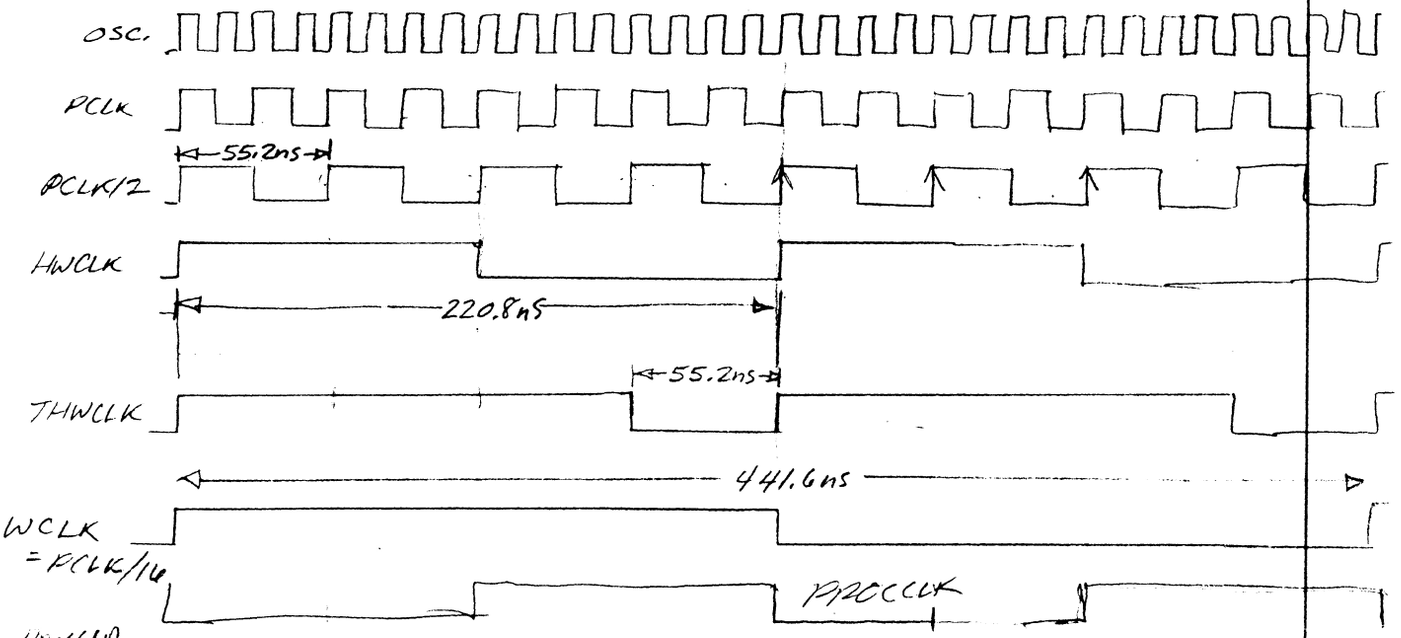
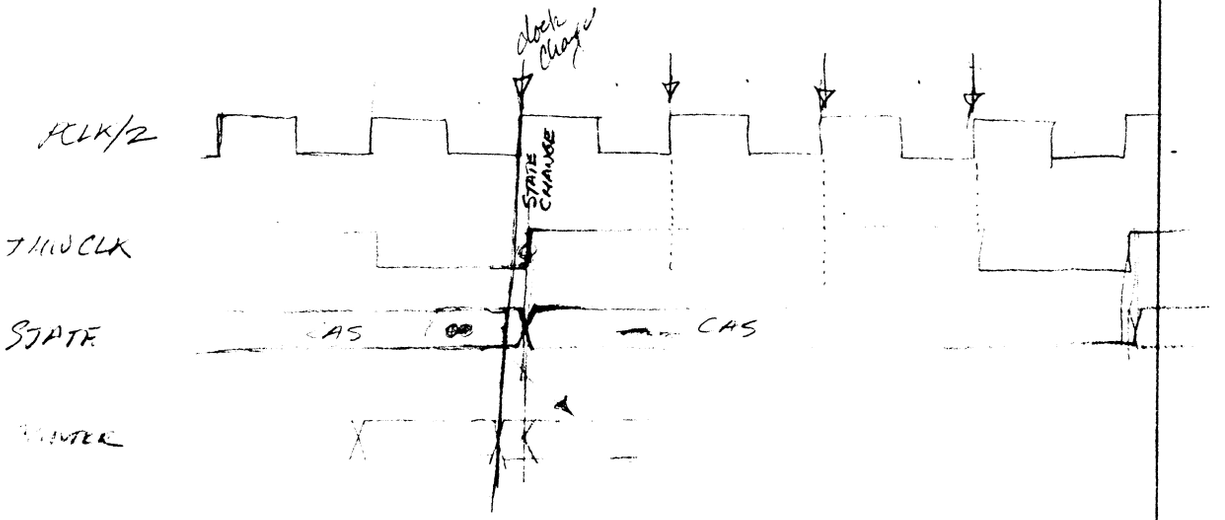
22-141 50 SHEETS
22-142 100 SHEETS
22-144 200 SHEETS
AMPAD

