

GRUNDLAGEN

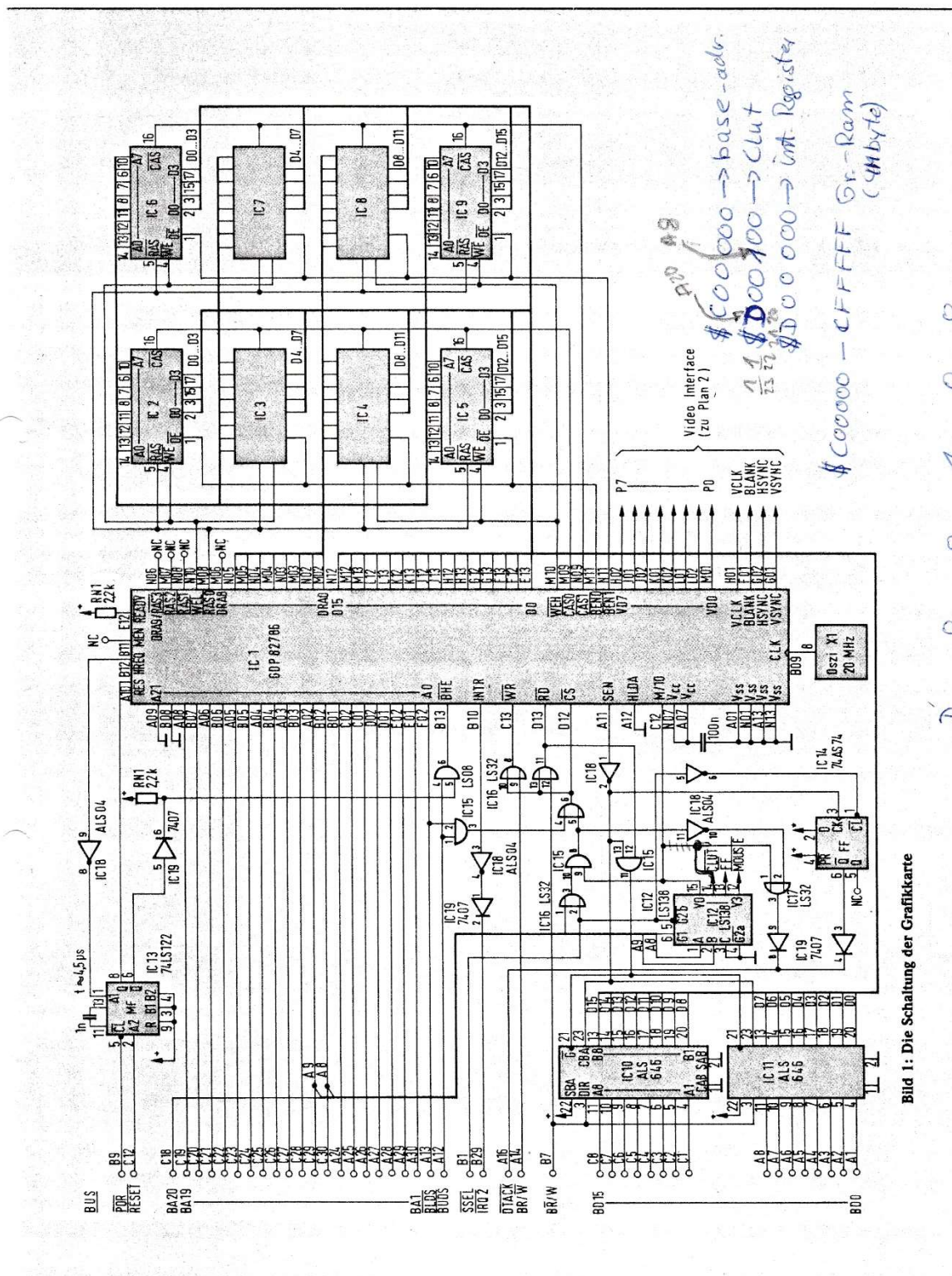


Bild 1: Die Schaltung der Grafikkarte

GRUNDLAGEN

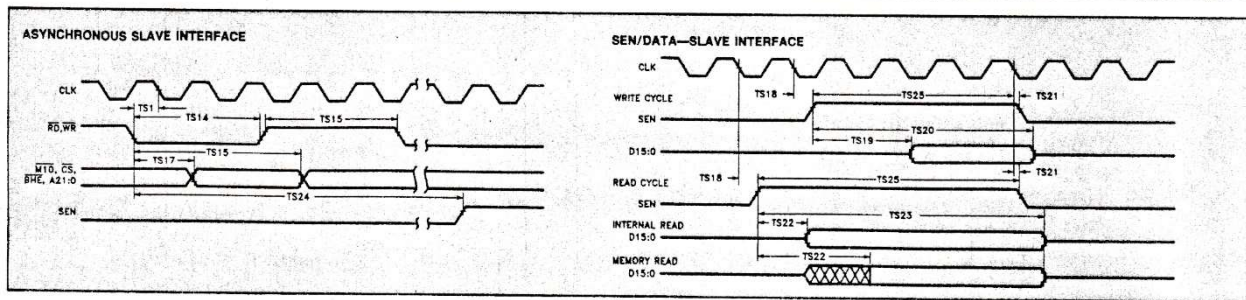


Bild 2: So sieht das Zeitverhalten des GDP 82786 bei asynchronen Buszugriffen aus

