

This article explains the effects and graphic modes achievable when GTIA and ANTIC graphic chips are appropriately warmed up. It also allows you to understand how to avoid such effects when they are not desirable and the chips warm up spontaneously. Described effects are related to delays of internal GTIA functions by one color cycle. This description contains also some information which can be found interesting by people trying to reverse-engineer internal GTIA schematics.

## **Terminology used further in the text.**

### **Known terms:**

**PRIOR** – GTIA chip control register residing at address \$D01B. Used for selecting GTIA mode as well as for setting PMG (sprite) priorities.

**GTIACTL** – alternative name for PRIOR register, used in source code of my programs.

**GTIA9** (properly GTIA1) – GTIA mode of horizontal resolution of 80 pixels, allowing for displaying 16 levels of luminance of one hue, obtained by writing \$40 (OR <priority>) into PRIOR register or by GR.9 command in Basic.

**GTIA10** (properly GTIA2) – GTIA mode of horizontal resolution of 80 pixels, allowing for displaying 9 independent colors, obtained by writing \$80 (OR <priority>) into PRIOR register or by GR.10 command in Basic.

**GTIA11** (properly GTIA3) – GTIA mode of horizontal resolution of 80 pixels, allowing for displaying 16 hues in one level of luminance, obtained by writing \$C0 (OR <priority>) into PRIOR register or by GR.11 command in Basic.

**GTIAX** – any of three GTIA modes.

**NORMAL** (properly GTIA0) – a name of mode of GTIA chip for normal non-GTIA graphic modes – mode obtained by writing \$00 (OR <priority>) into PRIOR register.

**PMG** – Player Missile Graphics – the name for Atari sprites.

**Playfield** – graphics layer including colors 0-3 (COLPF0-COLPF3) but not the background color COLBAK.

**CPU cycle** – a cycle of  $\phi$ 2 CPU clock, 1.77MHz in PAL, 1.79MHz in NTSC.

**Color cycle** – a cycle of OSC clock, 3.54MHz in PAL, 3.579MHz w NTSC. The cycle has timespan equivalent to one pixel of horizontal resolution of 160 pixels. Its length is half of CPU cycle. Color cycle indexes are equal to PMG objects positions. Entire scanline consists of 228 color cycles (114 CPU cycles).

**Lores pixel** – pixel of horizontal screen resolution of 160, equivalent to one color cycle.

**Hires pixel** – pixel of horizontal screen resolution of 320, equivalent to half of lores pixel and to half of color cycle.

**HBLANK** – invisible part of scanline including color cycles 222-227 and 0-33 (CPU cycles 111-113 and 0-16), 40 color cycles (20 CPU cycles) in total. During HBLANK PMG collision detection does not work.

**ACTIVE DISPLAY** – visible part of scanline including color cycles 34-221 (CPU cycles 17-110), 188 color cycles (94 CPU cycles) in total.

**VBLANK** – the time of vertical blanking comprising blank scanlines 248-311 in PAL/SECAM, 248-261 in NTSC and 0-7 in each of the systems. Includes VSYNC time. During VBLANK PMG collision detection does not work.

**VSYNC** – the time of vertical synchronization comprising 3 entire scanlines 275-277 in PAL/SECAM, 255-257 in NTSC.

**New terms (coined by me):**

DGM – Delayed Gtia Mode – the effect of shift (delay) of scanline content in GTIAX modes by half of pixel (1 color cycle) to the right.

DPS – Delayed Pmg and hSync – the effect of shift (delay) of PMG and horizontal synchronization pulse by 1 color clock. Due to hsync shift the visible effect is that playfield graphic is shifted to the left while PMG objects stay at normal position.

DPS2 – Delayed Pmg and hSync in every second (2) line – the effect being special case of applying DPS in every second line and giving a shift by one hires pixel to the left against normal position.

DGF – Delayed Gtia Functions – the name of the phenomenon of delayed GTIA chip functions comprising DGM and DPS effects.

DG9 – Delayed Gtia mode 9 – the mode being a result of applying DGM effect in GTIA9 mode.

DG10 – Delayed Gtia mode 10 – the mode being a result of applying DGM effect in GTIA10 mode.

DG11 – Delayed Gtia mode 11 – the mode being a result of applying DGM effect in GTIA11 mode.

**DGF effects.**

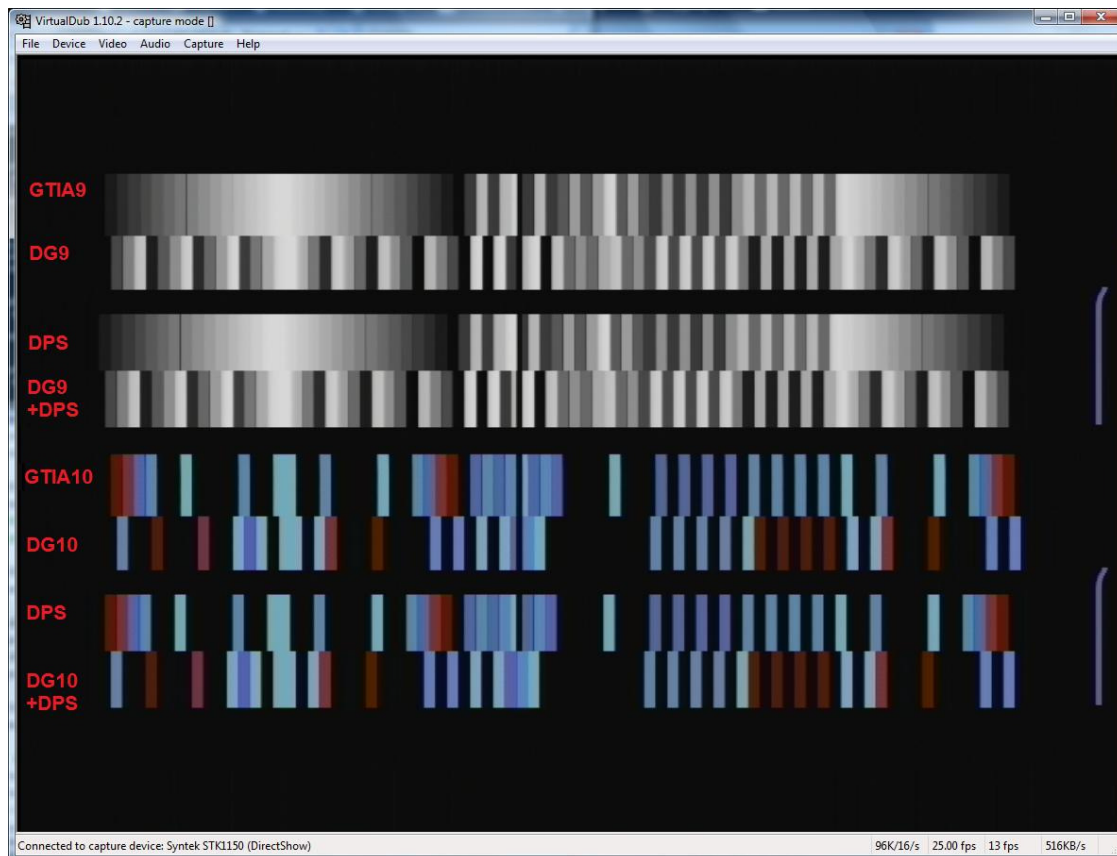
As I wrote in the introduction, this description is related to the effects of delay of GTIA chip functions by one color cycle and thereby the effects of scanline content shift by one lores pixel. These effects can be used to double the resolution of GTIA modes (80 pixels) if we apply horizontal interlace that is alternating display of two images where the line of image in one frame is not shifted and in the other is shifted by one lores pixel. Additional effect in this case is extending available color palette due to mixing of the halves of overlapping pixels of two images. The principle is the same as in HIP mode. Here, however, it is possible to mix GTIA9 lines and lores pixel shifted GTIA9 lines. In HIP it is mixing of GTIA9 and GTIA10 lines where GTIA10 line is always one lores pixel shifted against GTIA9 line. It was also found that there is a possibility of doubling the resolution of standard lores modes (160 pixels) on the same principle, that is, by alternating display of two images where the line of image in one frame is not shifted and in the other is shifted by one hires pixel. Hires pixel shift is achieved in a more complicated way than the lores pixel shift. This will be described later in the text.

There are two different ways to obtain one lores pixel shift of content of the line in GTIAX mode - DGM and DPS effects. There is a possibility to display each of them separately or both at the same time. Both effects belong to one category, which is called DGF.

A prerequisite for the occurrence of DGF is appropriate warming-up of GTIA and ANTIC chips. It is not known what the exact temperature of the chips is required but it is roughly known how much time it takes to achieve stable DGF effects at different temperatures of surrounding air. These temperatures vary between different machines and the total time needed for one machine to achieve stable DGF at the same temperature also changes. Due to the research I've learnt that at surrounding temperature of 25°C the time in minutes needed for achieving stable DGF varies between 150 and 250 and at 29°C between 30 and 40. The time needed may be reduced to 3 minutes when heating Atari with a hair dryer. In addition, to sustain a stable DGF for hours the surrounding temperature must be at minimum level of 25°C.

DPS stability may require higher temperature than the stability of DGM. DGF stability is the stability of both the DGM and DPS. It is likely that such machines exist where DGM is reachable but DPS is not.

It turns out that there are such machines, where DGF does not occur at all, that is, there is neither DPS nor the DGM. Personally, I have one such machine. This is the Atari 130XE produced in 1990, with a mainboard with 4 RAM chips and GTIA chip with defective GTIA modes. I'm not sure if DGF resistance is related to defective GTIA modes.



*DGF effects.*

## DGM.

DGM effect is that the pixels of GTIAX modes are shifted to the right by half a pixel (or one lores pixel). GTIA combines a pixel of resolution of 80 of two 2-bit words from AN1 and AN0 lines one color cycle too late, as referred to the data stream provided by ANTIC on AN2-AN0 lines. In other words, shifted pixel is combined of two least significant bits of the first pixel and two most significant bits of the second pixel. Two least significant bits of the first pixel become most significant bits of shifted pixel and the two most significant bits of second pixel become least significant bits of shifted pixel. Thus the data prepared for GTIAX modes will be displayed incorrectly when DGM effect is activated. This may be corrected in two ways. The first is to shift (delay) data provided by ANTIC by one pixel by setting HSCROL register properly (plus setting proper data address and activating horizontal shift in these lines of Display List, where DGM is enabled. The second way is just to change the image data in memory so that it is correct for DGM.

DGM effect may be enabled in any line of the screen, regardless of whether it is enabled in the other lines. In particular, the DGM may be enabled in every second line. This allows for obtaining a mode similar to the HIP but only using GTIA9.

DGM effect has been discovered in 1994 by Bryan - member of AtariAge community.

