

**Video Board XE
specification**

by electron of taquart

This is a very early and incomplete version.

Wstęp

Video Board XE (VBXE) jest urządzeniem będącym rozwinięciem układu GTIA komputerów ATARI XL/XE. Dubluje on funkcje układu GTIA dekodując niezależnie dane przesyłane przez układ ANTIC i dodając do wyświetlanych danych dodatkowe informacje. Stary układ GTIA pozostaje w komputerze pełniąc dotychczasową funkcję bez zmian, oraz zapewniając sygnał synchronizacji, którego VBXE nie musi generować.

VBXE generuje obraz poprzez oddzielne wyjście RGB sterowane przez przetwornik DAC True-Color ograniczony do palety Hi-Color tj. 65536 kolorów w formacie RGB565. (5 bitów koloru R, 6 bitów G i 5 bitów B).

Urządzenie składa się z następujących modułów:

- układu FPGA ALTERA ACEX 1K,
- pamięci FAST SRAM (10ns) o organizacji 512K x 8,
- przetwornika True Color DAC.

Układ FPGA oparty na RAM statycznym konfigurowany jest przez komputer ATARI z wykorzystaniem rejestru PORTA układu PIA 6520 (2 port joysticka). Niniejszy dokument nie jest opisem konfiguracji.

VBXE może zostać zamontowany tylko w komputerach wyposażonych w układ FREDDIE.

1. Możliwości układu

Poniższe możliwości i dalszy opis dotyczą VBXE zaprogramowanego plikiem konfiguracyjnym w wersji A0.

- wyświetlanie standardowego PLAYFIELD
- wyświetlanie zawartości własnego VRAM (OVERLAY) z ustalonym priorytetem względem PLAYFIELD
- Każdy piksel OVERLAY może przyjmować jeden z 256 kolorów
- 2 konfigurowalne palety 256 kolorów – jedna dla PLAYFIELD a druga dla OVERLAYA
- modyfikowanie kolorów PLAYFIELD poprzez tzw. mapę koloru umożliwiającą przydział każdemu kwadratowi o wielkości 8x8 lub 4x4 piksele własnego zestawu kolorów PLAYFIELD oraz innego priorytetu OVERLAYA względem kolorów PLAYFIELD. Dodatkowo możliwa jest zmiana rozdzielczości PLAYFIELD.
- Mechanizm FILL i BLITTERA wspomagający wyświetlanie obiektów na OVERLAY
- możliwość detekcji kolizji obiektów na OVERLAY
- stronicowanie OVERLAYA – wyświetlana jest 1 z 4 stron 128KB pamięci VRAM, na innej stronie VRAM w tym czasie można przeprowadzać transfery.
- stronicowanie mapy koloru – wybór 1 z 16 dostępnych stron.

