National® Brand

COMP BOOK

Richard Miller 1994

MPEG-1. Atari Corp

80 SHEETS • 10 x 7⅞ • 5 x 5 QUAD • 43-475

0 73333 43475 0

DENNISON STATIONERY PRODUCTS CO.
FRAMINGHAM, MA 01701-0344
Made in USA

On TOM read cycles, read data may be latched up to
38ns after WAIT ↑. will CC450 hold data that long? NO.

STATE

GC

SEL45
=TOM1, 1:

$\overline{AS}$, VI

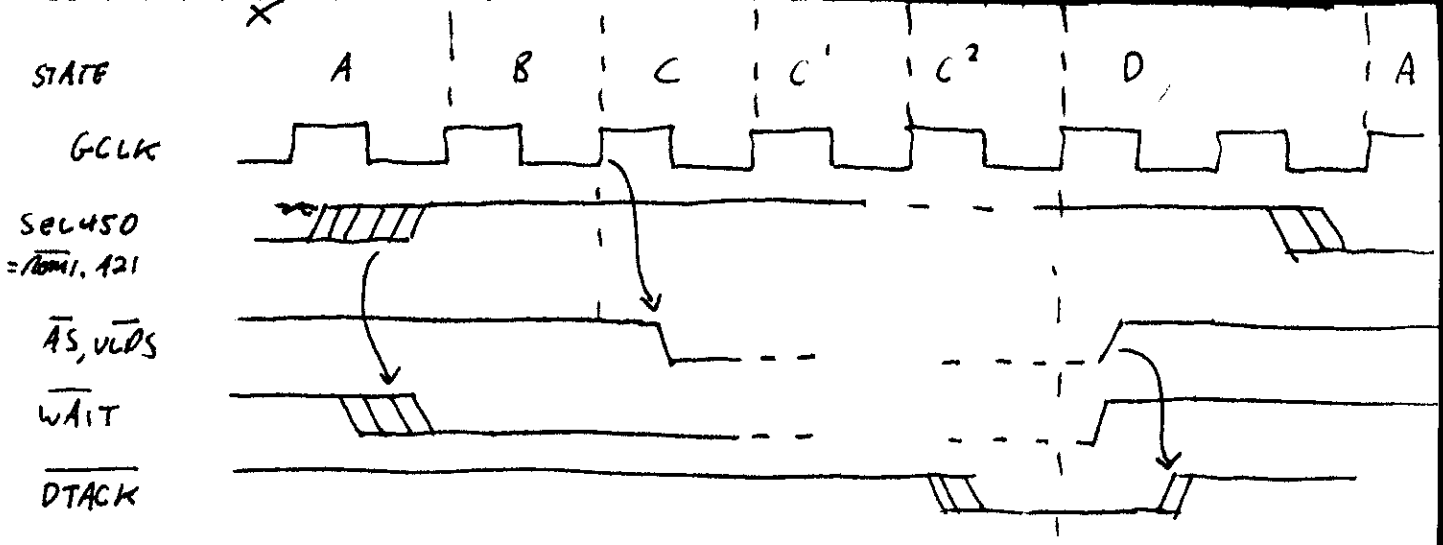
$\overline{WAIT}$

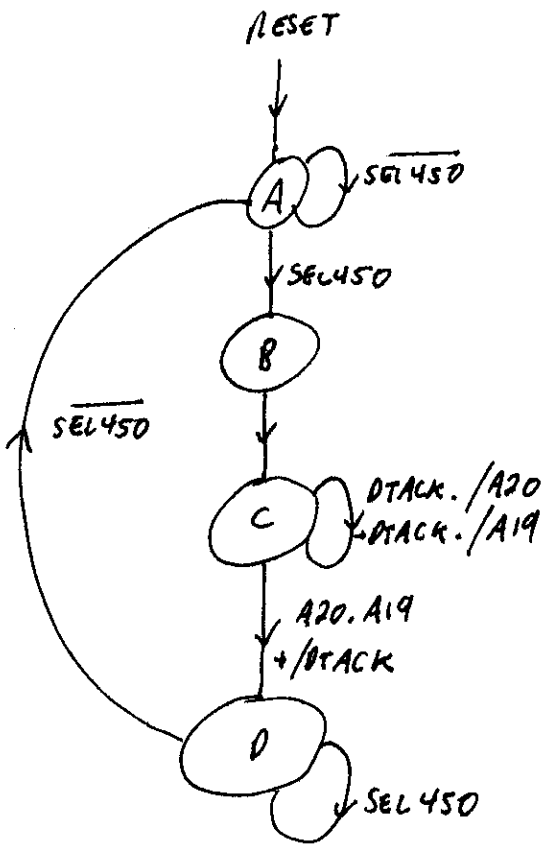
$\overline{DTACK}$

/W

/AS

XΦ





	AS	xφ	x1
A	1	0	0
B	1	0	1
C	0	1	1
D	1	1	0

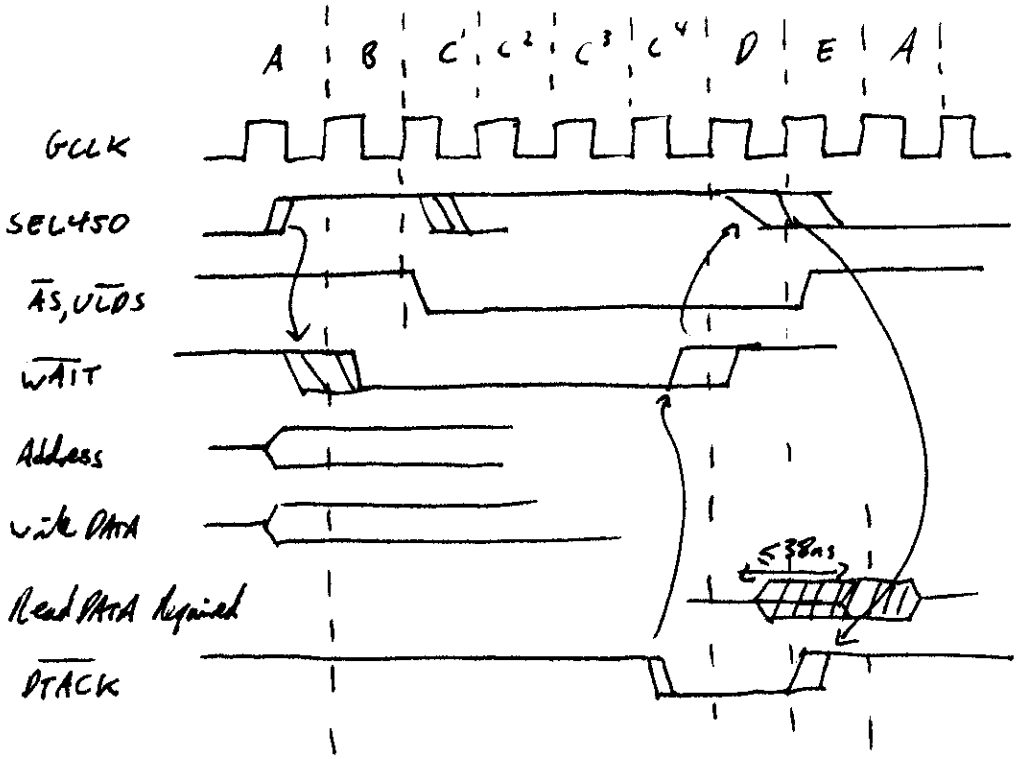
$$/WAIT = SEL450 . \overline{CMEM} . \overline{D}$$
$$= SEL450 . (/A20 + /A19) . (/x\phi + x1)$$

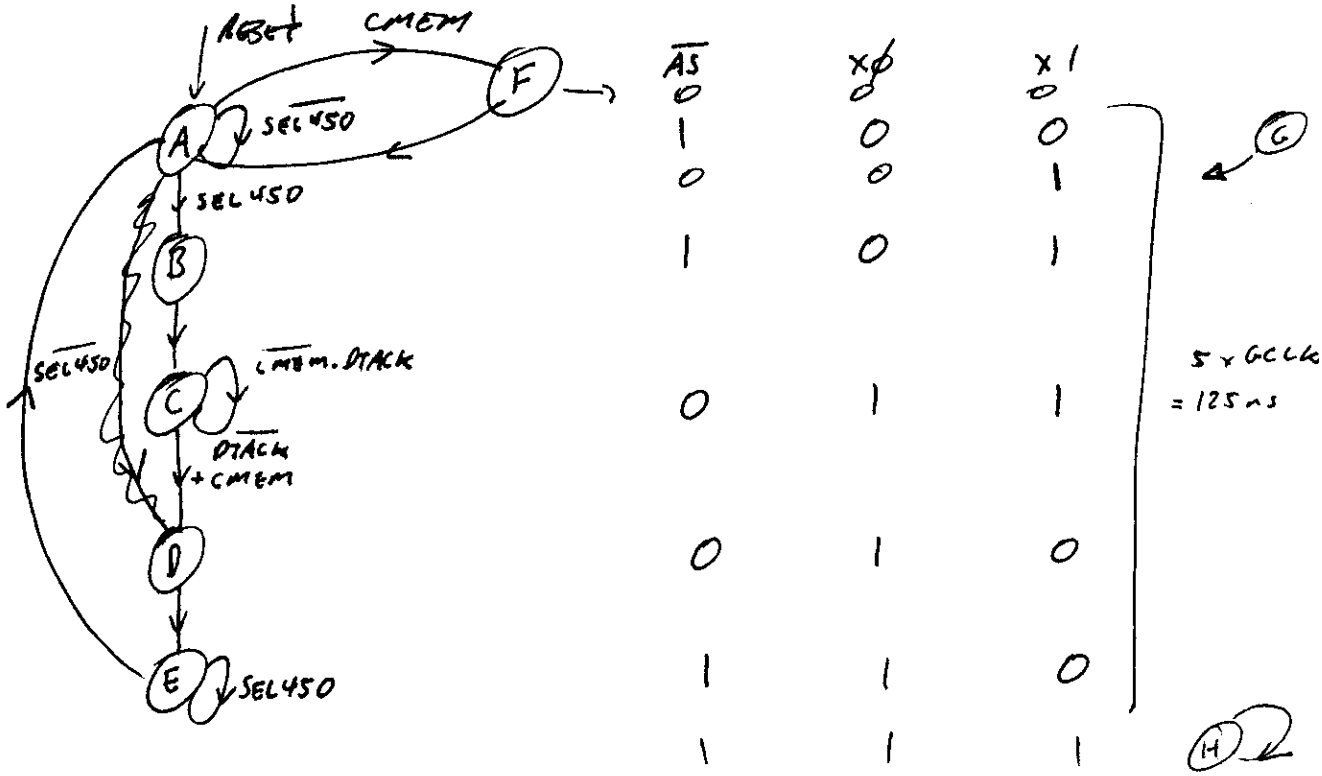
$$/AS = x\phi . x1$$

$$x\phi = B. + C.DTACK./A20 + C.DTACK./A19 + C.A20.A19 + C./DTACK + D.SEL450$$

$$+1 = A.SEL450 + B + C.DTACK./A20 + C.DTACK./A19$$

= U4D





/WAIT = SEL450. $\overline{CMEM}$. DTACK. $\overline{E}$

$X\phi :=$ RESET . B /

+ ~ . C . $\overline{CMEM}$. $\overline{DTACK}$ /

+ ~ . C . CMEM /

+ ~ . C . $\overline{DTACK}$ /

+ ~ . D /

+ ~ . E . SEL450 /

] = RESET . C

$X1 :=$ RESET . A . SEL450 /

+ . B /

+ . C . $\overline{CMEM}$. $\overline{DTACK}$ /

$AS :=$ $\overline{RESET}$ /

+ E . $\overline{SEL450}$

+ A . $\overline{SEL450}$

+ A . SEL450

+ D

+ E . SEL450

] =

+ G /

+ A /

+ D /

Back data to CC450

1. write CC450 DRAM test loop. ✓
2. write CC450 CMEM → DRAM test loop. ✓
3. Put V data into DRAM - check integrity of passes.

