Smaku Books Multimedia Publishing House

Wordmedia – Documents and Publications Site

ANTIC i GTIA Atari XL/XE oraz inne układy wizyjne / graficzne

Tworząc Atari XL/XE w Turbo Pascal 5.5 trzeba zrealizować w formie programowej między innymi układy elektroniczne odpowiedzialne za wyświetlanie grafiki na ekranie.

Zgodnie z opisem zawartym w dokumentacji inżynierskiej firmy Atari ™ do układów ANTIC i CGIA/GTIA, można przygotować wstępnie prostą procedurę w Turbo Pascal 5.5 realizującą jedną pełną ramkę ANTIC – czyli przygotowanie przez układ ANTIC całego jednego ekranu w trybie tekstowym lub graficznym Atari XL/XE do wyświetlenia na urządzeniu zewnętrznym.

```
procedure inc_pc_antic; { inc antic program counter }
begin
if reg_PCL_ANTIC+1<=255 then reg_PCL_ANTIC:=reg_PCL_ANTIC+1
else
if reg_PCH_ANTIC<255 then
begin
reg_PCH_ANTIC:=reg_PCH_ANTIC+1;
reg_PCL_ANTIC:=0;
end
else
begin { out of memory range – set to [0:0]}
reg_PCH_ANTIC:=0;
reg_PCL_ANTIC:=0;
end;
end;

procedure inc_msc; { inc memory scan counter }
begin
reg_BAH:=msch; reg_BAL:=mscl;
set_ab(1);
msch:=reg_BAH; mscl:=reg_BAL;

{ or use own independent inc full procedure like inc_pc_antic – check antic documentation to know how is for real }

end;
```

{ *ANTIC – Memory Scan Counter* }

```
{..}

{ SET CHAR MODES PARAMETERS }

{..}

procedure set_antic6_par;
begin
x_res:=4;
pixel_prop:=16; lines_per_pixel:=8;
end;

{..}

{ SET MEMORY MAP MODES PARAMETERS }

{..}

{ PLOT LINE }

procedure empty_line;
begin
c:=RAM[$2,$c8];
for b:=0 to (reg_IR_ANTIC and 112) shr 4 do
begin
reg_BAH:=0; reg_BAL:=0;
repeat
set_ab(1);
plot_point;
until (reg_BAH=1) and (reg_BAL=64);
VLC:=VLC+1;
end;
end;

{ VLC is of word type! forbidden! => change to two 8-bit registers VLCH, VLCL }
{ check ANTIC documentation IF VLC is increased one by one each line OR VLC:=VLC + lines_per_pixel (any-
way, running up each single scan line done makes work stable => use this method – no matter ANTIC docu-
mentation }
end;

{ ANTIC – Vertical Line Counter }

{..}

procedure plot_antic6_line;
begin
for b:=1 to x_res shl 2 + x_res do begin c:=RAM[msch,mscl]; plot_char1b_67; inc_msc; end;
end;
```

```
{..}

procedure do_ANTIC;
begin
reg_BAH:=DLISTH;
reg_BAL:=DLISTL; mem;
reg_PCL_ANTIC:=DB;
set_ab(1); mem;
reg_PCH_ANTIC:=DB;

NMIEN:=NMIEN or 128; { set 7-th bit of NMIEN }

x:=0; y:=0;

{ writeln(,Atari XL/XE ANTIC graph mode = ,,reg_IR_ANTIC and 15); }

VLC:=0; HSCRL_enabled:=false; VSCRL_enabled:=false;

repeat
reg_BAH:=reg_PCH_ANTIC;
reg_BAL:=reg_PCL_ANTIC; mem;
reg_IR_ANTIC:=DB;

antic_gr_mode:=reg_IR_ANTIC and 15;

if antic_gr_mode  = 0 then begin set_antic15_par; empty_line end   { b0 BLANK }
else
if antic_gr_mode = 1 then               { b1 JMP }
begin
inc_pc_antic; reg_BAL:=ram[reg_PCH_ANTIC,reg_PCL_ANTIC];
inc_pc_antic; reg_BAH:=ram[reg_PCH_ANTIC,reg_PCL_ANTIC];
reg_PCL_ANTIC:=reg_BAL; reg_PCH_ANTIC:=reg_BAH;
if reg_IR_ANTIC and 128 = 128 then do_DLI_IRQ;
if reg_IR_ANTIC and  64 =  64 then do_VLB
else begin set_antic15_par; empty_line; end;
end
else
begin
if reg_IR_ANTIC and 128 = 128 then do_DLI_IRQ;  { b7 DLI }
if reg_IR_ANTIC and 64  = 64  then            { b6 LMS }
begin
inc_pc_antic; mscl:=ram[reg_PCH_ANTIC,reg_PCL_ANTIC];
inc_pc_antic; msch:=ram[reg_PCH_ANTIC,reg_PCL_ANTIC];
end;

{ b5 przesuw pionowy VSCROL }
if reg_IR_ANTIC and 32 = 32 then VSCRL_enabled:=true else
VSCRL_enabled:=false;
{ b4 przesuw poziomy HSCROL }
```

```
if reg_IR_ANTIC and 16 = 16 then HSCRL_enabled:=true else
HSCRL_enabled:=false;

if antic_gr_mode and 8 = 0 then { CHAR modes }
case antic_gr_mode of { graph mode: 4 low bits of reg_IR_ANTIC }
02: begin set_antic2_par; plot_antic2_line; plot_crsr; end;
03: begin set_antic3_par; plot_antic2_line; end;
04: begin set_antic4_par; plot_antic4_line; end;
05: begin set_antic5_par; plot_antic4_line; end;
06: begin set_antic6_par; plot_antic6_line; end;
07: begin set_antic7_par; plot_antic6_line; end;
end
else   { GRAPH modes }
case antic_gr_mode of
08: begin set_antic8_par;  plot_line2b;  end;
09: begin set_antic9_par;  plot_line1b;  end;
10: begin set_antic10_par; plot_line2b; end;
11: begin set_antic11_par; plot_line1b; end;
12: begin set_antic12_par; plot_line1b; end;
13: begin set_antic13_par; plot_line2b; end;
14: begin set_antic14_par; plot_line2b; end;
15: case basic_gr_mode of
09:   begin set_antic15_1_par; plot_line2b; end; {BASIC 09}
10:   begin set_antic15_2_par; plot_line2b; end; {BASIC 10}
11:   begin set_antic15_3_par; plot_line2b; end; {BASIC 11}
64:   begin set_antic15_4_par; plot_line1b; end; { 320×192  16 color or other extended GTIA mode }
65:   begin set_antic15_5_par; plot_line1b; end; { 320×192 256 color or other extended GTIA mode }
else  begin set_antic15_par;   plot_line1b; end; {BASIC 08, 08+16, 08+32}
end;
end;

do_DMACTL;

{       putpixel(32,RAM[$2,$c8],RAM[$2,$c8]); putpixel(32,RAM[$2,$c8],RAM[$2,$c8]);}
{       putpixel(32,RAM[$2,$c8],RAM[$2,$c8]); putpixel(32,RAM[$2,$c8],RAM[$2,$c8]);}

end;

inc_pc_antic;

{ ANTIC scan line done }

until (reg_IR_ANTIC and 64 = 64) and (reg_IR_ANTIC and 15 = 1);
do_VBI_IRQ;
do_VSCRL; { inc vscrl_pos counter }
do_HSCRL; { inc hscrl_pos counter }

{ ANTIC frame done }
end;
```

