```
;VGA Generator

.include "M8515def.inc" ;@18,432MHz

.def _NOP               =r0
.def _Read              =r1
.def _Write             =r2
.def _Precharge =r3
.def _Activate  =r4
.def _NC3               =r5
.def _NC4               =r6
.def Reload             =r7

.def Color              =r8
.def Byte               =r9
.def R                  =r10
.def G                  =r11
.def B                  =r12
.def RX                 =r13


.def temp               =r16            ;Allgemeines Hilfsregister
.def loop               =r17            ;Allgemeiner Schleifenzähler

.def temp2              =r18            ;Allgemeines Hilfsregister

.def ZeileL             =r19
.def ZeileH             =r20
.def Bit                =r21

.def AdressLL   =r26            ;Spaltenadresse Low
.def AdressLH   =r27            ;Spaltenadresse High
.def AdressHL   =r28            ;Zeilenadresse Low
.def AdressHH   =r29            ;Zeilenadresse High

.equ Empfang    =0

.equ VStart     = 0                     ;Beginng von VSync in Zeile : 512 + x
.equ VLang      = 5                     ;VSync Länge in Zeilen
.equ Offs       = 15            ;Beginn des neuen Bildes, x Zeilen nach Ende von VSync

;IO Pins:
;PortA: Pixeldaten 0-7
;PortC: Adresse 0-7, RAMDAC Daten 0-7
.equ AdL                =PortC
;PortB: Adresse 8-11, BS0-1, RAMDAC WR, RAMDAC RD
.equ AdH                =PortB
.equ RS0                =1
.equ RS1                =0
.equ BS0                =4
.equ BS1                =5
.equ WR                 =6
.equ RD                 =7

;PortD
.equ Comm               =PortD
.equ RXD                =0
.equ TXD                =1
.equ DQM                =2
.equ CKE                =3
.equ CS                 =4
.equ RAS                =5
.equ CAS                =6
.equ WE                 =7
```

```
;Port E
.equ HSync = 0                  ;HSync direkt an Monitor
.equ VSync = 1                  ;VSync direkt an Monitor
.equ Blank = 2                  ;Blanking an den RAMDAC


; Befehle zur SDRAM Steuerung
        ;Command                        CKE                     CS              RAS
CAS                     WE              DQM
.equ Activate           =(1<<CKE)                                               |(1<<CAS)       |
(1<<WE)
.equ Read                       =(1<<CKE)                       |(1<<RAS)
|(1<<WE)
.equ Write                      =(1<<CKE)                       |(1<<RAS)
.equ Precharge          =(1<<CKE)                                               |(1<<CAS)
.equ ModeRegister       =(1<<CKE)
.equ NoOP                       =(1<<CKE)                       |(1<<RAS)       |(1<<CAS)       |
(1<<WE)
.equ AutoRefresh        =(1<<CKE)
|(1<<WE)
.equ SelfRefresh        =
(1<<WE) |(1<<DQM)
.equ SelfRefreshExit=(1<<CKE)                   |(1<<RAS)       |(1<<CAS)       |(1<<WE)        |
(1<<DQM)
.equ Deselect           =(1<<CKE)       |(1<<CS)
.equ BurstTerminate     =(1<<CKE)                       |(1<<RAS)       |(1<<CAS)
|(1<<DQM)

.org 0
        rjmp    RESET

.org OVF0addr
        out TCNT0, Reload
        rjmp Timer


.org URXCaddr                           ; Interruptvektor für UART-Empfang
        in RX, UDR
        sbr bit, (1<<Empfang)
        reti

RESET:
        ser     temp
        out PORTA, temp
        out     DDRB,temp
        out PORTB, temp
        out     DDRC,temp
        out     DDRD,temp
        out     DDRE,temp
        out PORTE, temp

        ldi temp, NoOP|(1<<DQM)
        out PORTD, temp

        clr temp
        out     DDRA,temp
        out PORTC, temp

        ldi temp, LOW(RAMEND)
        out SPL, temp
        ldi temp, HIGH(RAMEND)
        out SPH, temp

        ldi temp, NoOP
```

```
        mov _NOP, temp

        ldi temp, Read
        mov _Read, temp

        ldi temp, Write
        mov _Write, temp

        ldi temp, Precharge
        mov _Precharge, temp

        ldi temp, Activate
        mov _Activate, temp

        clr AdressLL
        ldi AdressLH, 192
        clr AdressHL
        ldi AdressHH, 192

        clr ZeileL
        ldi ZeileH, 192

        ldi temp, 9                                             ; 9 für 115,2kBaud bei 18,432MHz, 59
für 29,2
        out UBRRL, temp                                         ; Baudrate einstellen
        ldi temp, (1<<TXEN)|(1<<RXEN)|(1<<RXCIE)
        out UCsRb, temp

        ldi temp, 2                                                     ;Teiler auf 8
        out TCCR0, temp

        ldi temp, 256-72                                        ;32kHz Horizontalfrequenz
        mov Reload, temp
        out TCNT0, Reload

        sei

        rcall InitSD                                            ;SDRAM Initialisierung
        rcall InitSD


ldi temp, 54                                                    ;BMP Header entfernen
dump:
        sbrs bit, Empfang
        rjmp dump
        cbr bit, (1<<Empfang)
        dec temp
brne dump

        rcall InitRAMDAC                                        ;RAMDAC initialisieren und
Farbtabelle laden

        ldi AdressLL, 0
        ldi AdressLH, 192
        ldi AdressHL, 0
        ldi AdressHH, 192

rxloop:                                                                 ;Bild laden, SRDAM macht