$140000 → $1F0000

= TOP_VID.BIN
Compare with TOPGUN.BIN = Raw file

write </n> x,1

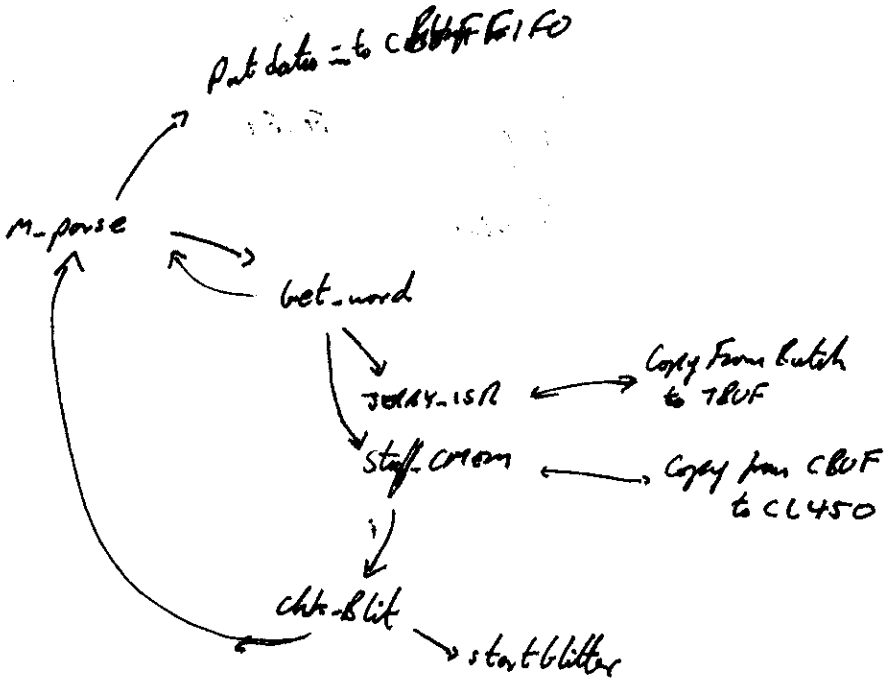
CC450 CMEM in DRAM is at offset $2880 + A0 0000 NO!

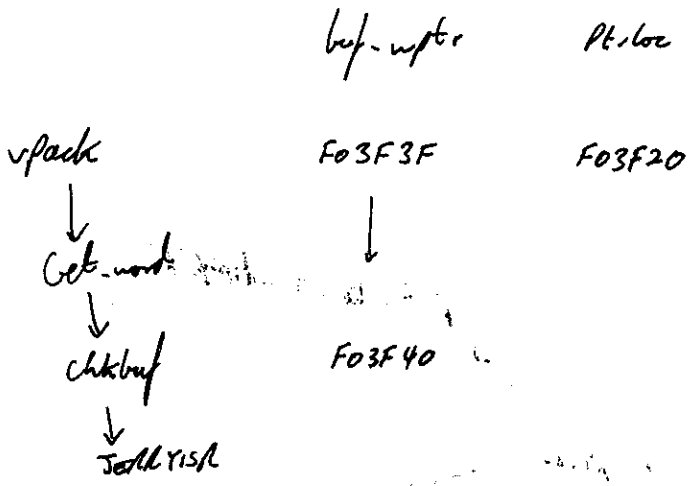
Try $5100 46k.

Raw CD track length = 2352 Bytes		Form1	Form2
+	12 Sync	12 Sync	12
	4 Header	4 Hdr	4
	8 Sub header	8 Subhdr	8
	2328 Data	2048 Data	2324 Data
	2352	4 EDC	4 Reserved
		276 ECC	
		2352	2352

10/20/94

1. Ignoring CMOM memories, get 30Hz frame rate running.




buf_wptr
Ptrloc
vpack
F03F3F
F03F20
Get_word
F03F40
chkbuf
JobLoop

~~Bandwidth:~~

Bandwidth again:

= 5880.

In 1/30 second, need to: - Read $\frac{176400}{30}$ Bytes from Butch

- Write up to $\frac{176400}{30}$ Bytes to CL450 cmem or Audio buffer

- display 860 frames of 240 lines

- Blit 240 lines from CL450 to VRAM.

33.3 ms.

↳

JUNT is 180 µs. in 1/30 second we get 185.2 interrupts.

→ for 268 line blits = 1.44 blits each 180 µs.

Every 180 µs, need 3 OP → 60 µs

~~2 Blits → 72 µs~~ 384 x 5 blits = 1930 → 72 µs

1 FIFO read → 12 µs

~~144 Bits~~

2 cmem writes 24

96 µs → having 84 µs for blits = 1.16 blits

We are usually seeing 50 ms/frame → 20 Hz

90 → 140 µs per line.

↳ x 268 lines = 37 ms

$\frac{50\text{ ms}}{268}$ = 186 µs per line on average

t1
t3

CD data

t0 = counter

spare
t2
t4
t5

SC-DBASE

SC-PERM-BUF = load 128 bytes in at start. | = $F03d80

25,420,1364

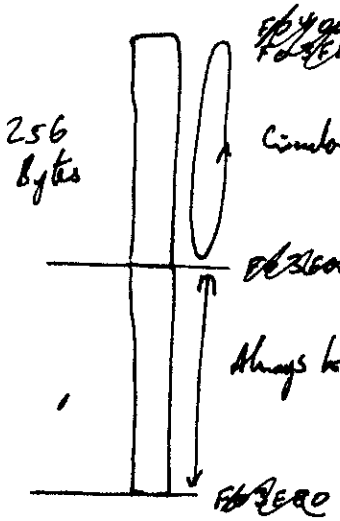
SKIPS Mark

Scra

256 bytes

116530

Scramble table:



F03F00
F03EFF

1110
F03E00
F03DFF

F03D80
1101

<s_head>
s_tail

Differs at $7C = 124 ×1
$98C → $944 = 2444 ×3] Longs
116530 12A4 = 4772

2328bytes 2328

$74	3 Longs	98C	12A4
78		990	12A8
7C		994	12AC

3rd byte ok!

1BBC	3 Longs
1BC4	
1BC4	

352 pixels = 176 Phase reads
+3 bits = 528 bits
÷26.4 = 19.8 μs.

In 1/25 sec, ÷180 μs → 222 bits.

In 1/30 second, 185 J-INT interrupts.

Also, OP needs to display 240 active lines, 20μs each.

OP

352 active pixels/line = overscanned → NTSC is 52.4μs.

8.24 $\frac{52.4\mu s}{352}$ = 148.9ns = 6.72MHz.

26.6MHz ÷ 4 = 6.65MHz = 150ns → 52.9μs/line.

RGB24, ÷ 4 clk → VMODE = 0110 0100 0011 = $643

240 active lines →
263 total lines → 23 lines of blanking per field.
= 7 × 180μs slots

On active lines:

Say 18μs

OP = 2μs × 3	60μs	
Both FIFOs SC	12μs	12
Cmem stuffing (16 words)	16μs	16
	88μs	28μ → 152μs

→ 92μs left.
= 1.27 × 72μs.

Say 1 blit per 180μs on active lines
2 blits " " on VBlank slots (14 total VBlank slots → 28 lines this way).

Leaving 240 − 28 lines to blit = 212

240 active lines ÷ 3 = 80 slots × 2 fields = 160 Line blits

In 160 × 180μs slots, need to blit 212 lines → 1.325 Blits/slot

To do

✓ 1. Put OP ISR into GPO

✓ 2. Only blit 240 lines worth

$100 = 256 lines = 50 ms = 20Hz
$F0 = 240 lines = 47 ms = 21.27Hz
$C8 = 200 lines = Not work!

3. Optimise CMOM writes
Batch + CS reads
Hsync/Vsync
Blits
] in s/w
] H/w too

✓

✓ 4. Do CMOM writes during blits

✓ 5. Separate Batch reads from CS, + optimise
~~Do CS reads during blits~~

6. Fix intricate vertical jitter

7. Center screen (again)

8. Store latest scanline count in buffer.

9. In Shaf comms, store to DRAM buffer.

MPEG Issues + ruines

- sc-tail initialisation. Is it OK to read dummy stuff?

10/27/04:

$\overline{AS}$ 1.2 µs

HSync 88 µs
→ 127 µs ← normally.

Vsync ~~88~~ 127 µs 48 ms

10/25

Performan

① On scope

② Hsy

VCC

③ $\overline{AS}$ b

with

my

a) BL

b) sa

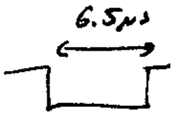
④ Readi

(For

10/25

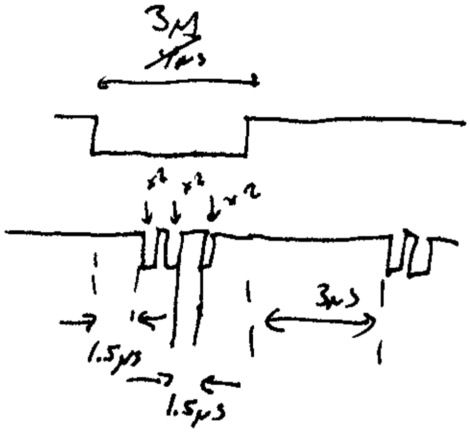
Performance tweaks:

(1) On scope: Hsync



Need 6 × VCLKs → 3+3 loadw
= 3 × 6 ticks = 18
+ 3 × 3 ticks (RAM) = 9
27 → 1µs!

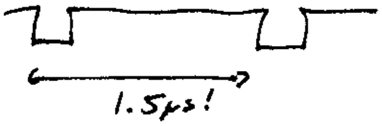
(2)



48ms VSYNCs.
256 line blits
→ 187.5µs for Hsync

→ OP 60
Blit 67µs
+ 6µs
<133µs>
+ 187
54µs
for BUTCH, CMEM

(3) $\overline{AS}$ GCC450



CMEM writes

With storew
loadw
rorq
storew
loadw
(1.5µs)

rorq no → No effect. → 15µs for the 24 bytes!

a) BLIT would take 8 read
6 write
10 ticks/word × 12 = 120 ticks = 4.5µs
+ about 3µs setup time. = 7.5µs.

b) Send 2 words at a time → 1.5µs for 2 words
- Does not work.

