banana plug 4 mm</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>850-012</td>
<td></td>
<td>Ribbon cable 4&quot; 50-pol</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>852-084</td>
<td></td>
<td>Yellow cable 0.22mm²</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>852C087</td>
<td></td>
<td>Red cable 0.22mm²</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>852-088</td>
<td></td>
<td>Black cable 0.22mm²</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Ribbon cable **modified** by only using following lines, to the 3 banana plugs:

<table>
<thead>
<tr>
<th>Pins on J1</th>
<th>Voltage</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>46,48,50</td>
<td>+12 V</td>
<td>Red plug to power supply</td>
</tr>
<tr>
<td>17,21</td>
<td>+5 V</td>
<td>Yellow</td>
</tr>
<tr>
<td>45,47,49</td>
<td>GND</td>
<td>Black</td>
</tr>
</tbody>
</table>

* Backlight - LED 2 types

**LED2** C000082-
Cable TFT LED backlight 905-098, 905-112, 905-116, 905-117
From board to display, extended. 46 cm
4 lines: LED cathode and anode for **2 loops**.
(Standard lengths: C000082-48 for 6.5": 19 cm, -69 for 10": 22 cm, -70 for 12": 25cm.)

**LED4** C000124-13
Cable TFT LED-backlight 15" 905-120
From board LED to display 8 lines: LED cathode and anode for **4 loops**.
Pins 1-2 normally power only anode 1 but power here anode 1 and 3 and so on, which seems to work fine. 27.5 cm
<table>
<thead>
<tr>
<th>Marking</th>
<th>CC Article nbr.</th>
<th>Type</th>
<th>Applicable for display model</th>
<th>Connect</th>
<th>Comments</th>
<th>Length</th>
</tr>
</thead>
</table>
| * Video signals -  
  Parallel | 1 type |
| PAR     | 859-100        | Digital display cable | 905-052, 905-098 | From board CCFL/LED to display. | 18 cm |
| * Video signals -  
  LVDS | 4 types |
| LVDS    | 2 pcs. 859-099 | LVDS display cable | 905-057, 905-097, 905-112, 905-116, 905-117 | From board CCFL/LED to display. | 50 cm |
|         |                | 20-pol |                              | Standard used for 5 of 8 LVDS displays. |        |
|         |                | Special made: | 2 cables joined. |          |          |        |
| 120     | 2 pcs. 859-099 | Special made: | 2 cables joined. | From board LED to display | 72 cm |
|         |                | Correct LVDS Cable |                            |          |          |        |

Note! As a consequence, the cable connector might be turned upside down. Metal pins shall be upwards!

Old type of cables but same article nbr., thereby the larger length.
### Arrival Control of TFT Displays

<table>
<thead>
<tr>
<th>Marking</th>
<th>CC Article nbr.</th>
<th>Type</th>
<th>Applicable for display model</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>065 XL10</td>
<td>2 pcs. 859-099</td>
<td>LVDS display cable 20-pol</td>
<td></td>
<td>46 cm</td>
</tr>
<tr>
<td>066 XL12</td>
<td>2 pcs. 859-099</td>
<td>LVDS display cable 20-pol</td>
<td></td>
<td>48 cm</td>
</tr>
</tbody>
</table>

**Special made,** 2 cables joined in different ways:

<table>
<thead>
<tr>
<th>905C065</th>
<th>XL 10.4” 18-bit RGB</th>
<th>905C066</th>
<th>XL 12.1” 24-bit RGB</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Carrier board</th>
<th>Display</th>
<th>Carrier board</th>
<th>Display</th>
</tr>
</thead>
<tbody>
<tr>
<td>+3.3V</td>
<td>1</td>
<td>+3.3V</td>
<td>1</td>
</tr>
<tr>
<td>+3.3V</td>
<td>2</td>
<td>+3.3V</td>
<td>2</td>
</tr>
<tr>
<td>GND</td>
<td>3</td>
<td>GND</td>
<td>3</td>
</tr>
<tr>
<td>GND</td>
<td>4</td>
<td>GND</td>
<td>4</td>
</tr>
<tr>
<td>Link 0-</td>
<td>5</td>
<td>Link 0-</td>
<td>5</td>
</tr>
<tr>
<td>Link 0+</td>
<td>6</td>
<td>Link 0+</td>
<td>6</td>
</tr>
<tr>
<td>GND</td>
<td>7</td>
<td>GND</td>
<td>7</td>
</tr>
<tr>
<td>Link 1-</td>
<td>8</td>
<td>Link 1-</td>
<td>8</td>
</tr>
<tr>
<td>Link 1+</td>
<td>9</td>
<td>Link 1+</td>
<td>9</td>
</tr>
<tr>
<td>GND</td>
<td>10</td>
<td>GND</td>
<td>10</td>
</tr>
<tr>
<td>Link 2-</td>
<td>11</td>
<td>Link 2-</td>
<td>11</td>
</tr>
<tr>
<td>Link 2+</td>
<td>12</td>
<td>Link 2+</td>
<td>12</td>
</tr>
<tr>
<td>GND</td>
<td>13</td>
<td>GND</td>
<td>13</td>
</tr>
<tr>
<td>CLKIN-</td>
<td>14</td>
<td>CLKIN-</td>
<td>14</td>
</tr>
<tr>
<td>CLKIN+</td>
<td>15</td>
<td>CLKIN+</td>
<td>15</td>
</tr>
<tr>
<td>GND</td>
<td>16</td>
<td>GND</td>
<td>16</td>
</tr>
<tr>
<td>Link 3-</td>
<td>17</td>
<td>Link 3-</td>
<td>17</td>
</tr>
<tr>
<td>Link 3+</td>
<td>18</td>
<td>Link 3+</td>
<td>18</td>
</tr>
<tr>
<td>MODE</td>
<td>19</td>
<td>MODE</td>
<td>19</td>
</tr>
<tr>
<td>SC</td>
<td>20</td>
<td>SC</td>
<td>20</td>
</tr>
</tbody>
</table>