He mentions about his discovery here:

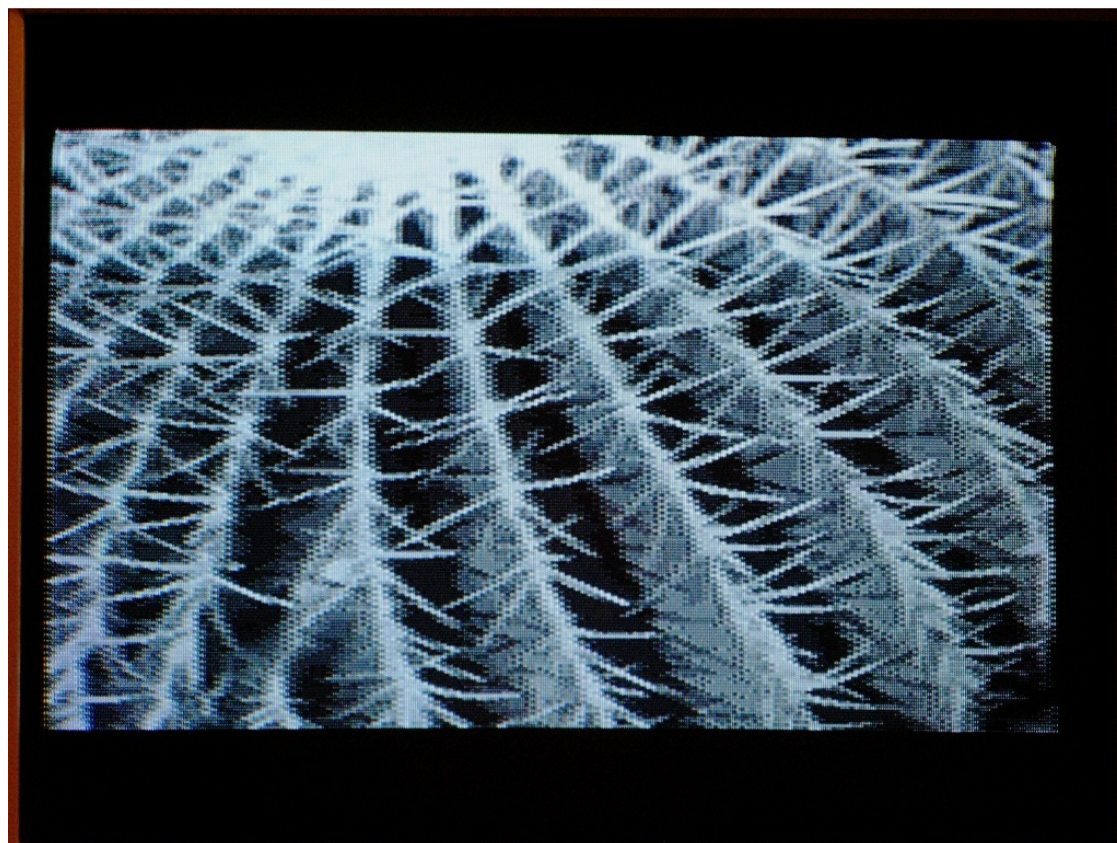
1. <http://www.atariage.com/forums/topic/136706-internal-antic-and-gtia-schematics/#entry1652806>
2. [http://www.atariage.com/forums/topic/136706-internal-antic-and-gtia-schematics/page\\_st\\_100#entry1676139](http://www.atariage.com/forums/topic/136706-internal-antic-and-gtia-schematics/page_st_100#entry1676139)
3. [http://www.atariage.com/forums/topic/136706-internal-antic-and-gtia-schematics/page\\_st\\_100#entry1677517](http://www.atariage.com/forums/topic/136706-internal-antic-and-gtia-schematics/page_st_100#entry1677517)
4. [http://www.atariage.com/forums/topic/136706-internal-antic-and-gtia-schematics/page\\_st\\_125#entry1677967](http://www.atariage.com/forums/topic/136706-internal-antic-and-gtia-schematics/page_st_125#entry1677967)

Link no. 3 leads to the post where two example pictures using DGM (precisely DG9) are attached together with screenshots from before and after the warming-up of graphics chips. Bryan called his

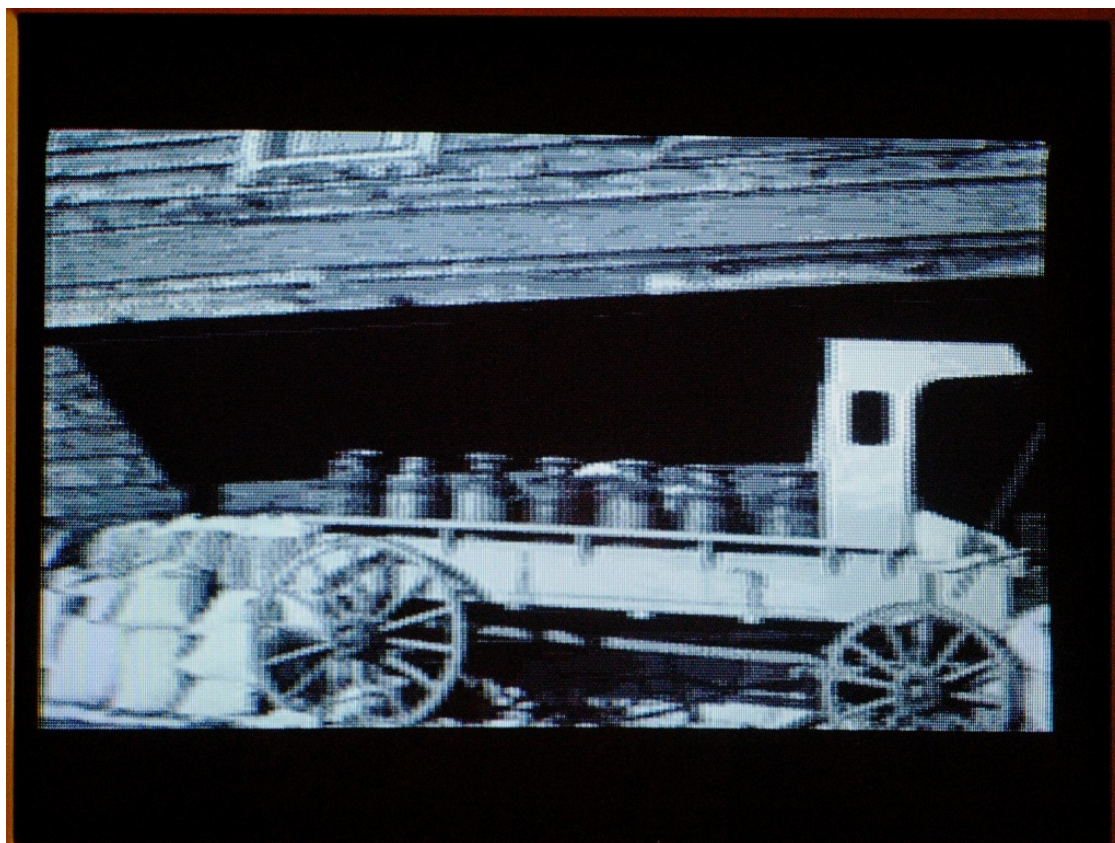
mode VZI - VertiZontal Interlacing. The mode is very similar to the HIP. In one frame the delayed by one color cycle are all the even lines and in a second frame the delayed are all the odd. As a method for correcting the data of incorrectly combined pixels Bryan uses horizontal shift with HSCROL.

Bryan's examples enable both DGM and DPS (precisely DPS2) – see paragraph "Relationships between DGF effects". DPS2 effect causes visible on the screen deflection of the image (and one hires pixel shift to the left!) but on Bryan's screenshots it is not visible. This means that on Bryan's machine the DPS was unreachable or the temperature was too low for DPS to be activated.

In Bryan's examples the DGM effect is triggered with GTIA9 mode, which on some machines significantly reduces the time of availability of the effect due to the higher temperature threshold (see paragraph "Influence of ways of triggering DGF effects on their stability").



*Picture in Bryan's VZI mode. A deflection visible at the top. Significant bottom part of the image shifted by one hires pixel to the left.*



*Picture in Bryan's VZI mode.*

## **DPS.**

DPS effect is that GTIA performs its functions i.e. overlaying PMG objects and generation of HSYNC pulse one color cycle later than normal, as referred to the data stream provided by ANTIC on AN2-AN0 lines. As a result HSYNC shift the effect on the screen is that the image is shifted to the left and PMG are at normal position.

Graphics shift effect as a result of HSYNC delay is not instant. It takes about 5 lines of the screen after activation of the delay to stabilize the shift. In these 5 lines visible is the "deflection" of the image that is a smooth transition between the normal and shifted by one lores pixel graphics or in other words between normal and shifted HSYNC pulse position. It is a feature of the display devices. I've observed this effect on several TV sets, C= 1084S monitor, AV capture card, AV to VGA converter and a projector, and I think that this feature is common to all display devices compatible with PAL/NTSC/SECAM systems. This feature causes that the shift cannot be achieved at any scanline independently of adjacent ones. Therefore the shift should be applied for entire screen or for multi-line blocks and each of them should be preceded and followed by at least 6 blank lines to hide the "deflected" areas of the image.

There is a problem related to DPS effect. In the first scanline without the DPS following the block of DPS-enabled lines PMG DMA does not work. The problem must be related to the return of GTIA from delayed to normal timing. Probably GTIA misses active edge of HALT signal that initiates the read of PMG data series from data bus.

DPS effect occurs regardless of the GTIA chip mode, that is NORMAL or GTIAX and in case of NORMAL also regardless of the mode specified by ANTIC, that is lores or hires.

I've discovered DPS effect in 2009.

## **DPS2.**

If DPS is applied in every second scanline then mentioned feature of display devices to smooth the transition from normal to shifted position of HSYNC pulse will average the shift of graphics in all lines in a position that is shifted exactly by half of lores pixel (or one hires pixel) to the left of normal

position. This effect is a special case of DPS effect. I called him DPS2. It is the basis for DGI graphics mode, which I presented in the KNIGHT demo. Please note that in the lines in which the DPS is applied (that is, in every second line) PMG objects are shifted one lores pixel to the right of desired position while in the other lines PMG are situated normally. It makes a bit complicated to use them to colorize the image. In addition, there is a problem of the lack of said PMG objects DMA which in DPS2 occurs in every second line and makes usage of PMG even more difficult. In summary, the problem of shifted PMG occurs in lines where DPS is on while the problem of lack of PMG DMA occurs in lines where DPS is off, so the two problems occur alternately in successive lines. The problem with lack of DMA may be overcome by writing data to PMG shape registers with CPU. However, this reduces the time available for other changes. While working on a KNIGHT demo it turned out that it is often possible to use the same byte of PMG shape in more than one line (in spite of pixel shift every second line), so that lack of DMA is not that onerous, but it obviously depends on particular image. In general, the development of raster program using DPS2 is very difficult.

## How to trigger DGF effects.

From programmer's perspective DGF effects are triggered in the following manner. **DGM effect** in particular line of the screen is triggered by switching from NORMAL to GTIAX within ACTIVE DISPLAY (CPU cycles 17-110). A necessary condition is that GTIAX must not be active earlier than in cycle 17. So writing to PRIOR (\$D01B) register, which causes switching to GTIAX, may occur in CPU cycle 16 the earliest. In this case the change affects GTIA in cycle 17. It does not matter what mode the GTIA was in during HBLANK. The first switch from NORMAL to GTIAX does matter and the other switches do not matter. The occurrence of HBLANK disables the DGM. This means that there is a need to trigger the DGM in each line where you want it to be visible. **DPS effect** is actually triggered in the same way, but it is visible only in the next scanline. Additional necessary condition for the DPS is that GTIAX must be active at the start of HBLANK (HBLANK preceding the line, where the effect is visible, so that the HBLANK occurring after triggering DGM). This condition is fulfilled when GTIAX is active in CPU cycle number 110 (color cycles 220 and 221). Switching from NORMAL to GTIAX must therefore take place no later than in cycle 109. The method of triggering DPS due to the additional condition is a special case of method of triggering DGM.

## Influence of ways of triggering DGF effects on their stability.

Described ways to enable DGF effects work in conditions of their full stability. When stability of DGF is achieved to considerable extent but not fully, obtaining stable DGM and DPS effects may be yet possible. Obtaining DGM requires then more than one CPU cycles in normal mode before switching to GTIAX (highest stability at 2-3 cycles) while obtaining DPS requires more than one CPU cycles in GTIAX mode before the cycle 110 and as low as possible (ideally zero) number of such cycles after the cycle 110. Following these principles, we can get stable DGF effects at some lower temperatures than those required for full DGF stability and sustain these effects for longer time. In other words, in this way we can lower the temperature threshold for the activity of DGF effects.

It matters with which GTIA mode or actually with what value written to PRIOR register we trigger DGM effect. Triggering here means the first switch from NORMAL to GTIAX within the ACTIVE DISPLAY. Activation means that the effect is really visible. There are GTIA chips for which triggering DGM with GTIA10, that is value \$80 OR <priority>, is much better, because the activation of DGM occurs at much lower temperatures than when using a different value. Activity of the effect sustains also for much longer, for example several hours at surrounding temperature of 22°C. In turn, the activity of DGM effect triggered with GTIA9, that is value \$40 OR <priority>, has a higher temperature threshold, appears much later and disappears much earlier. For example, it may last only for a few minutes at surrounding temperature of 22°C. The activity of DGM effect triggered with GTIA11, that is value \$C0 OR <priority>, has the highest temperature threshold. Therefore, the best way is to trigger DGM with GTIA10 and then possibly switch to another GTIA mode depending on which of DG9, DG10 or DG11 modes we want to display. There are also GTIA chips, for which temperature threshold of DGM effect activity is not dependent on value we use to trigger it.