Teraz testowanie i weryfikowanie, czy wersja programowa wstępna układu ANTIC nadaje się do dalszej pracy jako procedura dobra, tj. szybka, sprawna, ekonomiczna, prawidłowa w kwestii odwzorowania realnej pracy układu ANTIC, etc.

… aż uzyska się procedurę perfekcyjną, którą można będzie uznać za standard realizujący pracę układu elektronicznego ANTIC w formie programowej w odwzorowaniu bliskim 100%.

Na początek sprawdzenie, jak działa procedura w efektach oczekiwanych najbardziej przez użytkownika, ale i przez programistę w pierwszym etapie pracy nad programem, czyli ‚co widać / czy już coś widać / czy działa cokolwiek' w efekcie pracy programu.

Po ustawieniu odpowiednich kolorów, które mogłyby przypominać oczekiwane i prawidłowe…

(potem trzeba przygotować prawidłową paletę kolorów Atari XL/XE dla karty graficznej, z której korzystamy do realizacji programu: *Kolory Atari XL/XE i innych urządzeń…*)

… tryb tekstowy GRAPHICS 0 (ANTIC IR mode $02) wygląda jakby działał prawidłowo.

Czyli wstępnie gotowe, można pracować dalej.

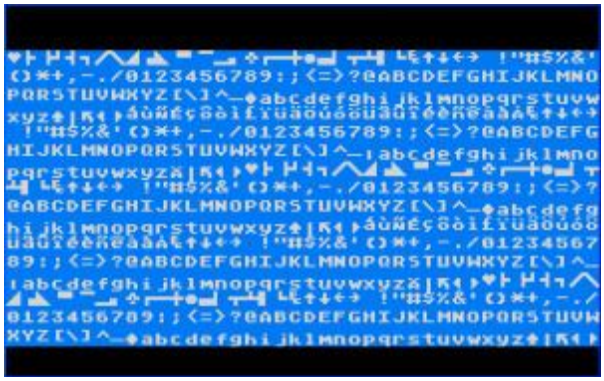
… analogicznie tryb GRAPHICS 1 (ANTIC IR mode $06)

… i GRAPHICS 1+16 (bez okienka tekstowego):

… potem tryb GRAPHICS 2 (ANTIC IR mode $07)

… i GRAPHICS 2+16 (bez okienka tekstowego):

Jeszcze dziwny tryb tekstowy ANTIC IR mode $03 (bez swojego numeru w instrukcji GRAPHICS Atari BASIC):



… potem zostaną sprawdzone kolejne tryby tekstowe i graficzne, z okienkami tekstowymi i bez okienek tekstowych, aż będzie widać, że procedura realizująca programowo pracę ANTIC działa prawidłowo w zakresie testowanym.