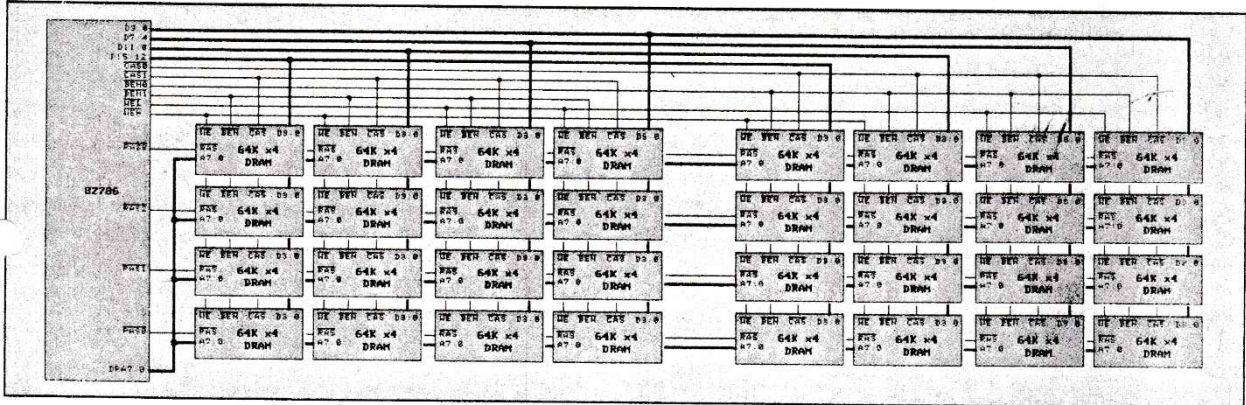


Bild 3: Massenhaft Bildspeicher: 4 Reihen von Interleaved Bänken mit 64K x 4 Bit DRAMs