(4) Reading Butch + Scramble values - 10/27 = 12µs.

(For 8 reads from Butch, + 8 reads from RAM. = 16 read cycles → 750 ns each!

Bugs history

CC for branch object wrong
VC==YPOS not be even valued

(STOP subroutine does not work!) ??

Perf History

		BUTCH		CMEM	
	$\overline{AS}$	↓ H_{sync}		↓ V_{sync}	
10/27	1.2μs	88-127μs		48ms	Using GAU for OLP Looping 68k in RIVU (STOP bit not set) 240+6+9 lines Blitted
"		12μs			" , 240 Blits 6+9 borders
"	720ns	12μs 87-127		46.36ms	68k = STOP
			10.8μs	→	for 12 words
	750ns	88-126		45ms	68k = STOP 240 Blits × 360 6+9 " × $10
			10.8μs		

	101
	6
	F7

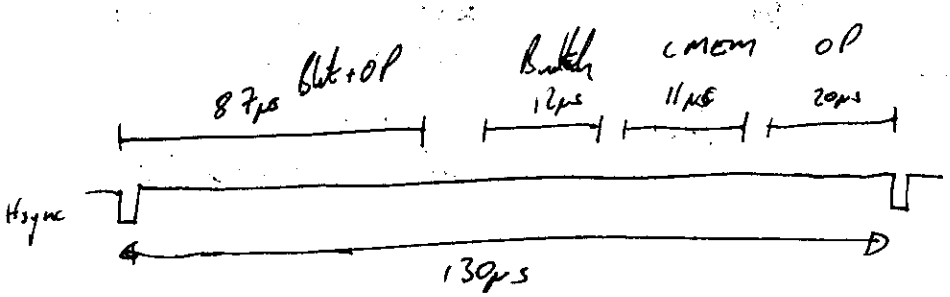
10/28 — 41ms — 101, 6, F7 CMEM writes during Blits

256 lines in 1/30 s → 130 μs/line 1.38 lines/180 μs 1.38 × 734 = 101

= 1/25 s → 156 μs/line 1.15 lines/180 μs 1.15 × 734 = 84.41

240 lines in 1/30 → 139 μs/line, 1.29 lines/180 μs 1.29 × 734 = 95 μs

1.29 × 70 = 90.3



101
6
F7

÷28.7

		<180>
Blit - 360 pixels, 5 bits each → 1800 bits	67.4 µs	112.6
Hsync + overhead	6 µs	106.6
CMEM - 32 bytes	22.5	~~[illegible]~~ 157.5
Butch sto	12	~~[illegible]~~ 145.5
OP	60	85.5

		<180>
OP	60	120
Butch +OH	12	108
CMEM	11	97

Blt setup

BLMDN~~[illegible]~~

BCOUNT bcnt ✓

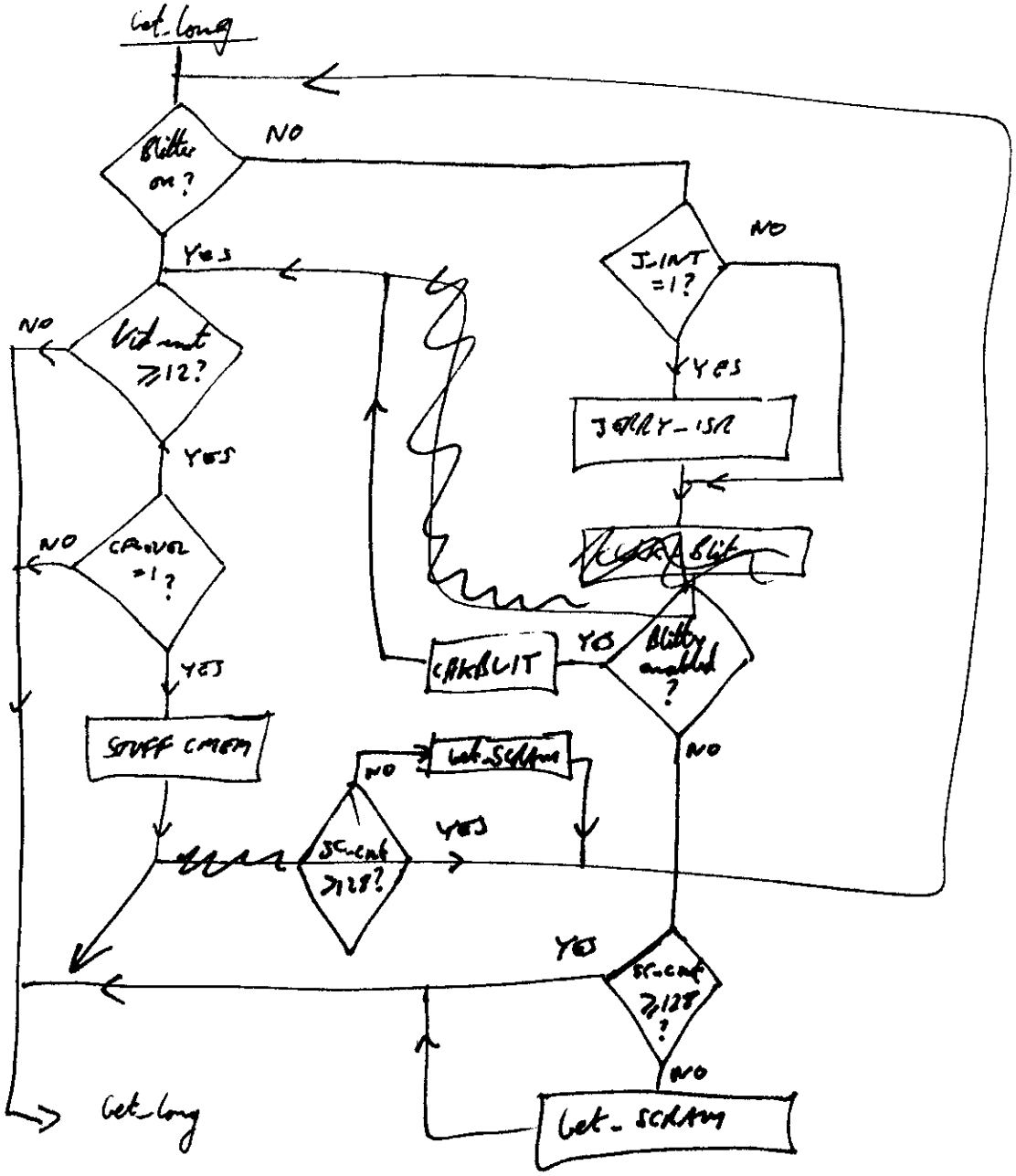
D~~[illegible]~~ADD

DLINCE dinc

M_PIX

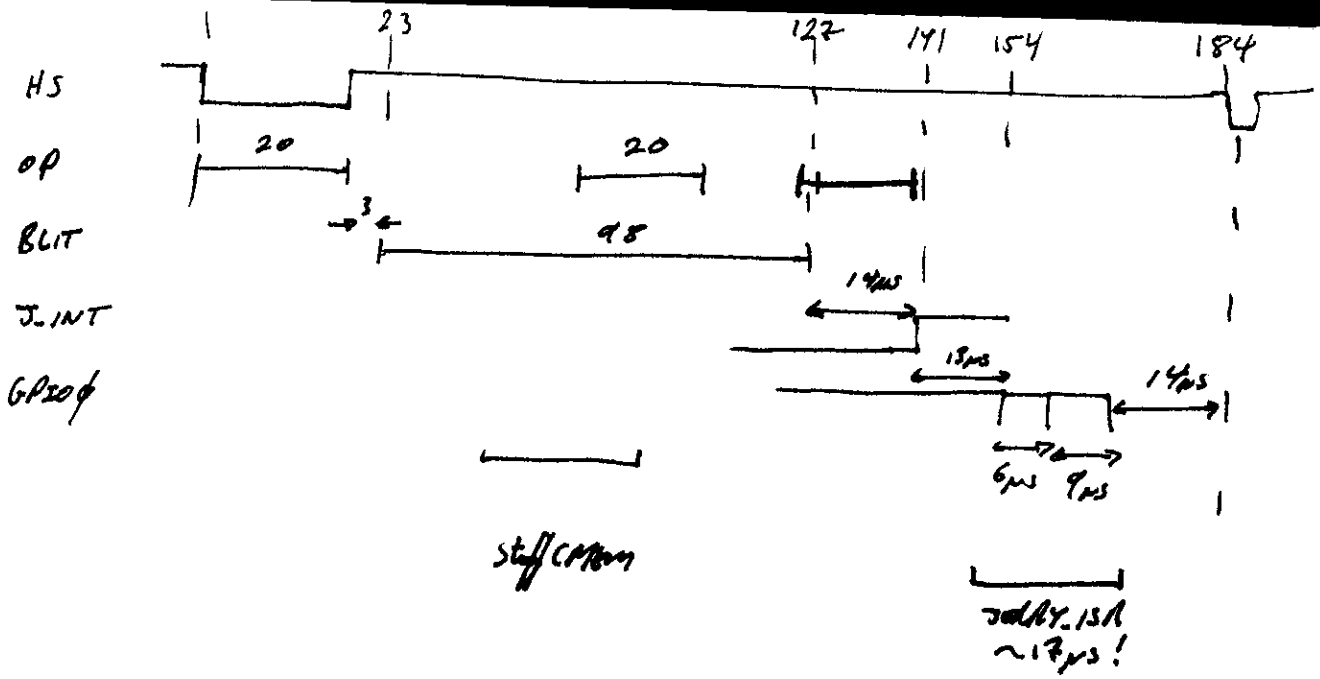
M_HS0
M_HS1

Memcon 1
monc 32
16
F32



extend GPU code load instructions which may pause for OP:

load (J-INT) 1 in 2 lines
load Blitter FIFO 1 in 3 lines



Chk_BLIT

48µs

OP -	60	
BLIT - +Comm	72	
Dead	13	—
J_ISR	17	—
Dead	14	—
	176	

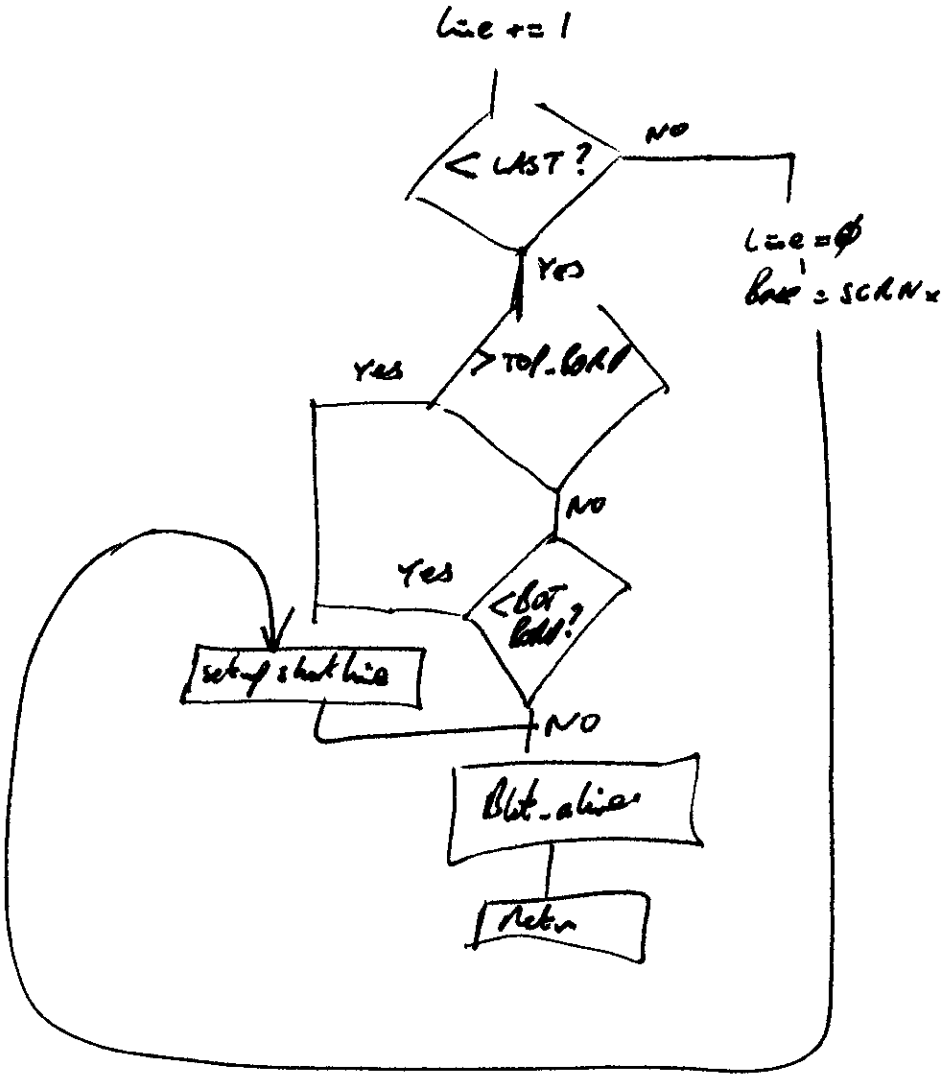
Perf Target is: 1 in 180μs approx (180)

Object processor	3×20 =	60 μs
Jerry ISR (both FIFOs)		8 μs
Steff CMOM (during blits)		4 μs
		72μs → 108μs

Blits are 68μs +6μs not. μ=74.
In 180μs, need 1.29 lines = 96μs

96
168 → 12μs spare.

CRT_BLIT


Line += 1
< LAST ?
NO
Yes
Line = 0
Row = SCRN x
Yes
> TOP_ROW
NO
Yes
< BOT ROW ?
setup shortline
NO
Blit_aline
return

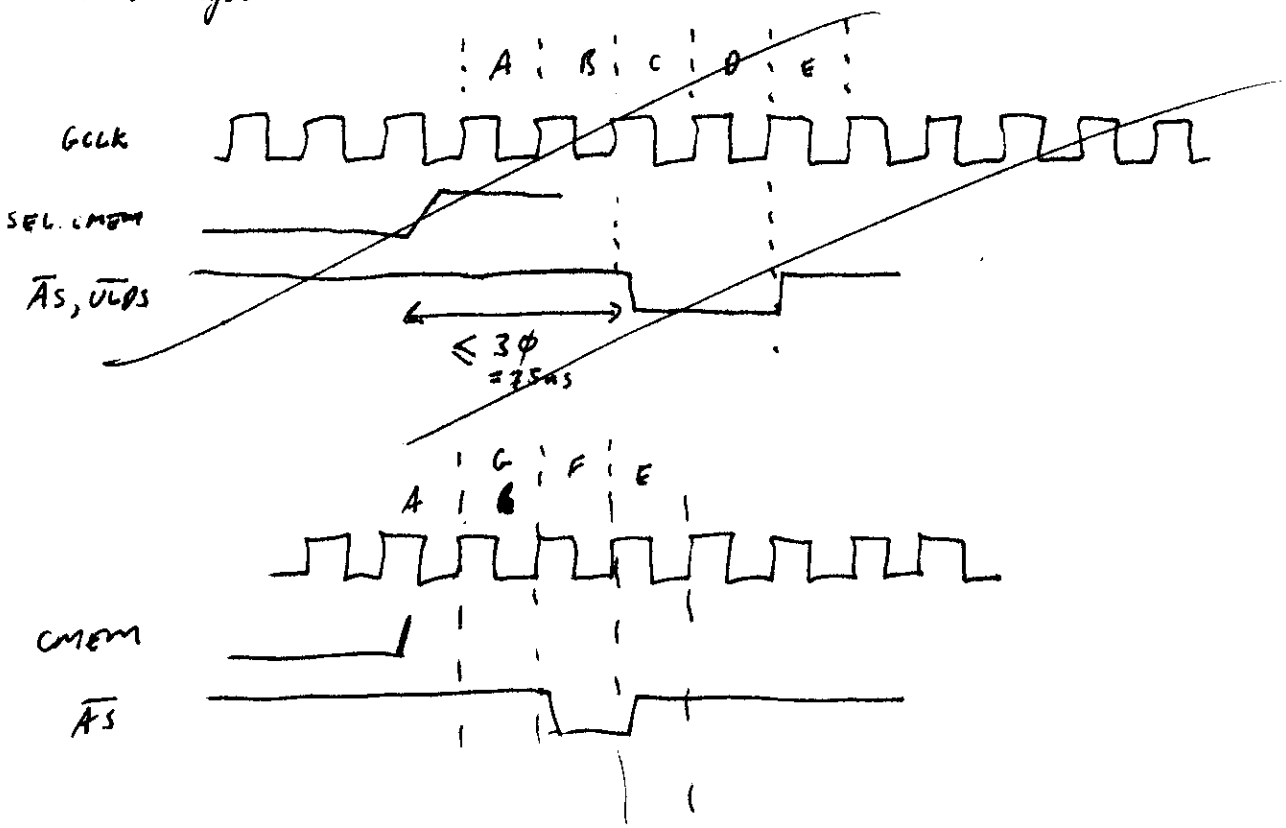
Byte 18 AND $0E ≠0 → Good sector

CDIEX 6

V		Volume descriptor	A	ASCII	
P		Path Table	B	Hex dump	
D	xx	Directory sector	F	Copy filetype	=Submode
S		Search for largest	S	Search largest	
R	xx	Show every sector	G	Play movie	
C	xx	Copy start sector			
		Size			
		Blocksize	(2324)		
		Rest file			

cdextrac filename track number → CDVideo

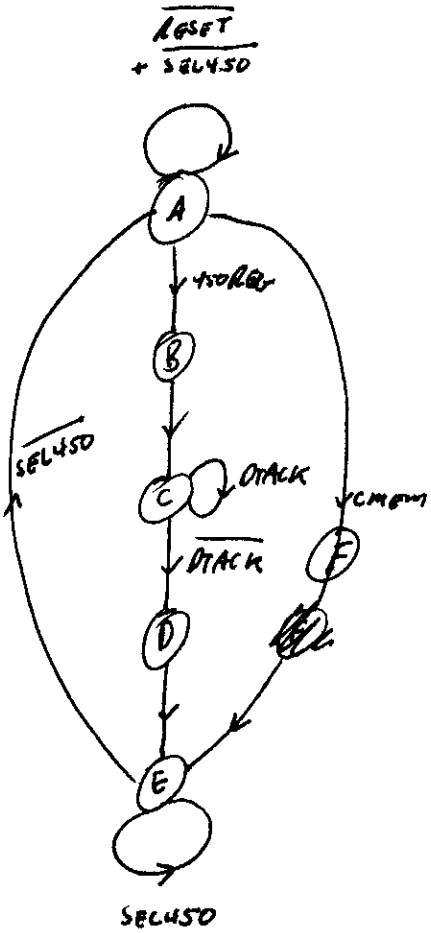
CMEM cycles


A
B
C
D
E
GCLK
SEL. CMEM
$\overline{AS}$, $\overline{ULDS}$
$\leq 3\phi$
=75ns

G
F
E
A
CMEM
$\overline{AS}$

U4F

11/11/94



	$\overline{AS}$	$X\phi$	$\bullet X1$
A	1	ϕ	ϕ
B	1	ϕ	1
C	ϕ	ϕ	1
[F	ϕ	ϕ	ϕ]
D	ϕ	1	1
E	1	1	0

$$AS := \overline{RESET} + A.\overline{SEL450} + E.\overline{SEL450} + E.SEL450 = E + D + F + A.REG$$

$$X\phi := C.\overline{DTACK} + D + E.SEL450 + F$$

$$X1 := A.REG + B + C.DTACK + C.\overline{DTACK} = C$$

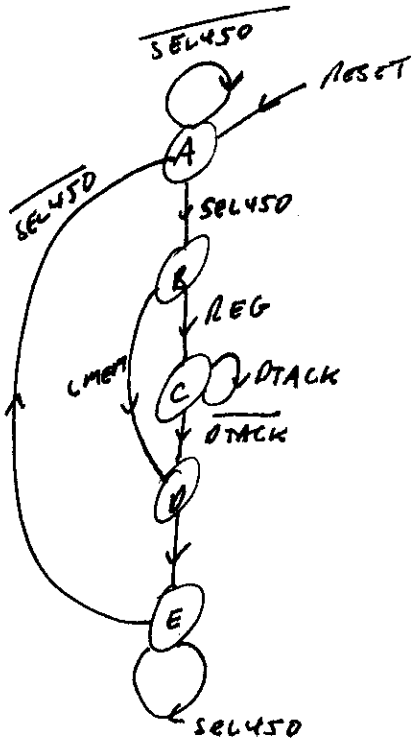
$$SEL450 = REG + CMEM = \overline{ROM1}.A21$$

$$REG = \overline{ROM1}.A21.\overline{A20} + \overline{ROM1}.A21.\overline{A19}$$

$$CMEM = \overline{ROM1}.A21.A20.A19$$

$$\overline{SEL450} = ROM1 + \overline{A21}$$

U4H.PAL



$\overline{AS}$	$X\phi$	$X1$
1	ϕ	ϕ
1	ϕ	1
ϕ	ϕ	1
ϕ	1	1
1	1	ϕ

→ U43.PAL

U44.PAL

SEL450 = $\overline{RMI} \cdot A21 + \overline{GP50\phi} \cdot \overline{WE}$

CMEM = $\overline{GP50\phi} \cdot \overline{WE\phi}$

$AS := \overline{RESET}$ ✓
$+ A \cdot \overline{SEL450}$
$+ E \cdot \overline{SEL450}$ ⟩ = A ✓
$+ A \cdot SEL450$
$+ D$
$+ E \cdot SEL450$ ⟩ = E

$X\phi := C \cdot \overline{DTACK}$ ✓
$+ B \cdot CMEM$
$+ D$ ✓
$+ E \cdot SEL450$ ✓

$X1 := A \cdot SEL450$
$+ B \cdot REG$
$+ C \cdot DTACK$ ⟩ = C ⟩ = B
$+ C \cdot \overline{DTACK}$ ⟩
$+ B \cdot CMEM$ ⟩

11-15-98

V0 to do

1. Tweek code to get back to 30Hz - Number of lines ✓
2. Use GPIO for comm, during line blits again - to get 30Hz
3. ✓ Double buffer screen
4. ✓ set Sync divide rate according to encoded rate
5. ✓ Change error handling, to restart 68k - interrupt
6. Handle - bitstream errors
   - comm overrun
7. Do audio parsing, + send audio data to Jerry
8. Read joystick at frame rate - PLAY
   STOP
   PAUSE
   FF
   REW
   SLOW MOTION
   FRAME STEP
9. Parse sector headers, + store (not Lisword)
10. White book CD handling
11. Parse CD type - ie eject Red book ..... + non MPEG discs.

