July 01, 2013 Severity: Information Eugeny Brychkov, RU

GR8BIT as a digital oscilloscope

Introduction

Background: In the course of development and testing of the GR8BUS master board, we faced the issue of finding out how VDP (video display processor) works *in backend* in order to implement its emulation within FPGA chip with output to the VGA monitor. Many people were queried for information, but it appeared that none of them have reliable information on how VDP accesses VRAM (video random access memory), and what is its memory access protocol and sequence.

Just using regular oscilloscope does not help much – not only depicted waveform images are required, rather than analytical information on what exactly happens on specific events at the VDP's backend.

Solution: For this purpose we decided to develop additional hardware and software, which allows reading and logging waveforms of multiple signals during specific timeframe, and provides it for further analysis. This oscilloscope can be used in other situations, having number of simultaneous parallel inputs only limited by the available and unused FPGA chip pins, and on-card available memory.

Note: all the sample files, programs and screenshots discussed in this document, are available for download from our site http://www.gr8bit.ru in Downloads section within *GR8BUS master board configuration modules* file.

1. The idea of the implementation of the oscilloscope

We want to have a complete picture how VDP works at the backend, thus sampling all changes of the signals of interest, and these signals include:

- $\overline{\text{RAS}}$ signal (row access signal), input to the DRAM memory chips, used to select row for further data read or for specific row refresh;

- $\overline{\mathrm{CAS0}}$ and $\overline{\mathrm{CAS1}}$ signals (column select signals 0 and 1), inputs to the DRAM chips, used to select specific column in the memory array for further reading or writing;
- BLNK signal (blanking) activated on blanking interval of the screen row display and serving as a marker for displayable line start;
- AD7...AD0 signals (address lines), which are multiplexed address information for DRAMs, selecting column or row of the memory array, depending on the condition of the $\overline{\mathrm{RAS}}$ and respective $\overline{\mathrm{CAS}}$ signals.

We need to sample all the changes of the signals above – except address line signals because they are only valid on activation of $\overline{\mathrm{RAS}}$ and $\overline{\mathrm{CAS}}$ signals.

As soon as we sample the data, we get it from the GR8BUS master board's memory buffer to the GR8BIT storage device, and load it to the PC into the excel spreadsheet, which will then analyze data, and at best draw waveforms of the signals under consideration and inform us about the pattern VDP accesses its VRAM.

2. The hardware part

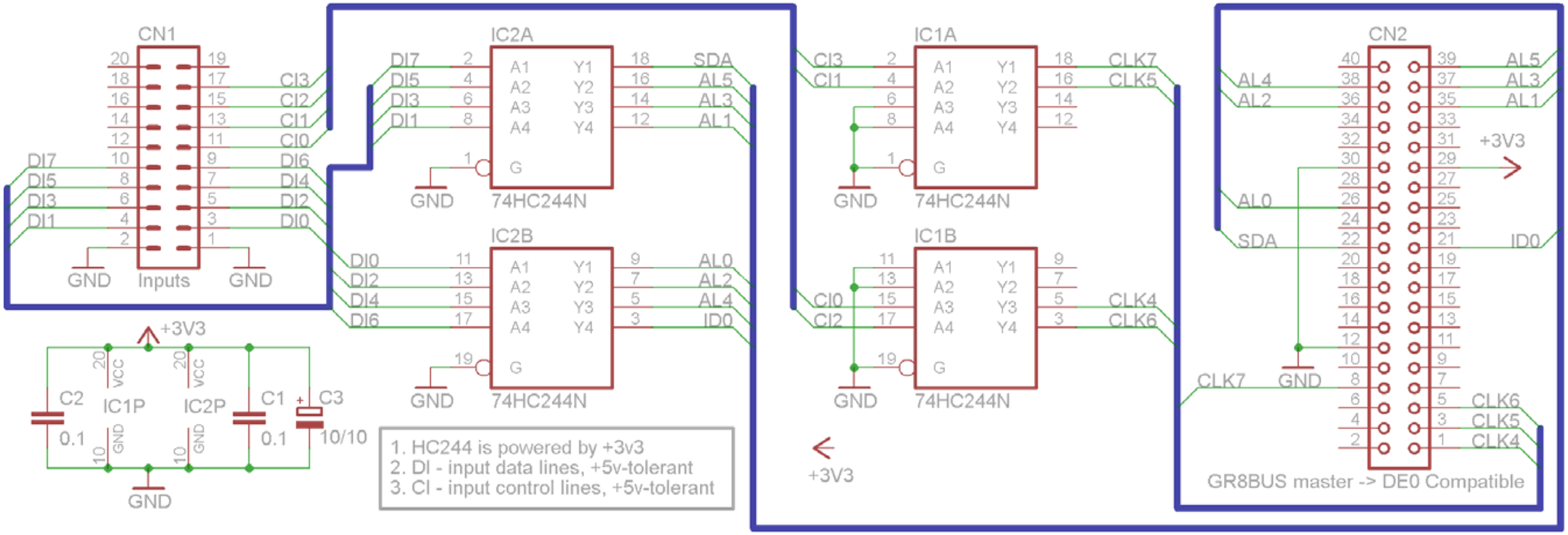
Schematic of the oscilloscope adapter board and cable are shown on fig. 1 and fig. 2 respectively.