In summary, to trigger DGF (DGM or DPS) it is recommended to use GTIA10 mode, that is value \$80 OR <priority>. Immediately afterwards during scanline you can change it if desired mode is different than GTIA10.

While writing programs using DGF effects it should be assumed that full stability of DGF is achieved by GTIA chip. If emulators would ever emulate DGF effects they should do it as if DGF stability has been reached fully. Full stability means that triggering DGF effect results in its immediate activation and visibility on the screen.

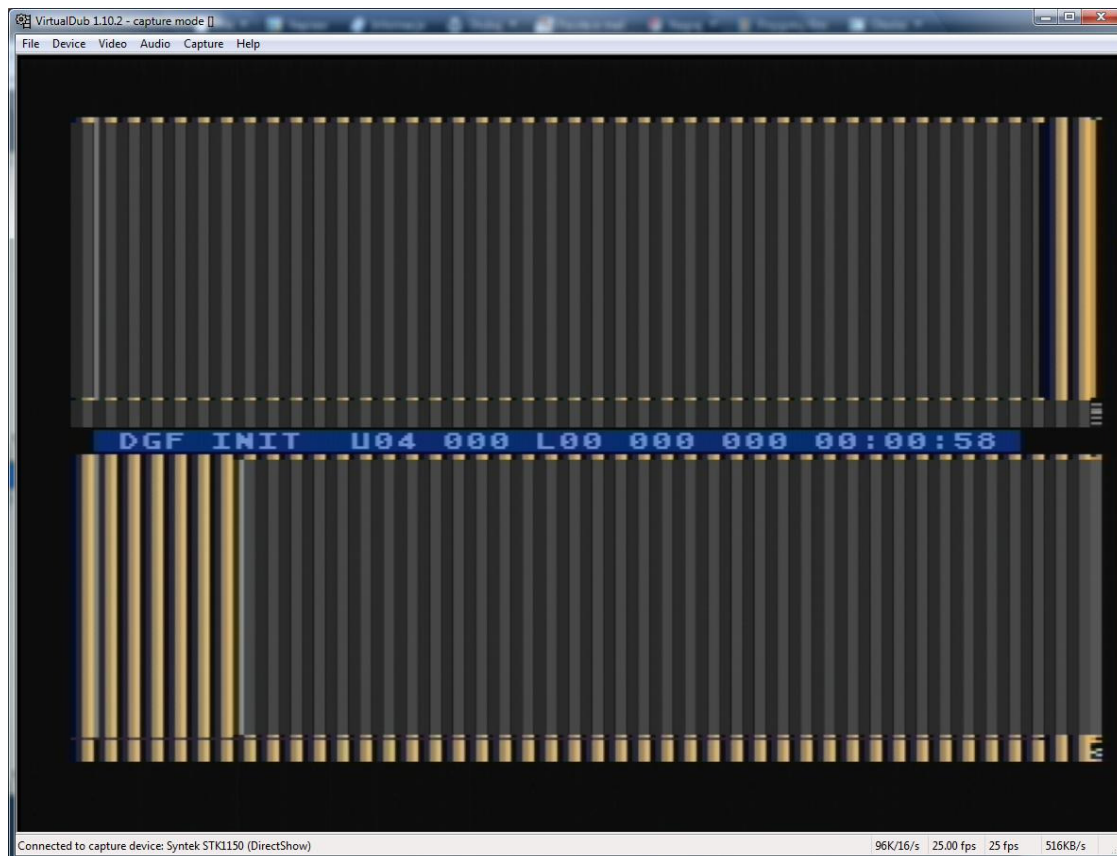
## **How to detect DGF effects activity and measure their stability.**

DPS effect may require a higher temperature of GTIA than DGM effect. Detection of stable DPS means that DGM is also stable. DPS detection can be accomplished by using mechanism of collision detection between PMG and graphics. It uses the fact that when DPS is active the PMG objects are shifted against graphics one lores pixel to the right as compared to their normal position when DPS is not active.

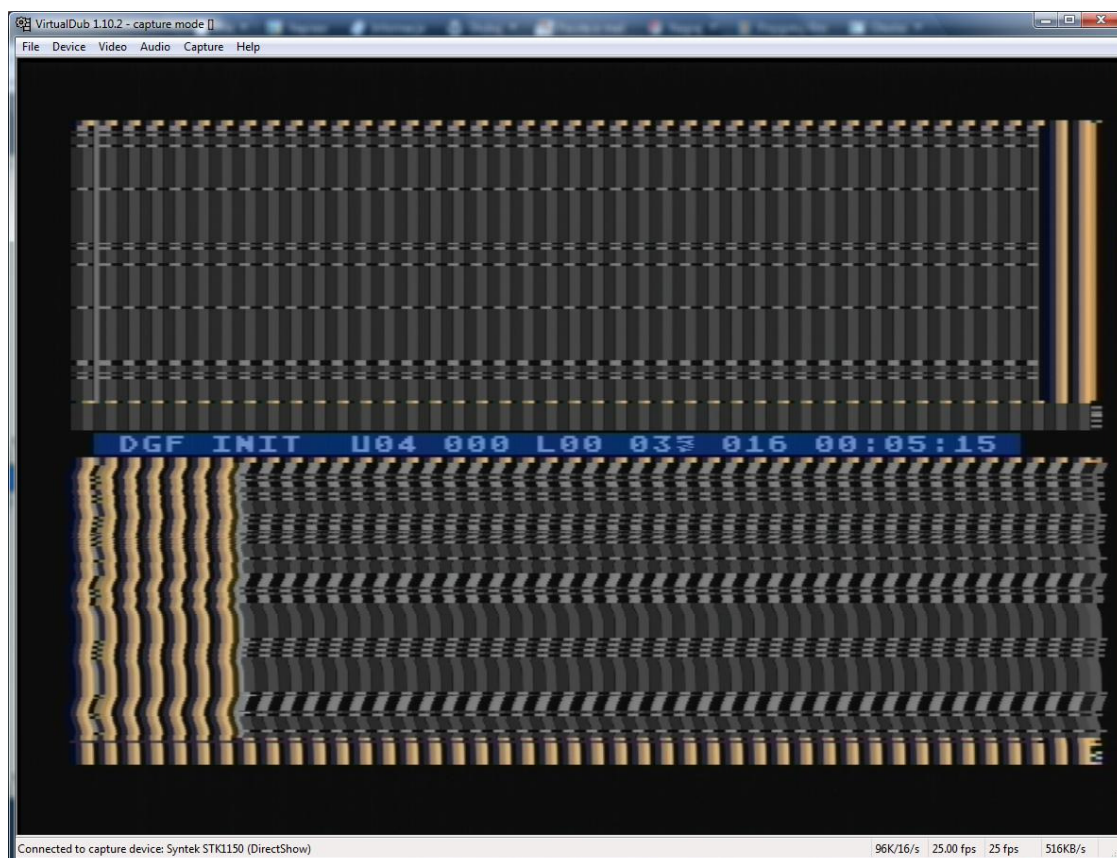
If in the program we use only the DGM effect then it is more reasonable to detect only the stability of this effect due to lower temperature threshold than for DPS. DGM detection is possible through the mechanism of collision detection between Missile objects and GTIA10 mode graphics. It uses the fact that, when DGM is active, boundaries of pixels of resolution 80 are shifted by one lores pixel and pixels values are different than when there is no shift (see paragraph "DGM").

The degree of stability of DGF effect is defined as the ratio of the number of scanlines in which DGF is active to the number of scanlines in which DGF is enabled. Number of checked scanlines should be as large as possible. In addition, the absolute stability can be considered only when it is maximal for several tens of consecutive frames.

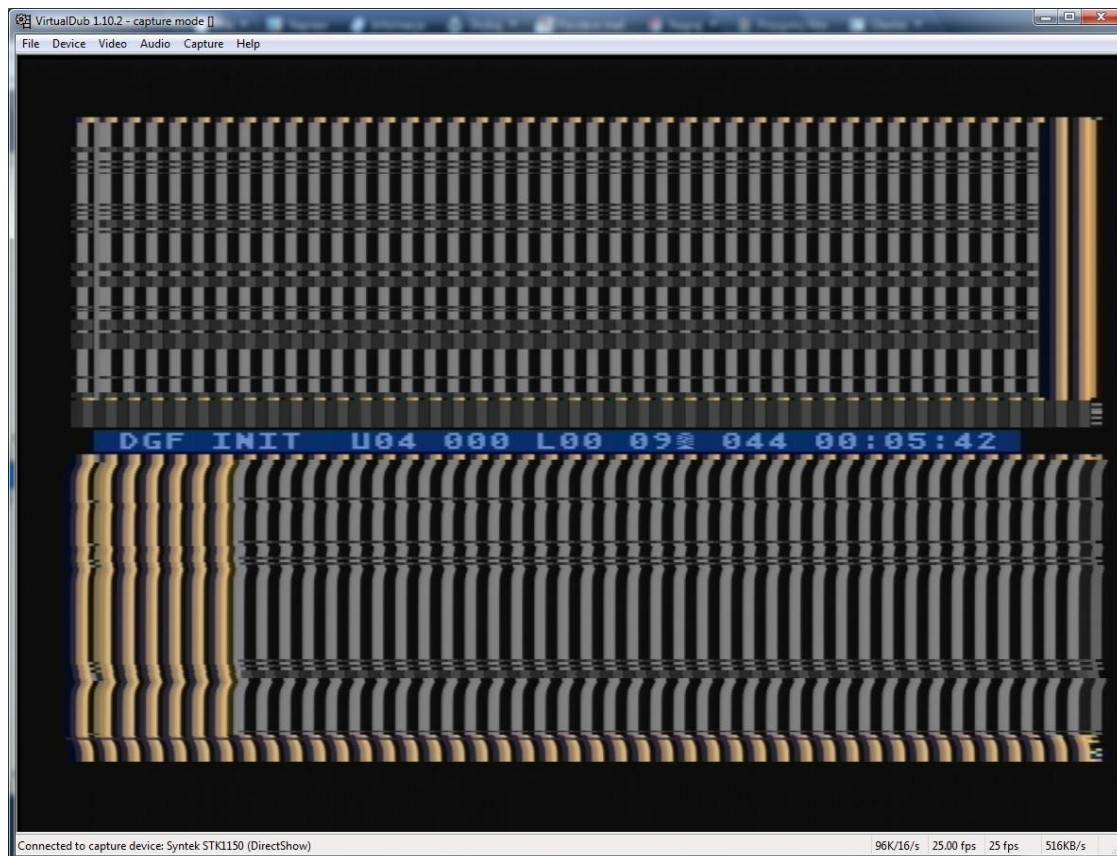
I wrote three short programs for detection of absolute stability of DGF effects. Program called DPSCHECK detects stability of DPS effect triggered with GTIA10. Program called DGMCHECK detects stability of DGM effect triggered with GTIA10. Program DGMG9CHK detects stability of DGM effect triggered with GTIA9. The programs do not have a RUNAD block and their purpose is to be added at the beginning of the .xex files which are programs that display graphics modes using the respective DGF effects.



*DGM effect enabled in upper part of the screen, DGM + DPS in lower part. The effects are not active.*



*Activity of the effects is visible. Stability of DPS effect in lower part of the screen has reached 35%.*



*Stability of DPS effect in lower part of the screen has reached 92%.*

## Relationships between DGF effects.

Although DGM and DPS effects are triggered in much the same way it is possible to obtain each of them separately. It is also possible to obtain them both at the same time. Everything depends on the moments where changes to PRIOR register occur.

In order to trigger **DGM effect only** switch from NORMAL to GTIAX in cycle 16 or later, as long as before the display of graphics, and switch from GTIAX to NORMAL in cycle 109 or earlier, as long as after the end of graphics display.

In order to trigger **DPS effect only** switch from NORMAL to GTIAX in cycle 109 or earlier, as long as after the end of graphics display, and switch from GTIAX to NORMAL in cycle 110 or later, as long as before cycle 109 of the next scanline.