H/w

1. New schematic
2. PCB layout
3. Environmental testing

1. Get Comm during blits working with Jerry Syncs
2. Test 32K of comm data

11-15-94 Conversation with Glen Ivey.

DEC_TOGGLE. Intended for VGA systems etc. Don't use

Errors. Maybe look at error interrupt

Full screen green. Probably due to HSYNC/VSYNC.

Need 5.5 μs HSYNC low time

9 lines bottom border will go away. Real soon now! Maybe by 11/23

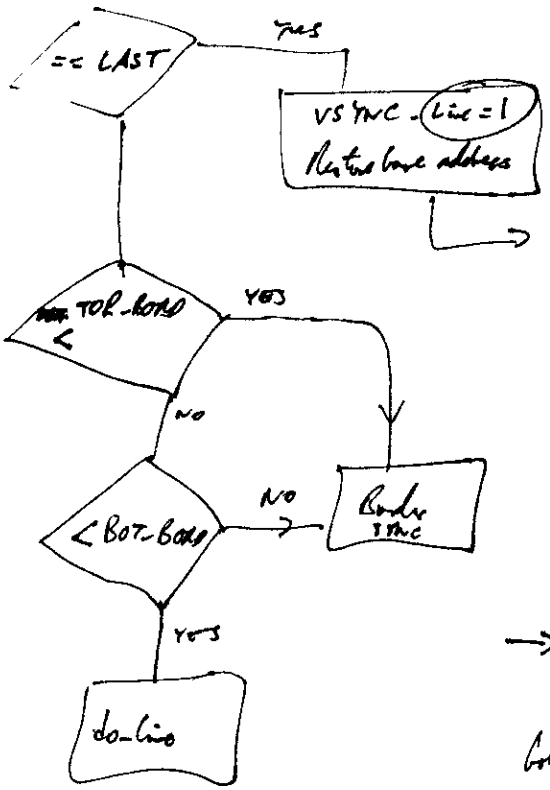
(v2.09)

2.06 is now 2.08

11-21-94 Keith - Customers

1. HSYNC < 2 or > 20 μs from VSYNC. ensure this is true
2. Keith will check VSYNC ⊓⊔ min pulse width
3. " - also check customer mode timing, + send latest timing parameters
4. - Try a s/w customer mode test, using Flashpoint stream. -
   - Check PcBlayout for xtalk
   - Check VOE contention possibilities.

Line += 1



Lines 2 → (TOP_BORD − 1) INCL

BOT_BORD → (LAST − 1) INCL

= Border

→ Top border = (TOP_BORD − 2 lines)

Bottom border = (LAST − BOT_BORD lines)

active = TOP_BORD → (BOT_BORD − 1) INCL
= BOT_BORD − TOP_BORD − 2

LAST = bottomborder + BOT_BORD

∴ TOP_BORD = Top border + 2

BOT_BORD = Active + TOP_BORD + 2

HSync	8 μs
+ op	20
Burst	2
Blk	72
op	20
op	20

82 μs !

$\frac{1}{44,100} = 22.676\,\mu s$

Picture rates:

φ	Hz	× Forbidden.	μs	÷1/44100 →
1	23.976	→	41,708.375	1839.339
2	24		41,666.666	1837.5
3	25		40,000.000	1764
4	29.97		33,366.700	1471.47
5	30		33,333.333	1470
>5	Forbidden			

7 μs error/line

$$\frac{\left(\frac{1}{Hz}\right)}{\left(\frac{1}{44100}\right)} = \frac{44100}{Hz}$$

1 video frame $= \frac{1}{23.976} = 41,708.375\,\mu s.$

72 minutes ~ 103680 lines (24 Hz)
× 7 μs = 0.72

Memory map:

388 lines
+6+9 = 303 × $600

= 71A00
Say $80000 ~ ~~65~~ 512k.

0 - 3FFF	ROM STUB	
4000 - 4FFF	COBIOS	
5000 -	Object list 68k Flags etc	
6000	~~TBUF FBUFFER~~	FBUF1
7000	~~TBUFFER2~~	TBUF1
8000 - 8FFF	68k stack	
~~9000 A000~~	~~bss~~ bss	
9000 - FFFF	.data + .bss	
1 0000 - 1 1FFF	68k code	
2 0000 - 2 FFFF	TBUF2	= 64k
10 0000 - ~~1F FFFF~~	Screen 1	
18 0000 - 1F FFFF	Screen 2	

MPEG Audio – Stephanie

00

Have tested with MPEG audio stream.

Layers I and II only.

32, 44.1, 48 kHz. Since we only have 44.1 interrupts.

1152 samples per channel per frame → 23 ms/frame
or 11.5 ms/frame/channel.

DRAM

1. Input FIFO — 256 words Read
2. Output buffer — 64 words write] 2 words at sample rate
3. — 32 words Read]
4. Windows Left — 480 word reads + 32 word writes] every 32 sample periods. Faster than this maybe do all this
   Right — " "] ×2 " "
5. Table — Read a few values at start of each 1152 sample frame for each channel.

DRAM ~~Reads~~ ~~1. Every~~ Sample interrupt → ~~1/4 to 1/8~~ × 4 bytes → 1 to 2 bytes

2.

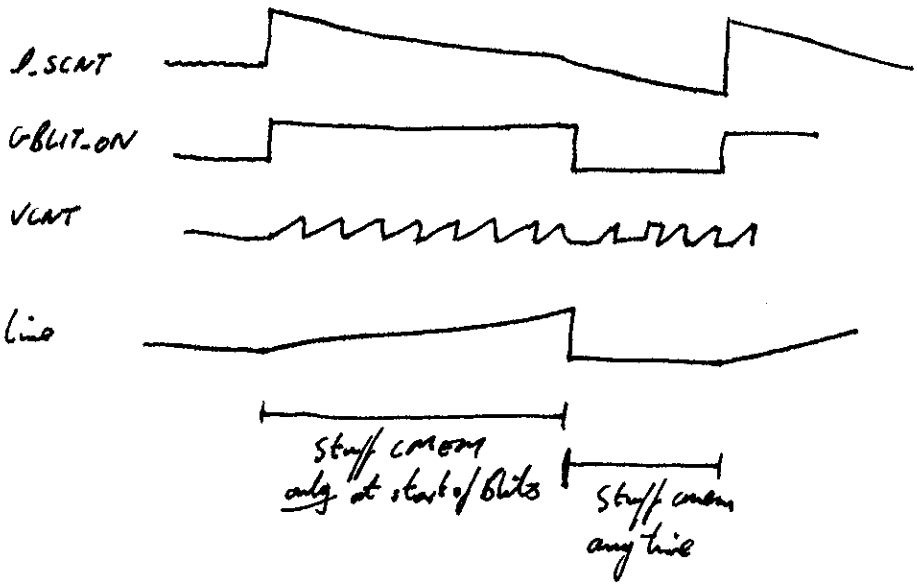
1. ½ to 1 Byte — Any — 1 Byte Read
2. — 4 Bytes Read] 4 words
3. — 4 Bytes write]

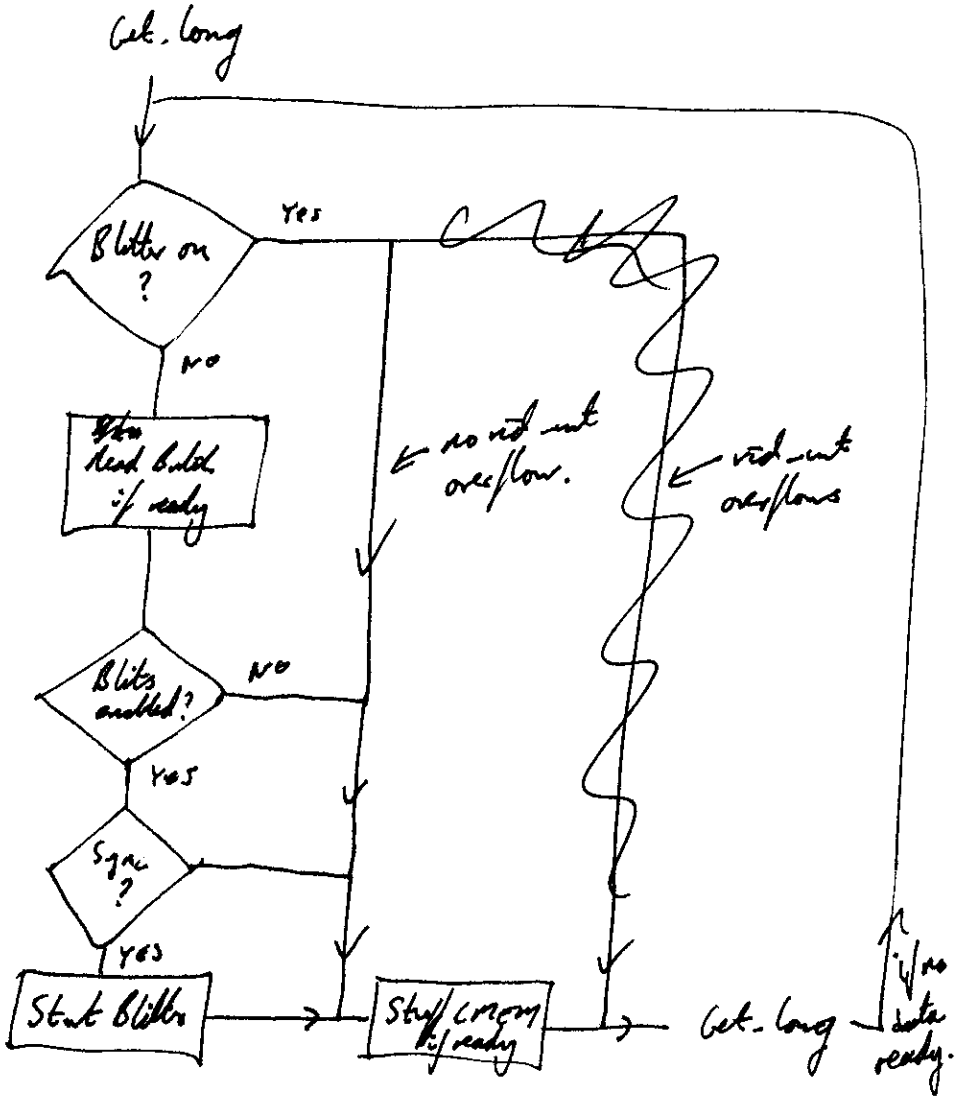
384 samples
→ 768
×2 = 1536

4.