As VDP has 5V signal level standard, we are unable to connect it directly to the FPGA chip input pins, which work as 3V3 level standard. 74HC244 chip is used as unidirectional signal level converter: it is powered by 3V3 supply from master board, thus has 3V3-level compliant outputs; at the same time this chip tolerates 5V-level inputs, at its 3V3 supply voltage treating levels of 5V TTL signal levels from VDP properly (both 0.8V of TTL low level and 2.4V of TTL high level match CMOS input level threshold at 3V3 scale).

All the gates of HC244 chip are always enabled, passing logical levels from the inputs (CI and DI) to the selected inputs of the FPGA chip (AL, SDA, ID0, and CLK). Control signals (CI) are passed to FPGA's CLK inputs to be used as trigger clocking signals in its configuration, data signals (DI) are distributed to the AL channel (audio DAC left channel, 6 bits) and SDA/ID0 signals (VGA monitor control signals). This distribution is deliberate, allowing still using master board's VGA signals for VGA monitor control (except SDA/ID0 signals) so that advanced FPGA configuration can display picture onto the monitor on-the-fly.

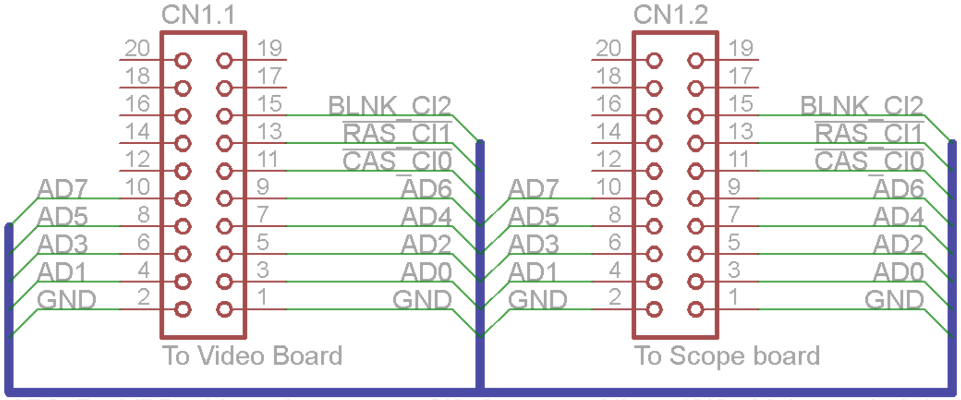
Cable connecting oscilloscope adapter board and source of the signals should be as short as possible, otherwise parasitic oscillations, reflections and noise in the wires, and will require termination (see *Single ended termination*).

Figures 3 and 4 show our prototype installed, and the location of connector installation on the video board. Figure 5 shows the wiring of the adapter board.