N.C.=No Connection
## Appendix G – Test setup depending on display

| CC Article no., Size, (- = Lead free, C = Indefinite) | Overlay always connected. | CCFL: Carrier board backlight is applicable only for 905-057, special supply must be added for the others. | LED: Carrier board backlight is working for all displays, so special supply is not relevant. | NOTE! 065 XL10 and 066 XL12 might be connected in wrong way with risk of damage. | VGA: 640x480 pix Pixel clk 25.175 MHz SVGA: 800x600 pix. Pixel clk 40 MHz XGA: 1024x768 pix. Pixel clk 65 MHz |
|---|---|---|---|---|
| Max. pixel defects in data sheet at page, N.I.=Not Incl. Manufacturer, Model no. on label. | Power supply +24V, Cable 24V always used for both types. | CCFL display cables are mounted from supplier. | |

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>905-052 Optrex, p. 21 T-51750GD065J-FW-ADN</td>
<td>CCFL or LED</td>
<td>Power supply +24 V Cable +24V Board 052</td>
<td>2 on display</td>
<td>PAR</td>
<td>VGA</td>
</tr>
<tr>
<td>905C065 NEC, NL10276BC20-04 10.4” N.I.</td>
<td>CCFL or LED</td>
<td>Power supply +12 V Power supply +5 V Cable XL Board XL</td>
<td>1 on display</td>
<td>065 XL10</td>
<td>XGA</td>
</tr>
<tr>
<td>-------------</td>
<td>-----------------</td>
<td>-------------------------------</td>
<td>----------------------</td>
<td>---------------</td>
<td>---------------</td>
</tr>
<tr>
<td>905C066</td>
<td>12.1” N.I.</td>
<td>CCFL or LED</td>
<td>1 on display</td>
<td></td>
<td>XGA</td>
</tr>
<tr>
<td>NEC,</td>
<td></td>
<td>Power supply +12 V</td>
<td></td>
<td>066 XL12</td>
<td></td>
</tr>
<tr>
<td>NL10276BC24-13</td>
<td></td>
<td>Power supply +5 V</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Cable XL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Board XL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>905-097</td>
<td>12.1” N.I.</td>
<td>CCFL or LED</td>
<td>2 on display</td>
<td></td>
<td>XGA</td>
</tr>
<tr>
<td>Mitsubishi,</td>
<td></td>
<td>Power supply +12 V</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AA121XJ01</td>
<td></td>
<td>Power supply +5 V</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Cable 097</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Board 097</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>905-098</td>
<td>6.5” p.20</td>
<td>LED</td>
<td></td>
<td>LED2</td>
<td>VGA</td>
</tr>
<tr>
<td>Optrex,</td>
<td></td>
<td></td>
<td></td>
<td>PAR</td>
<td></td>
</tr>
<tr>
<td>T-55465D065J-LW-A-AAN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>905-112</td>
<td>10.4” p.22</td>
<td>LED</td>
<td></td>
<td>LED2</td>
<td>SVGA</td>
</tr>
<tr>
<td>Optrex,</td>
<td></td>
<td></td>
<td></td>
<td>LVDS</td>
<td></td>
</tr>
<tr>
<td>T-55563D104J-LW-A-AAN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>905-116</td>
<td>12.1” N.I.</td>
<td>LED</td>
<td></td>
<td>LED2</td>
<td>XGA</td>
</tr>
<tr>
<td>Mitsubishi,</td>
<td></td>
<td></td>
<td></td>
<td>LVDS</td>
<td></td>
</tr>
<tr>
<td>AA121XK01</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>905-117</td>
<td>10.4” N.I.</td>
<td>LED</td>
<td></td>
<td>LED2</td>
<td>XGA</td>
</tr>
<tr>
<td>Mitsubishi,</td>
<td></td>
<td></td>
<td></td>
<td>LVDS</td>
<td></td>
</tr>
<tr>
<td>AA104XDD02</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>905-120</td>
<td>15” N.I.</td>
<td>LED</td>
<td></td>
<td>LED4</td>
<td>120</td>
</tr>
<tr>
<td>Mitsubishi,</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>XGA</td>
</tr>
<tr>
<td>AA150XT01</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Appendix H – Test instruction

Test setup procedure

Fig. H1. General test setup. Inputs VGA, SVGA and XGA have pull-up function within FPGA.