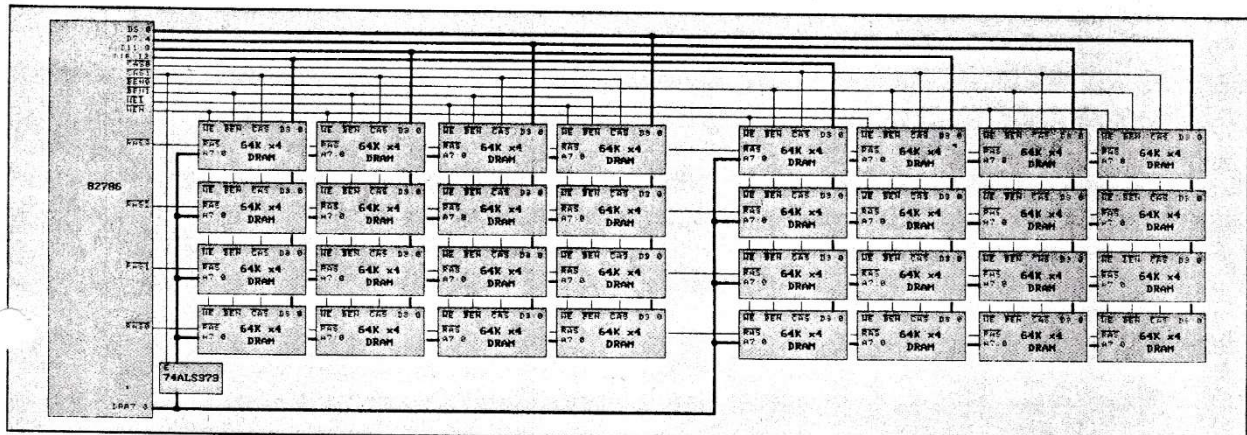
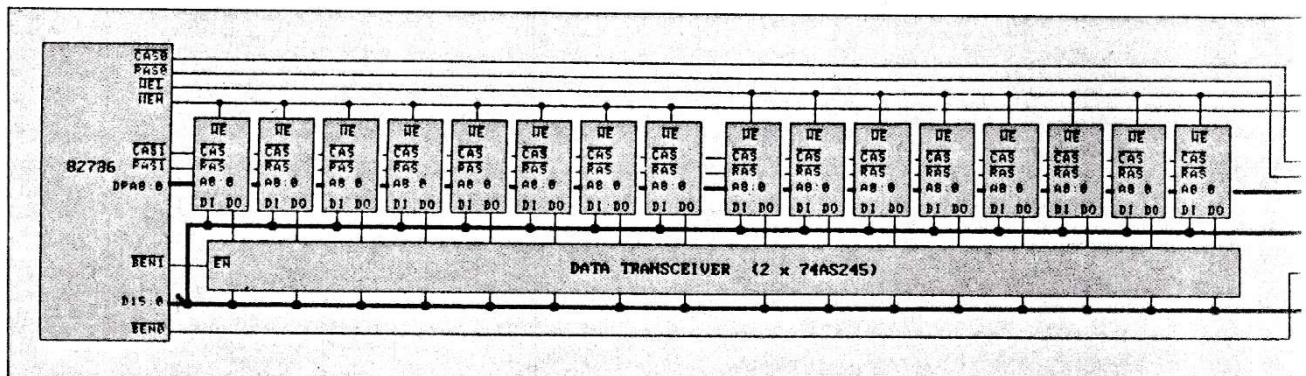
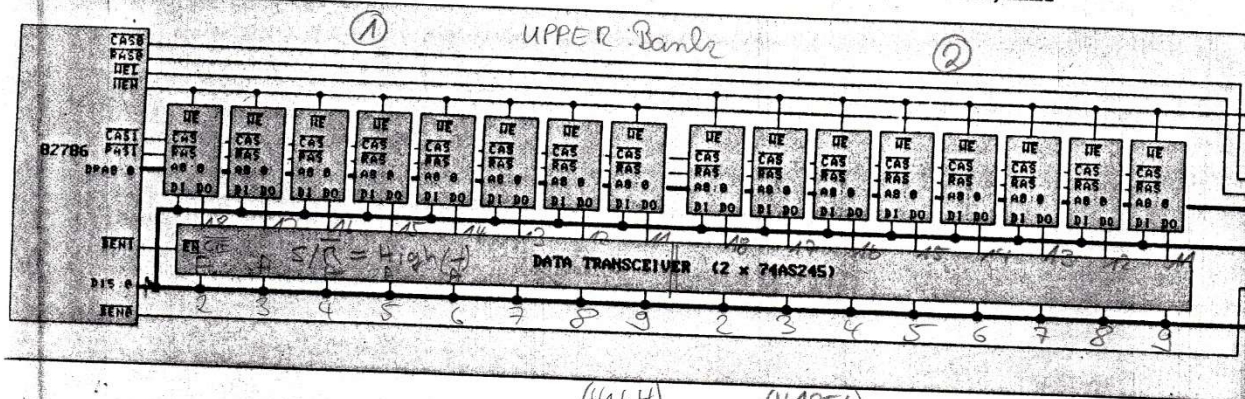


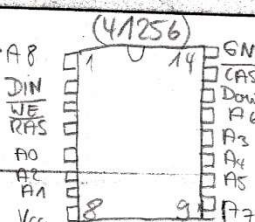
Bild 4: Bei Static Column DRAMs kommt zur Schaltung aus Bild 3 noch ein externer Zwischenspeicher (74 ALS 373) hinzu



Bei statischen Column DRAMs kommt zur Schaltung aus Bild 3 noch ein externer Zwischenspeicher (74 ALS 373) hinzu



(4164)
LD NC → A8
sonst
überall gleich!



März 1988

Ansprechen des DRAMs wird vom RD- und WR-Signal kontrolliert wird, daher werden diese nur in Verbindung mit einem Chip-Selekt aktiviert. Das

daß die fallende Flanke des SEN-Signales gut geeignet ist, um das DTACK-Signal zu erzeugen.