Jeszcze spojrzenie na Display List dla trybu ANTIC IR mode $06 (GRAPHICS 1) z okienkiem tekstowym:

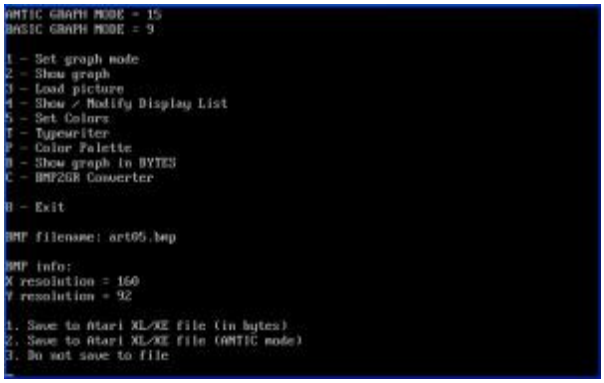
… trzeba sprawdzić adresy: początku Display List, skoków, obszarów pamięci obrazu i prawidłową budowę Display List, w tym ilość pustych linii ekranu, ilość linii danego trybu graficznego lub tekstowego, ilość linii okienka tekstowego, etc.

Budowa Display List dla każdego trybu standardowego ANTIC musi odpowiadać oryginalnym standardowym Display List używanym przez układy elektroniczne Atari XL/XE, oczywiste…

Jeszcze jeden tryb do sprawdzenia, graficzny tym razem z zestawu trybów rozszerzonych, spoza standardowego zestawu trybów tekstowych i graficznych ANTIC – w specyficznej współpracy układów ANTIC i GTIA.

GRAPHICS 9 (tryb graficzny GTIA na bazie ANTIC IR mode $0f):

Najpierw wczytanie jakiegoś ładnego obrazka z pliku BMP:

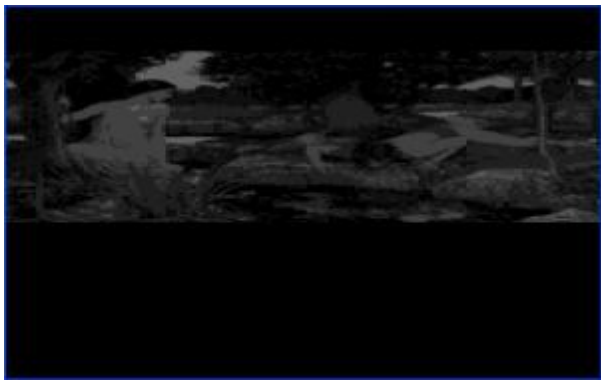


Jak widać obrazek w BMP ma rozdzielczość: 160×92 piksele.

Teraz podgląd obrazka w trybie GRAPHICS 9 Atari XL/XE za pomocą instrukcji ATARI BASIC

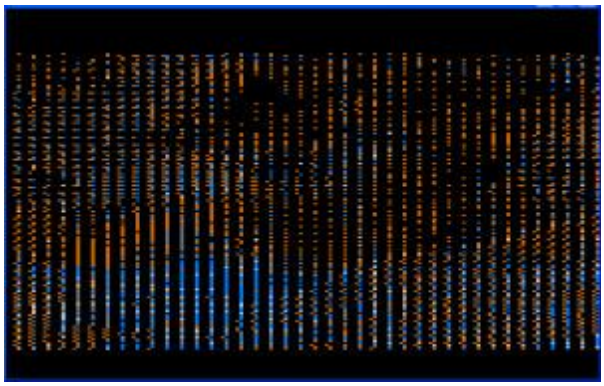
?#6;XLINE$

gdzie XLINE$ to zmienna tekstowa ATARI BASIC, która zawiera kolejne wartości bajów całej jednej linii poziomej obrazka:



Wygląda super. No to sukces. Można sprawdzać i testować parametry wszystkich trybów standardowych tekstowych i graficznych ANTIC + GTIA, oraz dowolnych trybów tekstowych / graficznych tworzonych przez użytkownika na bazie Display List modyfikowanych dowolnie.

Jeszcze podgląd tego samego obrazka w trybie graficznym GRAPHICS 9 po wczytaniu obrazka bajtami wprost do obszaru pamięci ekranu:



Wydaje się, że w trybie GRAPHICS 9 widać za dużo kolorów. Powinno być maksymalnie 16 odcieni, zawsze tylko jednego koloru => do sprawdzenia procedura malująca punkty trybu GRAPHICS 9 w kwestii kolorów (czy problem wynika ze źle ustawionej palety kolorów, czy z nieprawidłowych parametrów działania procedury, lub z nieprawidłowo ustawionych parametrów trybu graficznego, etc.).

Trzeba będzie zapisać obrazek wczytany z pliku BMP do trybu GRAPHICS 9 w formacie właściwym, żeby móc wprost wczytywać z pliku obrazek do pamięci ekranu dla trybu GRAPHICS 9, bez korzystania z instrukcji ?#6;, czyli bez konwertowania bajtów obrazka na dane obrazu wymagane przez tryb graficzny GRAPHICS 9.

Przy przechowywaniu obrazka w bajtach określających kolory pikseli – łatwo można dostosować obrazek do wyświetlenia w dowolnym trybie graficznym lub tekstowym Atari XL/XE korzystając jedynie z instrukcji ATARI BASIC: ?#6;XLINE$, gdzie XLINE$ to zmienna tekstowa ATARI BASIC, która zawiera kolejne wartości bajów całej jednej linii poziomej obrazka.