Fig. H2. Example of test setup (905-116: LED, LVDS, turned opposite way compared to Fig. H1).
The setup of a display for test will after the following procedure be a version of Fig. H1, the connections are depending on display model.

Fig. H2 shows an example of test setup: for the display 905-116.

Find the display model to be tested in App. G, column “TFT display”. Then perform the following action points, referring to the columns with same number and name in App. G. (If needed, extra information about the test equipment is found App. F.)

1. Carrier board
Check the type of carrier board needed, which is one of 3 alternatives:
- Backlight supply for CCFL needed
- Backlight supply for LED needed
- Type does not matter if special CCFL backlight supply shall be connected in 2.

Connect the cable from “COLOUR” push button at overlay to connector J3 at back of the correct carrier board, see Fig. H3. There is another identical connector on the board. J3 is the one closest to the board corner.

*Fig. H3. Connection of push button for colour change.*
2. Special CCFL backlight supply
- For displays with LED backlight this is not relevant, proceed to 3.
- For displays with CCFL backlight, check if special backlight supply is needed. If so, find the right equipment and connect it to the turned off power supplies, but not yet to the display.
The 3 different equipments for special backlight supply are seen in Fig. H4.

**Note:** For the display board 097 this is a **risky moment**, because there is a technical possibility to connect the ribbon cable turned the wrong way. But the cable should always be connected to the board, the way as in Fig. H4.

![Cable +24V and Board 052](image1.png)
![Cable and board XL](image2.png)
![Cable and board 097](image3.png)

*Fig. H4. Special CCFL Backlight supply. Names as in App. G.*

3. Backlight cable(s)
- Displays with CCFL backlight:
The backlight cables, 1 or 2 depending on display model, are mounted on the display at arrival. Connect them to the correct board. **Note:** For 095-052 this is a **risky moment**, because there is a technical possibility to connect to the carrier board. That would generate too high voltage. Instead, the backlight shall be connected to the special base board 052.
- Displays with LED backlight: Find the correct cable and connect it from carrier board to display.
Examples of both types of backlight and video connections are seen in Fig. H5.

![Fig. H5. To the left 905-057 with CCFL and LVDS, to the right 905-098 with LED and Parallel.](image)

4. Video cable

Find the correct cable and connect it from carrier board to display.

- For **parallel** video data there is only one type of cable, marked “PAR”.
- For **LVDS** video data, there are 4 different cables.

**Note**: For **all displays using LVDS** transmission this is a **risky moment**, because there are technical possibilities to make mistakes. To avoid damaging display or carrier board, this is very important:

- Use correct LVDS cable.
- Turn the LVDS cables “065 XL10” and “066 XL12” the correct way; carrier board end vs. display end.
- Turn LVDS cable “120” with cut-off sides of display connector with metal pins upwards. Also: if the unmodified connector is connected to the display it can cause bad connection. End towards display is marked “120 DISPLAY”.

The LVDS connectors for displays 905C065 and 905C066, for which the cables “065 XL10” and “066 XL12” are used, have another pin configuration, also compared to each other, see App. E and F. These cables are marked on yellow areas at the cable ends and in the middle, and should be kept in a special place. If they are used for other displays, similar damage might be done as if a standard LVDS cable is used on 905C065 and 905C066.

5. Resolution

Set the correct resolution by grounding 1 of 3 positions with a jumper at the double pin list on the carrier board (in Fig. H1 and H2 the resolution XGA has been set):

- **V** for VGA = 640 × 480 pixels
- **S** for SVGA = 800 × 600
- **X** for XGA = 1024 × 768

Finally, connect the power cable 24V from the turned off power supply for +24 V to connector X1 of carrier board, as seen in Fig. H2. Now the test setup is finished and the pixel test procedure can start.
Pixel test procedure

I.
Turn on the 24 V power supply and if needed also the power supplies for 5V and 12V.

II.
After a while the display should show the colour red. Inspect (at normal ambient light and distance, straight in front of the display) if some pixel has different colour. Press the overlay button below the marking “COLOUR” once. A colour change is done for every push of the button in a loop: red – green – blue – black – white – red and so on. Repeat the inspection for every colour.

III.
When the test is finished, turn off the power supplies and disconnect the display. If more displays of the same article number shall be tested, just connect the next display to the same backlight cable/connectors and video cable and proceed from point I. If a display with another article number shall be tested, repeat the test setup procedure.