In order to trigger **both effects at the same time** switch from NORMAL to GTIAX in cycle 16 or later, as long as before the display of graphics, and switch from GTIAX to NORMAL in cycle 110 or later, e.g. just before next switching to GTIAX.

## How to warm up GTIA (from programmer's perspective).

During GTIA warm-up the DPS effect must be enabled. The best is to enable it in as many scanlines as possible. Warming GTIA up when DPS is not enabled is ineffective. After such an inefficient heating up and then enabling the DPS or DGM it is still needed to warm up GTIA in order for both effects to gain visibility and stability, but then less time is needed.

During the warm-up triggering the DPS should use GTIA10 or GTIA9 mode. Warming up with use of GTIA11 is ineffective. If GTIA9 is used then some GTIA chips may require higher temperatures than while using GTIA10. It is recommended to use GTIA10 mode for GTIA warm-up.

## **Interesting observations and attempts to explain how DGF effects arise.**

### **Temperature.**

First of all, it should be noted that increase of temperature of integrated circuits causes increase of propagation times of the signals. This means, that in the conditions of the hot, some signals may be delayed. Such a delay is certainly the cause of DGF effects. The delay by exactly one lores pixel or one color cycle means that some flip-flop (or flip-flops) within GTIA holding chip's current state and clocked with 3.58MHz color clock latches its state one whole cycle later, or "misses" one active edge of the clock signal. Such a "missing" occurs once per scanline. It is not possible to fully explain the causes of DGF without detailed knowledge of the internal schematics of the GTIA.

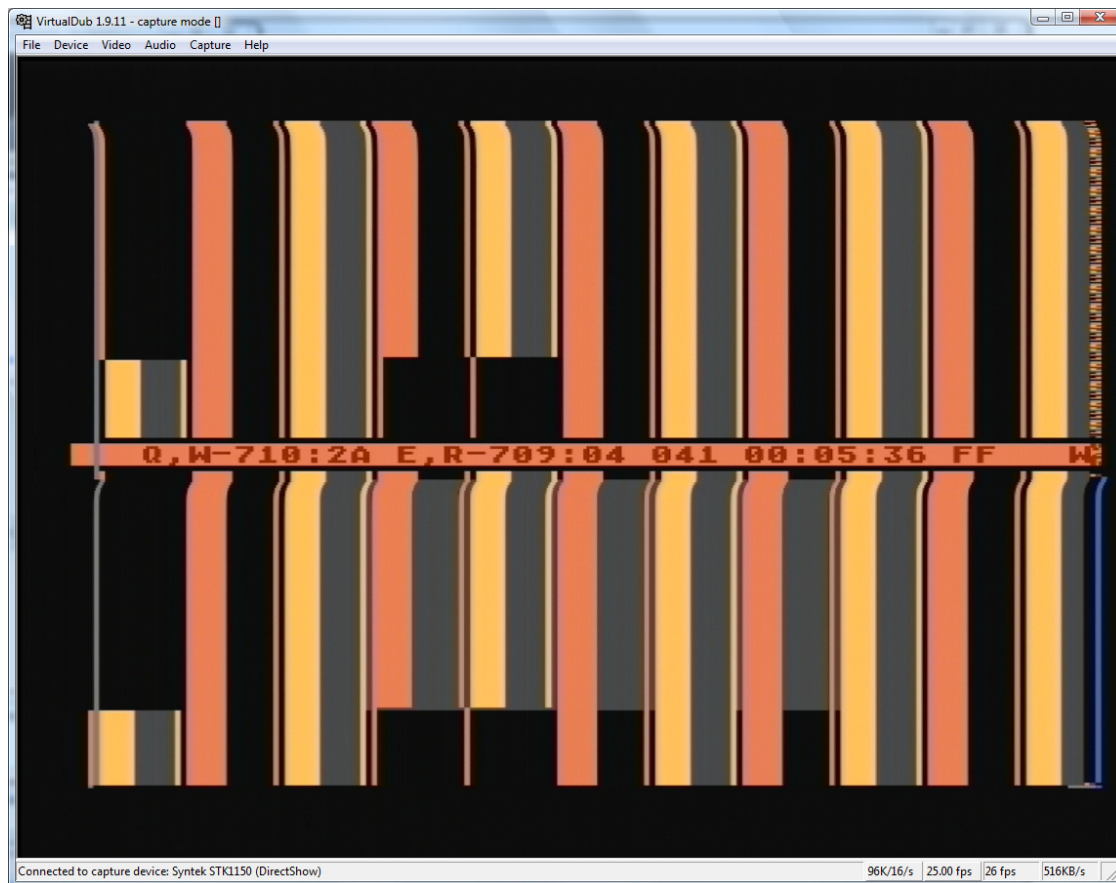
### **About synchronization of GTIA and ANTIC and its impact on DPS.**

Once, during experiments with so-called ANTIC bug in line 240 as well as the vertical and horizontal sync, I noticed that it is possible to affect horizontal synchronization of blanked lines (VBLANK) which results in various "deflections" of the image, but it is not possible to affect horizontal synchronization of 240 normal lines (of course if DGF is not active). In these normal lines, the image is always in the correct position. The conclusion of this observation is that GTIA synchronizes with ANTIC in each of the normal 240 lines, and does not synchronize in blanked lines. It is known that synchronization occurs when ANTIC sends through AN2-AN0 bus the HBLANK signal or binary value 010 (or 011 if the next scanline is to be in hires). Based on the recent conclusion it seems that ANTIC does not send HBLANK signal in blanked lines (VBLANK) or, what is more probable, ANTIC holds HBLANK signal value on AN2-AN0 bus constantly during entire VBLANK (with the exception of three lines where it sends VSYNC signal) and GTIA synchronizes with ANTIC only at the switch from ACTIVE DISPLAY to HBLANK. The latter option is supported by the fact that blank lines differ from normal lines because we cannot influence color of blank ones. It must therefore be the difference between the signal sent by ANTIC in blank and empty normal lines. The only possibility is that in empty normal lines it is BACKGROUND signal (binary 000) while in blank lines it is HBLANK signal.

The fact that under standard conditions GTIA displays correct image in terms of a correct horizontal sync in all scanlines (including blank scanlines) indicates that GTIA has internal counter which counts color cycles and allows to generate horizontal sync pulse at appropriate time, regardless of whether ANTIC sends HBLANK signal or not. This counter certainly is synchronized with HBLANK signal when it is sent by ANTIC (in 240 normal scanlines).

It seems that the cause of DPS effect is a delayed by one color cycle receiving HBLANK signal by GTIA and synchronizing of GTIA internal color cycle counter with this delayed signal. Another possible explanation is that GTIA "misses" the transition from ACTIVE DISPLAY to HBLANK because it runs in GTIAX mode and perhaps expects the transition at every second color cycle and thus does not synchronize, and that a one color cycle delay related to DGM causes also a delay of internal color cycle counter and thereby a delay of horizontal sync pulse.

I noticed that if DPS is enabled prior to the end of last 240th normal scanline and results in a shift of PMG and horizontal sync pulse in the next scanline, this shift remains for all blank scanlines (VBLANK) until the first (of 240 normal) scanline at the top of screen. Only here the DPS stops to be effective (unless it is enabled before first of normal scanlines). This observation confirms that GTIA does not synchronize with ANTIC during display of blank scanlines (VBLANK) and that it has an internal color cycle counter.



*DPS effect enabled only in the lower half of the screen. Return of image from shifted to normal position occurs only at the beginning of the next frame, that is at the top of screen. The shift remains for the duration of entire VBLANK even though the program does not do any switching of PRIOR register triggering DPS effect at that time.*

### **Does ANTIC influence DGF effects?**

During stabilizing the DGF effects appear and disappear randomly in particular lines of the screen. This is seen as shifting of line content by one pixel to the right or to the left. Let's call it pulsating. Additionally, in case of DPS effect a deflection of the image is seen nearby scanlines where the effect is active. In certain moments pulsating of the line looks completely independent of the others, but in other moments you may notice repeating the whole groups of lines that pulsate according to the same pattern. GTIA cannot distinguish the lines. It does not contain line counter and does not know which scanline is currently processing. GTIA relies only on the signals provided by ANTIC through AN2-AN0 bus. It seems that if GTIA was the only chip responsible for the delays, which result in DGF effects, these effects would appear and disappear at the same time in all scanlines. But it is not like that, and this fact indicates that ANTIC is co-responsible for generation of delays. I was able to observe an interesting case, which seems to prove it. Before I describe it, however, I need to make a short introduction. I conducted the research in a following way. I split the screen into two 100-line blocks separated by a dozen of lines. Each line was in ANTIC's mode \$0E. In each line of both blocks the DPS effect was enabled. In both blocks on the screen wide field DMA was enabled, which causes fetching 48 bytes per line in \$0E mode. In the Display List the LMS instruction (the one setting starting address of image data) was present only in the first one of hundred lines of each block. The amount of data needed for 100 lines is  $100 * 48 = 4800$ , which is more than 4kB. It is known that ANTIC's internal data pointer register is incremented automatically only within 4kB block (only bits 11-0 are incremented) which means that the pointer register consists of a 12-bit counter and a 4-bit regular register that holds constant most significant 4 bits of the address specified in the LMS. The 12-bit counter after reaching the highest value of \$FFF starts new turn, that is, starts counting from 0. Thus, if the 100-line block of image data started at boundary of 4kB memory block, e.g. from address \$5000, then exactly one start of new turn of the counter would have to occur and it would occur in line

of number  $(4096 \div 48) = 85$ . And now let's come back to the observation. I was able to observe a case where under conditions described above (the same in each line of 100-line block), the DPS effect was activated exactly in one line of each of 100-line blocks. After careful investigation, I found out that in both blocks it is the line immediately following the line where starting new turn of ANTIC's internal 12-bit data pointer counter occurs (remember that DPS effect is visible in the line following the line where it is triggered). This effect appeared as the first noticeable in the whole process of activating the DPS/DGF. Later on, the effect appeared successively in an increasing number of lines. In summary, the only factor which could cause that DPS occurred in only one of 100 lines of the block, is different delay (propagation time) of the signal sent by ANTIC through AN2-AN0 bus in case of starting new turn of ANTIC's internal data pointer counter than in case where there is no re-starting of ANTIC's counter. This observation makes me believe that ANTIC is co-responsible for generation of DGF effects.

### **HBLANK while DGF effects are active.**

Activity of sole DGM effect (i.e. when switching back from GTIAX to NORMAL takes place no later than in CPU cycle 109) has no impact on HBLANK.

Activity of DPS effect causes some changes. The duration of ACTIVE DISPLAY increases by one color cycle (exactly includes color cycle 222) from 188 to 189 cycles and consequently duration of HBLANK decreases from 40 to 39 cycles. HBLANK starts one color cycle later than normally - in cycle 223, and ends up as normal. In cycle 222 the display of graphics (in the same way as in cycles 220 and 221) and PMG objects works normally and PMG collisions are detected. In the line, where DPS is enabled for a first time (and will be visible only from the next line), the range of horizontal positions, where PMG objects are visible and where collisions detection works, is 34 – 222. It is equal to extended ACTIVE DISPLAY area. In the next lines, where DPS is active, the range of horizontal positions, where PMG objects are visible and where collisions detection works, is 33 – 221 (horizontal positions put to HPOSxx registers). This is due to the fact that the PMG are delayed by one color cycle. For example, the position 33 of the PMG is displayed in the color cycle 34. The fact that collisions detection includes PMG position 33, may be used for detecting activity and measuring stability of DPS effect.

In the first scanline without the DPS following the block of scanlines with DPS enabled the PMG objects DMA does not work, as mentioned in paragraph "DPS". This problem takes place precisely in first 40-cycle HBLANK following the 39-cycle one.