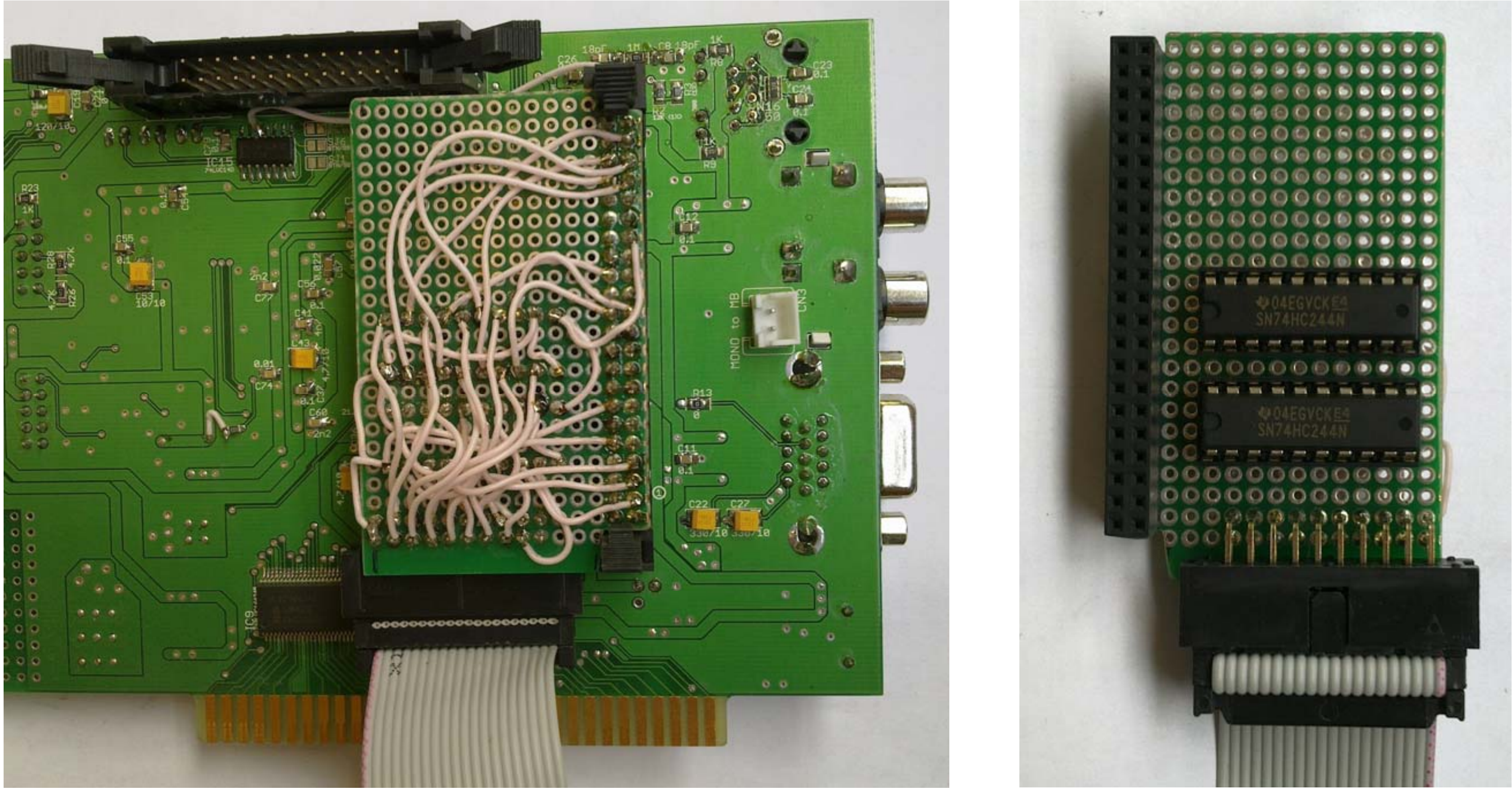
1. HC244 is powered by +3v3
2. DI - input data lines, +5v-tolerant
3. CI - input control lines, +5v-tolerant