Ω CLOCKS

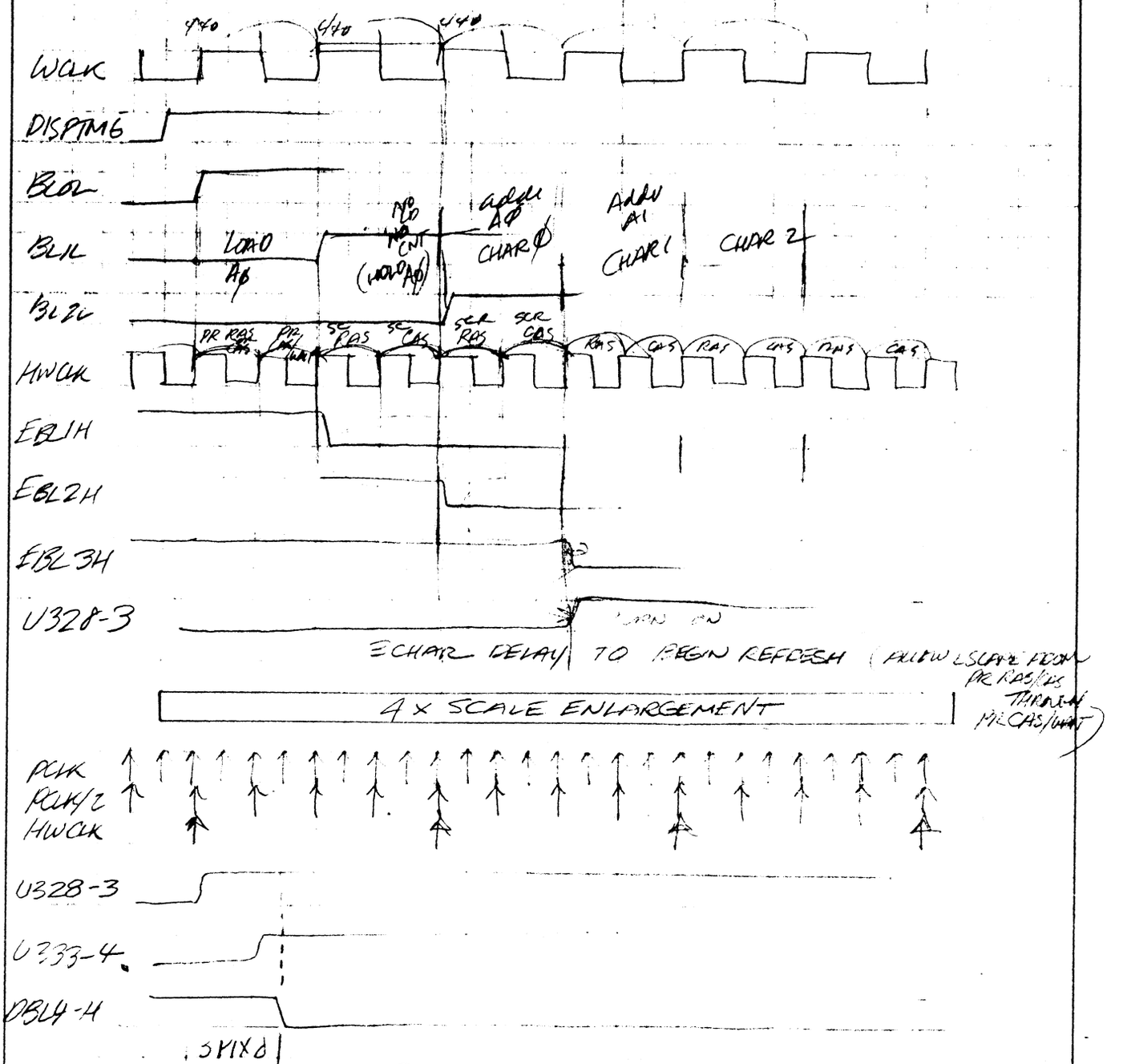
FREE RUNNING:

ECL {
- OSCILLATOR : 72.44 MHz
- PCLK : 36.22 MHz — 27.6 ns
- EPCLK/2 : 18.11 MHz — 55.2 ns — (TTL: PCLK/2)
- HWCLK: 4.53 MHz — 220.8 ns — (TTL; THWCLK)
- WCLK : 2.26 MHz — 441.6 ns — (6845 max clk freq : 2.5 MHz)





22-141 50 SHEETS
22-142 100 SHEETS
22-144 200 SHEETS
AMPAD



ECHAR DELAY TO BEGIN REFRESH

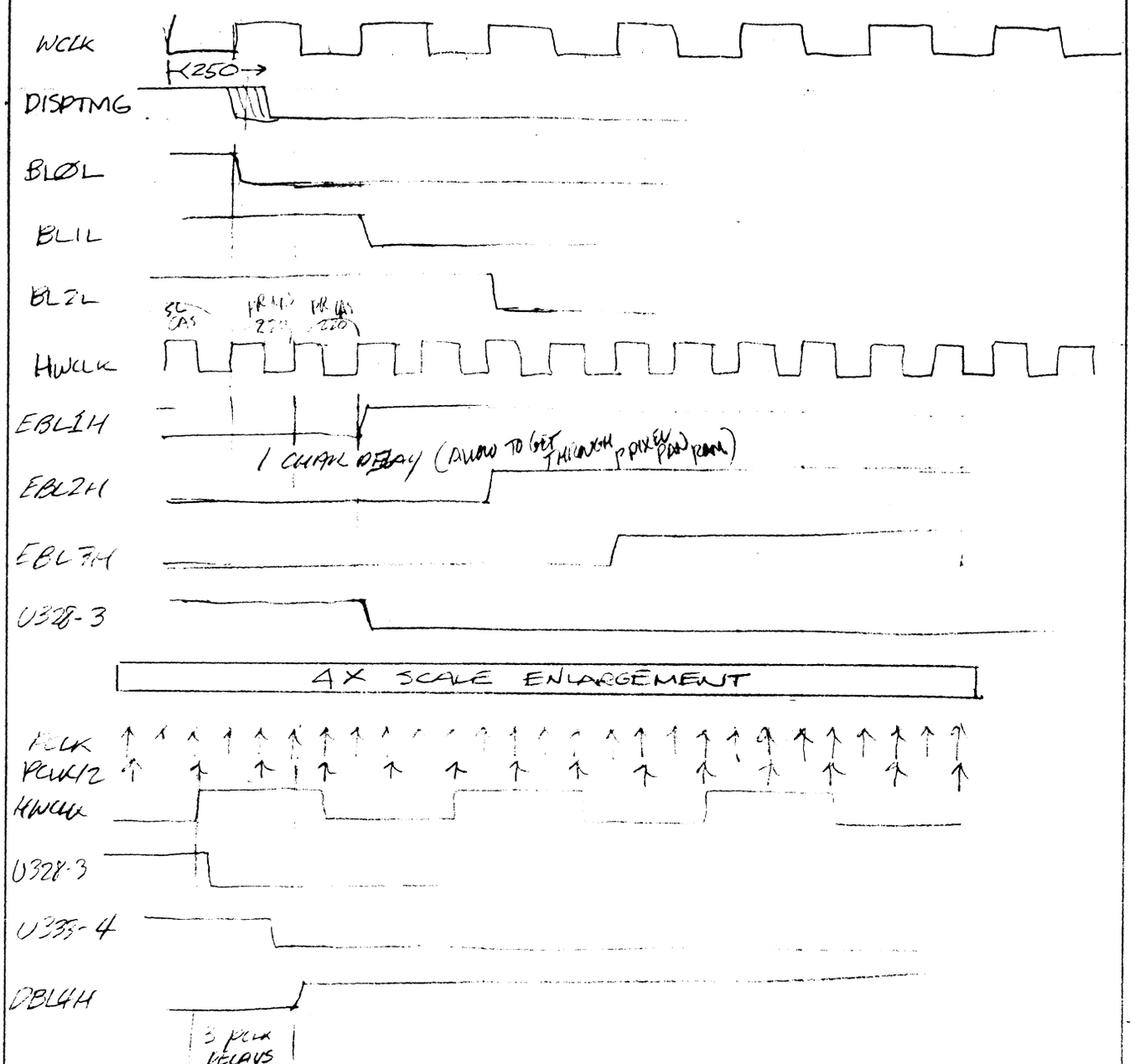
+ 3 PIXEL DELAY

1 DELAY OFF
3 DELAYS ON
NET 2 DELAYS

BLANKING - TIM.