### **Two types of HBLANK.**

This is an observation that has nothing to do with DGF but may be interesting for people trying to recreate the internal schematics of GTIA. As I mentioned earlier, during the horizontal blanking ANTIC sends through AN2-AN0 bus the HBLANK signal, that is binary value 010 if the next line is to be in lores or 011 if the next line is to be in hires. The HBLANK signal is set on AN2-AN0 outputs during CPU cycles 111-113 (end of line) and cycles 0-16 (beginning of next line). ANTIC fetches Display List instruction in cycle 1 and only then finds out what will be the mode in the next line. Thus ANTIC is able to set appropriate HBLANK signal value to AN2-AN0 outputs only in cycle 2. I believe that in cycles 111-113 and 0-1 ANTIC sets AN2-AN0 outputs to HBLANK signal value same as in the previous line and in cycles 2-16 the signal value appropriate for the current line. Both values will be different if a change of the mode from lores to hires or vice versa takes place, and the same if a change of mode does not occur. I think GTIA selects the mode i.e. hires or lores based on the value of HBLANK signal in the last color cycle belonging to HBLANK period. However, for GTIA it does not matter which of these two signal values is given by ANTIC when it comes to generation of signals (levels) of horizontal blanking and horizontal sync.

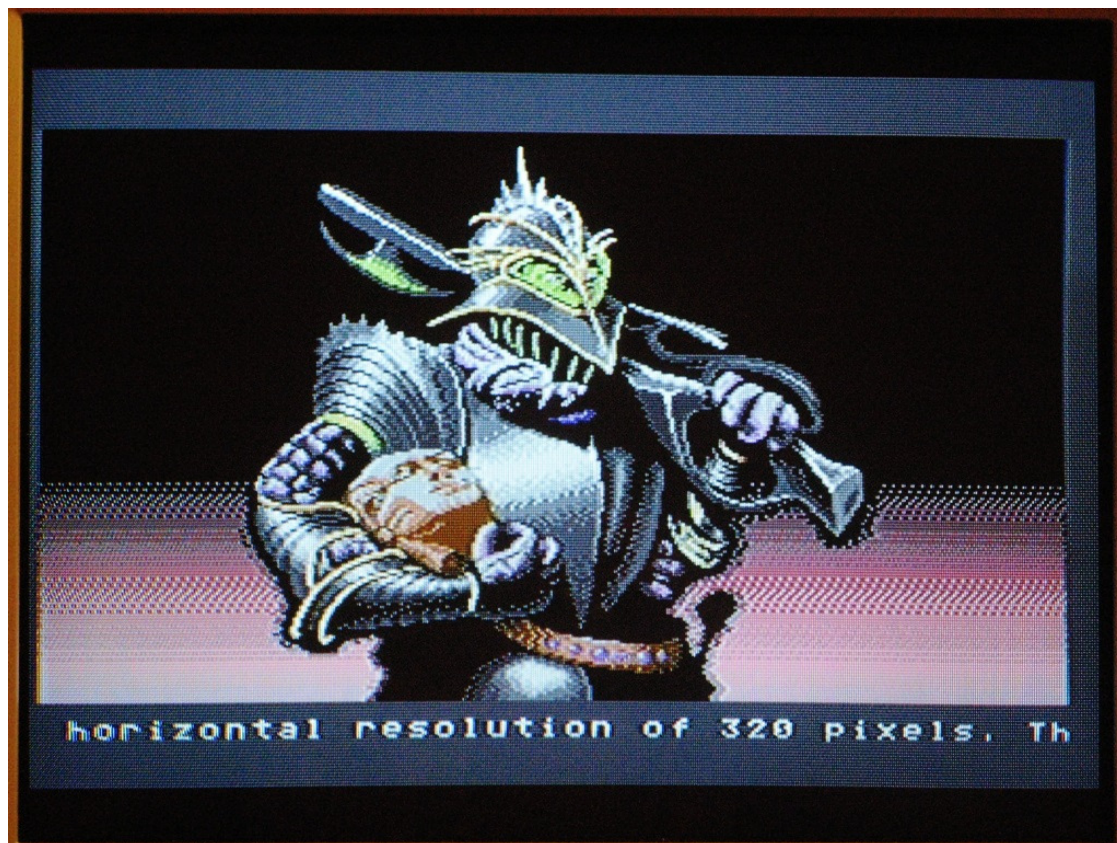
### **Graphics modes using DGF effects.**

Using DGF effects we can get some new graphics modes on 8-bit Atari. Most of them are interlace modes where one of two sub-images is horizontally shifted by half a pixel of particular resolution relatively to the other. This results in a doubling of horizontal resolution and increasing the number of colors or shades by mixing overlapping halves of pixels from both sub-images.

**DGI** – Delayed Gtia hires Interlace – a mode, which I demonstrated in the KNIGHT demo. It features horizontal resolution of 320 pixels (hires) for a standard width of the field. Formed by alternating display of two images in horizontal resolution of 160 (lores), that is, in ANTIC's mode \$0E or \$04, optionally colorized with PMG objects. In one of the images the DPS2 effect is used, giving a shift by one hires pixel to the left from its normal position.



*DGI – shot one (that's the version yet from WAP-niak party, with gothic font)*



*DGI – shot two*



*The sub-image shifted to the left by one hires pixel.*



*The not shifted sub-image.*

**DGX** – Delayed Gtia cross(X) interlace – a mode being a combination of DGI and regular vertical TV interlace (elaborated for Atari by Rybags). In this mode, the lines of sub-images do not overlap but are interlaced, and in result the vertical resolution is doubled. We can see the lores pixels stacked like bricks in a wall, that is, in one line not shifted, in the next one shifted half of length, and in the next one not shifted again, etc. The DGX mode has also been presented in KNIGHT demo. The used picture was not meant to be displayed in such way and so using the DGX here does not improve the quality of picture.

**D9I** – Delayed gtia9 Interlace – a mode that features horizontal resolution of 160 pixels for a standard width of the field. Formed by alternating display of two images in GTIA9 mode, that is, in horizontal resolution of 80 pixels. In one of sub-images the DPS effect is used, giving a shift by one lores pixel to the left from its normal position.

**D10I** – Delayed gtia10 Interlace – a mode that features horizontal resolution of 160 pixels for a standard width of the field. Formed by alternating display of two images in GTIA10 mode, that is, in horizontal resolution of 80 pixels. In one of sub-images the DPS effect is used, giving a shift by one lores pixel to the left from its normal position.

**D9X** – Delayed gtia9 cross(X) mode – a non-interlace mode where all scanlines are displayed in GTIA9 mode and every second scanline is shifted by half a pixel to the right by due to usage of DGM (DG9) effect. The pixels of horizontal resolution of 80 are stacked like bricks in a wall, that is, in one line not shifted, in the next one shifted half of length, and in the next one not shifted again, etc.

**D10X** – Delayed gtia10 cross(X) mode – a non-interlace mode where all scanlines are displayed in GTIA10 mode and every second scanline is shifted by half a pixel to the right by due to usage of DGM (DG10) effect. The pixels of horizontal resolution of 80 are stacked like bricks in a wall.

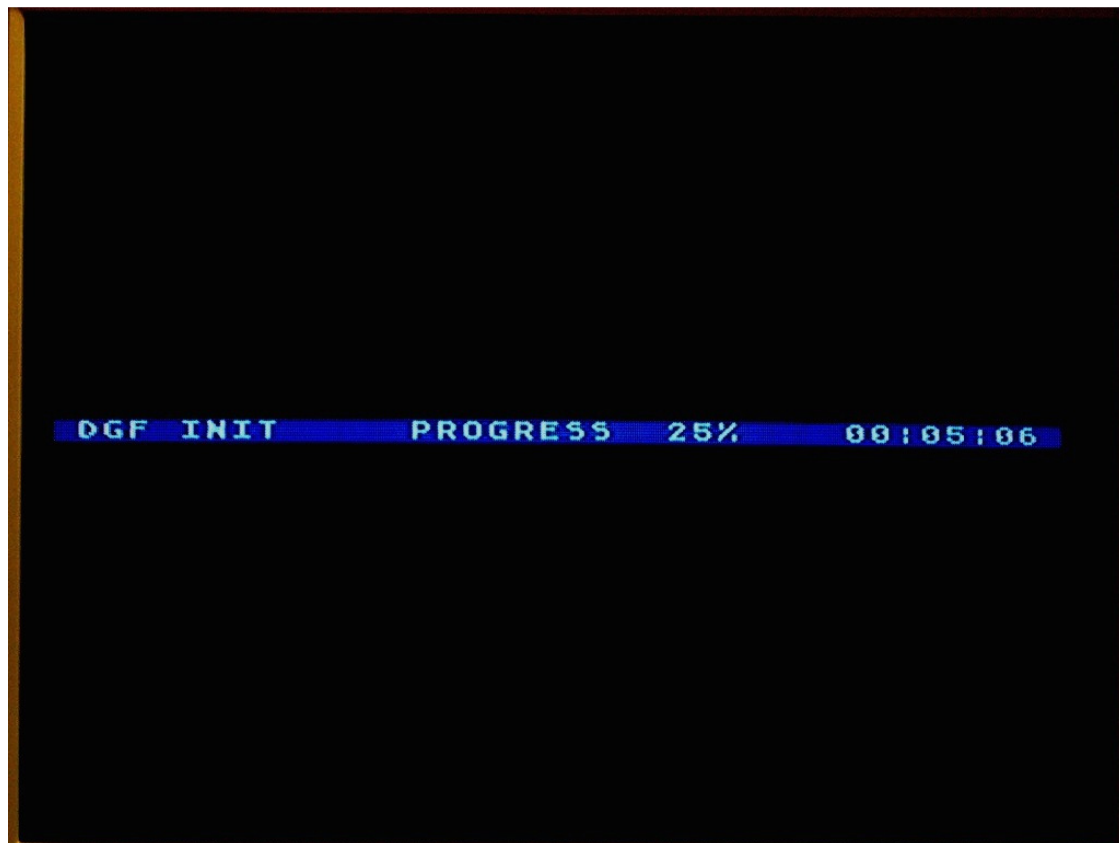
**D9XI** – Delayed gtia9 cross(X) Interlace – a mode that features horizontal resolution of 160 pixels for a standard width of the field. Formed by alternating display of two images in D9X mode where in one image the even lines are not shifted and the odd are shifted by half a pixel to the right and in the other

image the even lines are shifted half a pixel to the right and the odd are not shifted. This mode is very similar to the HIP mode, but here the GTIA9 mode is used in all scanlines. This is the same mode as VZI – VertiZontal Interlacing developed in 1994 by Bryan.

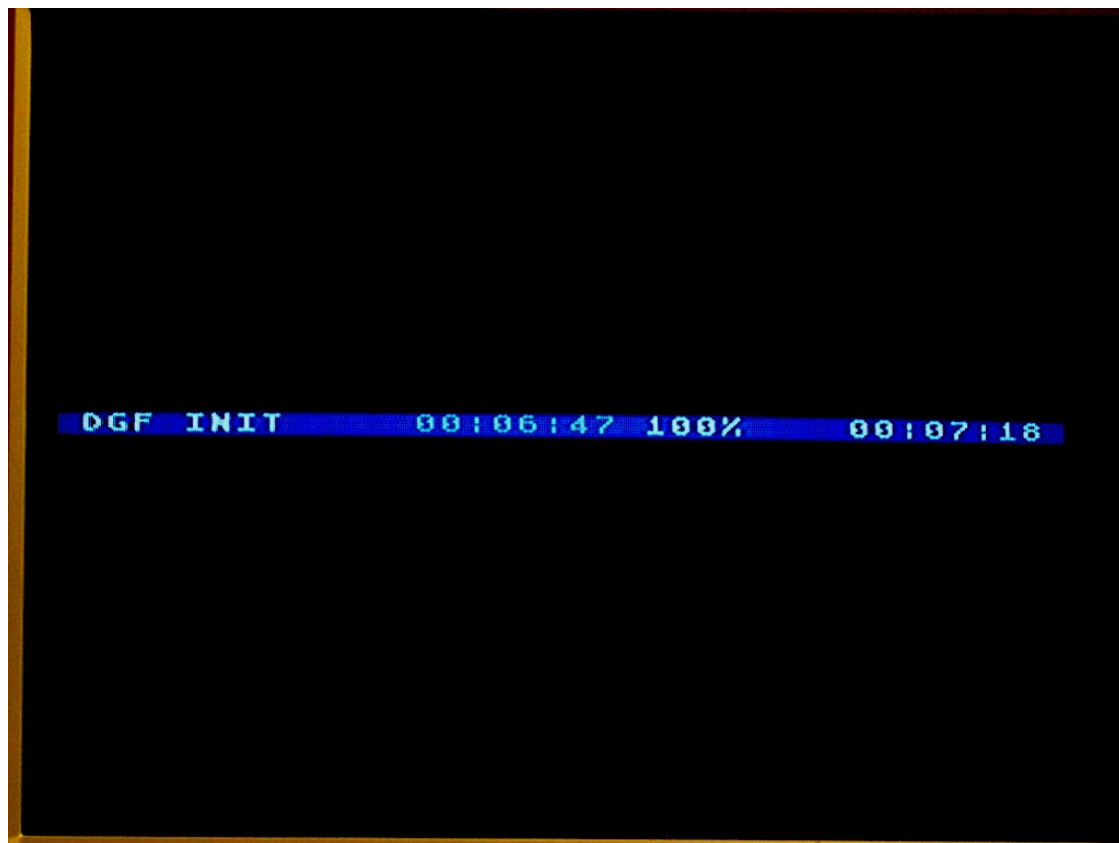
**D10XI** – Delayed gtia10 cross(X) Interlace – a mode that features horizontal resolution of 160 pixels for a standard width of the field. Formed by alternating display of two images in D10X mode, in a similar way as D9XI mode.