einen Autorefresh wenn er nichts zu tun hat
        ldi temp, AutoRefresh
        out Comm, temp
rxloopw:
        sbrs Bit, Empfang
        rjmp rxloopw
        out Comm, _NOP
```

```
        cbr Bit, (1<<Empfang)
        mov temp, RX
        rcall WriteByte

        adiw AdressLH:AdressLL,1
        cpi AdressLH, 192+2
        brne rxloop
        ldi AdressLH, 192

        adiw AdressHH:AdressHL,1
        cpi AdressHH, 192+2
        brne rxloop
        ldi AdressHH, 192

Start:

        ldi temp, (1<<TOIE0)                            ;Timer0 Interrupt aktivieren: HSync
        out TIMSK, temp

hier:
        rjmp hier

WriteByte:
        ser temp2
        out ddra, temp2
        out AdL, AdressHL
        out AdH, AdressHH
        out Comm, _Activate                                     ;Zeilenadresse laden
        out Comm, _NOP
        nop
        out AdL, AdressLL
        out AdH, AdressLH
        out portA, temp
        out Comm, _Write                                        ;Spaltenadresse laden und Byte
schreiben
        out Comm, _NOP
        out Comm, _Precharge
        out Comm, _NOP
        clr temp2
        out ddra, temp2
ret

InitSD:
        ldi loop, 0                                     ;1ms bei 16MHz
        clr temp
StartUpDelay:
        dec temp
        brne StartUpDelay
        dec loop
        brne StartUpDelay

        ldi temp, AutoRefresh           ;8 CBR Cycles
        out Comm, temp
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        out Comm, _NOP

        clr temp
        out AdL, temp
```

```
        ldi temp, Precharge
        ldi temp2, 4                        ;Precharge all Banks
        out AdH, temp2
        out Comm, temp
        out Comm, _NOP

        ldi temp, 7|32                      ;CAS Latency:2, Burst Length: Full Page
        out AdL, temp
        ldi temp, 192|2                     ;Burst Read, Single Write
        out AdH, temp
        ldi temp, ModeRegister
        out Comm, temp
        out Comm, _NOP
ret


InitRAMDAC:
        cbi portb, RS0                  ;Init Data Mask
        sbi portb, RS1
        ser temp
        out portc, temp
        nop
        nop
        cbi portb, WR
        nop
        nop
        sbi portb, WR
        nop

        cbi portb, RS0                  ;Reset LUT Adress
        cbi portb, RS1
        clr temp
        out portc, temp
        nop
        nop
        cbi portb, WR
        nop
        nop
        sbi portb, WR
        nop

        sbi portb, RS0                  ;Init LUT
        cbi portb, RS1

        clr loop
LUT:
colors:
        sbrs bit, Empfang
        rjmp colors
        cbr bit, (1<<Empfang)
        mov B,RX
        lsr B
        lsr B
colors1:
        sbrs bit, Empfang
        rjmp colors1
        cbr bit, (1<<Empfang)
        mov G,RX
        lsr G
        lsr G
colors2:
        sbrs bit, Empfang
        rjmp colors2
        cbr bit, (1<<Empfang)
        mov R,RX
```

```
        lsr R
        lsr R
colors3:
        sbrs bit, Empfang
        rjmp colors3
        cbr bit, (1<<Empfang)


        out portc, R
        nop
        cbi portb, WR           ;R
        nop
        nop
        sbi portb, WR
        nop
        nop
        nop

        out portc, G
        nop
        cbi portb, WR           ;G
        nop
        nop
        sbi portb, WR
        nop
        nop
        nop

        out portc, B
        nop
        cbi portb, WR           ;B
        nop
        nop
        sbi portb, WR
        inc loop
        brne LUT
ret

UART_TXD:                                           ; Start transmission of data (r16)
        sbis UCSRA, UDRE
        rjmp UART_TXD
        out UDR, temp
ret

Timer:
        cbi PortE, HSync                    ;H-Sync: Low
        cbi PortE, Blank                    ;Blanking: Low
        sbi PortE, VSync                    ;V-Sync: High

        out Comm, _Precharge            ;Stop Burst Read
        out Comm, _NOP

        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
```

```
        sbi PortE, HSync

        out AdL, ZeileL
        out AdH, ZeileH
        out Comm, _Activate
        out Comm, _NOP
        ldi temp, 245
        out AdL, temp
        ldi temp, 192+1
        out AdH, temp
        out Comm, _Read
        out Comm, _NOP

        ldi temp, 1
        add ZeileL, temp                        ;Zeilenzähler
        dec temp
        adc ZeileH, temp

        sbrs ZeileH, 1
        sbi PortE, Blank                        ;Blanking: off

        sbrs ZeileH, 1
        rjmp ExInt
ZeilenC:

        cpi ZeileL,VStart                       ;VSync Beginn ? (bei Zeile 512+VStart)
        brne Sel1
        cbi PortE, VSync                        ;V-Sync: 31,5kHz / 525 Zeilen = 60Hz
        rjmp ExInt
Sel1:
        cpi ZeileL,VStart+VLang         ;VSync Ende ? (bei Zeile 512+VStart+VLang)
        brne Sel2
        sbi PortE, VSync
        rjmp ExInt
Sel2:
        cpi ZeileL,VStart+VLang+Offs;Beginn von neuem Bild ? (bei Zeile 512+VStart+VLang+Offs)
        brne ExInt
        sbi PortE, Blank                        ;Blanking: off
        ldi ZeileH, 192
        clr ZeileL
ExInt:
reti
```