	15 word	Reads
+	15 word	-
+	1 word	write
+	1 -	-

65 Bytes Read
8 Bytes write] every 20 µs.


h_SCNT
GBLIT_ON
VCNT
Line
Stuff cmem only at start of Blits
Stuff cmem any time


Get_long
Blitter on ?
Yes
No
Read Blitd if ready
Blits enabled?
No
Yes
Sync ?
Yes
Strt Blitter
Stuff cmem if ready
Get_long
if no data ready.
← no vid_int overflow.
← vid_int overflows

11/22/94

Current code does cmem stuffing during blits when ~~blt~~ blits are active (2 tick writes) and not doing blits when blitting frame is over (5 or 6 tick writes).
($1875 = mouncam)

1. Looks like:

```
--> load (vid-tail), R0
    rorq #16, R0
    storew R0, (cmem)
    nop  ]
    nop  ] x6
    :    ]
    rorq #16, R0
    storew R0, (cmem)

    addq
    subq
<-- jr
    bclr
```

Probably become one AS ~~[illegible]~~, since rom1 stayed the same.

Everything else used is missing during non blit cycles.

2. Tried

```
--> load (vid-tail)
    rorq
    storew R0 (cmem)
    storew R0 (rom1)
    rorq R0
    storew R0 (cmem)
    storew R0 (rom1)

    addq
    :
<-- :
```

Gave a vid-cont overrun error #3.
Obviously takes too long

3. Tried same, but:

```
load (vid-tail)
rorq
storew (cmem)
storew (rom1)
rorq
storew (cmem)   <-- ie 2nd storew (rom1) removed.
addq
```

← OK ~~[illegible]~~ for TOPGUN (24Hz) = NG 112294.21P.

[SOME ERRORS AS EVERYTHING WARMS UP, ie by the time the guy opens door in red room]

4. Tried

```
storew   (cmem)
storew   (zero)
xorq
storew   (cmem)
storew   (zero)
```

Got vid-cnt overrun error (#3)
a/fr $24C cmem words.
line = $A4
FBLIT counts = 0 (ie first frame).

5. Same as ↑ but deleted.

Got vid-cnt overrun error #3.
a/fr $3FC cmem words
line = $C7
FBLIT counts = 0 (ie first frame)

6. Went back to #2 above, but changed Get_Long flow chart to test vid-cnt when blitter is static as well.

Very good! But still occassional data errors. (Try Gem)

30Hz Message - Error $C, line = 1, cmem words = $1218.
FBLIT = 1 → = frame overrun.

7. Same as #6, but set GPU to run flat out (ie no syncs).
A few data errors, and then back up after about 10 seconds
(probably missed BUTCH interrupt in JOLLY).

Current problems are:
1. A few data errors, if the cmem stuffed during blits
2. Not quite up to 30Hz (check actual rate)
3. Vertical lines jitters.

Current status is: About ~~30, 2 Hz~~ 31 Hz (35 Hz if no CMOS or Butch cycles)

Some data errors

Vertical jitter

Hangs after a while (due to ~~still~~ missing a Butch interrupt or doubtly? or else vid-int overrun.

Using → strew (CMOS)
strelm (Nmi)
~~nmq~~
strelm (CMOS)
ddq
⋮

Plus ASP writing to GBLIT-ON each frame.

$4800 50% at start.
50% on completion.

(510) 657 9201

2D6E9083 $40 0000 of Indecent Exposed, Disk1.

"

16MB of Raygun disk2

BBD30EE6 ""

Data Error possibilities:

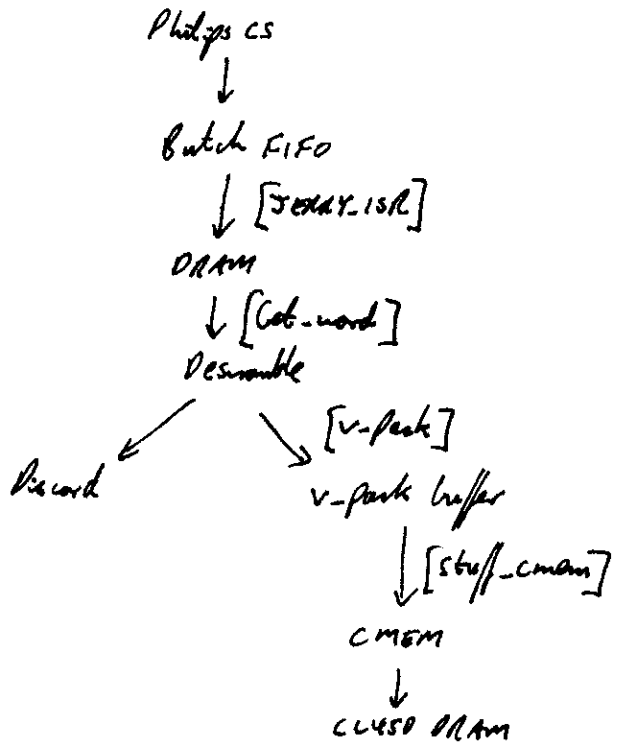
Setup + hold on - Address (A19, A20) X

Data looks good on 'scope.

R/W X

11/23/94. 1. Tied R/W = 0
A19, 20 = 1
] still errors.

2. Wrote 0 only to CMEM] Data bus looked OK.



Try a) Checksum in Get_long. Say, several MB.

b) Checksum in V-park - - -] 16MB checksum: Proved that data is OK.

11/28/94 / 12/1/94

✓1. Do checksum test in stuff-ovem to confirm last week's results. ✓CASUM notes (+2)

✓2. Design PAL for GPIO line, + test it.

✓3. Do audio code:
- Split screen with release bit in object
- Audio-park code
- Dummy Jerry ISR, to check bandwidth.

✓4. Sync from Jerry. Debug it

5. Vertical jitter problem

6. C3 error ~~recovery~~ recovery - just ignore, but keep a count!

7. PLAY, STOP, Pause, Rew, FF, Slow motion, Sing & step

8. Video CD filing system

9. AV Synchronisation. Accurate (ie non-drifting 44.1kHz syncs)

10. Squeeze some more bandwidth out.

New H/w:

Latched bits:
- HSYNC
- VSYNC
- RESET_450
- A19_450
- A20_450
- RW_450

DAC: o/ps:
- AS
- Xφ
- X1
- VCLK
- VOE
- Sφ
- S1
- WAIT
- ROM CS

9

Latch_Sel 1

10

i/ps:
- GCLK
- ATACK
- ROM1
- EA18
- 19
- 20
- 21
- [illegible]
- WEφ

8

.FBUF1 Patriot games, disk 2 - $D entries

sngl	LBN	
$206C3800	CA9	= pat2.tbf

Top Gun, disk 1 - $G entries

198900	9A9	controls.tf
40F800	C0A	for-presentes -
20BE1800	1699	top gun #1

VSP Sy

O_SCNT

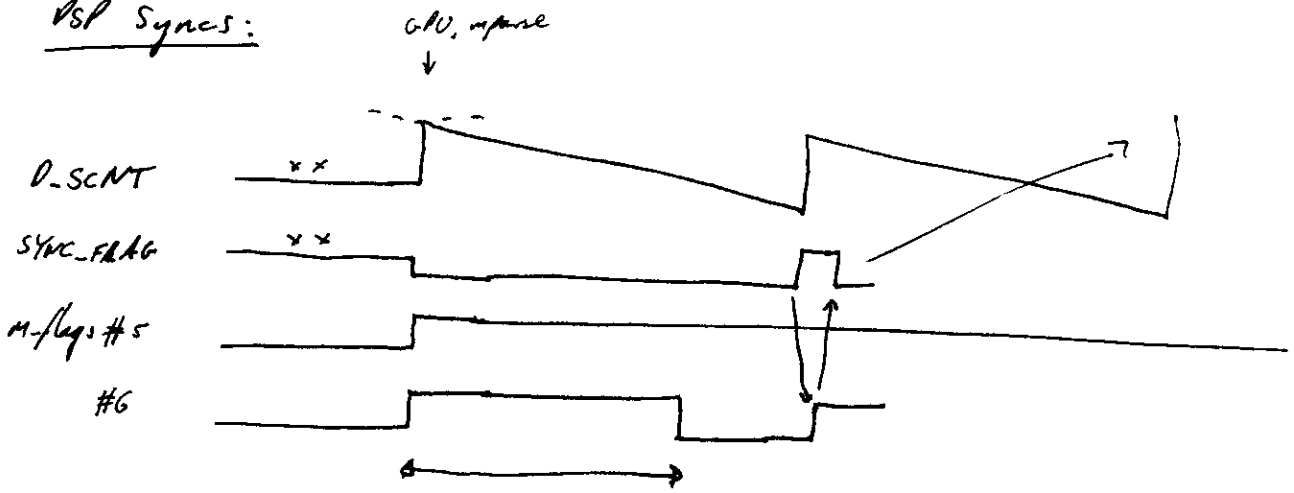
SYNC_FLAG

M-flags: #5

#6

OSP:

DSP Syncs:



DSP sets SYNC_FLAG
GPU clears it, and
sets m-flags #6.

GPU polls SYNC_FLAG

GPU polls m-flags #6

DSP: movq #1, R0
storew R0, (R1)

R1 = F03E08

F03E08 = 00

Jerry debug storew, store.

1.	storew to long aligned GPU RAM	No write is done.
2.	storew to long aligned +2 in GPU RAM	1st time, lower + upper word written same data. Then, just lower 16 bits are written to.
3.	storew to long aligned PRAM location	OK
4.	" " " +2 " "	OK
5.	store to long aligned GPU RAM	Not reliable. data flows into following long. Data from previous write is held.
6. [illegible] +2 GPU RAM		When storew to PRAM was removed, then it worked.

→ Always precede a stor with load x
or w,x

PAUSE command

Pause button detected

Set 'Pause' in m_flags

cdc_sync will kill CD stream after reading next header

Interrupt 68k

wait for 68k command

stuff_comm byte counter ≤ 0.

clear Pause bit

set header match bit

wait for our header again

clear header match bit

continue as before.

68k - CD Pause

- wait for Pause button to be released.
- Look for Pause -> continue in RAM mode, after CD_read

Play ->

Inquire buffer fullness. Set value for CPU to count to before Pausing again.

SingleStep

12/6/94

Audio Commands:

Begin	#1	- Initialise + start decoding with new data (eg after STOP)
Continue	2	- Re-start after Pause. Play. Stop with bit the initialisation.
Pause	1	- Pause, but keep data
Volume	3	- (includes MUTE - values 0 → 100)? High byte = Vol
Stop	0	- Mute, and stop decoding (Leaves Kernel), Initialise everything.

CMD is read every 1152 samples

D_SCNT
D_SMAX
D_SFLAG — GPU sets this to SMAX, then clears D_SFLAG to 0.

DSP needs minimum of 400 bytes to start.

A_TAIL - Initialised by DSP to $3 0000 - $3 7FFF. (with 'BEGIN' Command) Kept in DSP memory

A_HEAD.L - Kept at $3 8000. Always add 4, mask with FFF.C.