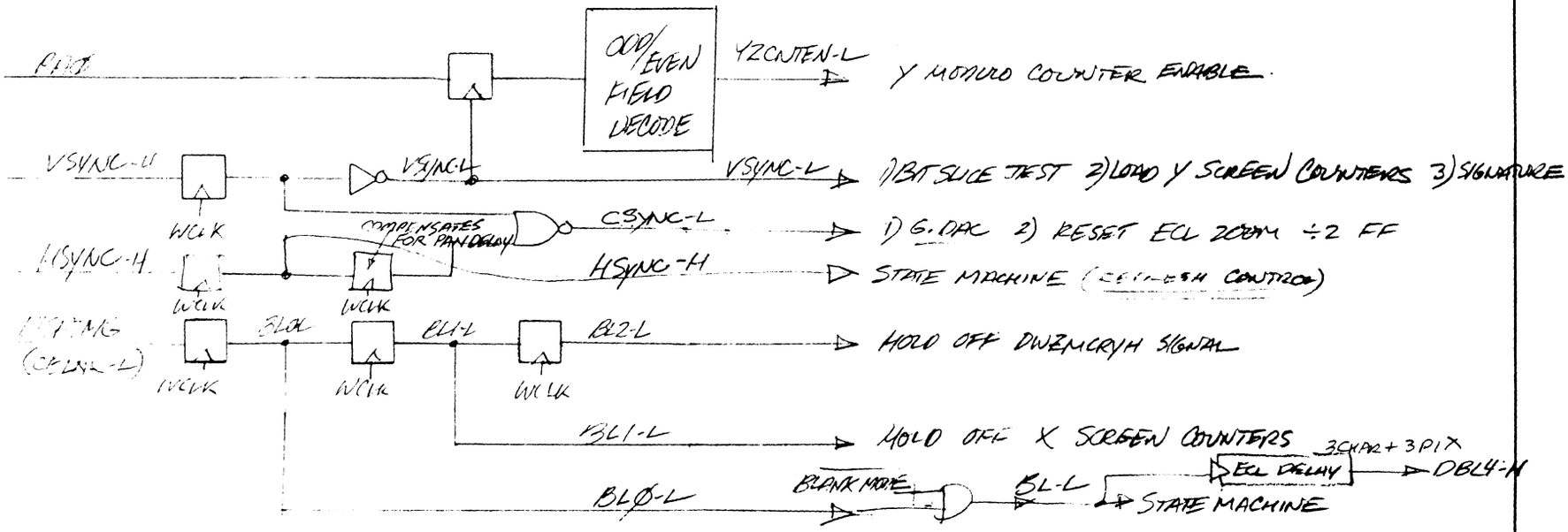
22-141 50 SHEETS
22-142 100 SHEETS
22-144 200 SHEETS
AMPAD

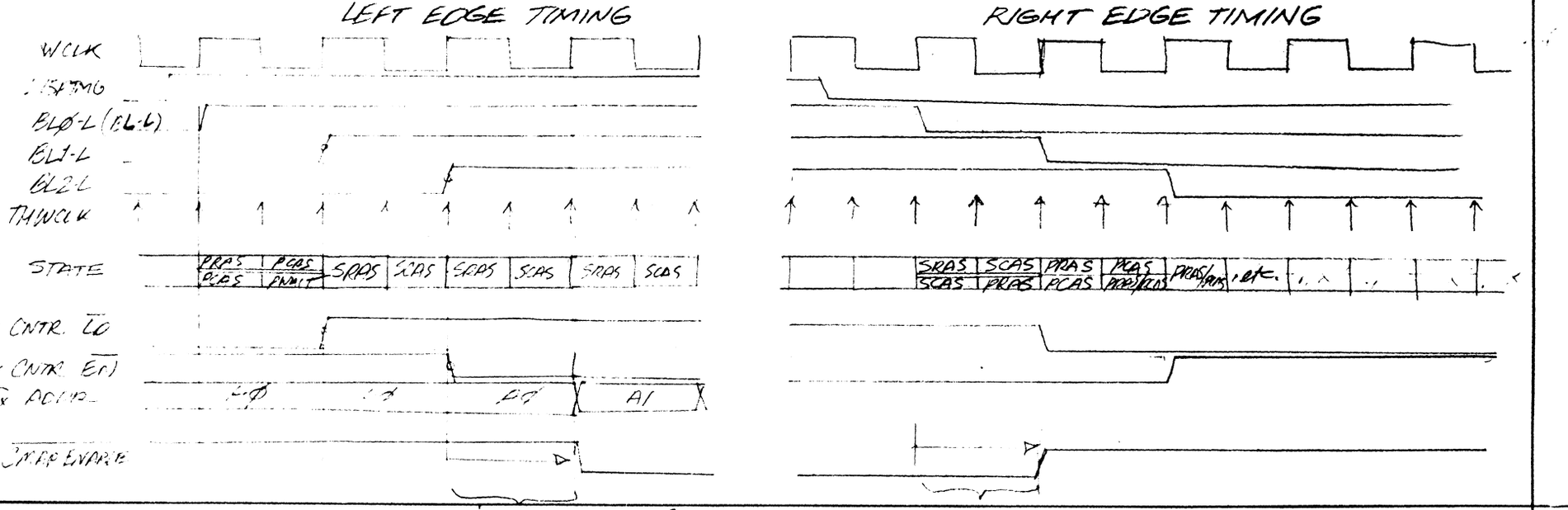
CRT CONTROLLER TIMING



BLANKING - INIT.

