MEMORANDUM
March 9, 1982

TO: Karl Guttag

CC: Kevin McDonough

FROM: Jerry Rogers

SUBJECT: 9918 and 9995 Design Mods for Personal Computer Division

Karl, the Personal Computer Division is looking at cost reducing the 99/4A computer using a 9995 derivative and 9918 with the by four dynamic RAMs. This necessarily implies design mods to both the 9995 and 9918 if not to maintain software compatibility with existing systems, then to modify pin timing for compatibility with the current generation of five old only dynamic RAMS. Please review the attached memos from Personal Computer describing the design mods they would like to see us undertake and give me your assessment as to the magnitude of the task and what kind of a schedule and expense plan we should commit to.

I need this by 3/12.

Jerry Rogers
Jerry Rogers

JR/mm

-MSG M#= 30637 FR=EMOS TO=DESN SENT=03/10/82 05:50 AM
R#=448 ST=C DIV=073 CC=0004 BY=EMOS AT=03/10/82 05:48 AM

| | | |
|---|---|---|
| TO: | JERRY ROGERS | DESN |
| CC: | RON NORRIS | MCPE |
| | JOE YEAGER | MCPE |
| | ROBERTO FANTECHI | FANT |
| | TOM PRESTON | EMOS |
| FR: | DEREK ROSKELL | EMOS |

RE: CUSTOMISED 9995 FOR HOME COMPUTER GROUP

JERRY,
I HAD A TALK WITH MIKE BUNYARD TO CONFIRM HIS REQUIREMENTS FOR A CUSTOM 9995. HIS TWO REQUIREMENTS ARE:

1) MOVE THE INTERNAL MEMORY TO BE LOCATED AT FF00 THRU FFFF AND LEAVE THE DECREMENTER AT LOCATIONS FFFA AND FFFB.

2) MOVE THE FLAG REGISTER OUTSIDE THE 9900 CRU MAP.

THE EASIEST CHANGE TO THE CURRENT PART WOULD BE TO LOCATE THE FLAG REGISTERS AT FEE0 THRU FEFE AND THE MID FLAG AT LOCATION FFDA.

I RECOMMEND WE TAKE THE FOLLOWING ACTIONS

1) MODIFY THE LAYOUT ON THE REV F PART (THE 9995 WITH THE OVERFLOW FIX) SO THAT A SINGLE LEVEL MASK OPTION (PROBABLY METAL) COULD GIVE US THE REV F PART AND THE CUSTOM PART FOR THE HOME COMPUTER GROUP.

2) MODIFY THE 95CR LAYOUT TO GIVE THE SAME OPTION.

THE CONSEQUENCES OF TAKING THESE ACTIONS AS I SEE IT ARE:

1) THE REV F PG RELEASE WILL SLIP BY ABOUT 2 WEEKS TO MID APRIL.

2) THE 95CR PG RELEASE WILL SLIP BY ABOUT 4 WEEKS TO 1ST AUGUST.

3) HOME COMPUTER GROUP WOULD BE CERTAIN OF GETTING PARTS BEFORE THE END OF THE YEAR AND COULD FOLLOW UP WITH 95CR PARTS WHEN THE CR GOES INTO PRODUCTION.

4) IF WE CAN GET 100-150 K DLRS FROM THE HOME COMPUTER GROUP WE CAN COVER OUR COSTS OF DOING THE MODS AND ALSO PAY FOR THE REV F PASS INCLUDING PERMANENT TOOLING.

MEMORANDUM
February 10, 1982

TO: Milton Kuser

COPY TO: Alan Lawson

SUBJECT: 9995 MODS FOR LCP

The following is a list of items that might be easily done to the TMS9995 that will allow us to delete a moderate amount of logic on the LCP, and recover some memory space.

- * Defeat the decoding logic on the internal 252 bytes of RAM based at >F000, and make this space available for outside world use. The top six bytes should function as they currently do.

- * Defeat the decoding logic on the 16 internal CRU flags. It is not necessary to give this space to the outside world, but I would like to have it.

- * Cause the control signal DBIN to stabilize at the same time as does MEMEN*. This task will probably be the toughest to handle, but this will bring the memory timing closer to that of the TMS9900.

These three items will make my life much easier in cutting support logic to the bone in the LCP.

Thank you.

Ouch,

MEMORANDUM
July 7, 1981

TO: Don Bynum<br>Alan Lawson<br>Milton Kuser

TO: Don Bynum
Alan Lawson
Milton Kuser

COPY TO: Jerry Rogers

FROM: Mike Bunyard

SUBJECT: THE 99/4(A) AND THE TMS9995

After spending some time thinking how to use the TMS9995 in the 99/4B, I have drawn some conclusions with respect to the present 99/4(A) that I feel should be documented for future reference. These conclusions revolve around the fact that the TMS9995 does not duplicate the timing of the TMS9900, and are listed as follows.

Please note that I forsee no problems in the 99/4B, as we have the chance to design around the points in question.

## Basic Timing

For a 12 MHz clock the TMS9900 has a state (cycle) time of 2/3us while the TMS9995 has one of 1/3us (which is one of the ways that it gains a considerable speed advantage over the TMS9900), but two 1/3us cycles for either memory or CRU are NOT equivalent to one 2/3us TMS9900 state time. The TMS9900 has a 500ns memory access while the TMS9995 has only a 448 ns memory access due to increased setup times and something else I don't know about. There is no margin in this for 450ns ROMs; thus, faster memories are required.

In Addition, the following items were noted.