Figure 1. Schematic of the oscilloscope adapter to be inserted into the GR8BUT master board’s expansion connector



AD0..7 = VDP address bus CI0..2 = control lines (CI0 - highest priority)

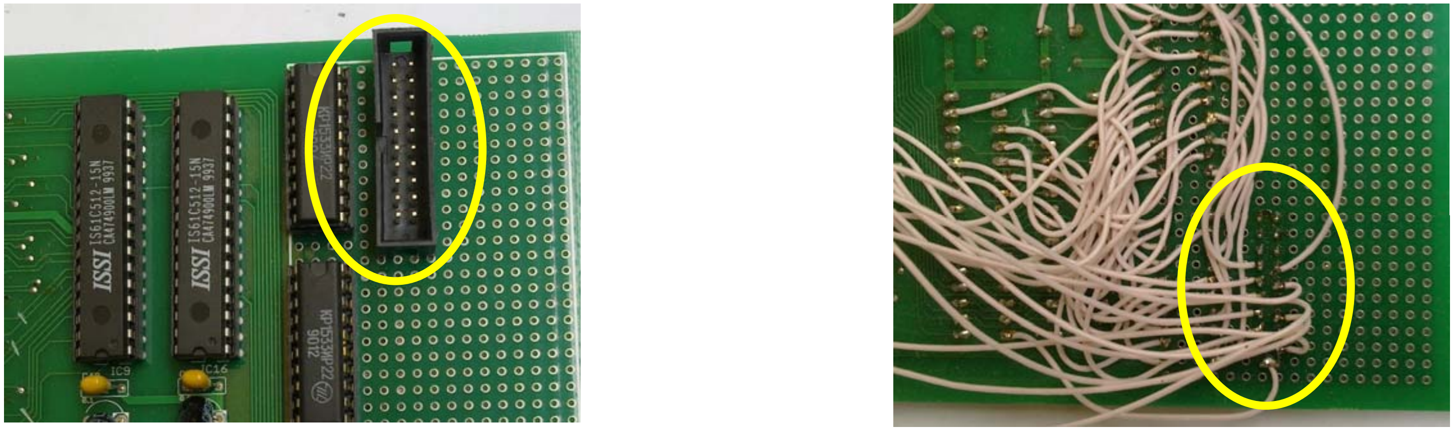
Figure 2. Schematic of the cable which interconnects video board and oscilloscope adapter board



(a) GR8BUS master board with oscilloscope adapter board (solder side) installed into the vertical connector

(b) Oscilloscope adapter board (component side)

Figure 3. Prototype of the oscilloscope adapter board



(a) Component side

(b) Solder side (view of our prototype board with some other modifications introduced)