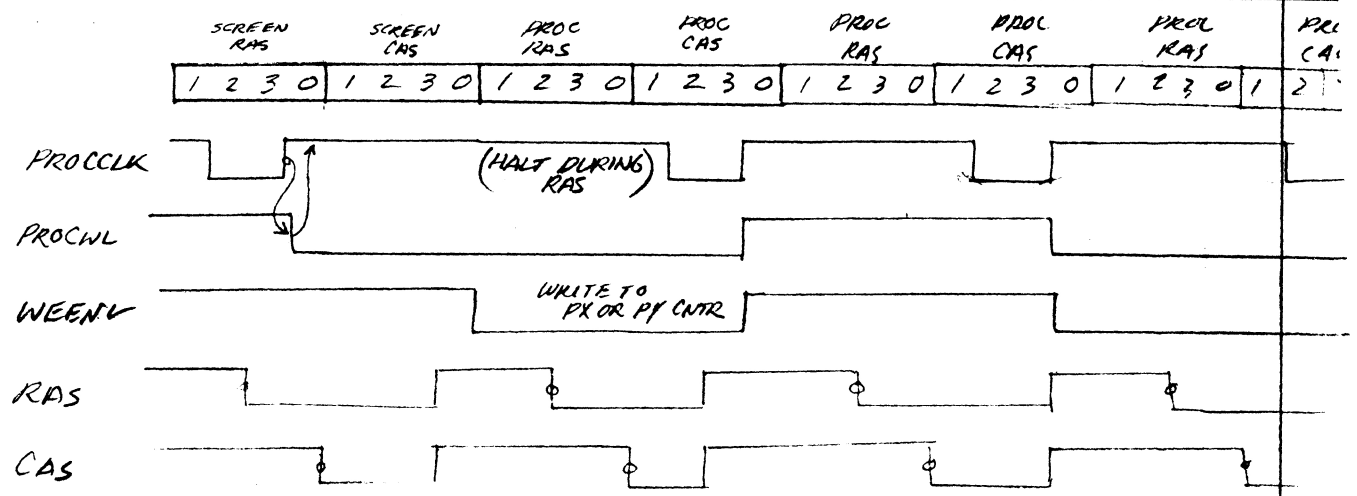
S2-400 VIDEO CONTROLS



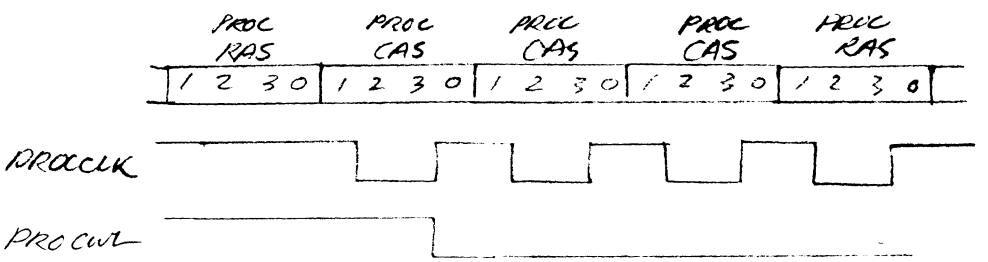


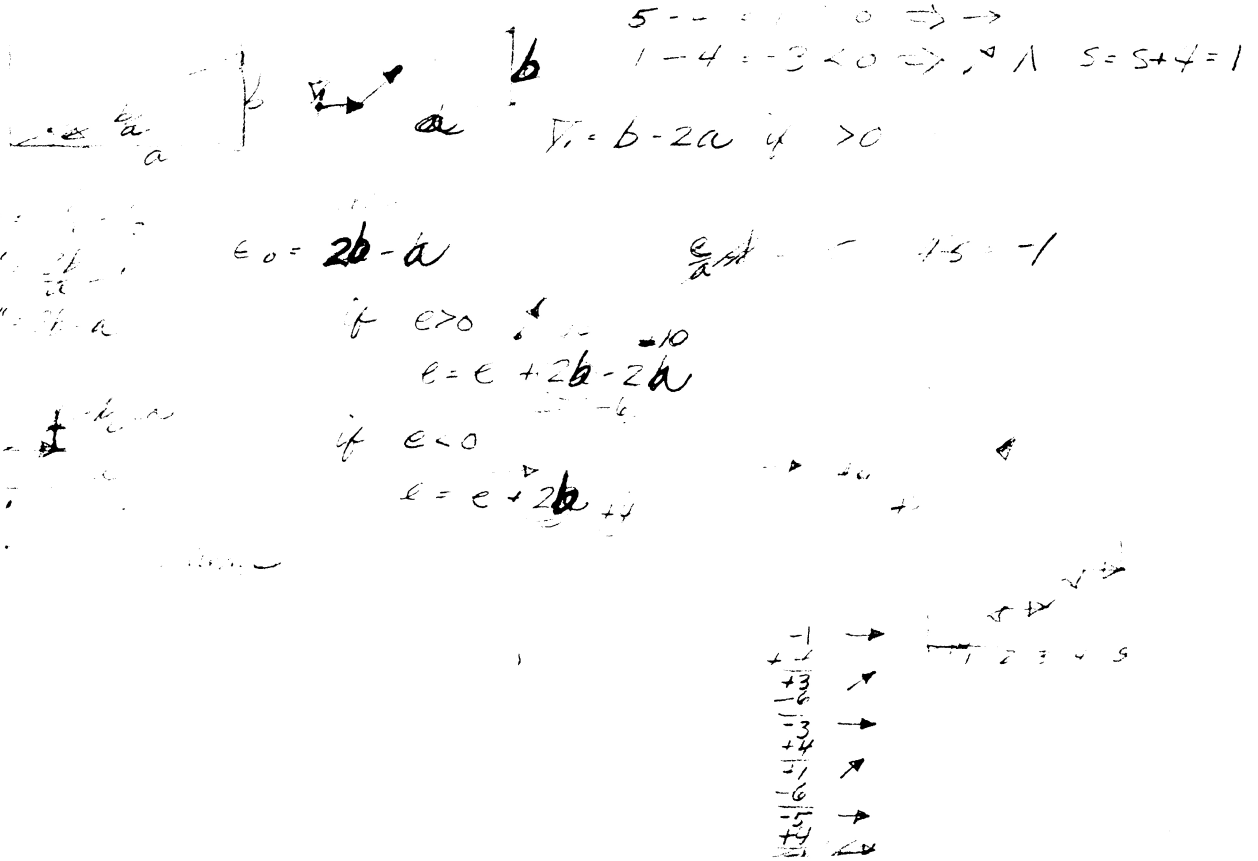
EXTRA DELAY TO COMPENSATE FOR 16 PIXEL DELAY IN RING BUFFER (PAN BY PIXEL)

