# EITF40 Digitala och analoga projekt

# 68008-VGA generator

Christian Grosse, nbi97cgr

2017-03-10

# Abstract

The goal of this project was to build a VGA generator based around a Motorola 68008 CPU. The system contains the CPU, a 10 MHz clock module, SRAM chip, EEPROM chip and 2 PAL (22V10) circuits. One of the PAL circuits is used for address decoding. And the second chips functions as an interface to a virtual VGA-chip. The system should generate a standard VGA resolution, 640x480 with a refresh rate of 60 Hz. Unfortunately, due to time constraints, the VGA part of the project was not completed on time.

# Table of contents

# 1. Introduction

This report presents the work done during spring-2017. Supervising teacher was Bertil Lindvall. The main goal of the project was to give the student experience with electronic components, tools and software, and to document the work done. At the end of the project a functioning prototype was presented and discussions with other students was conducted.

# 2. Project

The main purpose of the project was to construct and program a prototype within the bounds of the course, and according to a specification created by the student.

After some time spent on researching the VGA protocol, the goal of this project was decided to be a prototype for a simple VGA signal generator. The prototype was based on a Motorola 680008 CPU, with the added goal of generating all timing and VGA signals in software.

## 2.2 Specification

After a project goal was set the second important part was to create a specification and a project execution path that was followed during the project. The following list of desired project sub-goals was created:

- A VGA mode of 640x480 at 60 Hz is to be generated.
- The construction will have SRAM 512k.
- The construction will have EEPROM 128k.
- The construction uses a PAL (22v10) for address decoding.
- The construction uses a PAL (22v10) as a simple VGA chip interface.
- All signals and timing will be generated in software, 68000 assembly.

## 2.3 Project execution

The construction and testing of the prototype was divided into 3 phases in order to break down the total amount of work into smaller tasks and give a better overview of what needed to be done:

- Phase 1: Create a 68008 free-runner circuit. (See appendix A).
- Phase 2: Add EEPROM, SRAM and address decoding PAL circuit.
- Phase 3: Add virtual VGA (second PAL circuit).
- Phase 4 Add an R-2R ladder for analog red, green and blue signals.

In phase one a free-runner circuit was created and no software is needed. By connecting all data ines to ground, hardwire the control signals to default states and connect a clock source, the CPU will act like there is a memory present and that occupies the entire address space. The CPU will initialize the program counter to 0 and start executing the instruction 0x0000 which is ori.b #0,d0, over and over. The CPU will advance the PC after each instruction, this will count up the address pin signals, and by adding a LED to A19 a 2-3Hz blinking can be observed. This step was used to verify that the Motorola 68008 CPU was functioning correctly.

In phase two, some smaller assembly programs were developed to verify that the construction was working properly, and that the address decoding worked as expected. Also, programming of the PAL circuit was done.

In phase three, the second PAL circuit was added and in this step the actual assembly program that will handle timings and generation of VGA signals, was created.

In phase four, a simple R-2R ladder was constructed, to generate the analog values red, green and blue.

## 2.4 Hardware

The circuit diagram for the VGA generator can be found in Appendix A. Since all timing and generation of signals is done in software the amount of needed hardware components are very limited. The system can be split into smaller modules. CPU, reset, clock, SRAM 512k, EEPROM 128k, PAL circuits (address decoder and virtual VGA interface) and R-2R ladder (for analog signal generation). The CPU, a Motorola 68008, was overclocked with 2MHz from 8 to 10 MHz.

# 3. VGA

The VGA standard contains both All Points Addressable- (pixel addressable or resolution) and alphanumeric- (text) modes.

The signals of VGA that creates an image are not many and simple to understand. The signals are analog signals Red, Green and Blue, and the digital signals vertical blanking and horizontal blanking. The most difficult part is to generate all signals according to a specific timing diagram.

In this project all signals are generated in software, in an assembly program in EEPROM, which is executed by the Motorola 68008 CPU. The analog signals are generated with the help of a R-2R ladder.

Unfortunately the 68008 CPU is not made for fast simple data transfer because of the high cycle count per instruction. This means that the ability to generate a lot of pixels really fast is very limited. For example to generate a 640x480 pixel resolution a pixel clock of 25.175 MHz is needed and this also demands that pixel data can be generated each cycle.