Figure 4. Connector with source signals installed on the video board

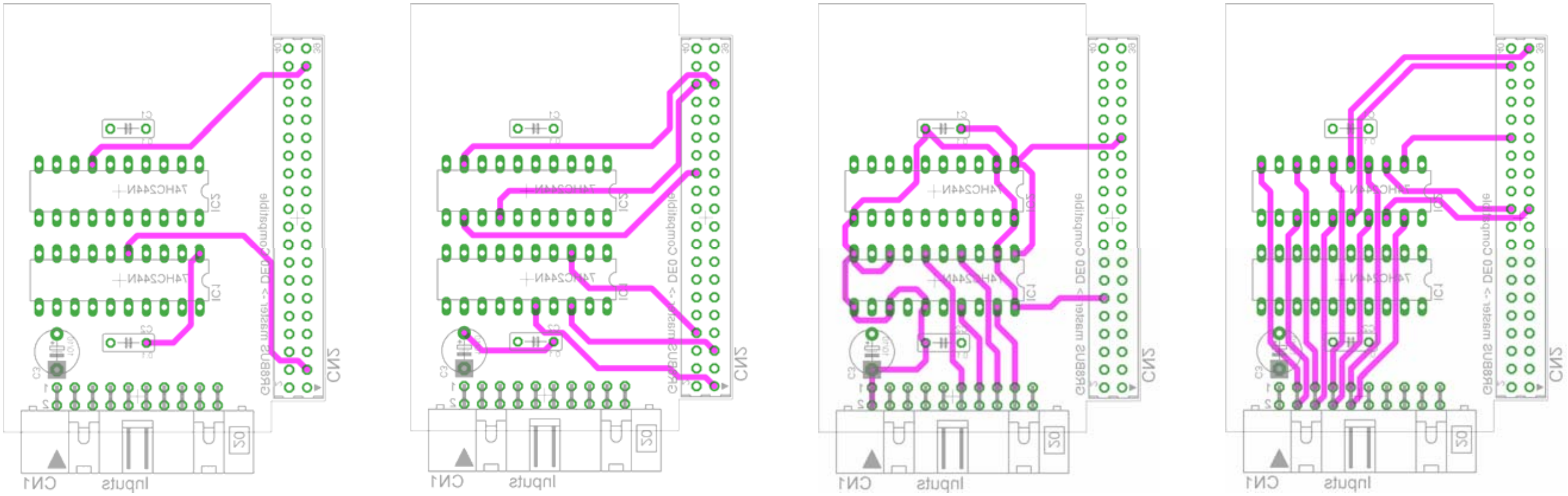


Figure 5. Scope adapter board wiring (solder/wiring side)

3. The FPGA configuration

FPGA configuration files consist of three parts: scope plugin, RAM buffer definition and PLL definition. Scope plugin configuration is to be included into the main project using "`include" command (do not forget to set "Verilog include file" type in the Quartus's *Setting* section.

Configuration is driven by the clock with short period sufficient to register every change on the control lines. As per VDP datasheet, minimal time between /RAS and /CAS signal activations is 40ns, and our test in SCREEN1 mode showed minimal value between states is 7 cycles of 7.759ns (in total about 54ns, see sample file *DATA_SCREEN1_29062013.TXT*).

Registration of the data on the change of control lines is performed in asynchronous way because minimal address hold-time on /RAS activation is just 12ns, and scope may miss the stable VDP address information in case it will be latched in synchronous way to PLL clock if its period is above 12ns.

4. The software part

Software part consists of two entities: Z-80 program which downloads collected data from the FPGA, and PC application based on Microsoft Excel Visual Basic which analyzes downloaded information and draws waveforms.

4.1. Data download program

Program called "SCOPE18.MAC", compiled into "SCOPE18.BIN", is an MSX-BASIC binary executable file, loadable with BLOAD command (execution start address is 0C000h). Initial implementation of the program was in pure BASIC, but it showed unacceptable level of performance, and we moved to pure assembler code in order to speed up port access, buffering and disk I/O operations. The source and executable are available for download from our website.

The download program works through API provided by structures configured into FPGA chip on GR8BUS master board. At the beginning it ensures that scoping is stopped resetting bit 7 of the port 34 (*samping_start*). Then it starts by setting this bit, and waiting for *sampling_end* to set. Then program reads final elapsed scoping time *cur_time* through ports 38-3B, and only then resets sampling (*samping_start* reset thus elapsed time gets reset). Then program is able to read records of data collected by setting record size in ports 30-31, and reading data from ports 32-24. All data is written into the file with name "DATA" created on the current drive.

To run the program, you set VDP into desired screen mode, fill its VRAM (e.g. drawing on the screen and setting sprites) and then either bload the program with ",r" option, or perform "DEFUSR=&hC000:A=USR(0)" if it was bloaded beforehand.

4.2. Excel-based analysis program

This application is purely PC-based, running Excel with Visual-Basic capability. The purpose of application is to process data downloaded by previous program, identifying timing of the control signal changes, and actual operations VDP performs, i.e. reads from VRAM and blanking.