PROCESSOR WRITE TO RAM

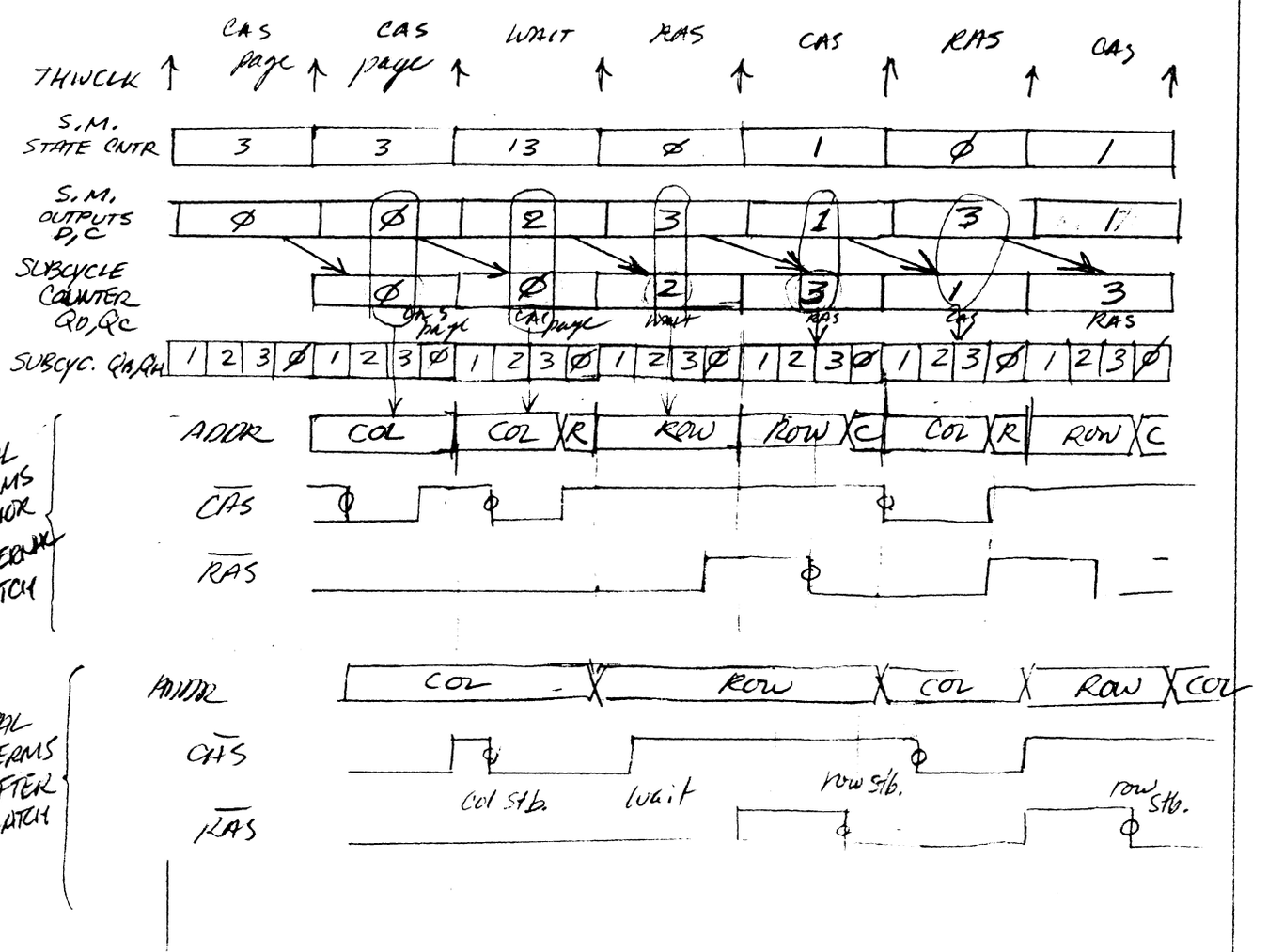


PAGE MOVE



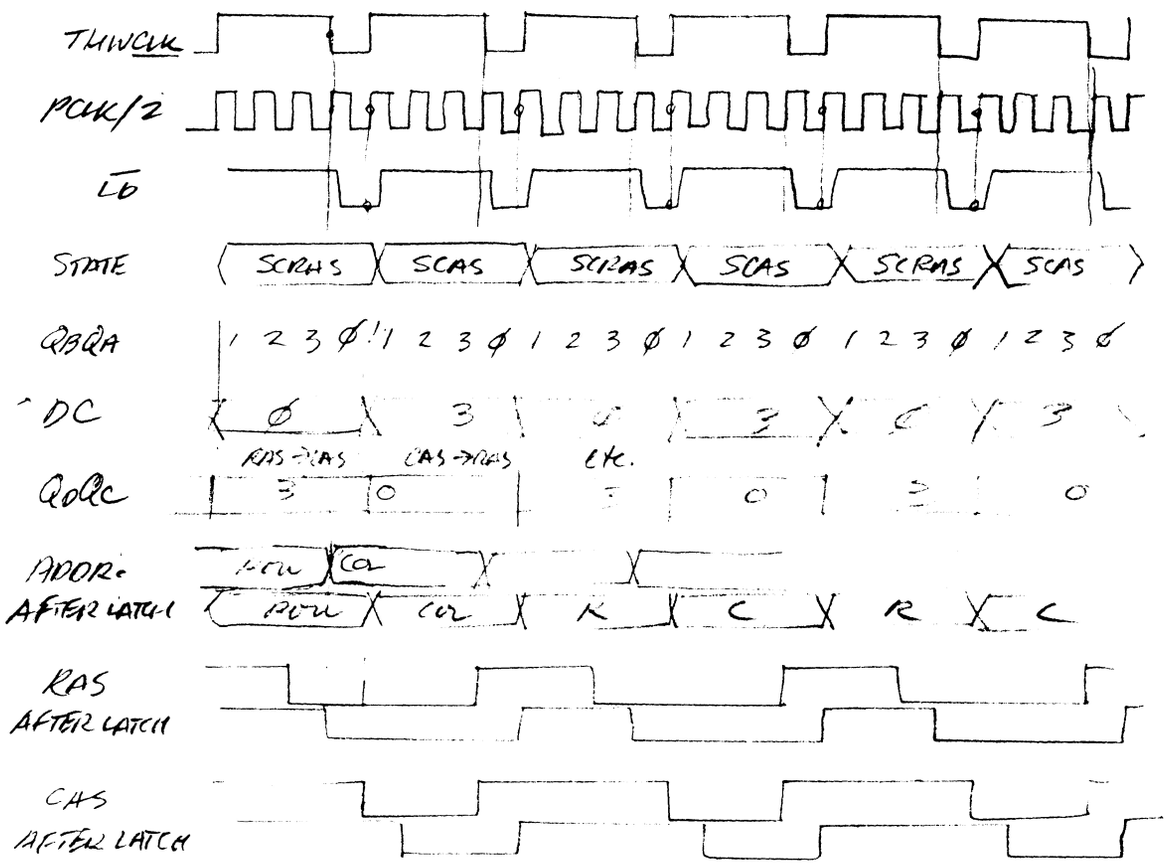


EXAMPLE : CAS-CAS-WAIT-RAS-CAS-RAS

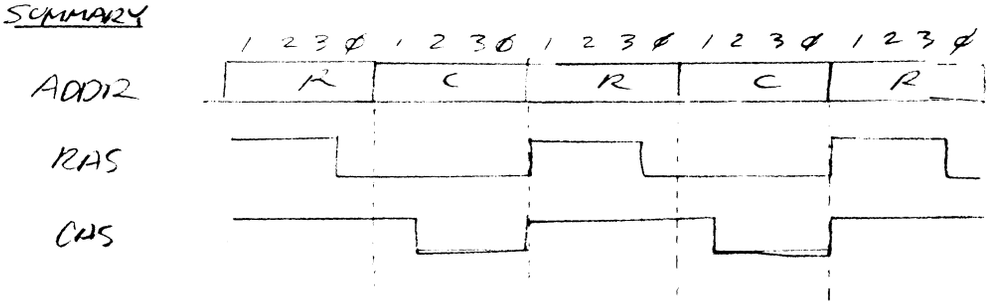


RAS - CAS - RAS - CAS ... etc:

SCREEN REFRESH RAS-CAS CYCLES



SUMMARY

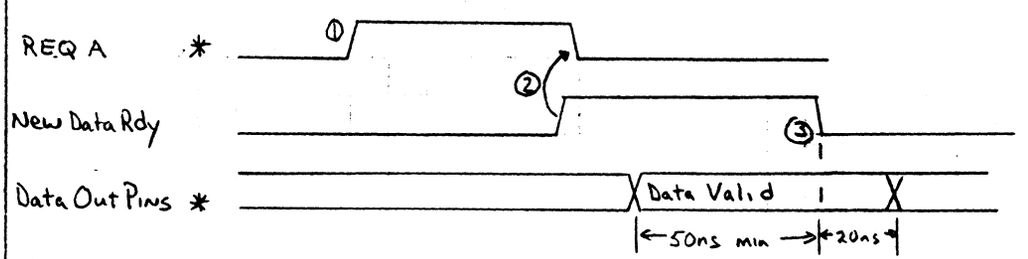


22-141 50 SHEETS
22-142 100 SHEETS
22-144 200 SHEETS
AMPAD

PGM I/O
DRV 11 TIMING

John Providenza

Data Out (Processor Write)



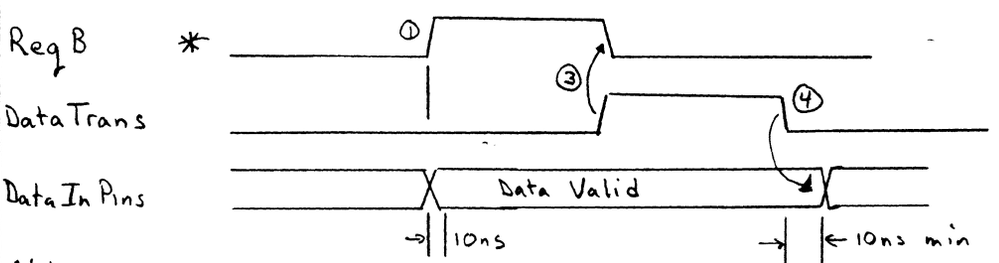
Notes:

① User asserts "REQA" telling CPU it can send more data.

② As processor begins to write new data out, user clears the "REQA" signal.

③ On the falling edge of "New Data Rdy" the output data is valid. Data stays valid until the next CPU write or an init occurs.

Data In (Processor Read)



Notes:

① User asserts "ReqB" telling CPU it has data ready.

② User sets up data before "Data Trans" occurs.

③ On rising edge of "Data Trans" the user clears the data ready request.

④ The CPU has finished reading the data on the falling edge of "Data Trans". The input data pins may now change.

