bcc	title THE PHASE 1 PHYSICAL MAP		prefix/class-number.revision P1PM / S-4.1	
checked Ulrich Sponnagel	authors Chuck Thacker Chuck Thacker	approval date	revision date 2/10/70	
checked		classification Specification		
approved		distribution Company Private		pages 12

ABSTRACT and CONTENTS

This document is a description of the Phase I CPU physical map. Interface to the CPU, timing, and implementation are discussed. This document is a complete revision of PHYSICAL MAP SPECIFICATIONS, PMAP/S-4, which is obsolete. Applicable schematics are:

1.)	L - 00310	Map Control Card #1
2.)	L - 00311	Map Control Card #2
3.)	L - 00309	Map register #1 & #2

The physical map is the section of the Model I CPU which translates virtual addresses to physical addresses in main memory. It consists of 4 printed circuit cards: two standard 64 word scratchpad cards and two control cards. In addition to address translation, the physical map also does protection ring checking and provides a path for data to the E1 bus which is used by the CPU for functions not related to the normal operation of the map.

The map communicates with the CPU via the Z, E1, and RØ busses. It is controlled by the CPU through a group of special functions. Some of the special conditions are decoded on the CPU special functions card, some are decoded at the map to save time. Most map operations require either one or two cycles for execution, although map operations may overlap two instructions.

The two basic operations performed by the map unit are address translation and ring checking. The address translation portion of the map consists of the 2 scratchpad cards, arranged as a 128 cell x 8 bit memory, (one cell for each 2K page in the address space of a process) plus the control card #2. Each cell of the scratchpad register contains the physical page number of the virtual page which is the address of that cell. During mapping, bits 6 through 12 of RØ (the virtual address) are used as an address by the scratchpads. The contents of the addressed cell is returned as the physical page number on E1(5) - E1(13). The control card #2 contains 128 3 bit registers: EF (the empty flag which indicates that the entry has not loaded since the map was last cleared), DB (the dirty bit), and PMRO (the read-only bit). These bits are addressed, loaded, and read similarly to the 8 bit registers, with the exception that all 128 EF bits may be set at once by the CPU and by the other processors in the system.

The data returned on E1 by the address translation process is in the following form:

E1 bit number	Significance
Ø	MRCE + EF (ring error or empty)
1	MRCE (ring error)
2	PMRO + $\overline{\text{DB}}$ (read only or not dirty)
3	PMRO (read only)
4	not used
5-12	8 bit physical page number from 8 x 128 memory
13-23	11 bit word address, directly from RØ(13)-RØ(23)

An entry is loaded into the map (by the CPU) by placing a 7 bit virtual page number on RØ(6) - RØ(12), the entry in Z(13) - Z(23), and executing the ZTOMAP special condition.

The bits of Z are loaded into the specified map register as follows:

Z(13) → EF

Z(14) → DB

Z(15) → PMRO

Z(16) - Z(23) → Physical page no.

The MRCE (map ring check error) bit returned in bits Ø & 1 of E1 by the mapping process is the result of the protection ring check performed by the map. The address space of a process is divided into 3 rings:

VIRTUAL ADDRESS	RING
0 - 37777	1: USER RING
403000 - 57777	2: UTILITY RING
400000 - 402777 and 60000 - 777777	3: MONITOR RING

It is an error for an instruction in one ring to access a higher ring.* At the beginning of the address calculation, the CPU loads the SOURCE REGISTER in the map with the address of the instruction. When the CPU maps the operand address, the ring of the source register is compared with the ring of the operand address (in RØ(6) - RØ(14)). If the ring number of RØ is greater than the ring number of the source register, MRCE is set and returned on E1 as described above.

* The full story on protection & rings & so on may be found in M1CPU/S-42, pp 4-10.

The source register is a 9 bit register. It is loaded from RØ(6) - RØ(14) when the MAPSS special condition is executed, or from Z(6) - Z(14) when LSR is executed.

The map contains one additional register, the ADDRESS REGISTER. This is a 7 bit register which is loaded from RØ(6) - RØ(12) (the virtual page number) whenever a mapping operation is done. The address register may be read onto the El bus by the CPU by executing the MAPAD special condition. This register is required since if the CPU detects an error in the mapping process, it must be able to recover the virtual page number which caused the error.

The special conditions which drive the map, their actions and timing are as follows:

1.) READS (MS = 56 B)

The source register is gated onto bits 6- 14 of the El bus. The special condition is decoded at the map unit. The delay from the MS field to the data's presence at El is 4 NAND levels, thus the execution of this instruction requires only one cycle.

2.) MAPAD (MS = 54B)

The address register is gated to bits 6 - 12 of El. Delay from MS to El is 4 levels. Execution requires one cycle.