DSP middle at 3 8000 - 3FFFF

$3 8004.w D_CMD DSP Command

F03000 EQU G_SFLAG.w (in fact F03002)

DSP .extras:

F1B244	D_SCNT.L	lsw = SCNT, msw = SMAX
3 8004	D_CMD.L	
F03002	G_SFLAG.w	
3 8000	A_HEAD.L	

$$\frac{n}{44100} = \frac{m}{23.976}$$

$$\frac{n}{m} = \frac{44100}{23.976}$$

or, $n \cdot 23.976 = m \cdot 44100$

$$n = \frac{m \cdot 44100}{23.976}$$

error is: ~~$\frac{n}{44100} - \frac{m}{23.976}$~~

~~$n \times 23.976 - m \times 44100$~~

error is: $\frac{n}{m \cdot 44100} - \frac{1}{23.976}$

12-8-94 PAUSE again.

[End of frame - detects 'Pause' button.
Set 'Pause' in m-flags
Disable Blits in - - (i.e Blitter OFF)
Return

↓

Get Sync will trip after next sector header
Interrupt 68k.

↓

WAIT for 68k

↓

'M_resume' command from 68k

[Assume CC450 + AUDIO
have at least 2350 bytes
free in their buffers!]

CD-I Filing system etc

	MSB		LSB		
1. word 0	SEC	MIN		In BCD	] CD-I Mode 2 Form 2 header block
word 1	MODE	FRAME		In BCD	

2. 0 MIN SEC FRAME] in HEX For CD BIOS

3. LBN = linear count of sector numbers.
For headers, and CD-BIOS, add 2 second (→ +150).

Filing system → Load Directory
↓
Get LBN of File
↓
Add 150 (2 seconds)

↙ Subtract 7
Convert to HEX
for CD BIOS

↘ Convert to BCD
for Header matching

BCD (Header Version)
↓
Convert to LBN

CDI File Structure.

TOC is not valid.

TRACK 1.

DISK LABEL. (Path Table)

ABN

Disk Label is at a fixed address = 0, 2, 16

Sectors are Mode 2 Form 2 → 2352 = 12
+ 4
+ 8
+ 2324
+ 4

Volume Structure Standard ID
= "CD-I␣".

PATH TABLE → list of all Directorys → ROOT

ROOT Directory → List of Files

→ other Directory → List of Files

Filing system gives LBNs.
BIOS and sector headers, use ABN = LBN + 150

VideoCD File Structure

TOC is valid.

TRACK 1 : 00:04:00 = INFO.VCD file
bytes 1-8 = "VIDEO_CD"

00:04:01 = ENTRIES.VCD
= the list of all MPEG tracks.

DSP controls

All tables
Scale Factors
Quantizers

every Frame (1152 samples), per channel

1152 samples = 26 ms: (store) 64 longs (Quantizers) written to DRAM
(store) 385 bytes ~~= 97 longs~~ (Scale Factors) written to DRAM
(load) 64 longs (Quantizer) Read from DRAM
then 12 × 64 words Read from DRAM
= 768 word reads
= 384 longs
(load) 314 words (Compressed Data) Read from DRAM
= 157 long reads

1 long read A_HEAD

1 long read MPEG_CONSTANTS

---

766 ~~bytes~~ loads (longs) in 26 ms.

→ 1 long memory access every 40 μs.

$$F_s = \frac{F_c}{2.(SCLK+1)}$$

$$2.F_s\,(SCLK+1) = F_c$$

$$\frac{F_c}{2F_s} - 1 = SCLK$$

need 1,411,200

300

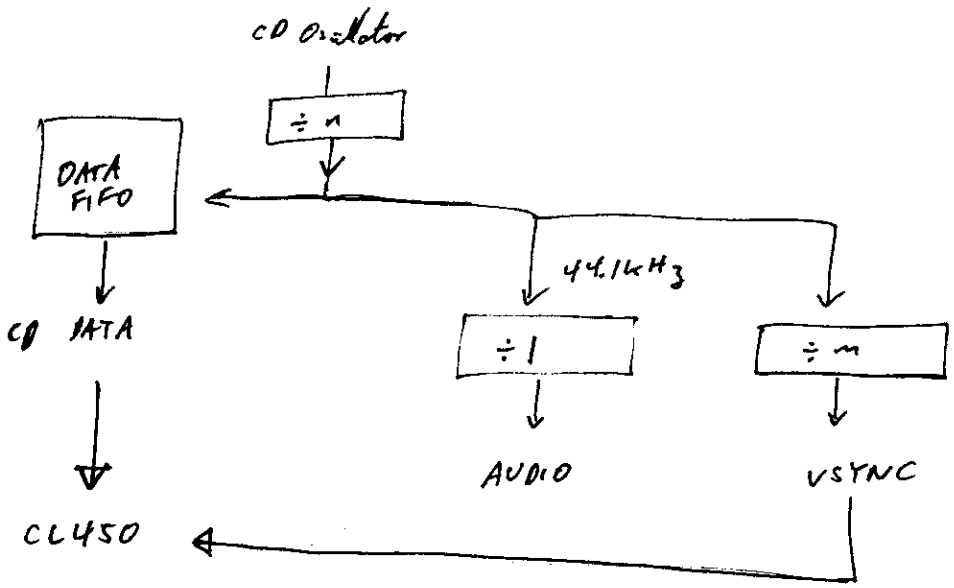
Use 8-4

7 SCLK

$15 SMODE

1/V Synch

A/V Syncronisation



Audio frames are 1152 samples → 26ms

Layer II

Get_Sync

Read Get 32×2 Quantization values (16bit)
Read Get 96×2 Scale factors (16bit)
Write 32×2 Quantization values

(One per Frame)

Read 96 Samples ×2
Read 96×2 scale factors

(in loop | Sparse RAM access)

Interval
Read window
Write sample to interval buffer

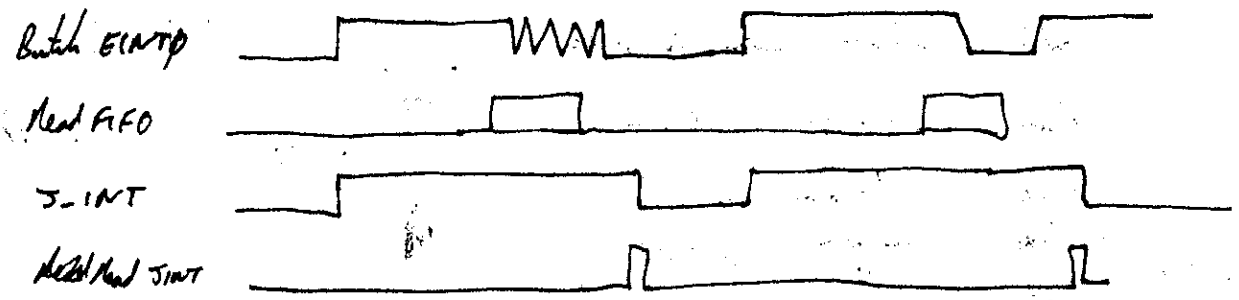
1. Clear
2. Write Quantization values
3. Sync Flag to GPU

RunCnt: Increments by 12 every frame

MIPS: How many samples out per 1/12 frame decompressed
(384 = 100%)

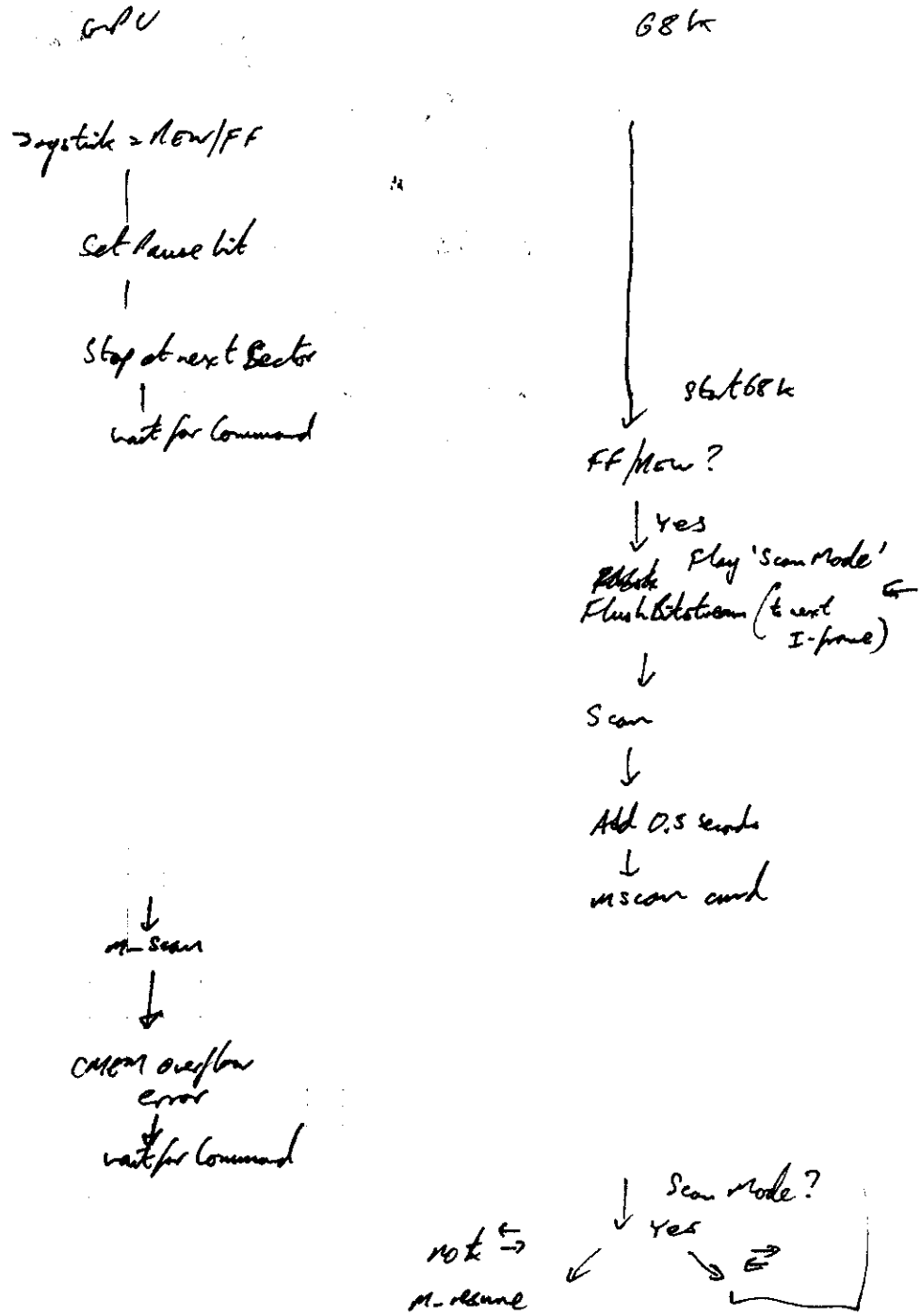
FF/LE

Hang:



2601

FF/REW ← →


CPU
68k
Joystick = REW/FF
Set Pause bit
Stop at next Sector
wait for Command
Set 68k
FF/REW ?
Yes
Play 'Scan Mode'
FlushBitstream (to next I-frame)
Scan
Add O.S search
mscan cmd
m_scan
CMEM overflow error
wait for Command
Scan mode ?
Yes
not ←→
m_resume

2/13/95 Issues: 1. DA interface. (Probably needs a delay stage)

2. VCLK enabled?

3. Pullup/Vcc?

MC32690D-3.3 -D is better. 48¢

-DT 67¢

10 weeks

Jim Chandler - 205 464 6821

2-13

Host I/F

D/A I/F

Je

CLK

Data (Jerry)

WS

Data (CL480)

2-13-95 CL480 design

Host I/F. On CL450, we could not use GPIO for all 450 host accesses, since there was not enough address space.