Przy przechowywaniu obrazka w bajtach właściwych dla danych wymaganych w obszarze pamięci ekranu danego trybu graficznego lub tekstowego, obrazek wyświetli się w pełni prawidłowo tylko w konkretnym trybie graficznym lub tekstowym Atari XL/XE.

Oczywistości.

[..]

Skoro podstawowe funkcje ANTIC działają już prawidłowo, można zająć się funkcją obsługi graczy i pocisków. Na początek wstępna prosta procedura w Turbo Pascal 5.5, żeby mieć od czego zacząć:

```
procedure do_DMACTL;
begin
reg_BAH:=DMACTLH;
reg_BAL:=DMACTLL; mem;

if DB and 32 = 32 then; { 1 = enable instruction fetch DMA }
if DB and 16 = 16 then  { 1 = 1 line P/M resolution }
else;               { 0 = 2 line P/M resloution }
if DB and 08 = 08 then  { 1 = enable player DMA }
if DB and 04 = 04 then  { 1 = enable missile DMA }
if DB and 02 = 02 then
if DB and 01 = 01 then { wide playfield DMA (192 color clocks) }
else                { standard playfield DMA (160 color clocks) }
```

```
else
if DB and 01 = 01 then { narrow playfield DMA (128 color clocks) }
else;           { no playfield DMA }
end;
```

Teraz można czytać dokumentację ANTIC Atari ™, lub dowolne inne opisy działania funkcji z dowolnych materiałów dostępnych gdziekolwiek, żeby zrealizować dobrą, prawidłową i sprawną procedurę w Turbo Pascal 5.5, która będzie wykonywana jako część ANTIC lub GTIA – zobaczymy z czasem, gdzie można umieścić tę procedurę, żeby działała sprawnie i prawidłowo, lub będzie napisane w dokumentacji do ANTIC lub CGIA/GTIA lub gdziekolwiek. Ważne, że musi działać, jak powinna, oczywiste.

Potem testowanie wszystkiego, co się da i jest do sprawdzenia w układach ANTIC i GTIA emulowanych i powinno być

READY

[]

[..]

Przy zabawach emulatorem Atari XL/XE, w tym emulowanymi układami ANTIC i GTIA, można na próbę z ciekawości ‚jak zadziała' ustawiać parametry pracy ANTIC i GTIA Atari XL/XE w sposób dowolny, który mógłby przynieść spodziewane / oczekiwane efekty, np.:

- próba zrealizowania trybu graficznego ANTIC IR mode $0f we współpracy z układem GTIA w taki sposób, żeby uzyskać parametry trybu graficznego 320×200, 256 kolorów.

```
procedure set_antic15_par; { ustaw parametry dla trybu ANTIC IR mode $0f }
begin
x_res:=64;
pixel_prop:=1; lines_per_pixel:=1;
end;

{ ustaw wartość zmiennej określającą ilość kolorów w danym trybie graficznym: }

num_of_colors:=256;
```

Emulator Atari XL/XE robi, co musi zgodnie z parametrami podanymi, oczywiste, stąd wynika, że w trybie GRAPHICS 8 z odpowiednimi parametrami i danymi przekazanymi na odpowiednio ustawioną pracę układu GTIA można uzyskać więcej niż na oryginalnych układach elektronicznych.

Wymagania sprzętowe do zrealizowania trybów wyższych niż możliwe na Atari XL/XE:

- dla trybu 320×200, 256 kolorów wartości do określenia koloru piksela muszą być bajtowe => dodać funkcję przerzucającą bajt danych dla jednego piksela wprost na GTIA do narysowania na ekranie

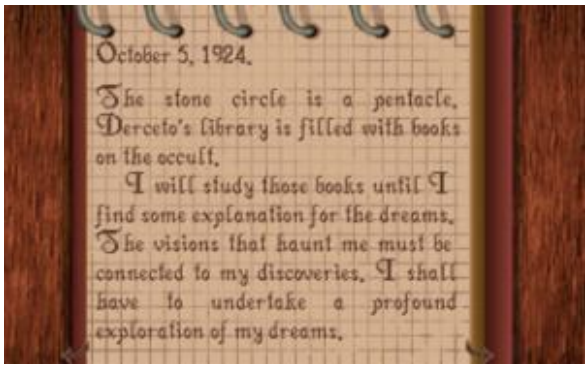
- GTIA musi obsługiwać ilość kolorów z zakresu oczekiwanego przez użytkownika

- dla wyższych trybów graficznych / tekstowych w większych ilościach kolorów, niż 256 ANTIC musi operować większymi strukturami danych dla określenia koloru piksela, a GTIA musi umieć odczytać taką daną i wysłać na ekran w kolorze prawidłowym z oczekiwanego zakresu kolorów

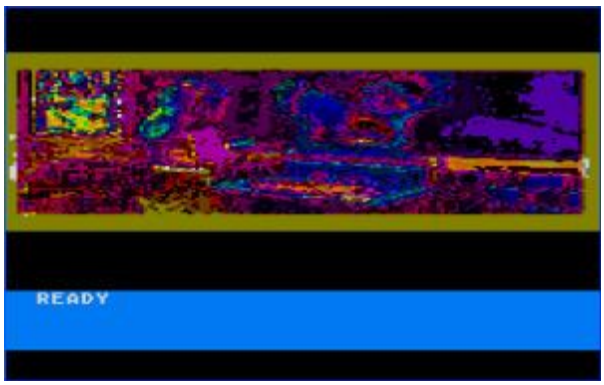
- wymagana pamięć RAM dla obszaru pamięci obrazu do obsługi trybu 320×200, 256 kolorów: 62,5kB, stąd warto przy takich trybach graficznych skorzystać z Atari 130XE

etc. proste, oczywiste.