2. Wykorzystanie pamięci VRAM

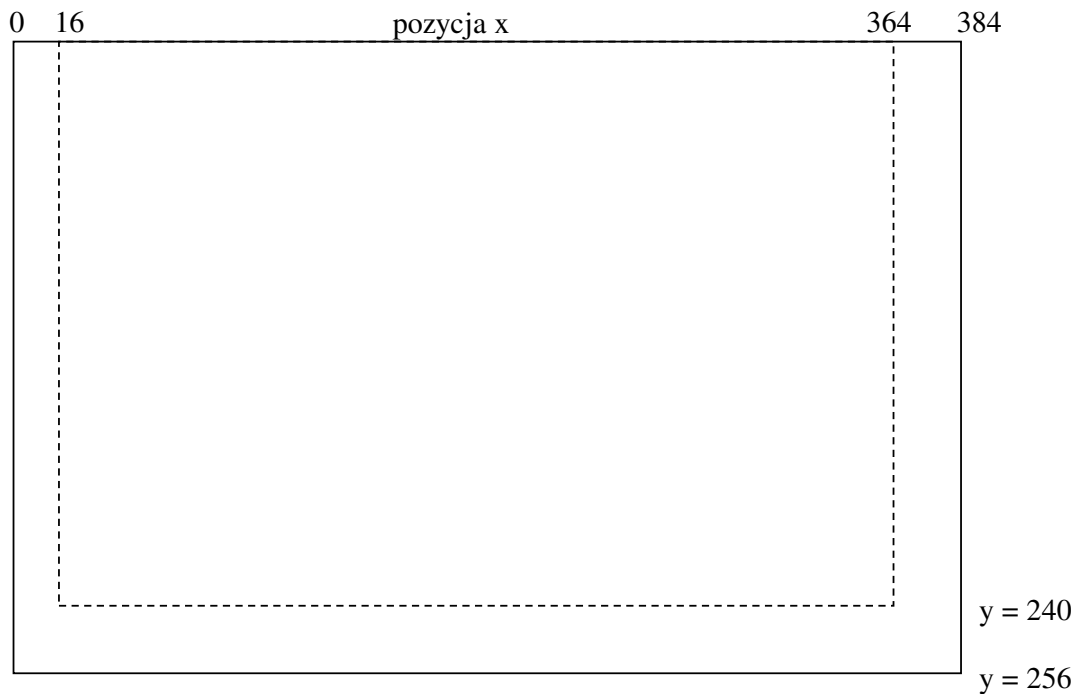
Układ wyposażony jest w 512KB szybkiej pamięci RAM (zwanej dalej VRAM), która służy do przechowywania danych do wyświetlenia w postaci OVERLAYA oraz danych mapy koloru. Pamięć podzielona jest na 4 banki Video dla OVERLAYA i 16 banków mapy koloru. Przełączanie banków odbywa się za pomocą rejestrów DISPAGE, WORKPAGE oraz CMAP_PAGE. W przeciwieństwie do standardowej pamięci Video komputera ATARI dane wyświetlane nie są ciągłe, tzn. każda linia zarówno OVERLAYA jak i mapy koloru oddalona jest od sąsiedniej o odległość 512 bajtów. Stąd krok adresu pamięci dla kolejnych punktów w pionie wynosi 512 bajtów. Umożliwia to łatwiejszą modyfikację tego adresu.

Poniższy rysunek przedstawia schematycznie podział pamięci na poszczególne banki.

<p>Adr :0</p> <p style="text-align: center;">VIDEO PAGE 0</p> <p>Obszar pamięci o wielkości 384x256 bajtów. Kolejne linie oddalone są o 512 bajtów. Pozostały obszar od adresu xx384 do xx511 zajmują dane mapy koloru, której kolejne wyświetlane linie też różnią się adresem o 512 bajtów. W rzeczywistości wyświetlane jest 348x240 pikseli (1 piksel = 1 bajt).</p>	<p>Adr :384</p> <p>CMAP PAGE 0</p> <p>CMAP PAGE 1</p> <p>CMAP PAGE 2</p> <p>CMAP PAGE 3</p>	<p>Adr :511</p>
<p style="text-align: center;">VIDEO PAGE 1</p>	<p>CMAP PAGE 4</p> <p>CMAP PAGE 5</p> <p>CMAP PAGE 6</p> <p>CMAP PAGE 7</p>	<p>128KB</p> <p>256KB</p>
<p style="text-align: center;">VIDEO PAGE 2</p>	<p>CMAP PAGE 8</p> <p>CMAP PAGE 9</p> <p>CMAP PAGE 10</p> <p>CMAP PAGE 11</p>	<p>384KB</p>
<p style="text-align: center;">VIDEO PAGE 3</p>	<p>CMAP PAGE 12</p> <p>CMAP PAGE 13</p> <p>CMAP PAGE 14</p> <p>CMAP PAGE 15</p>	<p>512KB</p>

3. Wyświetlanie OVERLAY

Poniższy rysunek przedstawia położenie widzialnego obszaru OVERLAY (przerywana linia) w każdym banku 128KB.



Zawartość pamięci poza obszarem zakreskowanym nie jest wyświetlana. Wyświetlany obszar ma wielkość 348x240 pikseli i całkowicie pokrywa zakres PLAYFIELD. Wyświetlana rozdzielczość odpowiada trybowi hi-res ANTICa generowanemu z częstotliwością 7,09 MHz w systemie PAL. Każdy bajt pamięci VRAM definiuje kolor jednego wyświetlanego punktu. Wartość bajtu staje się indeksem do 256-kolorowej PALETY2. Występują tu specjalne przypadki, gdy:

- bajt ma wartość 0 – wówczas wyświetlane jest PLAYFIELD (przezroczystość)
- bajt ma wartość 255 – wówczas wyświetlane jest PLAYFIELD ale kolor dobierany jest z PALETY2 zamiast z PALETY1. Umożliwia to robienie półprzezroczystych obiektów.