* The TMS9995 signal CLKOUT takes the place of PH3* of the TMS9900, and it is twice as wide as PH3*. Many peripherals use both the leading and trailing edge of PH3* for timing, and the additional pulse width leaves less timing margin.

* The control signal DBIN* of the TMS9995 is not timed identical to the DBIN of the TMS9900; DBIN* is now a function of CLKOUT, and becomes valid 167ns later than that for the TMS9900. This implies that there will be a

data bus conflict for this period for our peripherals during which abnormally high supply currents will result.

* Additional logic will be necessary to separate WE* and CRUCLK* with MEMEN* being the separating variable. While this logic is of the inexpensive variety, it still takes up PCB real estate, requires some supply power, and worse yet, adds própogation delay to these signals. The same thing is true for IAQ and HOLDA with HOLD* being the separating variable, but there is uncertainty in doing this. It depends upon when HOLD* is applied. This one is not a worry on the 99/4(A), though.

* CRUCLK goes from 83 1/3ns on the TMS9900 to 167ns on the TMS9995. This should present no problem on the 99/4(A) as I see it at this time.

Use of the TMS9995 Internal RAM

The TMS9995 internal RAM is located in the address space our Memory Expansion responds to, and when the TMS9995 accesses its internal RAM, external memory control signals are asserted. The MEU will respond, but will become very sick when the memory cycle is aborted at the end of 1/3us rather than 2/3us later as the MEU expects it to. This is the real KILLER in that existing product is already in the field that expects 1us memory cycles. We can provide the extra logic required to trap these addresses in the MEU, but again this is not especially simple (or pleasing for that matter).

The TMS9995 memory map has its Timer memory mapped in this manner, and of course it is disjoint from the internal RAM. Trapping all of this out is not my idea of fun when there are cost constraints involved. Even as argumentative as Acker is, he would say ugly things about having to arrange his code around these disjoint memory areas.

I, for the moment, think that the method of refreshing the present MEU would present conflicts with the TMS9995 accessing due to its 1/3us state times. It can get back to the MEU 1/3us earlier than the TMS9900 can.

## CRU Operation

With the exception of the previously mentioned 448ns CRU access, I don't see any problems in this area as the 99/4(A) doesn't decode CRUCLK as a function of the instructions involved....it simply uses only CRU types of instructions.

## Miscellaneous

The TMS9995 gets its MSBY of a word transfer first and the LSBY second. The 99/4(A) does the reverse which allows a settling time for DBIN and A14 as they are used in the VDP, sound chip, and GROMs before allowing the respective chip selects to go true. Extra logic will be required to prevent high speed accesses of these devices. This again amounts to dollars, space, and power.

Mike Bunyard

DESIRED FEATURES FOR NEW 9918-BASED VDP CHIP

Separate read and write pointers for VDP memory access. This will speed up and shorten code needed to copy a block of VDP memory.

```
   Current code                         Proposed code
   ------------                         -------------
   LOOP                                      {set up read address}
      {set up read address}                  {set up write address}
      MOVB @VDPRD,R0                    LOOP
      {set up write address}                 MOVB @VDPRD,@VDPWD
      MOVB R0,@VDPWD                         DEC  R1
      DEC  R1                                JGT  LOOP
      JGT  LOOP
```

Readable VDP registers. Would allow us to write system routines (e.g. automatic sprite motion) that would work for all application programs, not just those with tables in conventional locations. Currently that would require that copies of the registers be kept in fixed locations in RAM.

Readable VDP address. Facilitates subroutining and interrupt handling routines could save & restore address to make operation transparent to main program.
Implementation of (1) would mean that we need a readable read-address and a readable write-address.

Tunable colors. 16 colors at any given time is adequate, but software should be able to specify what those colors are.

---

9995 instructions needed

BLS GD    Branch & link via stack

    PC → *R11+
    GD → PC

RTS    Return via stack

    DECT R11
    *R11 → PC

block mode

```
-MSG M#=  39368   FR=JTTL TO=DESN   SENT=02/23/82 07:12 PM
     R#=260 ST=C DIV=072 CC=3654 BY=JTTL AT=02/23/82 07:12 PM
```

TO:  DESN - JERRY ROGERS
CC:  TJFM - JOHN BRYANT
JLIN - M. SHIBUYA / T. IWATA / S. YAMASHIRO

RE:  TMS9918A DESIGN CHANGE REQUEST

DO YOU HAVE ANY TMS9918A DESIGN CHANGE PLAN?
TMS9918A HAS TMS4116 DRAM INTERFACE CAPABILITY, BUT
DOESN'T HAVE TMS4164 DRAM.
AT NEAR FUTURE, TMS4164 WILL BE MORE POPULAR AND MORE COST
PERFORMANCE PRODUCT THAN TMS4116.
THE FOLLOWING POINTS ARE OUR REQUIREMENT.

1) TMS9918A SHOULD HAVE TMS4164 DRAM INTERFACE CAPABILITY.

2) IF TMS9918A WILL BE ABLE TO USE TMS4164 AS VRAM, PROCESSOR SHQULD BE ABLE TO ACCESS 64K VRAM DIRECTLY.
BECAUSE TMS9918A WILL ONLY USE 16K MEMORY OF 64K VRAM FOR VIDEO DISPLAY. AND THE OTHER MEMORY CAN BE USED AS MAIN PROGRAM MEMORY.

PLS ADVISE US ABOVE SUBJECT.

BEST REGARDS,
SHIGEO OKAMOTO     JLIN    13654     KH

what about 4416