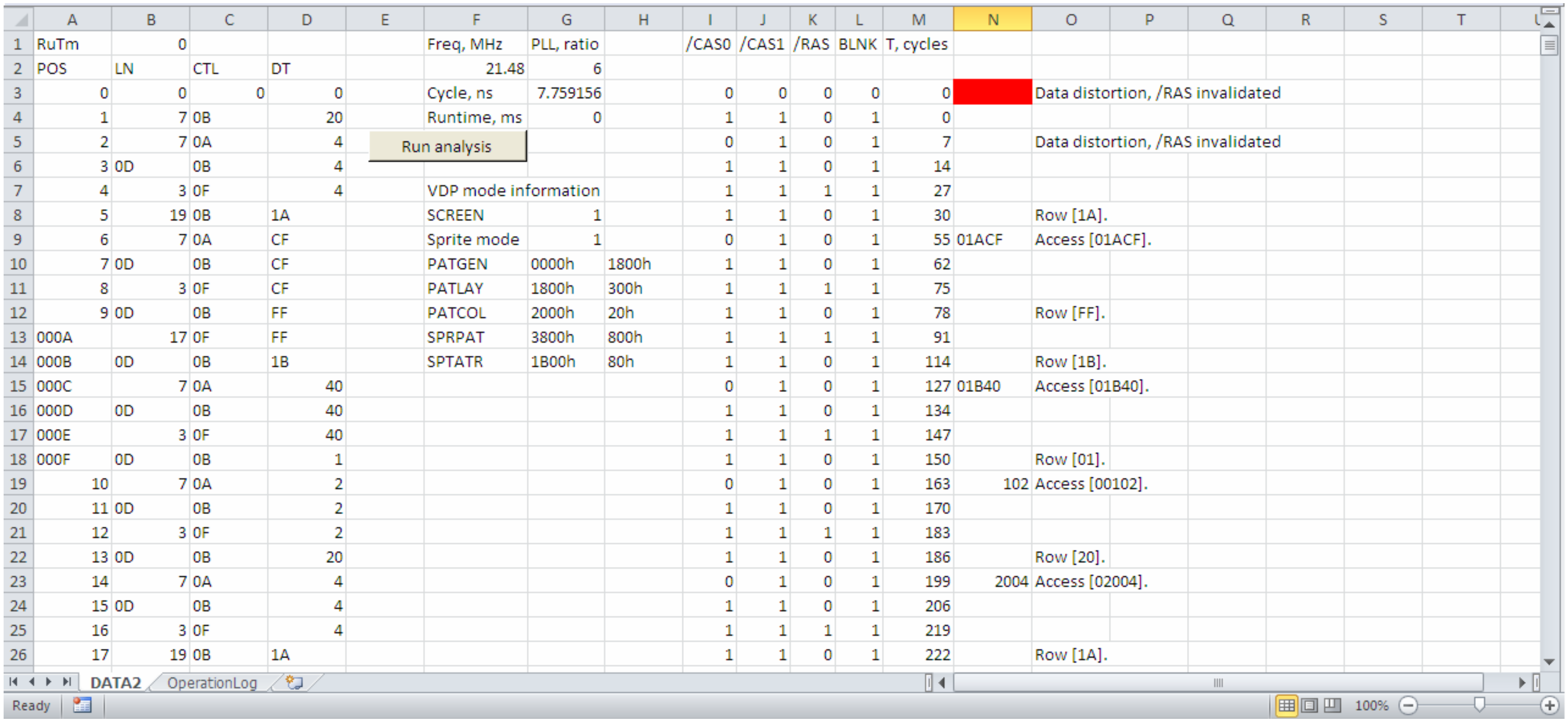
Figure 6 shows screenshot of the output of the analysis program (file: scope18_29062013.xlsm). Columns A-D are raw data, columns I-L are states of the signals decoded from column C, column N indicates VDP access of the address specified in the cell, and column O is a verbal explanation of the operation and other diagnostic information. If you will look into the file itself, which is 8400h (33794) lines length, you will that somewhere at file's line 10020 screen display operation ends and VDP enters screen blanking state (beyond the log file). Important information about blanking interval is that VDP performs regular VRAM chip refreshes, and sometimes performs memory reads.