Adres każdego piksela można obliczyć wg wzoru:

$$\text{adr} = x + 512 * y + 128\text{KB} * \text{bank}, \quad \text{gdzie } x = 0 \dots 383 \text{ a } y = 0 \dots 255, \text{ bank} = 0 \dots 3$$

Po zmianie wyświetlanego banku za pomocą rejestru DISPAGE wyświetlanie nowej zawartości rozpocznie się po najbliższej synchronizacji pionowej, czyli nie trzeba czekać ze zmianą banku na synchronizację, wystarczy wpisać nowy bank, a przełączy się on w odpowiednim momencie.

// do napisania

4. Mapa kolorów
5. Transfer danych do VRAM poprzez IO
6. Operacje FILL i BLITTERA na VRAM

Address	When write	When read
\$d600	MODE	MODE
\$d601	ADRXL	COLXL
\$d602	ADRXH	COLXH
\$d603	ADRY	COLY
\$d604	XCOUNT	COL_COLOR
\$d605	YCOUNT	0
\$d606	DISPAGE	DISPAGE_NOW
\$d607	WORKPAGE	0
\$d608	START	BUSY
\$d609	SRCA0	0
\$d60a	SRCA1	0
\$d60b	SRCA2	0
\$d60c	SDATA	0
\$d60d	MAIN_PRIOR	0
\$d60e	BLT_CONFIG	BLT_CONFIG
\$d60f	-	0
\$d610	PAL_ADRL	0
\$d611	PAL_ADRH	0
\$d612	PAL_DATAR	0
\$d613	PAL_DATAG	0
\$d614	PAL_DATAB	0
\$d615	CBANK_ADR	0
\$d616	CBANK_DATA0	0
\$d617	CBANK_DATA1	0
\$d618	CBANK_DATA2	0
\$d619	CBANK_DATA3	0
\$d61a	CMAP_HSCROLL_L	0
\$d61b	CMAP_HSCROLL_H	0
\$d61c	CMAP_VSCROLL	0
\$d61d	CMAP_PAGE	0
\$d61e	CMAP_SPLIT	0

MODE – główny rejestr konfiguracyjny VBXE

Bity 0-1 – tryb pracy układu

00 – tylko wpisywanie ręczne danych do pamięci poprzez rejestr SDATA. Służy do transferu danych pamięć główna -> pamięć VBXE. Transfer w tym trybie polega na wpisywaniu kolejnych bajtów z automatyczną zmianą adresu docelowego zdefiniowaną przez bity 2-3 rejestru MODE. Adres docelowy inicjowany jest poprzez rejestry SRCA0, SRCA1 i SRCA2.

01 – tryb FILL. Bloki pamięci Video zostaną wypełnione wartością ostatnio wpisaną do rejestru SDATA. Pozycję początkową bloku określają rejestry ADRXL, ADRXH (pozycja X = 0...511), ADRY (pozycja Y = 0...255). (Pozycje X i Y odpowiadają adresowi w pamięci wg wzoru $ADR = X + 512 * Y$). Wielkość bloku w osi X i Y określają rejestry XCOUNT i YCOUNT. Tryb przydatny szczególnie do kasowania starych sprite'ów (gdy SDATA = 0). Wypełnianie bloku rozpocznie się, gdy do rejestru START zostanie wpisana wartość różna od 0. W trakcie trwania operacji rejestr BUSY przyjmuje wartość 1. Po zakończeniu rejestr BUSY przyjmuje wartość 0. Tryb FILL jest 2-3 krotnie szybszy od trybu BLITTER.

1x – tryb BLITTER. Bloki pamięci Video zostaną wypełnione zawartością przepisaną z pamięci VBXE spod adresu określonego przez rejestry SRCA0, SRCA1 i SRCA2. Pozycję początkową bloku docelowego określają rejestry ADRXL, ADRXH (pozycja X = 0...511), ADRY (pozycja Y = 0...255). Wielkość bloku w osi X i Y określają rejestry XCOUNT i YCOUNT. Wypełnianie bloku rozpocznie się, gdy do rejestru START zostanie wpisana wartość różna od 0. W trakcie trwania operacji rejestr BUSY przyjmuje wartość 1. Po zakończeniu rejestr BUSY przyjmuje wartość 0. Na pracę w tym trybie wpływa rejestr BLT_CONFIG.