1. Zwei Speicher-Bänke (mit je 16 Bit) werden zueinander phasenverschoben angesteuert, die Daten werden dann kur

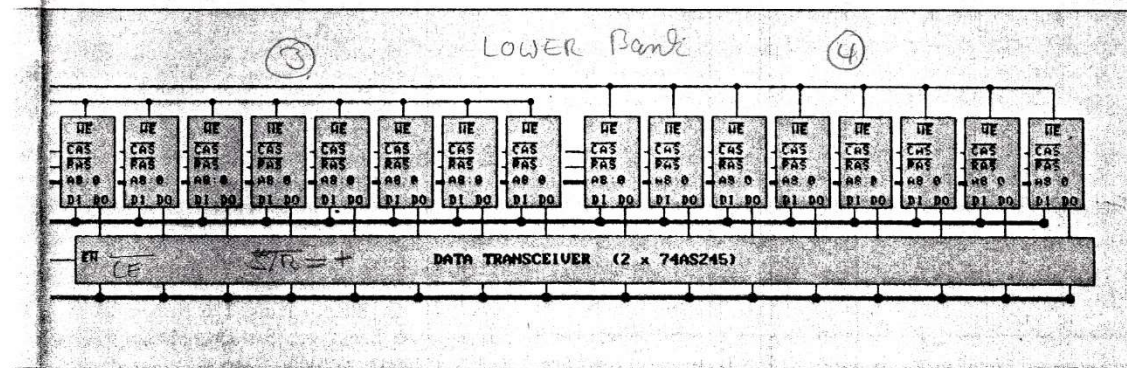


Bild 5: DRAMs ohne Output-Enable müssen über Tristat-Treiber geführt werden: Hier zwei Interleaved Bänke mit 256K x 1 Bit

März 1988

mc

8

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
a								Vcc+sv	GND	GND	NC	NC	NC	NC	NC	NC	NC	NC	NC	NC	NC	CAS0	BEND	DAS	D14	D13	D12	D11	D10	D9	D8	
b								Vcc+sv	GND	GND	NC	NC	NC	NC	NC	NC	NC	NC	NC	NC	NC	CAS1	BEM	D7	D6	D5	D4	D3	D2	D1	D0	

Sync-CN: Bottom View

Front View

View:

Connection

	2	4	6	8	10
Blank	Blank	Hsync	Vsync	Csync	Blank
GND	Hsync	Vsync	Csync	GND	
1	3	5	7	9	

GRUNDLAGEN

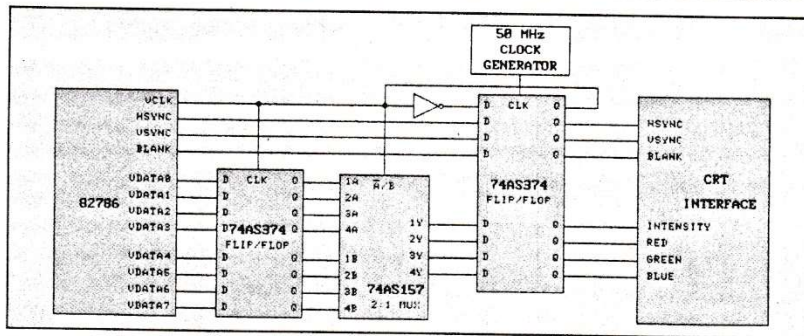


Bild 6: Die externe Taktverdopplung ermöglicht Punktfrequenzen bis 70 MHz

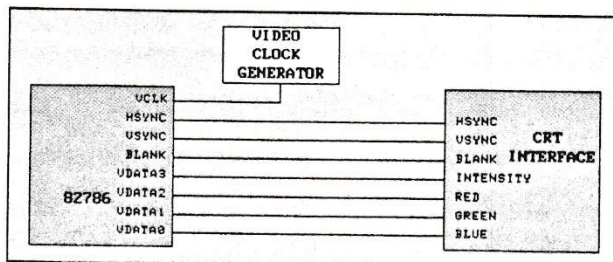


Bild 7: Einen Monitor mit TTL-Eingängen kann man direkt an den GDP 82786 anschließen ...

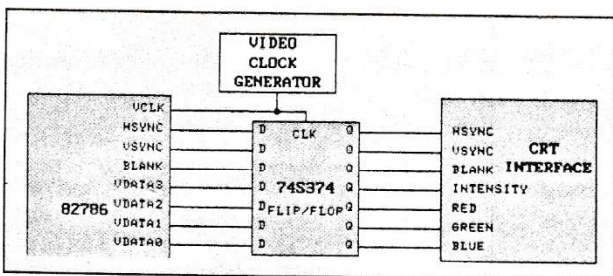


Bild 8: ... zum Schutz des teuren Grafikprozessors ist der Einsatz eines Treibers vor dem Monitor-Ausgang empfehlenswert