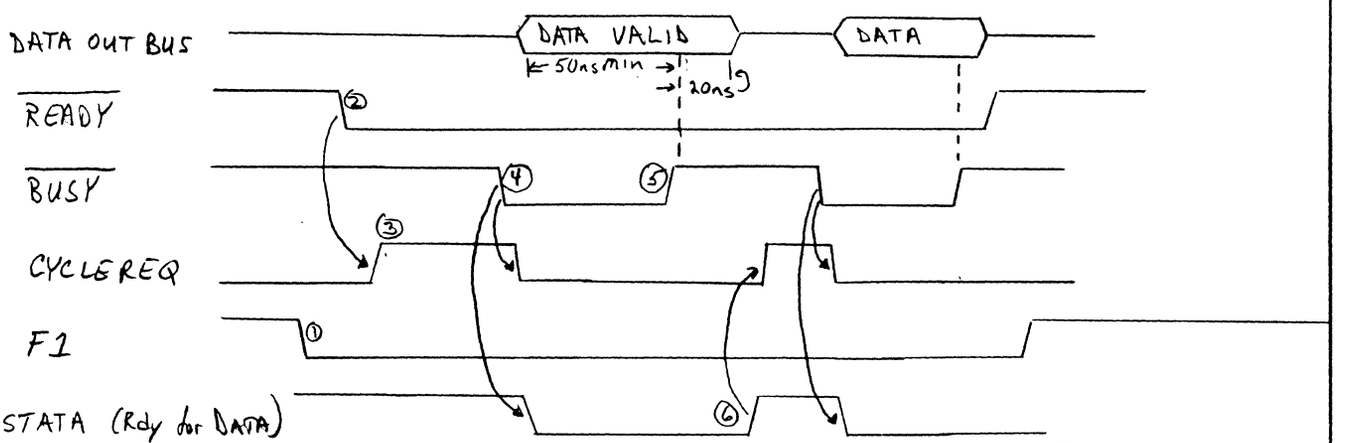
* signal from user to DRV11

42-381 50 SHEETS 5 SQUARE
42-382 100 SHEETS 5 SQUARE
42-389 200 SHEETS 5 SQUARE
NATIONAL

DR11W TIMING – DMA

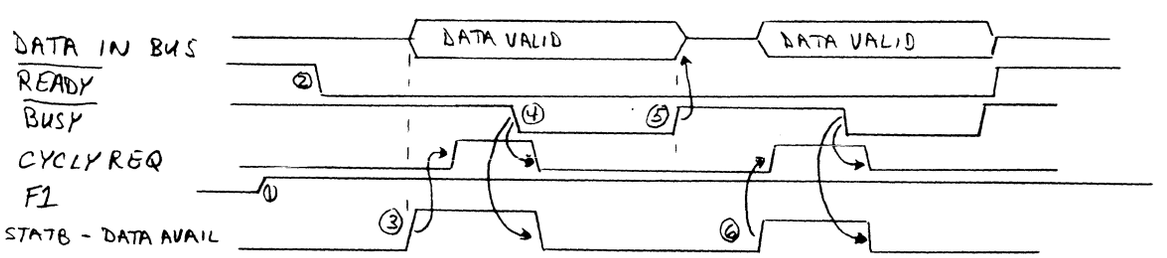
4-14-83 SRP

DATA OUT – HOST WRITE



① HOST sets F1 bit to Ø SIGNALLING A "DATA OUT" DMA OPERATION
② HOST sets "GO" bit in the CSR, this says we can start DMA cycles
③ After Ready is set, if the system is ready to accept data (STAT A = 1) CYCLE REQ will be set.
④ When the HOST starts it's DMA cycle, it sets $\overline{\text{BUSY}}$. This causes CYCLE REQ to be cleared. NOTE THAT STAT A is also cleared.
⑤ ON THE rising edge of $\overline{\text{BUSY}}$ a word of data is latched into the interface.
⑥ When the Omega is ready for another word, STAT A is set, which causes another CYCLE REQ to occur

DATA IN – HOST READS



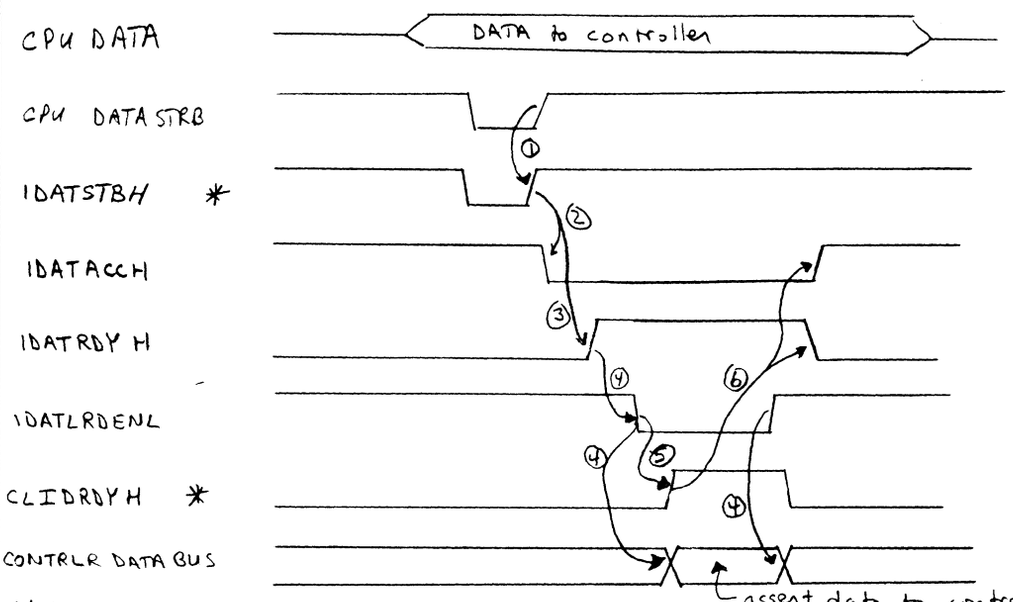
① HOST sets F1 to a 1, we'll do data → host
② HOST sets "GO" bit of CSR indicating that we can start DMA requests
③ When a WORD of data is available, STAT B is set which causes CYCLE REQ to be set.
④ When $\overline{\text{BUSY}}$ is set, CYCLE REQ is cleared as is STAT B
⑤ DMA cycle done, data is no longer needed, Omega can load next word
⑥ When the next word is available for the host to read, STAT B and CYCLE REQ are set and another cycle can begin

NOTE THAT CYCLE REQ is always held off for about a micro second after READY is set.

22-141 50 SHEETS
22-142 100 SHEETS
22-144 200 SHEETS
AMPAD

Red Controller Data from CPU/memory Timing

2-2-82
John providenza



Notes:

1) When computer interface has data for display controller, it latches the data into the data latch and initiates the transfer by a rising edge on "IDATSTBH"

2) this causes "IDATACCH" to go low until the data is accepted by the interface. If multiple bytes are unpacked from a word without wiggling "CLIDRDY", this line won't go high until "CLIDRDY" makes a positive transition.

3) "IDATRDYH" goes high when the display controller synchronizes the request internally.

4) When the micro-code wants to read the data onto the bus, "IDATLRDENL" goes low to enable the tri-state buffers onto the data bus.

5) If this is the last byte available to be transfered, assert "CLIDRDYH" to terminate the transfer sequence, else another xfer sequence will start.