## Programs.

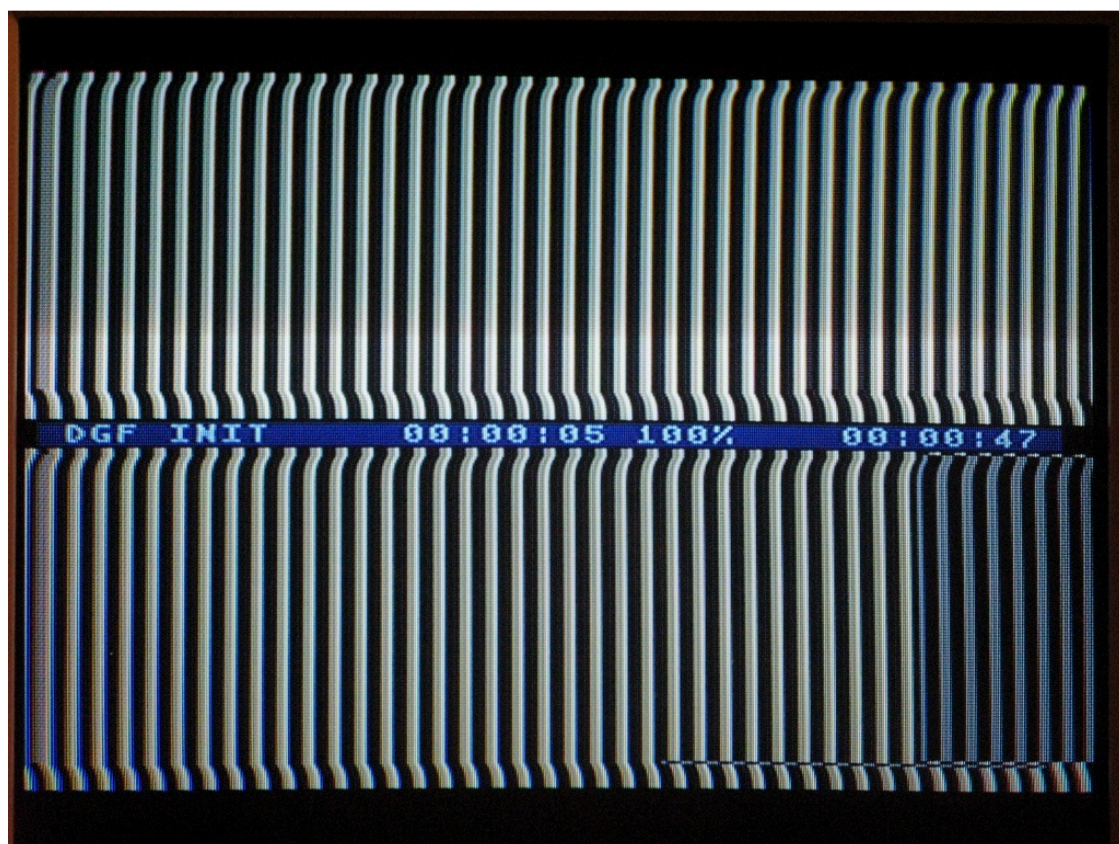
DGF INIT (DGFINIT.XEX) - a program that initiates DGF effects while rising temperature of GTIA and ANTIC chips, according to description given above. The program also measures the stability (availability) of DGF effects and shows it as a percentage. DGF effects are triggered with recommended GTIA10 mode, and measurement results are relevant to the effects triggered with such method only. The DGF effects triggered with other methods e.g. using GTIA9 may still be unavailable when the program shows the 100% stability. The program, in order to detect DGF effect activity in particular scanline, uses the mechanism of collisions detection between PMG objects and graphics. When stability of DGF effects reaches 100% an audible alarm is activated and total time of reaching full stability is displayed. The alarm signal lasts for about 1 minute and can be turned off earlier by pressing any key. Pressing B causes displaying of background graphics used by the program while initiating DGF and allows for observation of behavior (pulsating) of particular scanlines during process of DGF stabilization. Pressing Esc causes exiting to DOS.



*DGF INIT program in action. The stability of DGF (DPS and DGM) effects has reached 25%.*



*The 100% stability reached in 6 minutes and 47 seconds.*



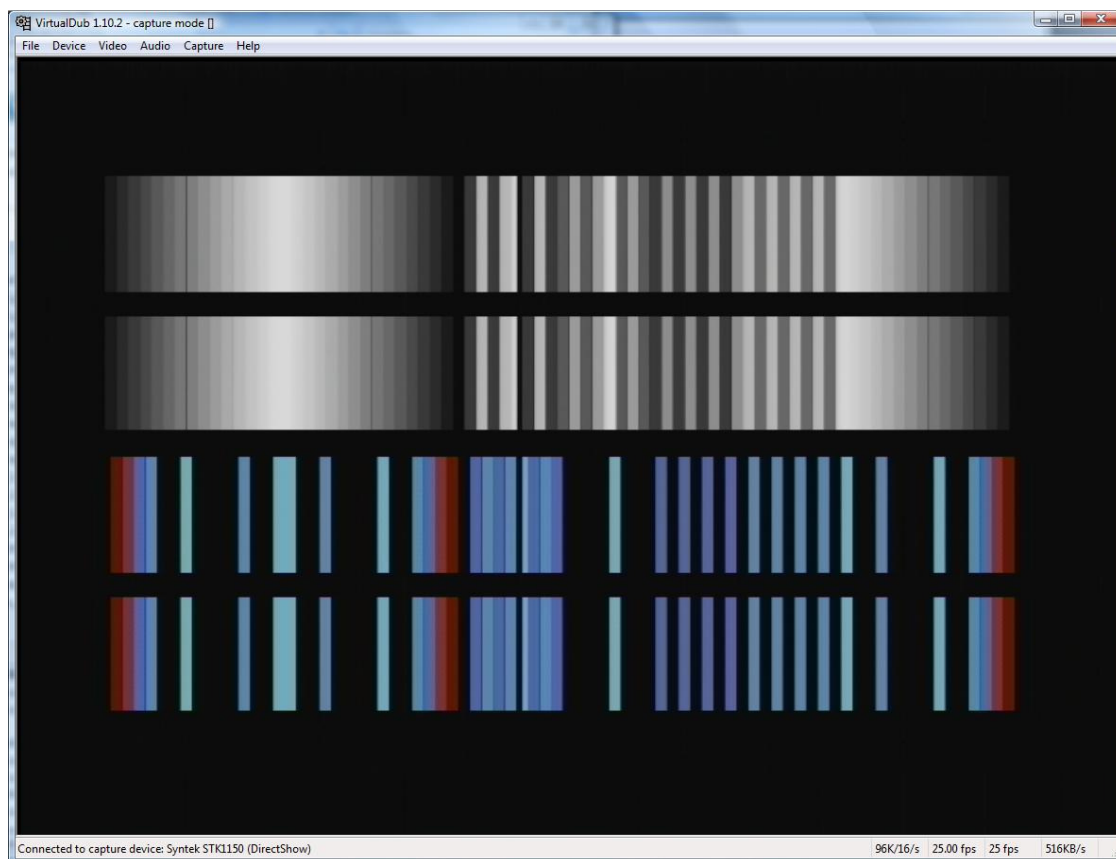
*After pressing B you can observe deflection of the image during reaching DGF effects stabilization.*

DPS CHECK (DPSCHECK.XEX) – a program that checks if DPS effect (triggered with recommended GTIA10) is stable (available) in 100%. The program does not have a RUNAD block and its purpose is to be added at the beginning of the .xex files being programs that display graphics modes using the DPS. If the program detects that the DPS is not stable (available) in 100% then it will display an appropriate message and return immediately to DOS in order to avoid further loading and starting the main program.

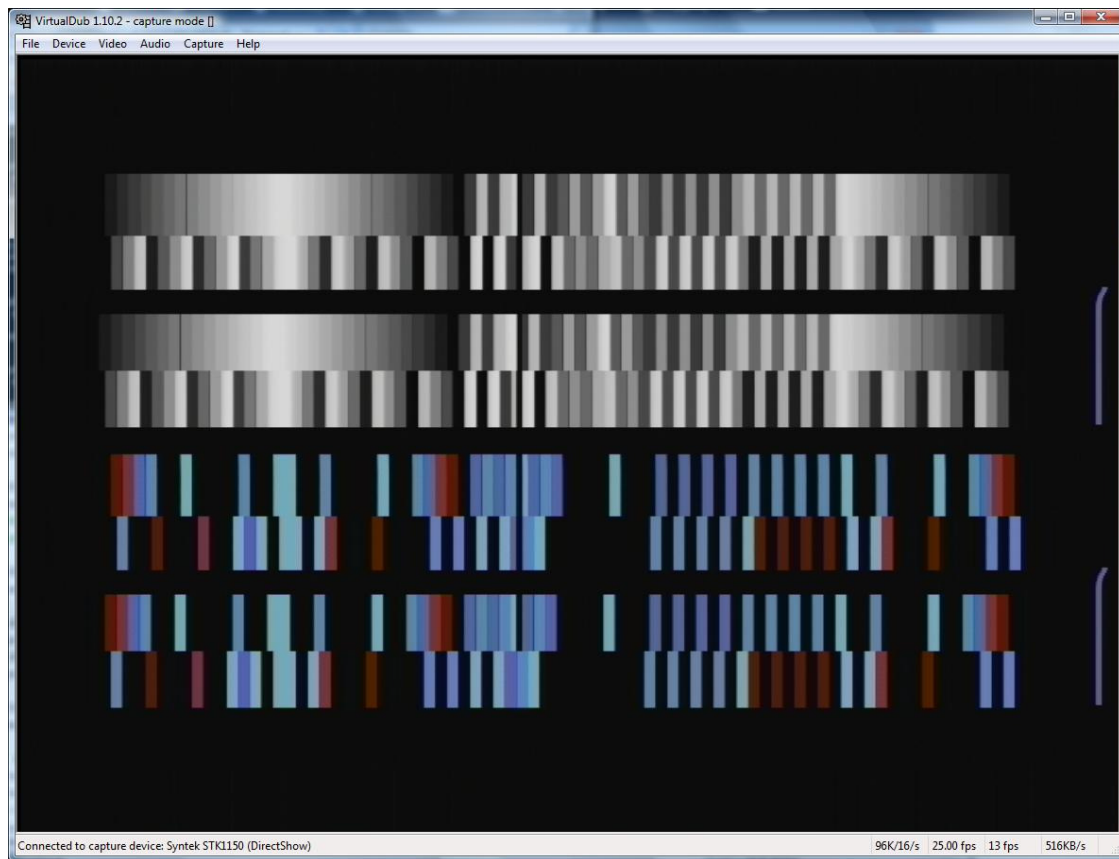
DGM CHECK (DGMCHECK.XEX) – a program that checks if DGM effect (triggered with recommended GTIA10) is stable (available) in 100%. Similarly to DPS CHECK, the program does not have a RUNAD block and its purpose is to be added at the beginning of the .xex files being programs that display graphics modes using the DGM.