UWAGA: wszystkie rejestry konfiguracyjne (ADRXL, ADRXH, ADRY, SDATA, SRCA0, SRCA1, SRCA2, XCOUNT, YCOUNT, WORKPAGE, BLT_CONFIG) są podwójnie buforowane, czyli po uruchomieniu operacji rejestrem START można je skonfigurować do nowego zadania nie czekając na zakończenie operacji (BUSY = 0).

Zatrzymanie dowolnej operacji (gdy BUSY = 1) jest możliwe poprzez zmianę tych dwóch bitów MODE na dowolną inną wartość.

Bity 3-2 – określają sposób automatycznej zmiany adresu przeznaczenia danej w trybie 00

00 – adres docelowy zwiększany jest o 1 (+ 1 pixel w osi X)

01 – adres docelowy zmniejszany jest o 1 (- 1 pixel w osi X)

10 – adres docelowy zwiększany jest o 512 (+ 1 pixel w osi Y)

11 – adres docelowy zmniejszany jest o 512 (- 1 pixel w osi Y)

bit 4 – nieużywany

bit 5 – rozdzielczość mapy koloru

0 – 4x4 pixele

1 – 8x8 pixeli

bit 6 – włączenie mapy koloru

0 – wyłącz mapę koloru

1 – włącz mapę koloru

bit 7 – wyświetlanie danych Video z pamięci VBXE

0 – wyłącz

1 – włącz

Stan tego bitu nie wpływa na wyświetlanie danych przychodzących z ANTICA, jedynie wyłącza transfer z pamięci VBXE. Dane są wyświetlane ze 128 KB strony określonej przez DISPAGE.

ADRXL, ADRXH – Pozycja X dla najbliższej operacji wpisywania, FILL lub BLITTERA. ADRXL (młodsza część) ma szerokość 8 bitów, ADRXH ma szerokość tylko jednego bitu.

ADRY - Pozycja Y dla najbliższej operacji wpisywania, FILL lub BLITTERA. ADRY ma szerokość 8 bitów.

XCOUNT, YCOUNT – wielkość bloku docelowego w pikselach. Pomniejszona o 1.Np. gdy chcemy uzyskać blok o wielkości 32x16 pikseli wpisujemy tu wartości odpowiednio 31 i 15.

DISPAGE – zmiana wyświetlanej strony pamięci Video (128 KB z 512 KB) DISPAGE = 0..3
Po wpisaniu danej do DISPAGE wyświetlana strona pamięci ulegnie zmianie dopiero po najbliższej synchronizacji pionowej. DISPAGE_NOW przy odczycie podaje zawsze aktualnie wyświetlaną stronę pamięci.

WORKPAGE – strona 128 KB pamięci VRAM, na której wykonujemy operacje FILL lub BLITTERA.
WORKPAGE = 0..3

START – wpisanie wartości różnej od 0 rozpocznie operację FILL lub BLITTERA. Później zawartość rejestru BUSY = 1 informuje o trwaniu operacji. BUSY = 0 informuje o jej zakończeniu. Zatrzymanie operacji możliwe poprzez zmianę dwóch najmłodszych bitów MODE.

SRCA0, SRCA1, SRCA2 – adres liniowy dla danych źródłowych BLITTERA lub adres docelowy wpisywanej danej w trybie MODE 0. (SRC0 – część najmłodsza)
Dane dla BLITTERA mogą pochodzić z dowolnego miejsca 512 KB RAM, Dane są ciągłe, tzn. najpierw pierwsza linia od lewej do prawej później bezpośrednio druga linia od lewej do prawej itd. Każdy bajt danych określa jeden piksel docelowy. Wartość 0 oznacza punkt przezroczysty. Wartość 255 oznacza punkt, który wyświetla dane PLAYFIELD ale z paletą kolorów OVERLAYA.

SDATA – dana do wpisania do pamięci dla trybów MODE 00 i MODE 01. W trybie MODE 00 po wpisaniu do rejestru SDATA adres określony przez SRCA0, SRCA1 i SRCA2 jest automatycznie modyfikowany zgodnie z ustawieniami bitów MODE bit2 i bit3. Można dzięki temu wpisywać kolejne dane bez potrzeby ręcznej modyfikacji adresu docelowego.

MAIN_PRIOR – główny rejestr priorytetu OVERLAYA względem danych PLAYFIELD z ANTICA. Znaczenie mają tylko dwa najmłodsze bity.