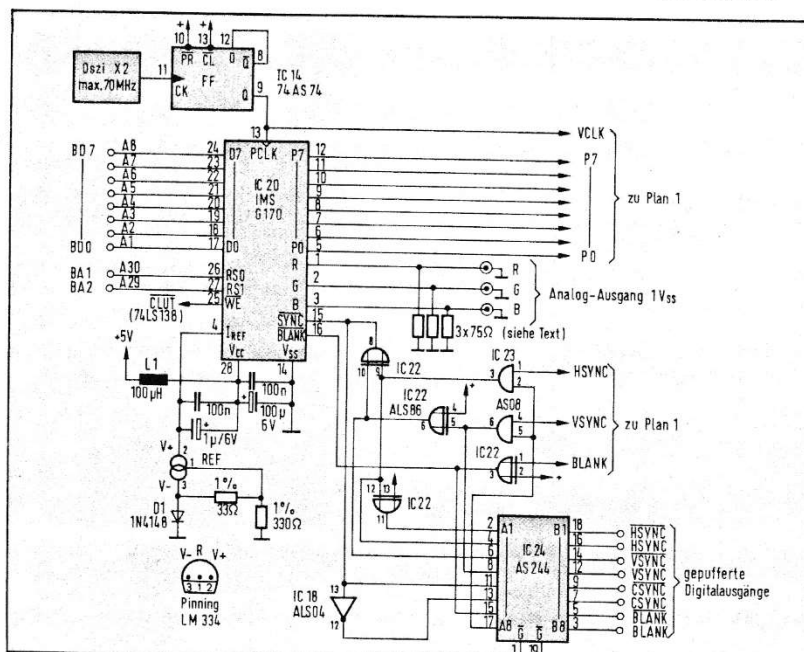


Bild 9: Viele Farben mit nur wenigen Bauteilen: Die Inmos-CLUT IMS G170 macht's möglich

nacheinander in die jeweilige Bank übertragen

2. Die Verwendung von Fast-Page-Modus-RAMs

Während der erste Punkt keine Mehrkosten erzwingt, da man meistens ohnehin relativ viel Bildspeicher verwenden möchte, gehen die Fast-Page-Modus-RAMs – oder alternativ Static-Column-RAMs – ins Geld.

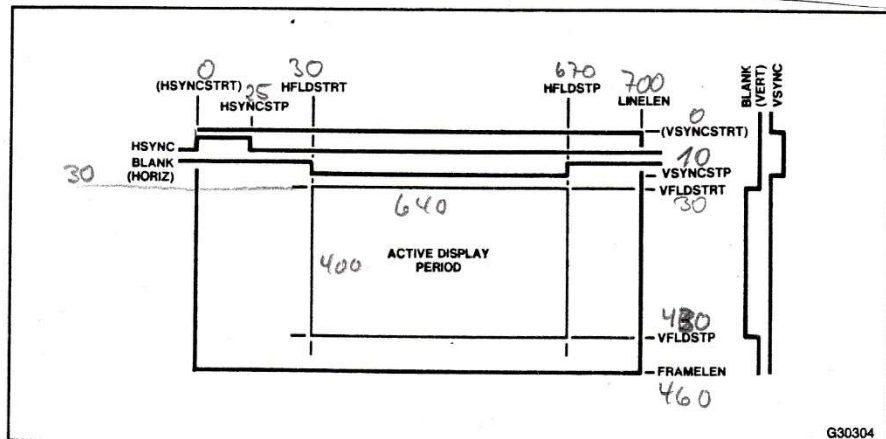
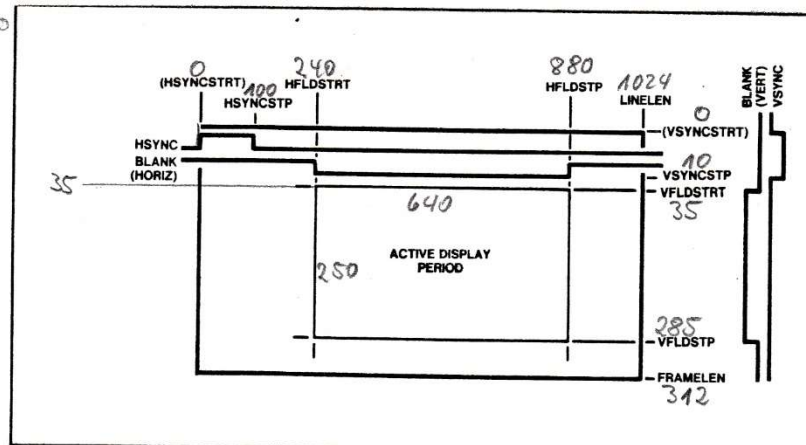
Man kann sich beim Preis des Speichers also nach der Anwendung richten: Verwendet man eine RAM-Bank mit 16 Bit, so erhält man einen Datendurchsatz von 10 MByte/Sekunde. Mit zwei phasenverschobenen Speicher-Bänken (interleaved) oder Fast-Page-Modus-RAMs ist der Datendurchsatz auf 20 MByte pro Sekunde begrenzt. Beim Einsatz von zwei phasenverschobenen Bänken von Fast-Page-Modus-RAMs liegt der maximale Durchsatz bei knapp 40 MByte pro Sekunde. Die Art der verwendeten RAM-Bausteine hängt also von folgenden Faktoren ab:

1. gewünschte Auflösung bzw. maximale Punktfrequenz
2. gewünschte Bittiefe bzw. Farbzahl pro Punkt.