## 3.1 VGA timing for 640x480 resolution mode.

In order to generate a stable VGA signal in software, each cycle of each line and frame has to be accounted for, to make sure the signal does not drift or simply not get shown. Each frame contains 419200 cycles and each second has 60 frames. This is a lot of cycles and losing one might kill the image.

A frame is constructed by an active part and a blanking part. The active area contains actual pixel data and the blanking part is legacy (and more or less wasted time) from the time when a CRT display needed time to move the cathode ray to a new position. Modern displays do not have a cathode ray and does not need the time in the blanking part, but it is kept to keep the standard and make sure older displays works.

About 20% of the frame or about 100000 cycles are free time for the CPU, that can be spent on other things than generate the actual signals. Each line has 800 cycles and the first 640 are pixel data. After that there is the horizontal front porch for 16 cycles, then comes the horizontal sync signal 96 cycles (active low) and last the horizontal back porch for 48 cycles.

This line structure is then repeated for a total of 480 lines. Each frame has 514 lines, but only 480 is displayed, the rest is vertical blanking time. After the final line that is displayed (line 480) comes the vertical front porch for 11 lines.

Then comes the vertical sync signal for two lines and finally the vertical back porch. Each frame is generated at a rate of 60 Hz.

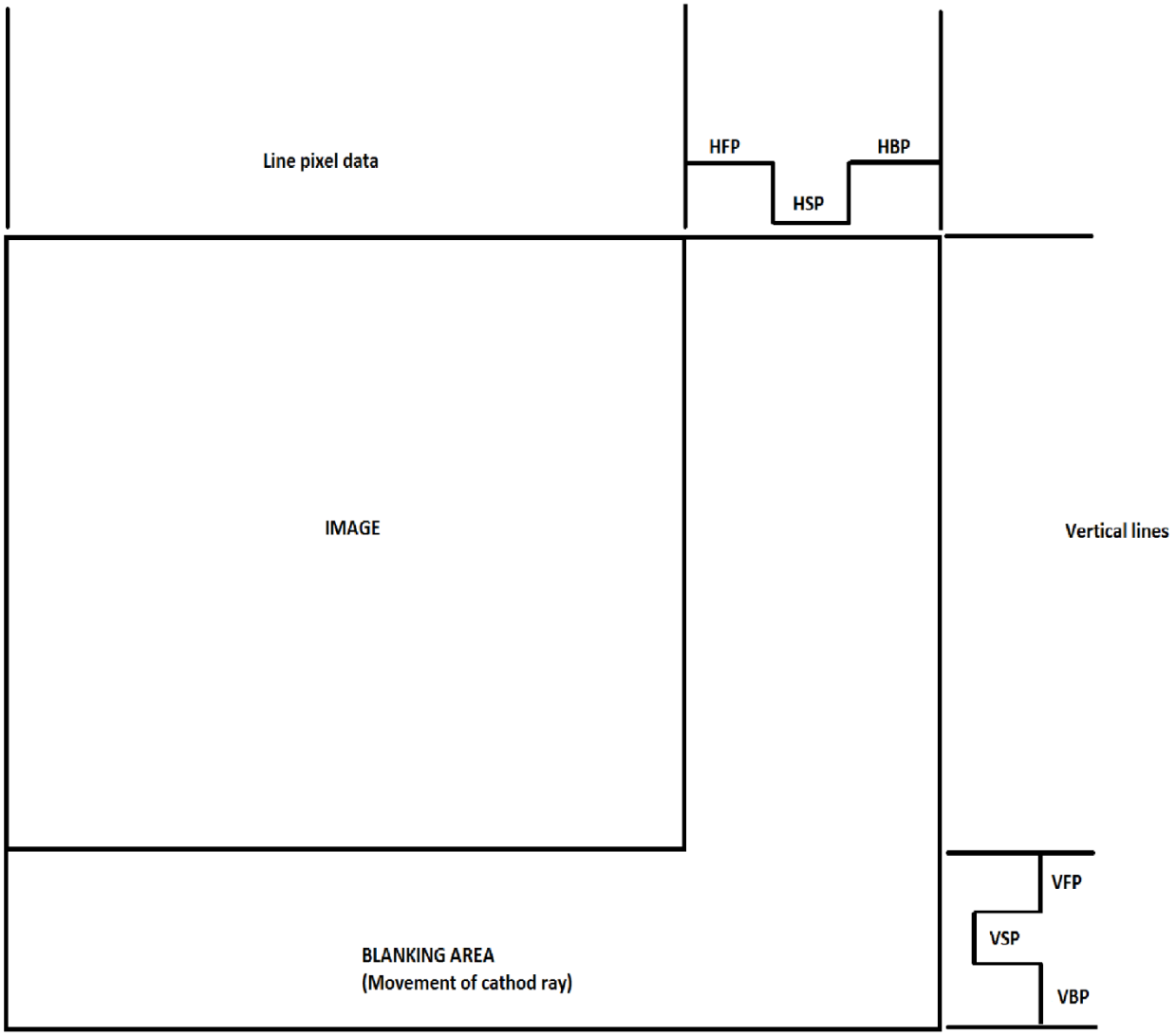


**Diagram 1. How a frame is generated in VGA.**

# 4. Results and discussion

This course has helped to develop a greater understanding of how to develop hardware and software in unison. After a long learning process, with a lot of new elements, such as tools, software, components and accompanying datasheets, a final construction was completed.