Quick look at the display period will reveal that VDP reads from location 100-107h, which is decimal 256-263 and represents patterns of the *space* character in the VDP pattern generator table – at the time of scoping bottom of the screen was blank – filled by spaces (character #32 decimal).

We also can see that from the scoping start VDP sequentially reads VRAM from 1ACFh (row 9) till 1AFFh (row 9948); this area represents the pattern layout of the screen (range 1800h-1AFFh) in SCREEN1 mode.

Throughout whole cycle of the line processing (from one line blanking period till next one, for example lines 8673-9420), VDP reads Y-coordinate of every sprite (1B00h, 1B04h, 1B08h etc.), most probably identifying if sprite to be displayed and preparing for next line sprite image display. During this test we refer to, we had sprites enabled, but no active sprites displayed, so it is not possible to see from the log how and when exactly VDP reads pattern of the sprite for display. We will have to make another sampling for this purpose.

On the second tab called *Operation log* of the Excel sheet you can see the data and graph for the period starting from relative time, identified by value *oplogstart* of the VB script, for 8000 clock ticks. Scaled picture of the graph is shown on fig. 7. Nice, isn't; it?



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	RuTm	0				Freq, MHz	PLL, ratio		/CAS0	/CAS1	/RAS	BLNK	T, cycles		
2	POS	LN	CTL	DT		21.48	6								
3	0	0	0	0		Cycle, ns	7.759156		0	0	0	0	0	🟥	Data distortion, /RAS invalidated
4	1	7	0B	20		Runtime, ms	0		1	1	0	1	0		
5	2	7	0A	4					0	1	0	1	7		Data distortion, /RAS invalidated
6	3	0D	0B	4					1	1	0	1	14		
7	4	3	0F	4		VDP mode information			1	1	1	1	27		
8	5	19	0B	1A		SCREEN	1		1	1	0	1	30		Row [1A].
9	6	7	0A	CF		Sprite mode	1		0	1	0	1	55	01ACF	Access [01ACF].
10	7	0D	0B	CF		PATGEN	0000h	1800h	1	1	0	1	62		
11	8	3	0F	CF		PATLAY	1800h	300h	1	1	1	1	75		
12	9	0D	0B	FF		PATCOL	2000h	20h	1	1	0	1	78		Row [FF].
13	000A	17	0F	FF		SPRPAT	3800h	800h	1	1	1	1	91		
14	000B	0D	0B	1B		SPTATR	1B00h	80h	1	1	0	1	114		Row [1B].
15	000C	7	0A	40					0	1	0	1	127	01B40	Access [01B40].
16	000D	0D	0B	40					1	1	0	1	134		
17	000E	3	0F	40					1	1	1	1	147		
18	000F	0D	0B	1					1	1	0	1	150		Row [01].
19	10	7	0A	2					0	1	0	1	163	102	Access [00102].
20	11	0D	0B	2					1	1	0	1	170		
21	12	3	0F	2					1	1	1	1	183		
22	13	0D	0B	20					1	1	0	1	186		Row [20].
23	14	7	0A	4					0	1	0	1	199	2004	Access [02004].
24	15	0D	0B	4					1	1	0	1	206		
25	16	3	0F	4					1	1	1	1	219		
26	17	19	0B	1A					1	1	0	1	222		Row [1A].

Figure 6. Beginning of the sample output analysis file, generated by the GR8BIT oscilloscope system

