APPENDIX C

PROJECT: Z8000

REPORT TITLE: Proposal for a New MMU Chip

AUTHOR: Madeline Shaw

DATE 3/30/83

DISTRIBUTION

Shiraz Shivji
Robert Russell
Frank Hughes
Pete Bowman

ACTION: For information and comments only.

1. Introduction

Need for the support of non-segmented Z8000 execution under a UNIX-like environment has lead to the use of three Z8010 memory management units on Zilog's ZEUS cpu boards. The intent of this paper is to propose a new mmu chip that can replace these three mmus and the additional logic needed to control them.

The nonsegmented software created under UNIX is divided into 3 logically distinct sections: code, data, and stack. A code section may occupy as much as 64K bytes and the combined size of the data and stack section may also be a maximum of 64K. Or, the sum of all three sections may be restricted to 64K.

The hardware difficulties arise in supporting the stack and in having the code and data areas in two different physical sections. Both problems are from the segment lines of the Z8001 being fixed during the execution of nonsegmented code; only one segment number is presented to the mmu for three logically separate segments. Unfortunately, the Z8010 mmu cannot be used to describe segments with both an upward growing heap and a downward growing stack. Nor can three distinct sections using the same logical segment number be supported by a single Z8010 segment descriptor.

By adding additional mmu registers and opcodes and using section descriptors instead of segment descriptors, the original design of the Z8010 can be modified to acheive the desired effect. These changes are briefly summarized below:

- o Section Address Register (SAR).
- o Segment Mode Register (SMR).
- o Three section descriptors per segment.
- o Elimination of the multiple segment table mmu configuration based upon the N/S- line of the cpu.
- o Support for 16 segments per mmu.
- o A modified "upper range select" feature to allow a maximum of 4 mmus to be used to support segments 0 through 63.

The discussion in the remaining pages presents further details and is organized to be read with the Z8010 MMU Technical Manual.

2. Pinout

The pinout shall be identical to the Z8010.

3. Registers and Flags

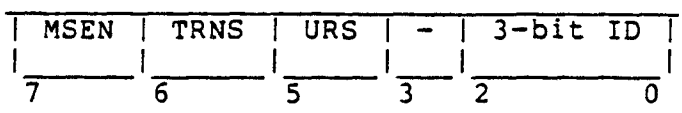
A. Control Registers

1: (SAR) Segment Address Register. 6 bits.

2: (DSCR) Descriptor Selection Counter. 2 bits.

3: (SCR) Section Address Register. 2 bits.

4: Mode register with bits. 8 bits.



Bit 3, originally NMS, has no meaning here and accept any value.

Bits 4 and 5, formerly URS and MST, are used as a two-bit upper range select to specify what range of segment numbers shall be recognized by the mmu. Note, address translation is not enabled and the mmu outputs are tri-stated when the segment lines do not match the selected range.

URS = 00 => segments 00H-0fH
= 01 => segments 10H-1fH
= 10 => segments 20H-2fH
= 11 => segments 30H-3fH

5: (MLR) Segment Mode Register (low). 8 bits.

Bit 0 represents segment 0, bit 1 represents segment 1, and so forth through segment 7. The bit determines the number of distinct sections in the segment. If the bit is clear, the corresponding segment has only one section. If the bit is set, the corresponding segment has three sections.

~~Descriptor data in all three mmu sections of a single section segment are identical. A write to MLR which changes a segment from multi-section to~~

~~single-section causes section descriptor 0 data to be duplicated in the descriptors of section 1 and 2 of that segment.~~ A write to MLR which changes a segment from single-section to multi-section and vice versa leaves descriptor data unaltered. Of course, MLR bits that are not toggled by writes do not affect descriptors. (*) SEE NOTE on page 5.

6: (MHR) Segment Mode Register (high). 8 bits.

Bit 0 represents segment 8, bit 1 represents segment 9, and so forth through segment 15. MHR functions similarly to MLR.

B. Status Registers

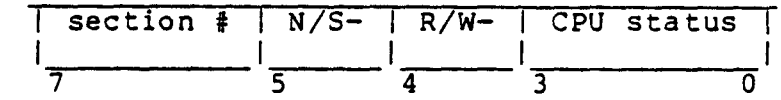
1: Violation Segment Number. 8 bits.

2: Violation Offset (high byte). 8 bits.

3: Instruction Segment Number. 8 bits.

4: Instruction Offset (high byte). 8 bits.

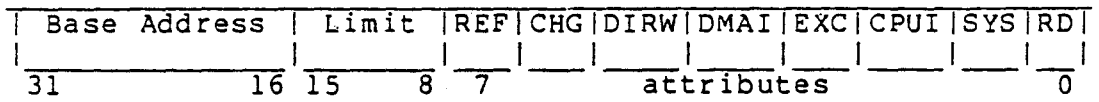
5: Bus Cycle Status + Section Number. 8 bits.



The section number is the section in use when the violation occured.