During this project different tools were used to accomplish the task of constructing a prototype. Main tools used were PAL assembler for the PAL circuits. EASy68K assembler and EASYBIN tool for 68000 assembly programming and binary file generation. For programming of circuits, such as PAL and EEPORM, a programmer, SmartProg2, with software from Elnec was used. One very valuable tool was the Saleae logic probe (with PC software) that hade the possibility to measure 4 digital signals at the same time.

As it turned out the 680008 is not a good choice to act as central part of a VGA generator. And unfortunately, due to time constraints the software part of the VGA generator (assembly code for signal generation), was not finished in time but will be concluded at a later date.

# Apendix A-Circuit diagrams

//TODO:

# Appendix B-Program listings

```
device 22V10
AS    1     'Address strobe, active low'
DS    2     'Data strobe, active low'
RW    3     'Read/Write'
A15   7
A16   8
A17   9
A18   10
A19   11
GND   12
DTACK 14
CSDIOD      15

WERAM 19
OERAM 20
CSRAM 21

CSROM 22
OEROM 23
VCC   24

start
CSRAM /= /AS * A17 * /A18 * /A19 +
         /AS * /A17 * A18 * /A19 +
         /AS * /A17 * /A18 * A19 +
         /AS * A17 * A18 * /A19;

OERAM /= /DS * RW;
WERAM /= /DS * /RW;

OEROM /= /DS * RW;
CSROM /= /AS * /A17 * /A18 * /A19;

DTACK /= /CSROM + /CSRAM;
End
```

**Listing 1. RAM-ROM address decoder PAL equations.**

```
device 22V10
CLK   1     'Address strobe, active low'
A18   2
A19   3
D0    4
D1    5
D2    6
D3    7
D4    8
D5    9
D6    10
D7    11
GND   12
AS    13
CS    14

VBL   15
HBL   16
B0    17
B1    18
G0    19
G1    20
R0    21
R1    22
DTACK 23
VCC   24

start
CS = /AS * A18 * A19;
R0 := D0;
R1 := D1;
G0 := D2;
G1 := D3;
B0 := D4;
B1 := D5;
HBL := D6;
VBL := D7;

DTACK /= /AS * A18 * A19;

End
```

**Listing 2. VGA PAL equations.**

**//TODO: Assembly listing.**

# References

Elnec: https://www.elnec.com/en/

Saleae: https://www.saleae.com/?gclid=COWrysvTw9ICFcWkGAodymgAIQ

PAL Asm: http://www.eit.lth.se/fileadmin/eit/courses/edi021/PDF_files/Pldasm.pdf

EASy68k: http://www.easy68k.com/

VGA timing: http://www.tinyvga.com/vga-timing