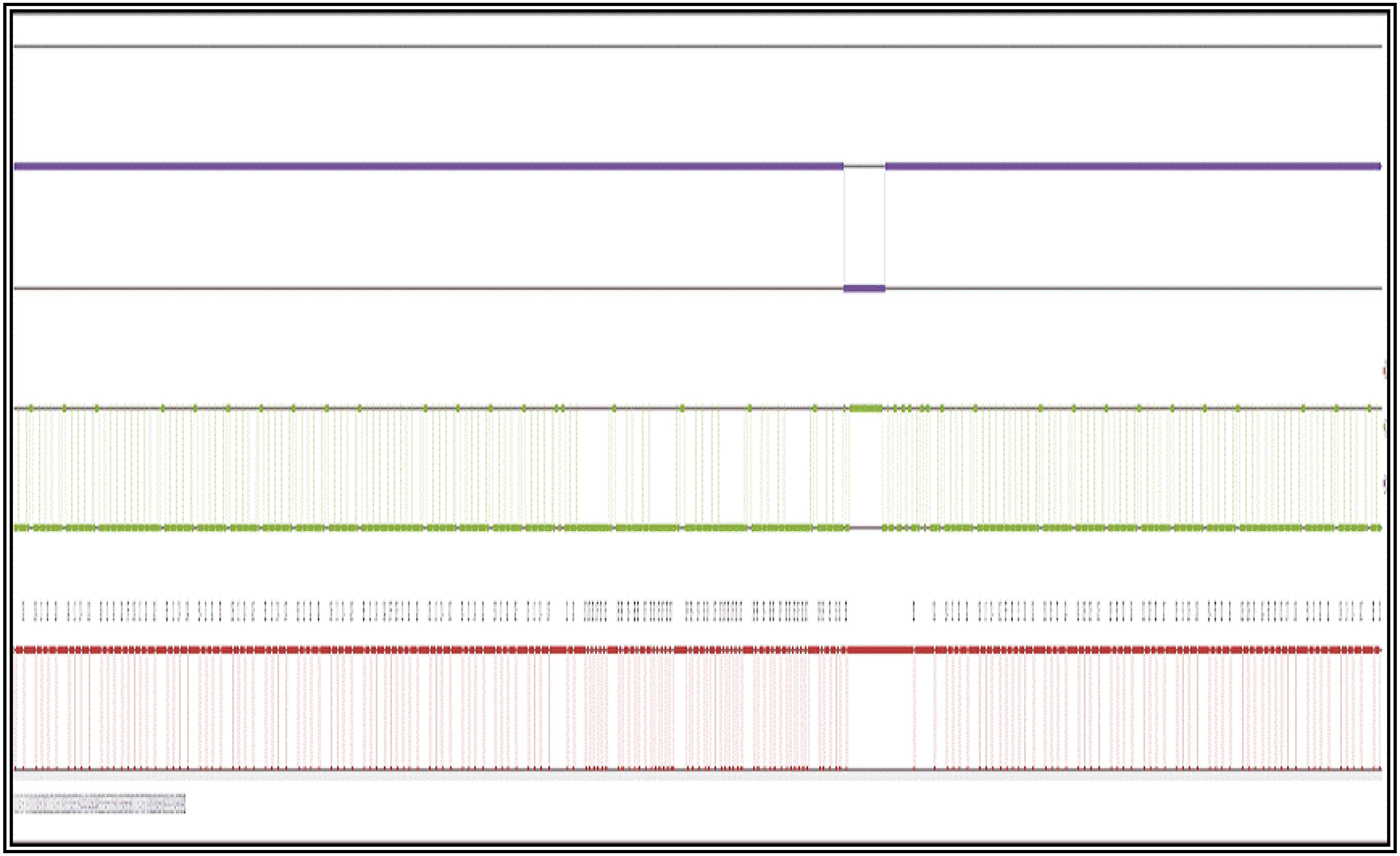


Fig. 7. An output of the Excel program for SCREEN1 mode – significantly scaled vertically and squeezed horizontally. Top (blue) signal is BLNK blanking signal, middle (green) signal is $\overline{\mathrm{RAS}}$, and bottom (red) is $\overline{\mathrm{CAS0}}$.

Sources and information for further reading:

- Brychkov, E. (ed.), ASCII Corp., Nippon Gakki Co. (2010) *V9938 Programmer's Guide*, available online at http://www.gr8bit.ru/Documentation/V9938_programmers_guide.pdf (accessed 30-Jun-2013)

End of KB0013 “GR8BIT as a digital oscilloscope”.