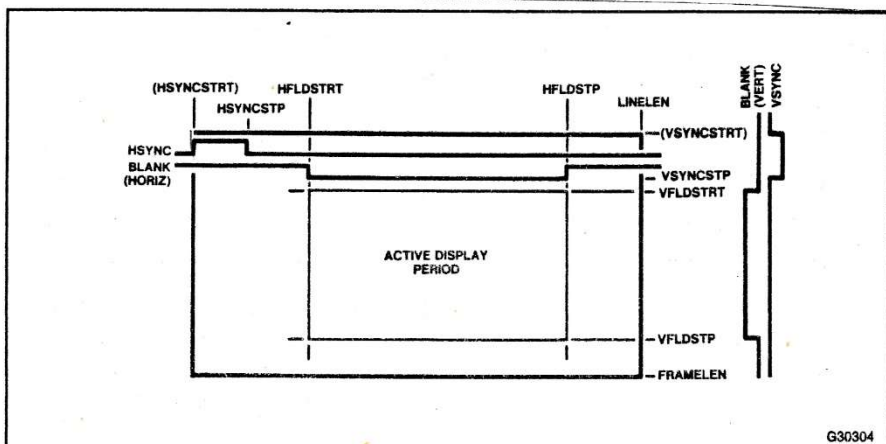
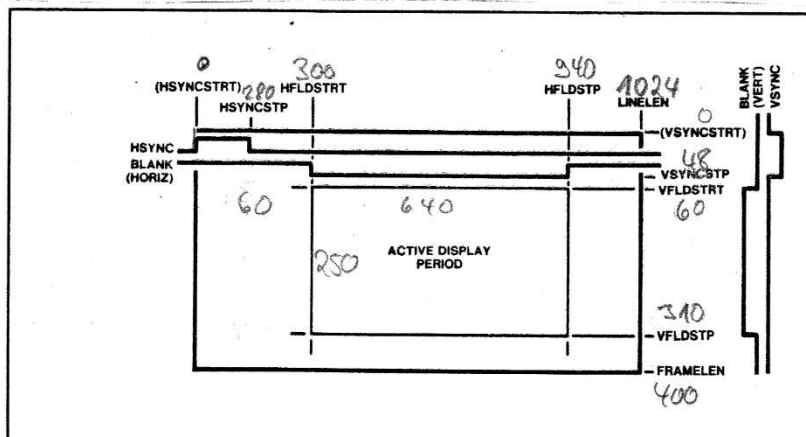
Dazu ein paar Beispiele: Für einen Standardbildschirm mit 640 x 300 Punkten Auflösung reicht eine Punktfrequenz von 16 MHz gut aus. Arbeitet man mit 8 Bit pro Punkt, so braucht man einen Datendurchsatz von 16 MByte pro Sekunde. Am billigsten ist daher die Verwendung von zwei phasenverschobenen Speicher-Bänken, wie sie Bild 1 zeigt. Braucht man nur 4 Bit pro Punkt, so kommt man auch mit einer einzelnen RAM-Bank aus.

Man muß bei der Bandbreite natürlich immer etwas Reserve lassen, da ja neben dem Display-Prozessor auch noch der Plotprozessor, der Window-Prozessor und die Haupt-CPU auf den Speicher zugreifen wollen. Läßt man zu wenig Luft, so verbleibt für diese nur die vertikale Rücklaufphase, da die horizontale für den Refresh der DRAMs verwendet wird. Das hätte merklliche Geschwindigkeitseinbußen zur Folge. Zurück zu den Beispielen: Für ein professionelles Textsystem soll mit einer Auflösung von 1024 x 800 Punkten bei 60 Hz Bildfrequenz (noninterlaced) gearbeitet werden. Dazu ist eine Punktfrequenz von fast 70 MHz notwendig. Arbeitet man mit einem Bit pro Punkt – also schwarz/weiß –, so reicht noch eine normale RAM-Bank aus. Wünscht man zwei Bit pro Punkt – also vier Graustufen bzw. Farben – so benötigt man schon zwei RAM-Bänke und beim Maximum von vier Bit pro Punkt (16 Farben) müssen zwei Banks von Fast-Page-Modus-RAMs eingesetzt werden.