6: Violation Type Register. 8 bits.

C. Section Descriptor Registers



- o Attributes are identical to the Z8010 attribute bits.
- o DIRW is automatically 1 for section 2 (stack) of a multi-section segment.
- o DIRW is automatically 0 for section 0 (code) and section 1 (data) of a multi-section segment.

- o DIRW is programmable form single section segments.
- o The limit field of section 1 descriptors is used as the "break" register of the ZEUS 3-mmu configuration.

(*) Note: Although changing a segment from single to multi-section or from multi-section to single section leaves section descriptor data unchanged. However, in order for single section segments to work properly, all section descriptors must contain identical data. This can be accomplished by a single write to any of the sections of the "single" section segment.

4. Address Translation

4.1. Conditions

Address translation is performed when several conditions are met in the order listed below. This list is somewhat simpler than the Z8010 conditions.

1: The address is generated during a memory access. I.e., the status lines ST0 - ST3 are 1000, 1001, 1010, 1011, 1100, or 1101. Otherwise, A0 - A23 are tri-stated.

2: The MSEN flag is enabled in the mode register. Otherwise, A0 - A23 are tri-stated.

3: The TRNS flag in the mode register is set. Otherwise, A0 - A23 are tri-stated.

4: A0 - A23 are tri-stated if segment line SN6 is not zero.

5: A0 - A23 are tri-stated if segment lines SN0 - SN5 do not match the range of segment numbers selected by the URS bits.

4.2. Address Translation Process

Once translation conditions are met, bits SN0-SN3 of the segment lines are used to specify a segment. The status lines ST0-ST3 are used to choose a section. If ST0-ST3 are 11XX, an instruction fetch is intended and causes section0 to be selected. Otherwise, section 1 or section 2 is picked according to the following rules in the order of importance listed as shown.

1: Select section2 when the CPUI bit of section1's attribute field is set.

2: Select section1 when doing a DMA access.

3: Compare the high order byte of the logical offset to the limit field of section1. If greater than the limit field, section2 is selected.

The method of selecting a segment's section descriptor

can be made the same for multi-section and single-section segments. This is possible because all sections of a single section segment contain identical values in each field.

Once the section descriptor is determined, address translation proceeds similarly to the Z8010.

5. Opcodes and Operations

The opcode number is listed first followed by a brief description of the operation. Starred (*) items indicate new opcodes.

5.1. Segment and Section Descriptor Registers

(1) 08: R/W BASE field of section descriptor.
09: R/W LIMIT field of section descriptor.
0A: R/W ATTRIBUTE field of section descriptor.
0B: R/W descriptor, all fields.

(2) 0C: R/W BASE field of descriptor. Increment SAR.
0D: R/W LIMIT field of descriptor. Increment SAR.
0E: R/W ATTRIBUTE field of descriptor. Increment SAR.
0F: R/W descriptor, all fields. Increment SAR.

(3) *18: R/W BASE field of descriptor. Increment SCR.
* 19: R/W LIMIT field of descriptor. Increment SCR.
* 1A: R/W ATTRIBUTE field of descriptor. Increment SCR.
* 1B: R/W descriptor, all fields. Increment SCR.

(4) *1C: R/W BASE field of descriptor. Increment SCR, SAR.
* 1D: R/W LIMIT field of descriptor. Increment SCR, SAR.
* 1E: R/W ATTRIBUTE field of descriptor. Increment SCR, SAR.
* 1F: R/W descriptor, all fields. Increment SCR, SAR.

Note, the current value of SCR and SAR select the particular descriptor. In groups (3) and (4), where SCR is incremented after the read or the write, SCR is reset to 0 if it is 2 prior to the operation. In group (4), SAR is not incremented unless SCR is 2 prior to the operation.

Writing to any section field of a single section segment causes the same fields of the other two sections to be written with that value.

5.2. Control Registers

00: R/W MODE register
01: R/W Segment Address Register (SAR).
20: R/W Descriptor Selector Counter (DSCR).
* 10: R/W Section Address Register (SCR).
* 12: R/W Low Segment Mode Register (MLR).
* 21: R/W High Segment Mode Register (MHR).

Note, an attempt to write 3 to SCR leaves SCR unchanged.

5.3. Status Registers

02: Read Violation Type Register (VTR).
03: Read Violation Segment Register.
04: Read Violation Offset (high) Register.
05: Read Bus Cycle + Violation Section Register.
06: Read Instruction Segment Register.
07: Read Intsruction Offset (high) Register.

5.4. Set/Reset Commands

15: Set all CPU-Inhibit flags of all sections.
16: Set all DMA-Inhibit flags of all sections.
11: Reset VTR.
13: Reset SWW flag in VTR.
14: Reset FATL Flag in VTR.

5.5. Reserved

22-FF: Not assigned.

6. Internal MMU Error States

The proposed mmu will have the same internal mmu error states as the Z8010 as described in Appendix A of the Z8010 MMU Technical Manual.