3.) ZTOMAP (MS = 53B)

This special condition loads the map register specified by RØ(6) - RØ(12) from bits 13-23 of Z. The instruction of which it is a part must be at least 2 cycles long. The data on Z(13) - Z(23) is written into the map register during the second cycle of the instruction. The strobe which writes into the map is generated in a flip flop, MWS. It is one cycle less one clock width in length:

set MWS = STATEA·ZTOMAP.MK3

clear MWS = MWS·MK2

MK3 and MK2 are generated from the CPU's UKl on the Control #1 card. MK2 exists from t = 80-100 nsec., MK3 from t = Ø to 20 nsec both relative to a system clock interval. The signal ZTOMAP' is generated on the CPU special function card.

4.) LSR (MS = 34B and MS = 52B)

The source register is loaded from bits 6- 14 of Z. The instruction must be at least one cycle long. The clock which loads the register is generated from the signal LSR', which is decoded at the special function card. This signal also gates data from the Z bus to the source register:

SR Clock = LSR·MK2

Gate data from Z to SR = LSR

5.) COMAP (MS = 74B)

This instruction is executed by the CPU to clear its own map. The instruction length is arbitrary (1,2, or 3 cycles). Other processors clear the map by the same mechanism used by this special function. Clearing of the map requires 4 cycles. Any attempt to load the map during the time it is being cleared will be ineffective, and any attempt to map during this time will result in a returned entry with ER = 1.

The reason clearing requires 4 cycles is that drive limitations on the I.C. with which EF is implemented prohibit writing into more than 4 of the 16 cells simultaneously. Clearing is done by addressing one fourth of the cells in the 8 I.C.'s constituting EF during 4 successive intervals, writing a 1 during each cycle. The normal address path is disabled during this time, to ensure that only 4 cells rather than 8 are addressed at any time, which could otherwise be the case if the CPU attempted to map while clearing was in progress.

6.) MAP (MS = 37B)
MAPSS (MS = 51B)
MAPFETCH (MS = 33B)

These 3 special conditions map the virtual address on RØ into a physical address on E1 as described earlier. They also do ring checking between RØ and the source register. MAPSS also loads the source register from bits 6 - 14 of RØ at the end of the first cycle of the instruction. MAPFETCH starts a fetch to central memory at the end of the instruction which includes the special condition (this activity is implemented on the special function card, not on the map).

Timing for these instructions is shown in fig. 1. The flip flop GO is the signal which gates the map entry to the E1 bus. It is set at the end of the first cycle during which the special condition exists:

$$\text{set GO} = (\text{MAP} + \text{MAPSS} + \text{MAPFETCH}) \cdot \overline{\text{GO}} \cdot \text{MK2}$$

The GO flip flop is cleared at the end of the instruction which includes TE1Y. This is usually the instruction which includes the special condition, but overlapping is possible:

$$\text{clear GO} = \text{GO} \cdot \text{TE1Y} \cdot (\text{I2CLOCKA} + \text{I2CLOCKB})$$

TE1Y, I2CLOCKA, and I2CLOCKB are a gating signal and clocks generated in the CPU. I2CLOCKA or I2CLOCKB occurs at the end of every instruction (rather than at the end of every cycle).

Ring checking is implemented by continuously comparing the ring of the source register with the ring of RØ, and generating a signal (BADRING) if a violation is detected. At the end of every cycle during which GO = Ø, BADRING is copied into the flip flop MRCE (map ring check error). The copying is inhibited while GO exists, so that the data presented to the CPU cannot change. A flip flop is required since if the operation being done is a MAPSS, the source register is loaded from RØ(6) - RØ(14) at each MK2 time as long as the MAPSS special conditions exists.

Fields and registers used in the map are summarized in Fig. 2.

The physical map has several other features which are not normally used by which are included for completeness:

1.) One-for-one mapping:

The control card #1 has a two position switch marked 'NORMAL' and 'ONE-FOR-ONE'. The one-for-one position is a debugging aid which causes the map to return RØ(5) - RØ(23) to the CPU on E1(5) - E1(23) whenever a mapping operation is done. E1(Ø) - E1(4) are Ø when this feature is used. Note that this allows the CPU to supply a 19 bit address on RØ, which is one more bit than normal.

2.) Gating to E1:

The map control card #1 has two sets of inputs, (MONE - MSIX, GE1(14)', and GE1(20') - GE1(23)') which are used by the CPU's special functions to pass data to E1. GE1(14)' - GE1(23)' are passed directly onto E1, MONE - MSIX are encoded into their octal equivalents before being put on the bus.