2800
 Framelen 50Hz vert.
 640 → 16MHz 640dots
 Linelen 15.625Hz hor.
 8bpp
 250x640 Pixel
 2 Ram Banks interleaved



G30304



G30304

C0402E

GRUNDLAGEN

```

start
; setup IF0 vector-#
; get var-pointer
; Wichtig: Alles PC-relativ!
; setup global pointer

move.w #intvec,$fff+4
lea 37q(pc),a1
move.l #a1,intvec*4
lea 000 variables(pc),a6

lea entry(pc),a1
move.l a1-8(a0,d7.w),a1
lea ginit(pc),a2
move.l a2,(a1)+
lea move_abs(pc),a2
move.l a2,(a1)+
lea move_rel(pc),a2
move.l a2,(a1)+
lea line(pc),a2
move.l a2,(a1)+
lea circle(pc),a2
move.l a2,(a1)+
lea rectan(pc),a2
move.l a2,(a1)+

movea.l 4(a0,d7.w),a0
movea.l a0,a1
adda.l #gopoff,a1-000000
move.l a0,base(a6)
clr.w busy(a6) 00005

move.b #30,biuctr+1(a1)
move.b #0,biuctr(a1)
move.w #0,(a1)
move.w #1011,dramctr(a1)
move.w #18,refrctr(a1)
move.w #11110,disprio(a1)
move.w #100011,igrprio(a1)
move.w #101000,extprio(a1)

lea allregs(pc),a2
movea.l a0,a3
adda.l #ldall,a3
move.w #data/en/2-1,d0
move.w (a2)+,(a3)+
dbra d0,loop

move.w #idlow,dppara1(a1)
move.w #idup,dppara2(a1)
move.w #C_LDALL,dppocode(a1)

move.w #video,defvdata(a1)

movea.l a0,a2
adda.l #clutoff,a2
move.b #ff,5(a2)
moveq #0,d0
move.b #0,1(a2)

move.b d0,d1
and.b #3f,d1
move.b d1,3(a2)
move.b d1,3(a2)
move.b d1,3(a2)
addq.b #1,d0
bne loop

movea.l a0,a1
adda.l #framebuf,a1
move.w #38000/4-1,d0 224KByte
move.l #01010101,d1
move.l d1,(a1)+
dbra d0,cl0

clr.w d0
trap #2

; LINK-Befehl
; LOADALL-Befehl

; Karte in Slottabelle suchen
; falls nicht gefunden, dann
; Fehlermeldung ausgeben

; und abbrechen

; Das Programm verschiebt nun den später residenten Teil an das
; Ende der TPA und setzt die TPA-Grenzen herunter. Interrupt-
; routine und Graphikfunktionen bleiben somit ständig verfügbar.
; In der Slottabelle wird die Adresse der Einsprungtabelle für
; die Graphikfunktionen abgespeichert.

; get TPA limits
moveq #3f,d0 03ED0(A)
lea tpa6(pc),a2
move.l a2,d1
trap #2

lea start(pc),a4
movea.l 6(a2),a3 821(A)
suba.l a2,a3
adda.l a4,a3 7740
move.l a3,6(a2)

move.w #211,(a2)
moveq #3f,d0
trap #2

sub.l a4,d1 37748

movea.l a3,a2
move.b (a2)+,(a2)+
subq.l #1,d1
bne.s copy
jmp (a3)

; A3 is new start address
; transfer text to new
; address
; now jump to new address

; A3 is new start address
; transfer text to new
; address
; now jump to new address

; Der letzte JMP-Befehl läuft bei 'start' weiter, allerdings
; bereits unter der neuen Adresse des Programmes am Ende der TPA.

```

```

lea    (a0)
lea    variables(pc),a1
lea    (a1)
movea.l    base(a1),a0
adda.l    %gdppoff,a0
tst.w    gpstat(a0)
tst.w    biuctr(a0)
clr.w    busy(a1)
movea.l    (a7)+,a1
movea.l    (a7)+,a0
rte

```

- * Plotfunktionen
- * Es folgen nun einige Plotfunktionen. Sie können von Assembler oder C aus angesprochen werden. Dazu werden zunächst alle Befehle im Graphikspeicher gefolgt von einem HALT-Befehl angelegt. Beim Aufruf des jeweiligen Befehls werden dann lediglich die Parameter im GSP-Speicher eingetragen und der Graphikprozessor wird gestartet.
- * Es werden die Adressregister A0 und A1 sowie das Datenregister D0 benutzt. Diese sind unter C Scratchregister und müssen nicht benutzt werden.

```

ginit
gini
lea variables(pc),a0
tst.w busy(a0)
bne.s gini
st busy(a0),a1
moves.l base(a0),a1
adda.l #combuf,a1
lea gintab(pc),a0
#gintab/2-1,d0
move.w #gintab/2-1,d0
move.w (a0)+,(a1)+
d0,gini0
dbr a
lea variables(pc),a0
moves.l base(a0),a0
adda.l #gdproff,a0
move.w #comlow,glinklo(a0)
move.w #comup,glinkup(a0)
move.w #C_LINK,gprocode(a0)
move.w rts