Here, we use GPIO for all registers.

GPIO 0 = CL480 host port
EA(4:2) = " " " addresses

GPIO 1 = Misc latched bits

Rom1, $\overline{EA18}$ = Rom

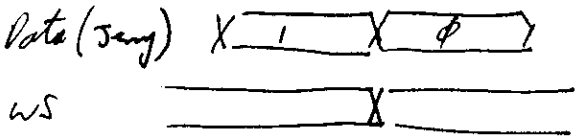
Rom1. EA18 = Pixel port

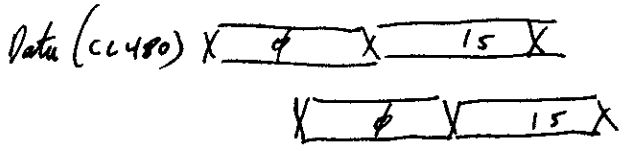
Batch will prevent Batch register accesses getting through.

CFLEVOL goes to EEPROM E2DATA bit.

D/A I/F.

Jerry needs:





→ Delay data by 1 bit, + edge.

→ Delay data by ½ bit ↑ edge

480 PCB layout notes

1-27-95

1. Add 10μF Tant Cap between VDD3 + GND, vcc at U8 and U7.
2. Fatten up VDD3 traces out of U8.
3. 40MHz part is 3.3V type! U5/8 changes from VCC to VDD3. Extend VDD3 plane out to U5/8.
4. U7/3 near = Rev 480 = U4/16
5. UCASIN/LCASIN (U7/50, U7/45). Connect to U3/28, so trace length is same as CAS signal, back to U7/50, 45.

John - I have inserted the numbers for the gaps:

pclk to ROMCS1 valid externally	28	measured Toshiba Tom assumed worst case as 2 mA
take away pclk to internal clock	-10	intelligent guess
ROMCS1 to Butch ROMCS1 output	13.9	Tim's / Orbit's spice figures
ROM1 CS at GAL to VOE on CL480	15	15ns GAL, combinatorial term (This can be faster ...)
VOE on CL480 to Data out (50pf)	15	CL480 data book, into 50pF
ED to D delay through AC245 or Jerry 2	6	Jerry 2 assumed worst case
D to internal Tom load bus	10	intelligent guess

	77.9 ns	

We are out by 2.9ns in this scenario.

C-Cube says 15ns is pessimistic. I can apparently use 10ns. I will get this in writing. Also, the GAL can do better than 15ns. I can use 10ns parts for example (and I will!).

This way we get an extra 10ns, meaning a 7ns margin in total. I think this is OK since eve rything is worst case. What do you think?

Richard

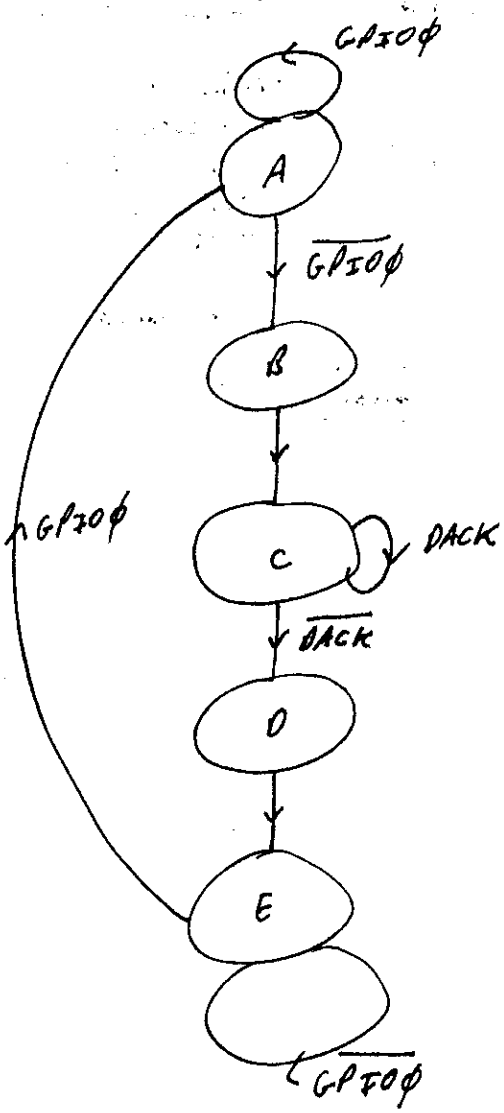
CL480 Data Timings 3-10-95

	ns	
PCLK to ROM1 out, 2mA Toshiba	28	Measured
Minus PCLK to internal clk	-10	Guess
ROM1 Both in to ROM1 out	13.9	Tim Dunn, OrCAD Spice
ROM1 at GAL to VOE	15	-15ns GAL Combinatorial
VOE to Data on CL480	15	CL480 data book 50pF load.
ED to D data (Jany2 > AC245)	6	JM
D to internal Tam at latches	10	JM Guess
	~~79ns~~	

~~2~~ clocks = 75ns !

CC480 - P12B

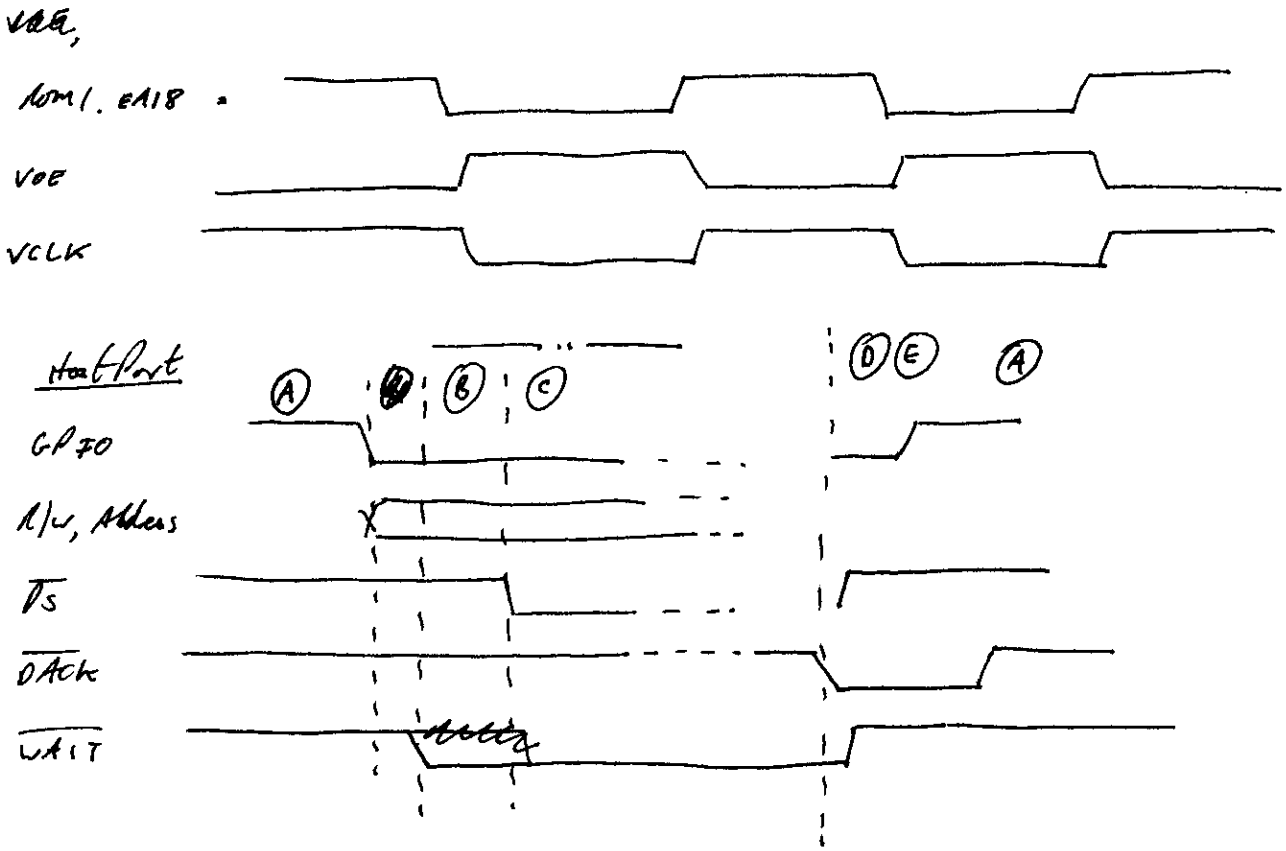
MV1.3ES



	$\overline{DS}$	$\overline{WAIT}$	Xϕ	X1
A	1	1	0	0
B	1	ϕ	0	0
C	ϕ	ϕ	0	0
D	1	(?) 1	1	0
E	1	1	1	1

U44a.PAL

romcs ✓
latil = GPIO ✓



$\overline{DS} = B_a + C \cdot DACK$

$\overline{WAIT} = A \cdot \overline{GPIO\phi} + B + C \cdot DACK$

$X\phi = C \cdot \overline{DACK} + D + E \cdot \overline{GPIO\phi}$

$X1 = D + E \cdot \overline{GPIO\phi}$

- Turn-off time for CL480 host bus → assert wait for longer
- RDE WE0 is too late. Need separate address spaces for reads/writes
- Assert WAIT earlier. Use asynchronous decode
- Gray code the state machine

B
r t

Get CD data

153 kBytes/s = 6.5 μs/byte

Line = 64μs

Switch Fig0 ½ pull after 8 longs = 32 bytes

= 209μs ≈ 3.2 lines.

29 bytes

9.792 bytes/line. = 3 mem cycles
6 ticks = 0.6μs

Put CD data

9.792 bytes/line = 10 memory cycles
6 ticks = 2.26μs

Get Video data

320 pixels = 320 memory cycles/line.

4 tick O.P. cycles = 1280 ticks.

= 48μs

3	30
6	60
9	58
12	56
15	54
	52
	50
	⋮
	36
	34
	32
	62

From RICHARDS Notes. of 26th Aug '94

CD Data Rate 44100 x4 = 176400 byts/s

÷60 = 2940 byts/NTSC frame.

÷50 = 3528 byts/PAL frame.

5·7 μs/byte

182 μs / 32 byts.

Batch copy ½ full after 2·8 lines.

11·29 byts/line.

Get CD data 3 mem cycles. 0·6μs/line. (6 tick)

Put CD data

11·2 byts/line 12 mem cycles (6 tick)

= 2·7μs

Get Video data.

352 pix = 352 mem cycles/line.

4 tick O.P. cycles (3+1 wait) = 53μs

Leaves 7·7μs.

HSYNC ?
VSYNC ?