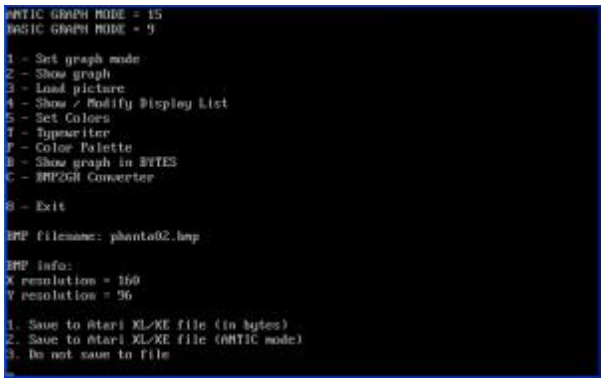
Efekty ustawiania parametrów pracy ANTIC i GTIA Atari XL/XE według pomysłów własnych mogą być zadziwiające:



W trybie GRAPHICS 9 z 256 kolorami bez ustawionej odpowiednio palety kolorów, przykładowy screen z gry Phantasmagoria wygląda jak poniżej:

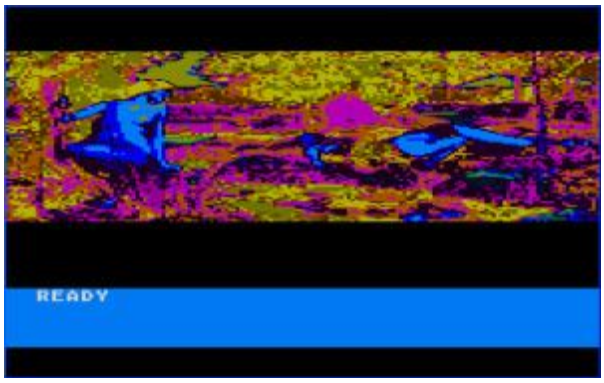


Informacja o pliku BMP:

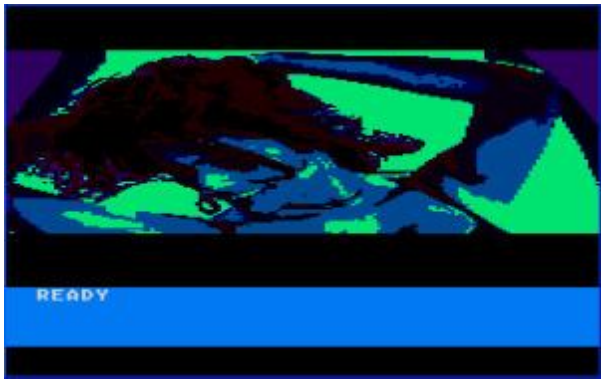


Po ustawieniu palety kolorów właściwej dla wyświetlenia prawidłowo screena z gry, ekran bez okienka tekstowego w trybie o rozdzielczości 160×96 powinien wyglądać jak poniżej:

Jeszcze na próbę, sprawdzenie obrazka z wcześniejszych przykładów w ilości kolorów 256 możliwych do wyświetlenia, dodatkowo z okienkiem tekstowym, w trybie GRAPHICS 9:



I jakiś inny obrazek, żeby mieć pewność, że tryb działa prawidłowo automatycznie:



Ślicznie. 8 kolorów w trybie GRAPHICS 9. Dostępnych, tak, czy inaczej 256 kolorów z palety Atari XL/XE – jest czym się bawić – można robić super gry w dowolnym trybie graficznym lub tekstowym Atari XL/XE w pełnej palecie 256-ciu kolorów dostępnych jednocześnie na ekranie, czyli super zabawa gotowa.

Jak widać: brakuje kursora pod READY, czyli procedura emulująca ANTIC i GTIA nie obsłużyła kursora w trybie GRAPHICS 9, mimo, że udało się wyświetlić okienko tekstowe (modyfikując standardowo Display List) w trybie GRAPHICS 9 – prawdopodobnie kursor pojawia się jedynie w określonych trybach graficznych, bez względu na obecność linii trybu tekstowego GRAPHICS 0, czy pełnego okienka tekstowego standardowego.

Paleta kolorów ustawiona dla Atari XL/XE, stąd wartości kolorów pobrane z pliku BMP wyświetlanego na ekranie są trudne do określenia w kwestii doboru właściwych kolorów – jak widać na przykładzie powyżej.

Tak, czy inaczej – sukces. Jest nad czym i z czym pracować dalej, jak widać.

… i inny obrazek jakiś, dla pewności, że wszystkie tryby działają poprawnie w sposób automatyczny, na przykład w trybie GRAPHICS 7 (ANTIC IR mode $0d):

Bardzo ładnie, przy zabawie kolorami można uzyskać inne palety, inne kolory, akurat wyszło automatycznie, jak widać, zakres zabawy jest zawsze określony dostępnymi trybami tekstowymi i graficznymi Atari XL/XE, oraz wyborem funkcji rysującej linie skaningowe (dane 1 bitowe, 2 bitowe, 4 bitowe, 8 bitowe).