00 – OVL, PF0, PF1, PF2, PF3 - najwyższy priorytet
01 – PF0, OVL, PF1, PF2, PF3
10 – PF0, PF1, OVL, PF2, PF3
11 – PF0, PF1, PF2, PF3, OVL – najniższy priorytet

gdy włączona jest mapa koloru (MODE bit 6) wówczas rejestr MAIN_PRIOR traci znaczenie. Priorytety ustalone są niezależnie dla każdego pola z mapy koloru.

BLT_CONFIG – konfiguracja blittera.

bit 7 – 1 = włączona detekcja kolizji podczas blitu (spowalnia blitter do 50%)
bit 6 – 1 = wykrywanie ostatniej kolizji zamiast pierwszej
bity 5-4 – podział palety

00 – 1 x 256 kolorów (każdy bajt danych definiuje kolor 0...255)

01 – 4 x 64 kolory (przepisywane są bity 5...0 z danych + bity 1..0 z BLT_CONFIG (można dzięki temu zmieniać kolor bez zmiany źródła danych !)

10 – 8 x 32 kolory (przepisywane są bity 4...0 z danych + bity 2..0 z BLT_CONFIG (można dzięki temu zmieniać kolor bez zmiany źródła danych !)

11 – 16 x 16 kolorów (przepisywane są bity 3..0 z danych + bity 3..0 z BLT_CONFIG (można dzięki temu zmieniać kolor bez zmiany źródła danych !))

bity 3..0 bity zastępcze koloru (patrz wyżej) przepisywane na miejsce starszych bitów danej podczas bitu.

PAL_ADRL, PAL_ADRH – numer koloru w paletach (0..511). Kolory 0..255 określają paletę dla PLAYFIELD, kolory 256..511 określają paletę dla grafiki wyświetlanej z pamięci VBXE.

PAL_DATAR, PAL_DATAG, PAL_DATAB – składowe R, G, B koloru wybranego przez PAL_ADRL(H). Składowe R i B mają po 5 bitów (najstarsze są ważne) Składowa G ma 6 bitów (najstarsze ważne). Daje to 65536 dostępnych kolorów. Uaktualnienie koloru w paletce następuje dopiero po wpisaniu składowej B. Wówczas też numer koloru określany przez PAL_ADRL(H) zwiększany jest automatycznie o 1 i można wpisywać dane dla następnego koloru z palety.

CBANK_ADR – wartość 0..31 określa jeden z 32 banków kolorów PLAYFIELD używanych przez mapę koloru. Bank 0 nigdy nie jest używany. Po wpisaniu wartości do CBANK_ADR można modyfikować kolory PLAYFIELD poprzez rejestry CBANK_DATA0 CBANK_DATA3

CBANK_DATA0, CBANK_DATA1, CBANK_DATA2, CBANK_DATA3 – odpowiadają rejestrom PF0, PF1, PF2 i PF3 GTIA. Po wybraniu banku przez CBANK_ADR tu wpisujemy odpowiednie kolory. Po wpisaniu wartości do CBANK_DATA3 CBANK_ADR zwiększane jest automatycznie o 1.

CMAH_HSCROLL_L, CMAH_HSCROLL_H – scrolling poziomy mapy kolorów. wartość 0..511 określa przesunięcie w pikselach względem PLAYFIELD. CMAH_HSCROLL brany jest pod uwagę przy każdej synchronizacji poziomej (można zmieniać co linię).

CMAH_VSCROLL - – scrolling pionowy mapy kolorów. wartość 0..255 określa przesunięcie w pikselach względem PLAYFIELD. Wartość brana pod uwagę tylko przy synchronizacji pionowej oraz przy zapisie do CMAH_SPLIT.

CMAH_PAGE – aktywny bank pamięci mapy koloru. Podobnie jak DISPAGE, CMAH_PAGE powoduje zmianę aktywnego banku po najbliższej synchronizacji pionowej. CMAH_PAGE = 0..15

CMAH_SPLIT – wpisanie tu dowolnej wartości spowoduje przepisanie nowych zawartości CMAH_VSCROLL oraz CMAH_PAGE bez czekania na synchronizację pionową począwszy od następnej linii. Umożliwia to przesuwanie mapy koloru tak, aby możliwe było prawidłowe pokrycie kolorami obrazu przy niestandardowej display-list. np. zawierającej pojedyncze puste linie.