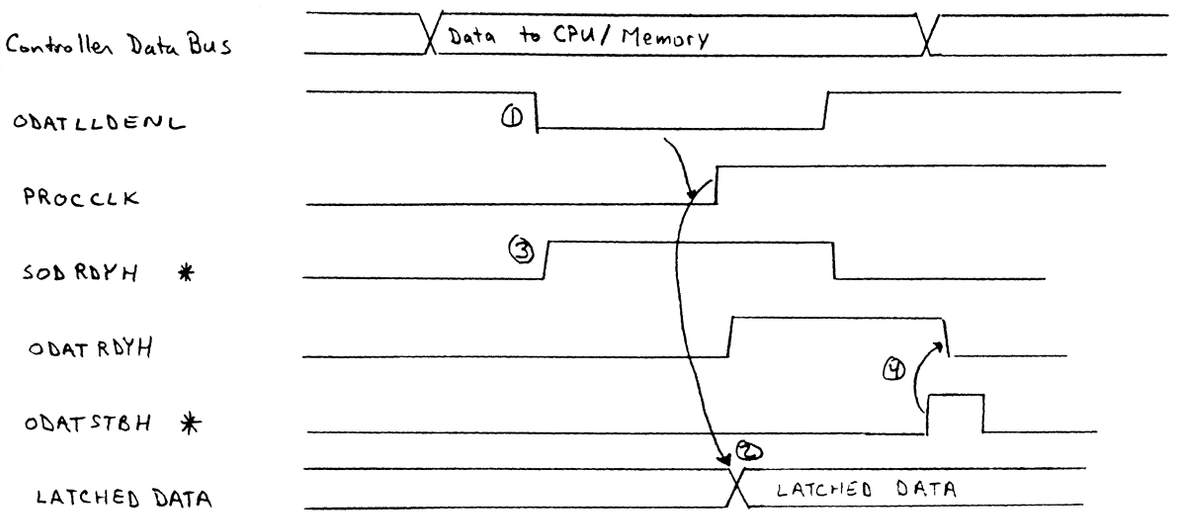
6) When "CLIDRDYH" goes high, the "IDATRDYH" signal from the display will be released as will be "IDATACCH"

* signal to display controller

42-381 50 SHEETS 5 SQUARE
42-382 100 SHEETS 5 SQUARE
42-389 200 SHEETS 5 SQUARE
NATIONAL

Red Controller Data to CPU/Memory Timing

2-3-82
John Providenza



NOTES

① The output cycle starts with the falling edge of "ODATLLDENL".

② Latch the data from the display when "ODATLLDENL" is low and a rising edge of "PROCCLK" occurs. A 74LS377 fits this nicely.

③ IF the interface can only accept this byte now, assert "SODRDYH". This tells the display not to send any more data for now. If "SODRDYH" is held low, the display is free to send more data at it's leisure.

④ If "SODRDYH" was asserted in step ③, the rising edge of "ODATSTBH" clears "ODATRDYH" informing the display that it may send another data byte to the interface.

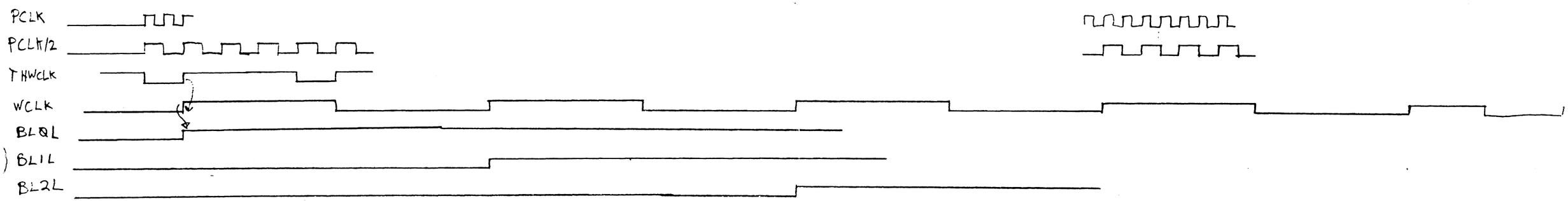
* signal to display controller

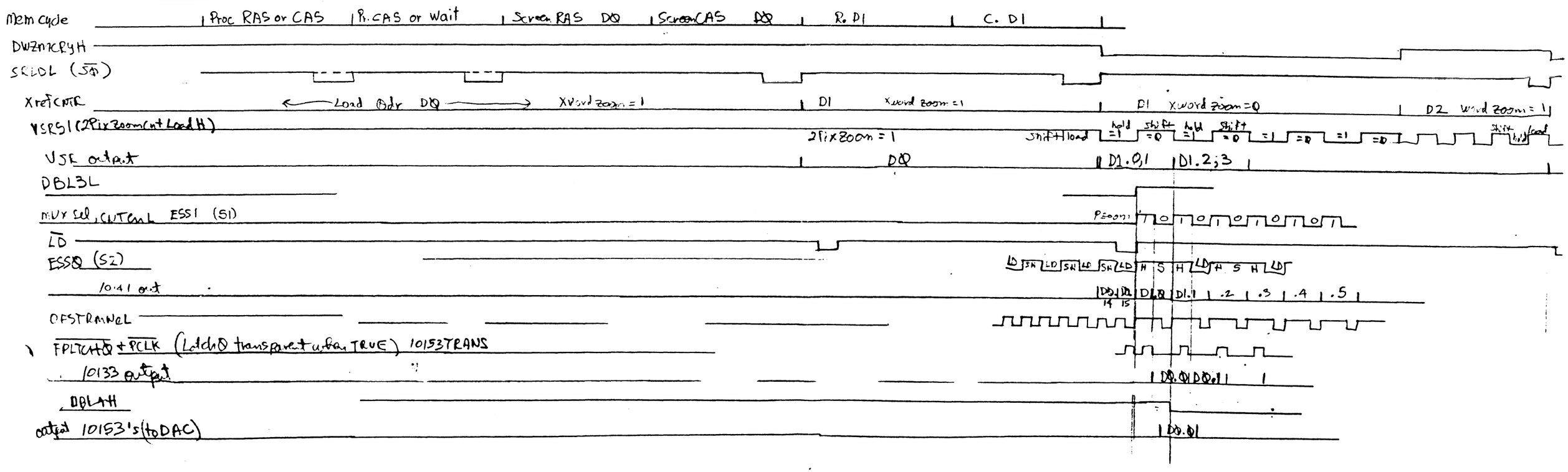
42-381 50 SHEETS 5 SQUARE
42-382 100 SHEETS 5 SQUARE
42-389 200 SHEETS 5 SQUARE
NATIONAL

Horiz Pan + Zoom Timing

RB 8/27/81

period	CLOCKS	delay
28	PCLK	0
56	EPCLK/2	
225	HWCLK	
56ns	PCLK/2	8
225	THWCLK	
450	WCLK	48
	BLQ, 1, 2 L	76
	VBLANKL	116
	BL3	17





8/30/81 Revision