DGM G9 CHECK (DGMG9CHK.XEX) – a program that checks if DGM effect (triggered with GTIA9) is stable (available) in 100%. Similarly to DPS CHECK, the program does not have a RUNAD block and its purpose is to be added at the beginning of the .xex files being programs that display graphics modes using the DGM.

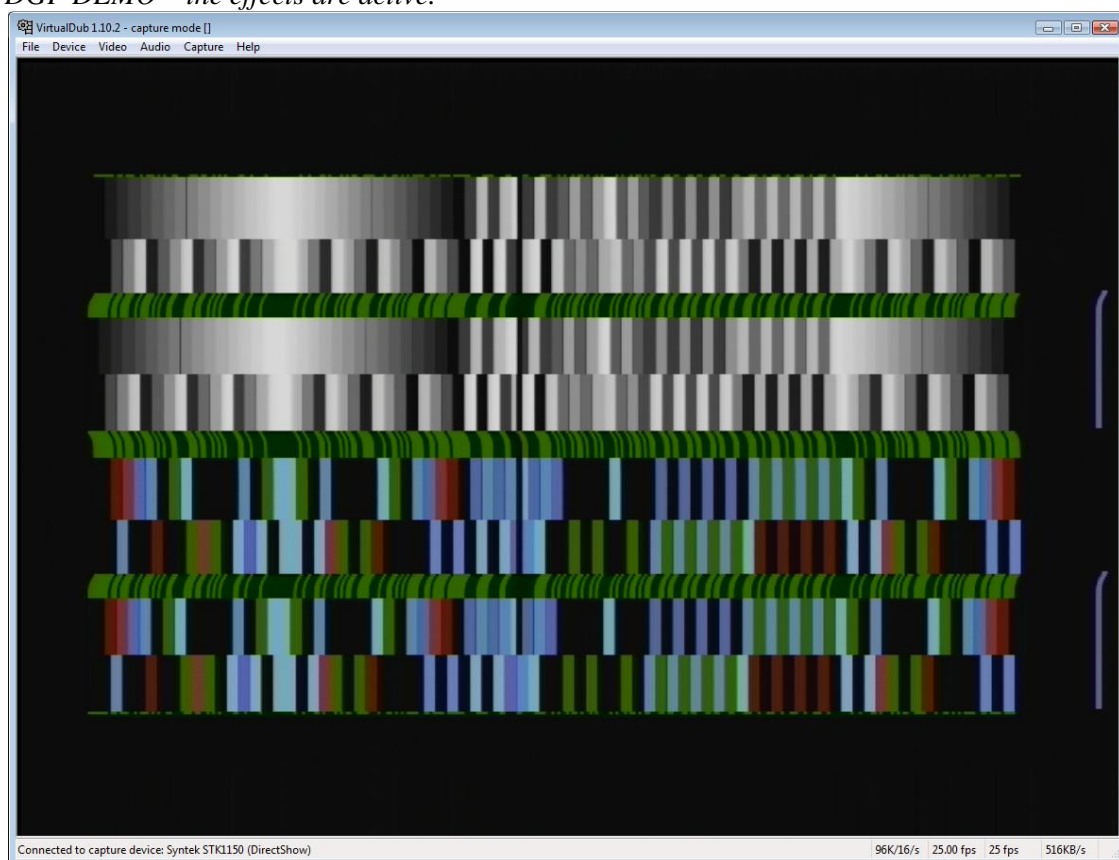
DGF DEMO – a demo of DPS and DGM effects applied in GTIA9 and GTIA10 modes. Pressing space bar reveals hidden deflected parts of the image. Pressing Esc causes exiting to DOS. I encourage to look at the source code of the program. In Src\ldgfmde folder there are separate source files for DPS, DG9 and DG10 effects as well as for their combinations.



*DGF DEMO – the effects are inactive.*



*DGF DEMO – the effects are active.*



*DGF DEMO – visible deflected parts of the image.*

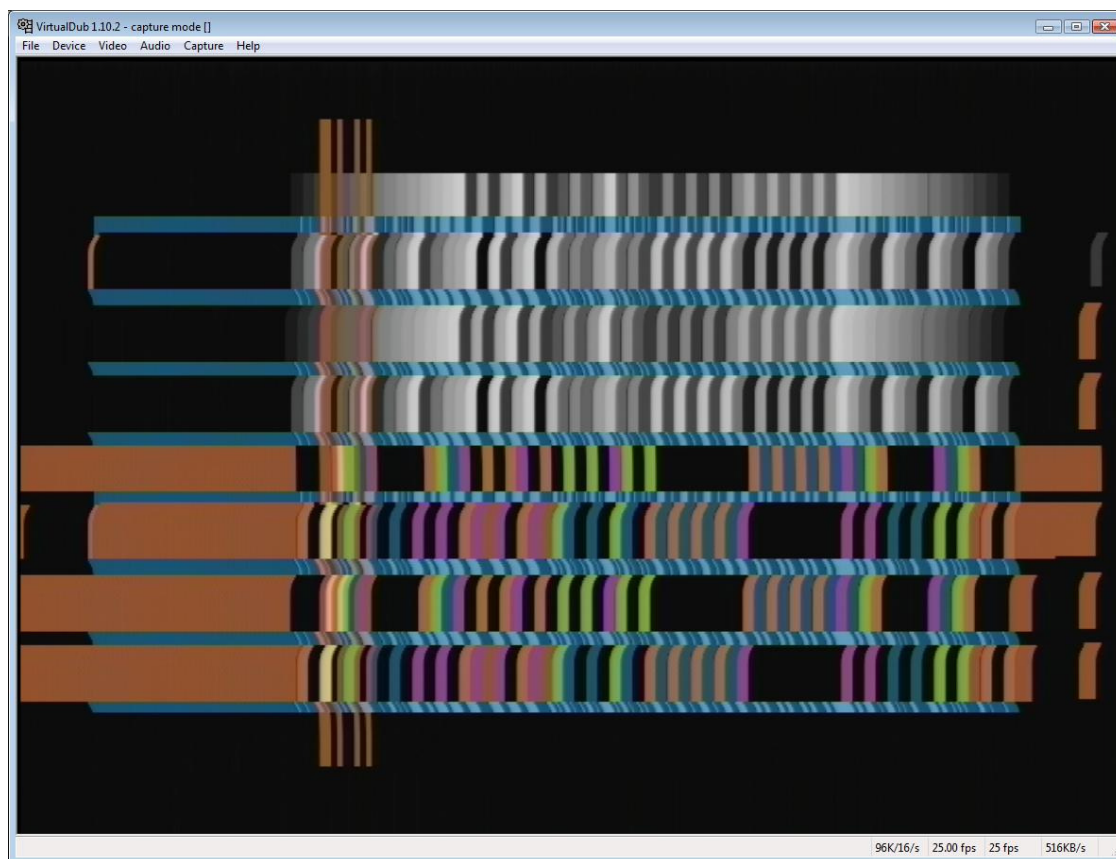
KNIGHT – a demo of DGI and DGX modes, using "Knight" picture by Tomasz "Levi" Lewandowski created for C64 in FLI interlaced mode. It also allows for display of two sub-images solely. The program can detect the lack of full stability of DGF, and in such case blocks the display of DGI and DGX modes as well as the sub-image using DPS2 effect which would display incorrectly due to differences in the horizontal positions of PMG objects in every second scanline. Pressing 1 will cause display of only the sub-image shifted by one hires pixel to the left, that is the one using DPS2 effect, pressing 2 - only the sub-image without a shift, 3 - the full image in DGI mode, 4 and 5 - two variants of image in DGX mode (400i), which differ in that which of sub-images is displayed higher and which lower. Pressing space bar pauses scrolling text. Esc causes exiting to DOS.

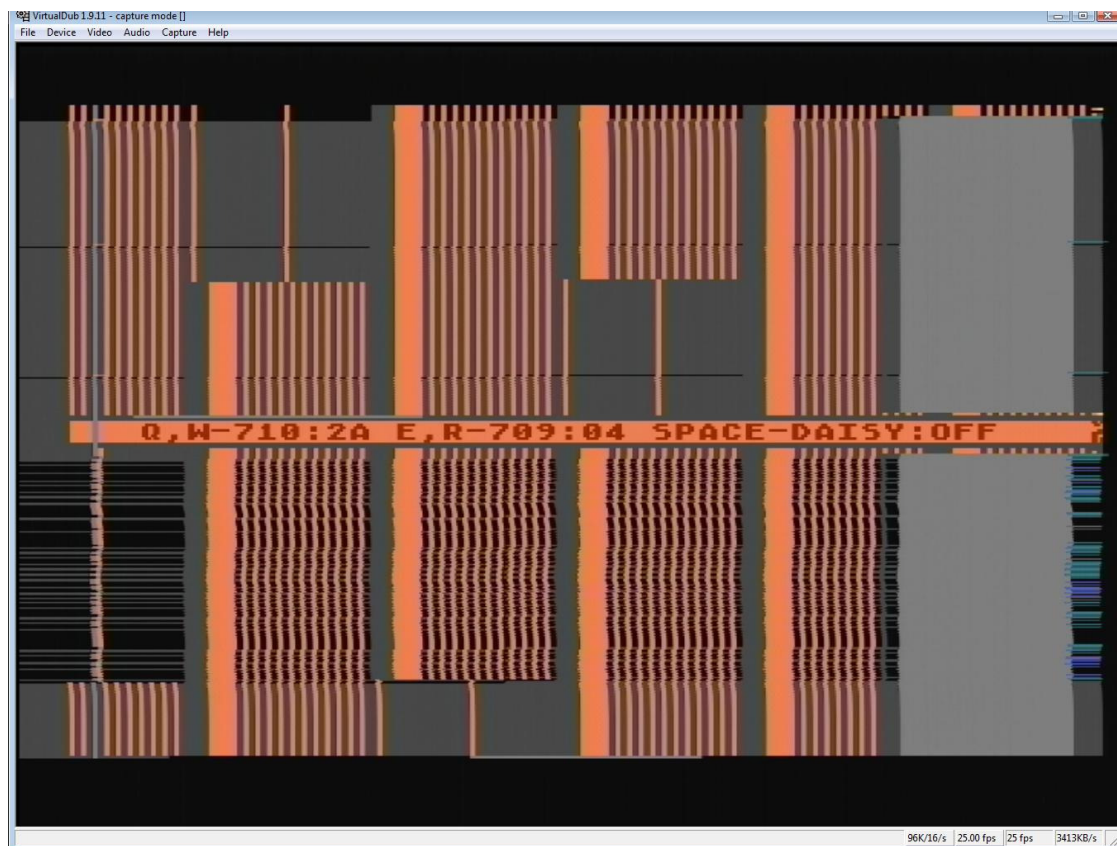
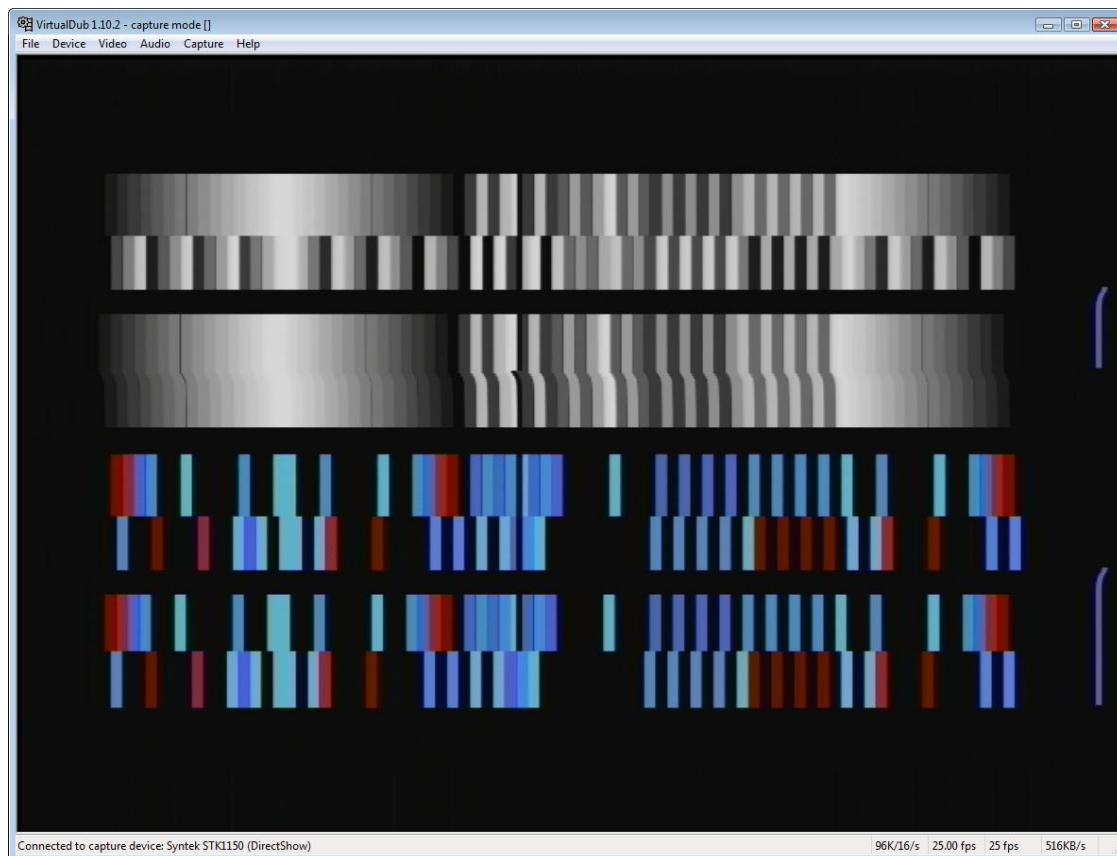
CACTUVZI and FARMVZI – programs that display images in VZI (D9XI) mode developed by Bryan. In these programs triggering the DGM effect is done with usage of GTIA9 mode, so that on some machines the effect will be relatively short-lived (see paragraph "Influence of ways of triggering DGF effects on their stability"). One can add DGMG9CHK.XEX at the beginning of both programs in order to add a mechanism of detection of full stability of DGM effect.