W przykładzie powyżej kursor widać, bo tryb GRAPHICS 7 jest standardowo z okienkiem tekstowym, gdyby wyłączyć okienko tekstowe, uruchamiając tryb GRAPHICS 7+16, nie byłoby okienka w ogóle, oczywiste…

No to sukces, reszta to zabawa, jak kto lubi, oczywiste…

Podsumowania na dalszą pracę w zakresie architektury 8 bit:

- żeby uzyskać w trybach graficznych Atari XL/XE więcej kolorów, niż standardowo, potrzebne są dodatkowe dwie funkcje rysujące linie skaningowe w układzie ANTIC: line_4b (4 bity na kolor, 16 kolorów) i line_8b (8 bitów na kolor, 256 kolorów)

- do wyświetlenia obrazka w 256 kolorach w rozdzielczości 320×192 punktów z użyciem funkcji line_8b potrzebna jest sensowna pojemność pamięci RAM do przechowania bajtowej informacji o kolorze dla każdego piksela obrazka, czyli około 64kB pamięci więcej, niż posiada Atari XL/XE – czyli w kwestii wystarczającej pamięci RAM do realizacji pomysłu potrzebny jest Atari 130XE

- przystosowanie układu GTIA do obsługi kolorów 4-bitowych i 8-bitowych przy wyświetlaniu pikseli na ekranie – na podstawie informacji podawanych z układu ANTIC z możliwością ustawiania w sposób programowy palety dostępnych 256-ciu kolorów, poprzez modyfikowanie rejestrów składowych każdego koloru Red, Green, Blue (żeby nie być ‚uwięzionym' w raz określonej na stałe standardowej palecie 256-ciu kolorów Atari XL/XE)

Atari 130XE z przystosowanymi układami ANTIC i GTIA pozwala uzyskać tryby graficzne i tekstowe Atari XL/XE w 256-ciu kolorach wyświetlanych jednocześnie na ekranie.

Istnieje również możliwość uzyskania trybów graficznych o podwojonej długości standardowej linii skaningowej, maksymalnie do rozdzielczości 640×384 w 4-ech kolorach – po przystosowaniu układów ANTIC i GTIA do pracy w trybach omawianych przykładowych.

Przykładowe numery trybów graficznych i tekstowych dla rozszerzonych układów ANTIC i GTIA:

0 do 15 + 16 + 32 (standardowe tryby – dla danych 1-bitowych i 2-bitowych)

64 do 79 + 16 + 32 (rozszerzone tryby – dla danych 4-bitowych i 8-bitowych)

128 do 143 +16 + 32 (rozszerzone tryby – podwojona długość linii skaningowej dla danych 1 i 2 bitowych)

192 do 207 + 16 + 32 (rozszerzone tryby – podwojona długość linii skaningowej dla danych 4 i 8 bitowych)

… lub dowolnie według specyfikacji projektanta.

Dalsze ewentualne pomysły sensowne na pracę z grafiką Atari XL/XE mogą zacząć się dopiero od architektury 16-bit.

[..]

Po takim sukcesie zabaw z parametrami emulowanych w Turbo Pascal 5.5 układów elektronicznych ANTIC i GTIA Atari XL/XE, pomysły pojawiają się kolejne:

- rozdzielczość trybu graficznego podwójna, niż maksymalna Atari XL/XE, czyli 640 dla linii skaningowej (poziomej), maksymalna ilość linii pionowych zależy zawsze jedynie od odpowiedniej ilości linii danego trybu graficznego umieszczonych w Display List, czyli w zależności od pojemności pamięci RAM, która jest do dyspozycji

- kolejne rozdzielczości poziome mogą wynikać z podwajania sukcesywnie maksymalnej wielkości linii skaningowej ANTIC/GTIA, czyli 320×2=640, 320×4=1280, 320×8, etc. przy mnożniku ‚2', lub inne dowolne sposoby ustawiania rozdzielczości poziomej

- przy poszukiwaniu maksymalnej sensownej użytkowo rozdzielczości pojawia się pomysł znalezienia rozdzielczości i wszystkich możliwych parametrów dla układów ANTIC i GTIA przy korzystaniu z 64 bitowych struktur, zamiast 8 bitowych typowych dla Atari XL/XE

Tak powstaje sukcesywnie naturalnie poprzez pomysły realizowane / sprawdzane jak działają – specyfikacja techniczna układu ANTIC 64-bit oraz GTIA 64-bit.

Proste, oczywiste.

Wnioski i notatka do dalszej pracy:

- zestaw funkcji rysujących linię skaningową dla trybów graficznych, line_1b (2 kolory, 1 bit na kolor), line_2b (4 kolory, 2 bity na kolor) rozszerzyć o funkcje analogiczne dla większej ilości kolorów, tj. przykładowo line_4b (16 kolorów, 4 bity na kolor), line_8b (256 kolorów, 8 bitów na kolor) i w następnym etapie rozszerzania zestawu funkcji rysujących linię skaningową – przejście do struktur wyższych niż jeden bajt, opisujących informację o kolorze piksela

- rozdzielczości opisać funkcją liniową ogólną (ax+b), lub kilkoma konkretnymi, np.: x_res shl 2 + x_res (linia rozdzielczości z mnożnikiem ‚2' typu 5-10-20-40-80-320-640-1280-etc.), lub x_res shl 2 (linia rozdzielczości z mnożnikiem ‚4' typu: 4-8-16-24-32-64-128-256-512-1024-etc. z dostępnymi wartościami pomiędzy wartościami przykładowymi podanymi wyżej), itp.