3.) Expansion capability:

In order to allow references by an (as yet unspecified) IPU, the scratchpad cards and the control card #2 have the capability of being addressed by a second source (other than RØ). On the scratchpad cards, this source is selected by using the SPFZ special condition decode, and additional gating on KS'. On the control card 2, the signals IPREF and EUREF will be used to select the source of the address.

When the IPU is specified, the control card #1 will have to be augmented or redesigned to include map request priority unsnarling, etc.

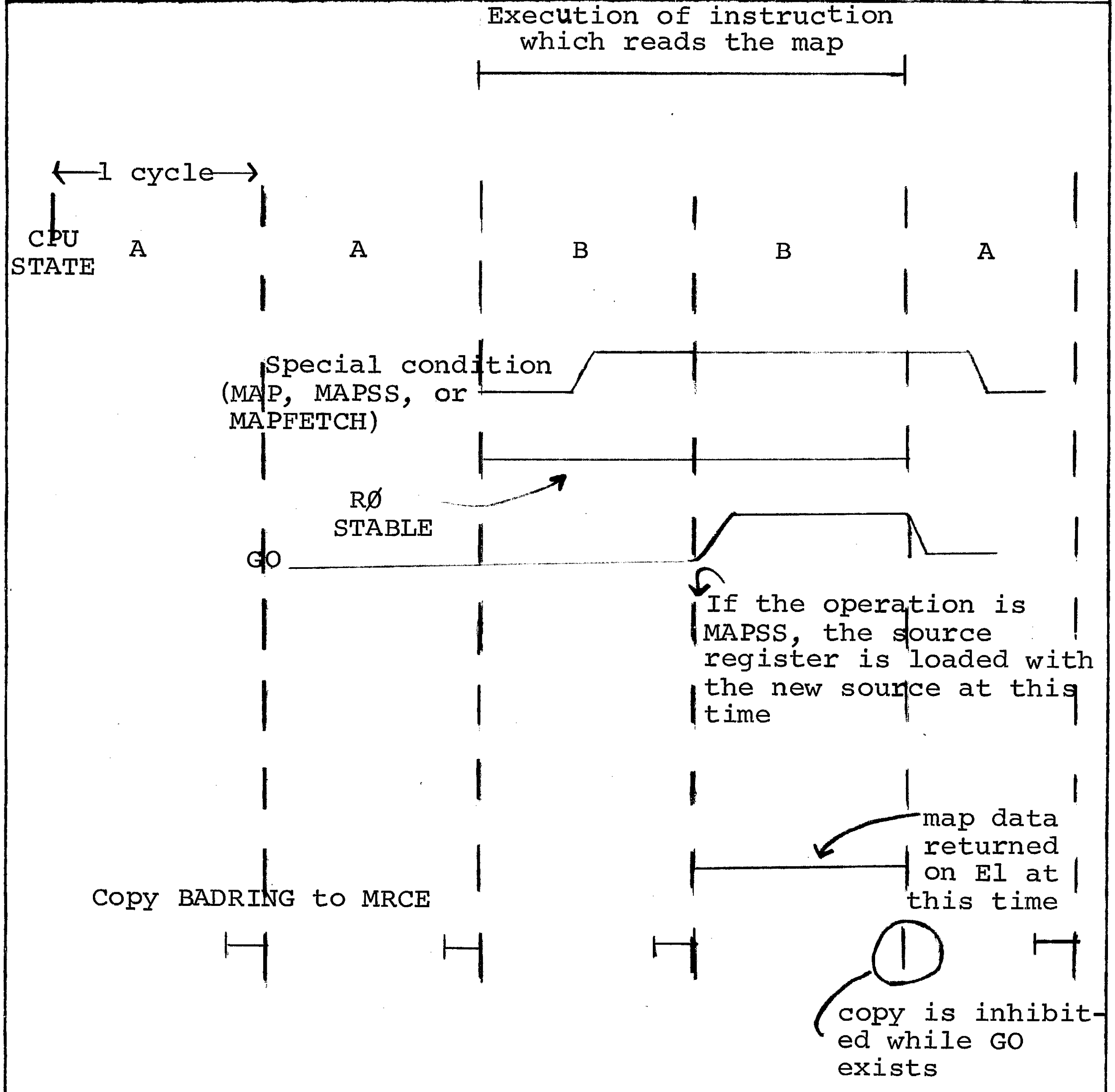
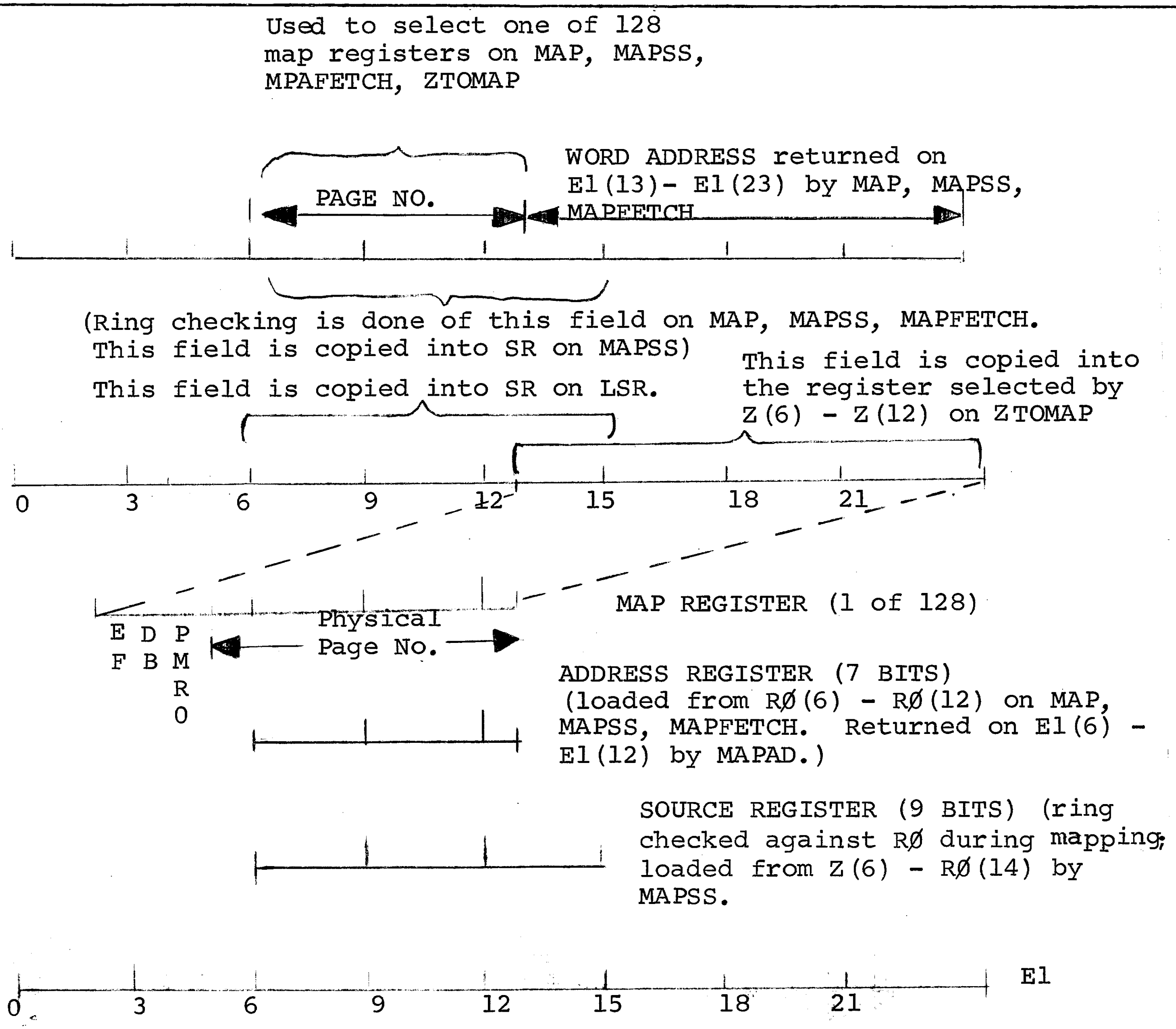

Execution of instruction which reads the map
1 cycle
CPU STATE
A
A
B
B
A
Special condition (MAP, MAPSS, or MAPFETCH)
RØ STABLE
GO
If the operation is MAPSS, the source register is loaded with the new source at this time
map data returned on El at this time
Copy BADRING to MRCE
copy is inhibited while GO exists

FIGURE 1 Map Timing



Used to select one of 128 map registers on MAP, MAPSS, MPAFETCH, ZTOMAP

WORD ADDRESS returned on E1(13)- E1(23) by MAP, MAPSS, MAPFETCH

(Ring checking is done of this field on MAP, MAPSS, MAPFETCH. This field is copied into SR on MAPSS)

This field is copied into SR on LSR.

This field is copied into the register selected by Z(6) - Z(12) on ZTOMAP

MAP REGISTER (1 of 128)

ADDRESS REGISTER (7 BITS) (loaded from RØ(6) - RØ(12) on MAP, MAPSS, MAPFETCH. Returned on E1(6) - E1(12) by MAPAD.)

SOURCE REGISTER (9 BITS) (ring checked against RØ during mapping; loaded from Z(6) - RØ(14) by MAPSS.