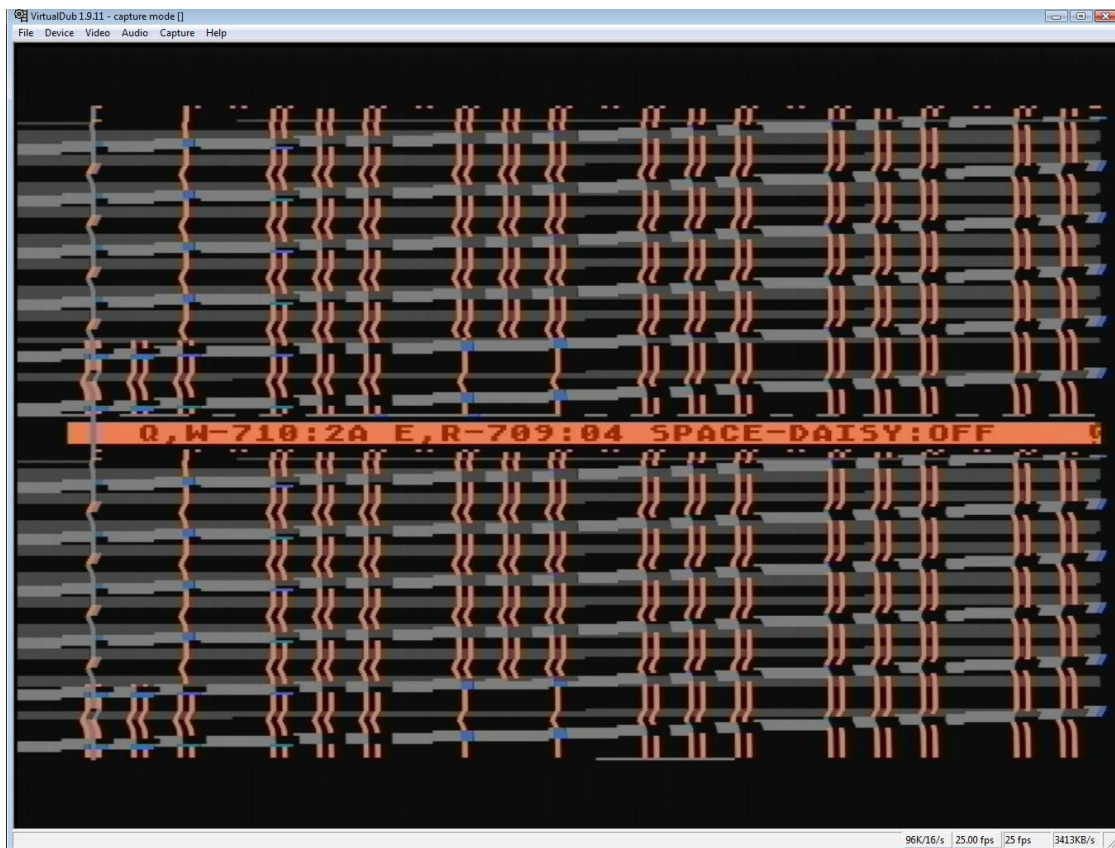
All of these programs written by me need no more than 48kB of RAM and do not use RAM under OS ROM, so that they can work well even on Atari 800. My programs run on both PAL and NTSC machines.

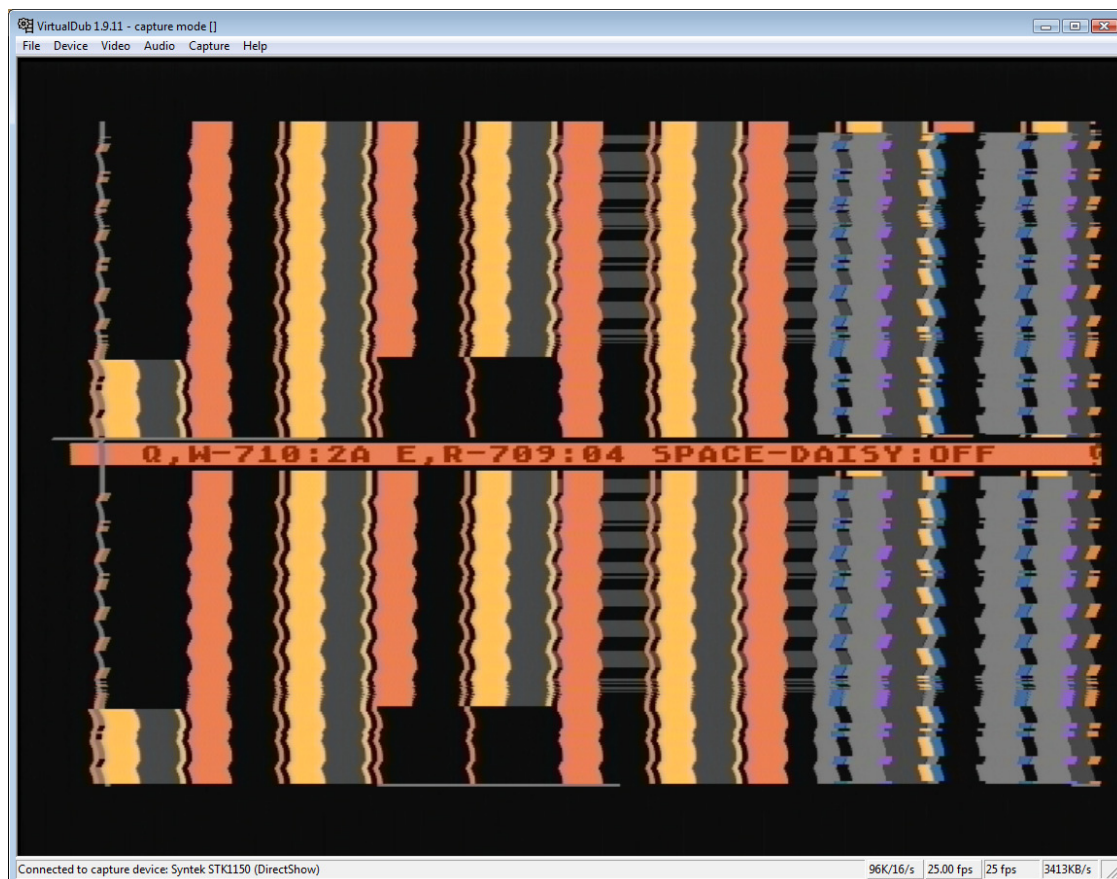
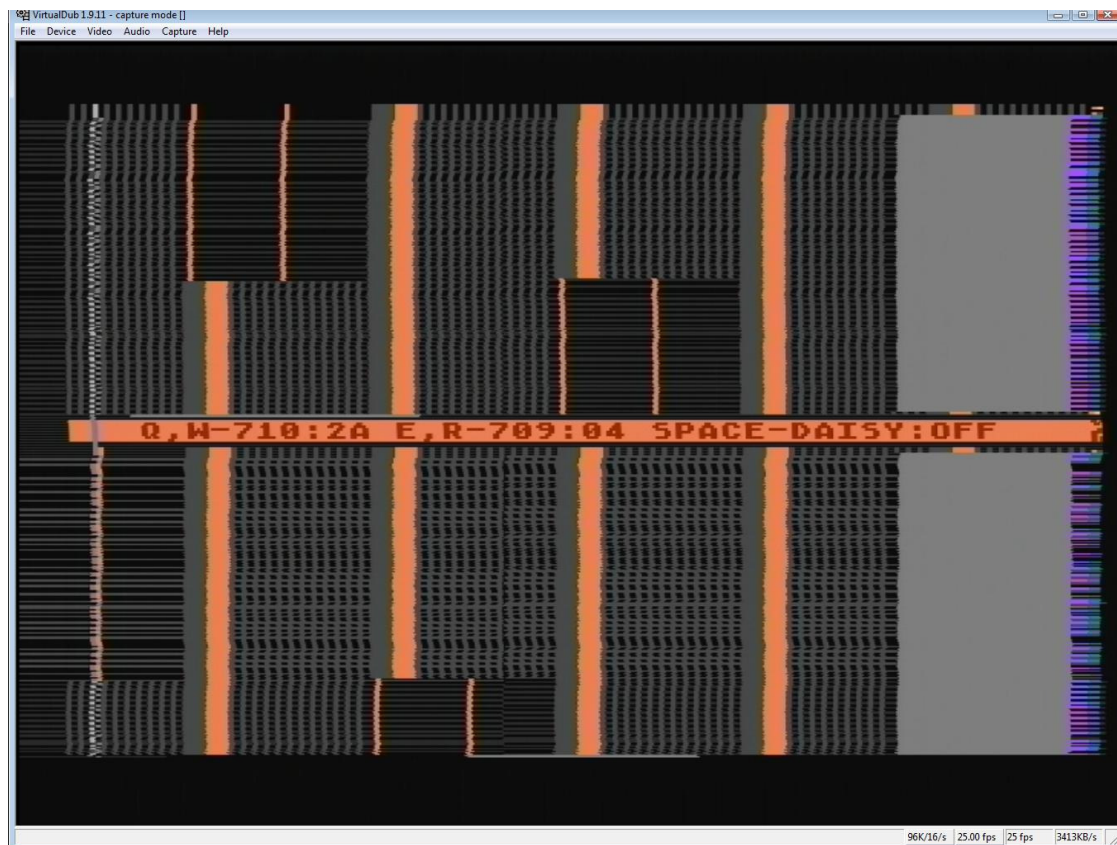
## Interesting screenshots.

Below are a few of many screenshots taken during work on DGF.









## **This is the end.**

Described DGF effects and based on them graphics modes cannot be used in serious applications due to the limited availability (long time of waiting for spontaneous warming up of the graphics chips, the need to warm-up these chips, the absence of DGF effects in some machines). However, I think it was worth it to investigate and describe the phenomenon of DGF, simply just to get to know it and explain one of the most hidden secrets of Atari. In addition, watching DGI mode – the hires with lots of colors on a standard Atari has brought me great joy, and only for that mode it was worth it to carry out such a research.

Pawel Rosowski