- ustalić, czy parametr określający rozdzielczość dla danej funkcji wyliczającej realną rozdzielczość jest 1B (bajt), czy 2B (dwa bajty) – określić zakresy rozdzielczości dostępnej na 1B, 2B, etc. – dla danych funkcji

- zachować 100% kompatybilność pracy układów ANTIC-64 i GTIA-64 z układami ANTIC i GTIA Atari XL/XE

- ‚przejście' z pracy ANTIC i GTIA Atari XL/XE do ANTIC-64 i GTIA-64 ustalić na wzrost ilości dostępnych kolorów w standardowych trybach ANTIC i GTIA poprzez ‚uwolnienie' zakresu dostępnych kolorów z równoczesnym zastosowaniem właściwych funkcji rysujących linię skaningową w miejsce standardowych line_1b, line_2b dla tych samych trybów graficznych standardowych ANTIC i GTIA

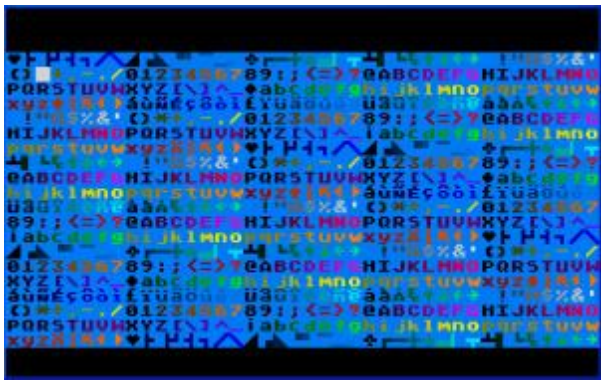
- z powyższego stanu pracy ANTIC-64 i GTIA-64 dalsze rozdzielczości i tryby graficzne / tekstowe, mieszane, etc. z uwolnioną paletą kolorów – dostępne zgodnie ze specyfikacją określoną dla IR 64-bit.

IR 8-bit ANTIC + GTIA => uwolnienie palety kolorów i zastosowanie właściwych funkcji rysowania linii skaningowej dla palety kolorów / struktur danych dot. obrazu => IR 64-bit ANTIC-64 + GTIA-64.

Cosmos READY[]

Teraz testowanie nowych trybów Atari XL/XE dla ‚uwolnionej' palety kolorów (64-bit zamiast 8-bit).

Przykład trybu ANTIC IR mode $02 (GRAPHICS 0) z paletą 256 kolorów dostępnych jednocześnie na ekranie:

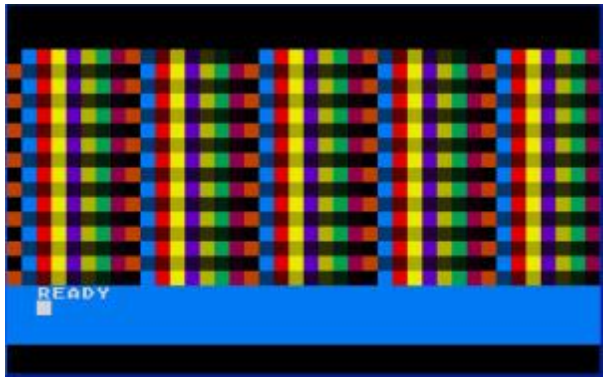


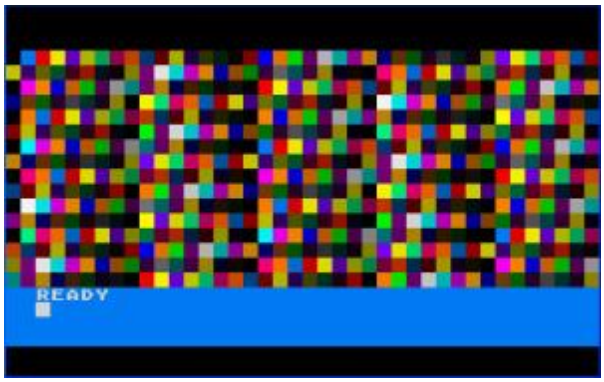
Przykład trybu ANTIC IR mode $02 (GRAPHICS 0) z paletą 8-miu odcieni wybranego koloru:

Przykład trybu ANTIC IR mode $02 (GRAPHICS 0) z paletą 16-tu odcieni wybranego koloru:

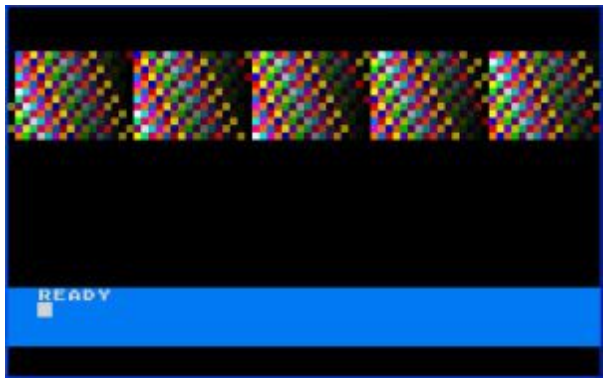
Analogicznie pozostałe tryby tekstowe / graficzne Atari XL/XE.