```

```

move_abs    lea     variables(pc),a0
              base(a0),ai
              #combuf,ai
              #amovop-gintab,a1
              busy(a0)
              mai
              busy(a0)
              base(a0),a0
              #gdgpoft,a0
              adda.l 4(a7),2(ai)
              move.w 4(a7),4(ai)
              #cmilow-amovop-gintab,gplink(a0)
              #comup,gplinkup(a0)
              #c_lnk_rpopcode(a0)
              move.w #c_lnk_rpopcode(a0)

```

```

move_rel
lea
movesl
adda.1
adda.1
adda.1
tst.w
bne.s
mr1
st
movesl
adda.1
variables(pc),a0
base(a0),a1
#combuf,a1
#movop-gintab,a1
busy(a0)
mr1
busy(a0)
base(a0),a0
#gqpcoff,a0

```

```

line
lea    variables(pc,%a0)
movea.l %a0,%a1
#combuf,%a1
adda.l #lineop-gintab,%a1
tst.w  %a0
bne.s  ll
st      %a0
movea.l %a0,%a0
adda.l #gdpoff,%a0
move.w  4(a7),2(a1)
move.w  6(a7),4(a1)
move.w  %a0+lineop-gintab,gplinkto(%a0)
move.w  %a0+gplinkup(%a0)
#C_LINK,gpopcode(%a0)
rts

```

```

circle
lea    variables(pc),a0
movea.l base(a0),a1
adda.l #combuf,a1
adda.l #circop-gintab,a1
tst.w  busy(a0)
bne.s  c1
st      busy(a0)
movea.l base(a0),a0
adda.l #gdboff,a0
adda.l 4(a7),2(a1)
move.w #comlow+circop-gintab,gplinklo(a0)
move.w #comup+gplinkup(a0)
move.w #C_LINK,gpopcode(a0)
rts

```

```

rectan
lea    variables(pc),a0
movea.l base(a0),a1
movea.l #combuf,a1
adda.l  #rectop-gintab,a1
tst.w   busy(a0)
bne.s   rect1
st       rect1
movea.l base(a0),a0
adda.l  #gdpoff,a0
movea.w 4(a7),2(a1)
movea.w 6(a7),4(a1)
movew   #comlow+rectop-gintab,gplinklo(a0)
movew   #comup,gplinkup(a0)
movew   #C_LINK,gpopcode(a0)
rts

```

```

variables    ds.1 1 ← 000000
             ds.w 1 ← busy-Flag
entry        ds.1 6
             : 6 Funktionseinsparungen
             : Speicherplatz für Variablen
             : Offsets für Variablen

```

```

*****
*   Daten Definitionen
*
*   Es folgen Video-Parameter für folgende Konfiguration:
*   Pixeltakt: 16 MHz, 2 interleaved RAM-bank's
*   Auflösung: 640 x 250 Punkte, 8 Bit pro Punkt
*   Abtastfrequenzen: horizontal 15.625 Hz, vertikal: 50 Hz
*****

```

HSync 25kHz
VSync 78Hz



```

*****
*
*   Einfacher Treiber für den Intel-Graphiprozessor GDP 82786
*   Erstellt am 6.12.1987 von H. Völzke
*
*****

* System Definitionen

slottab      equ      $fff800          ; Basisadresse der Slottabelle
kennung      equ      $07028000       ; Kennung der Graphikkarte
intvec       equ      $80              ; Interruptvektor

*****
* variable Definitionen

dvideo       equ      0                ; Default Video Farbe

*****
* Speicher Definitionen für 256 KByte Video-RAM
* Der Graphikspeicher von 256 KByte wird wie
* folgt aufgeteilt:
* $00000 ... $03FFF: 16 KByte Puffer für Displayprozessor
* $04000 ... $07FFF: 16 KByte Puffer für Graphikprozessor
* $08000 ... $3FFFF: 224 KByte Bildspeicher

ldall        equ      $00000           ; Speicher für GDP Parameter
ldlow        equ      $0000           ;
ldup         equ      $0              ; aufgeteilt in Worte

combuf       equ      $04000           ; Speicher für Plotkommandos
comlow       equ      $4000           ;
comup        equ      $0              ;

framebuf     equ      $08000           ; Bildspeicher
framelow     equ      $8000           ;
frameup      equ      $0              ;

gdpoff       equ      $100000          ; Offset für GDP I/O-Bereich
clutoff      equ      $100100          ; Offset für CLUT

*****
* GDP Definitionen, Offsets für interne Register

intrel       equ      0
biuctr       equ      4
refrctr      equ      6

```

Bild 1: Das Grundgerüst eines Treibers unter CP/M-68k – leicht zu erweitern und als Beispiel sicher aussagekräftiger als viele Worte

mc

April 1988