Przykładowo ANTIC IR mode $08 (GRAPHICS 3) w 4, 16 i 256 kolorach:





Inny przykład – ANTIC IR mode $0a w 16 i 256 kolorach:



ETC.

[..]

Po ‚przejściu' na IR 16-bit w miejsce IR 8-bit, wszystkie tryby tekstowe i graficzne Atari XL/XE można przykładowo podwoić w długości linii skaningowej i analogicznie liczba linii w pionie.

W kolejnych ‚przejściach' na kolejno dodawane IR 8-bit, do zestawu poprzednich IR 8-bit dochodzi się do IR 64-bit, czyli pełen IR dla ANTIC-64 / GTIA-64.

Cosmos READY working visible.

[..]

Dostosowanie ANTIC Atari XL/XE do pracy w architekturze 64-bit nie wymaga żadnych zmian w układzie: rejestry, adresy, końcówki, etc. – wszystko natychmiastowo pracuje w architekturze 64-bit po podłączeniu układu ANTIC do systemu.

Pełne profesjonalne wykorzystanie układu ANTIC Atari XL/XE w architekturze 64-bit jest możliwe po:

- uzupełnieniu linii adresowych i linii danych w układzie ANTIC do adresowania i przesyłania danych korzystając z rejestrów 64-bit zamiast standardowo 8-bit

- ewentualnym (dodatkowo) określeniu pracy ANTIC-64 dla pozostałych 56-ciu bitów na rejestrach / adresach i ew. innych użytecznych końcówkach dodanych do układu

Przykładowo jeden IR 8-bit rozszerza się do 8-miu IR 8-bit, czyli trybów graficznych ANTIC w wersji 64-bit ANTIC Atari XL/XE może być 13^8-13 więcej niż w ANTIC standard, lub więcej:

$02-$0f x $02-$0f x $02-$0f x $02-$0f x $02-$0f x $02-$0f x $02-$0f x $02-$0f

Dodając tryby graficzne GTIA: 3^8-3 trybów więcej, lub więcej.

Ilość i jakość oraz zastosowanie i efekty zależą od sposobu wykorzystania pozostałych pustych 56-ciu bitów rejestrów ANTIC-64.

Analogicznie puste linie, skoki, skrole poziome, pionowe, etc. – dotyczą każdego dodanego IR 8-bit osobno lub mogą pozostać jedynie na jednym standardowym IR 8-bit.

Paleta kolorów 64-bit złożona z 32-bit kolorów i 32 bit odcieni (analogicznie do ANTIC / GTIA standard: paleta 256 kolorów złożona z 4 bit kolorów i 4 bit odcieni).

Gracze i pociski analogicznie.

ETC.

Super maszyna graficzna, oczywiste.

Proste.

Przykłady konkretnych rozwiązań – Układ ANTIC-64.

- tryby tekstowe, graficzne (multiplikowane, inne)

- Multi Display List, Multi Screen Display

- własny / niezależny RAM, ROM, GPU

- pełna obsługa In/Out Video w czasie rzeczywistym z przetwarzaniem i współpracą z układem POKEY-64 oraz systemem Atari-64

- inne

Zgodnie z dokumentacją układu ANTIC-64 ,2015 smakubooks.com / 64-bit.eu.

[..]

ANTIC 64-bit READY.

GTIA 64-bit READY.

,2015

Podsumowania:

Lista funkcji rysujących linię skaningową:

line_1b – 1 bit na kolor, 2 kolory
line_2b – 2 bity na kolor, 4 kolory
line_4b – 4 bity na kolor, 16 kolorów
line_8b – 8 bitów na kolor, 256 kolorów
[..]
line_64b – 64 bity na kolor, [2^64] kolorów

Opis rozdzielczości poziomej trybu graficznego / tekstowego (długość linii skaningowej):

horiz_pixel_res_1B – 1 bajt na określenie długości linii skaningowej
horiz_pixel_res_2B – 2 bajty na określenie długości linii skaningowej
horiz_pixel_res_nB – n bajtów na określenie długości linii skaningowej

Funkcje określające realną długość linii skaningowej wg funkcji liniowej typu ax+b, np.:

{ linia rozdzielczości typu 5-10-20-40-80-160-320-640-1280-2560-etc.: }
horiz_pixel_res_nB shl 2 + horiz_pixel_res_nB

{ linia rozdzielczości typu 2-4-8-16-32-64-128-256-512-1024-2048-etc. z wartościami pośrednimi: }
horiz_pixel_res_nB shl 2

Funkcje określane dynamicznie lub konkretnie na układach elektronicznych.

[..]

Literatura:

1. Asembler 6502 / Jan Ruszczyc ; Stołeczny Ośrodek Elektronicznej Techniki Obliczeniowej., Warszawa, Wydaw. SOETO, 1987

2. ANTIC C012296 (NTSC), REV. D, Atari Inc., 10.1982

Wersja PDF artykułu (ostatnia aktualizacja z dn. 25 października 2017)