bcc	title: Micro-Processor Parity Checking Logic	prefix/class-number.revision: MPPC/S-14
checked: Mel	authors: Chuck Thacker	approval date: 9-5-69 / revision date:
checked:		classification: Specification
approved: J. T. C.		distribution: Company Private / pages: 3

ABSTRACT and CONTENTS

This document describes the parity check circuitry used in the BCC microprocessor.

Microprocessor parity card

For purposes of parity generation and checking, the data (from the Microprocessor M register) is broken into two fields, composed of bits Ø-11 and bits 12-23. The parity generator examines these two fields, and outputs a 1 if the field contains an even number of 1's.

When a store is done, the two parity bits are gated to the central memory with the GDM signal from the memory interface. The two parity bits are sent directly to the local memory without gating, as is data. Since data resides in the microprocessor M register for at least 1 cycle before being gated to central memory, or being strobed into the private memory's data register, the parity generator has 100 nsec. to become stable before its output will be used.

When a fetch is done, the two parity bits from memory are stored in two latches at the time the M register is loaded with data. There are four latches, two for the parity bits from central memory, two for the parity bits from private memory. There are two parity error latches, one for errors in central memory references, one for private memory errors.

The parity error latch for private memory will be set 100 nsec after MCLOCK2 occurs if there is a parity error. The parity error latch for central memory will be set 100 nsec after MCLOCK1 occurs, providing the memory interface is not in state 1 or state 2. The reason for this restriction is that MCLOCK1 is generated whether the request is satisfied or not, and it would be inappropriate to call the lack of a transfer a parity error.

The latches on the parity card may be tested by two branch conditions, one to test for local memory parity errors, one for central memory parity errors. The parity error latches are reset at the end of every branch instruction. The outputs of the error latches are routed to the system warning registers, where they are sampled at the end of every interval. Care must be taken when using the branch conditions on the parity error latches, since the latches are not set on an error for 1 cycle after the memory has stopped.

An instruction sequence like:

```
FETCH
GO TO ZILCH IF PARITY ERROR
```

checks the parity of transfers which occurred prior to the fetch, since the memory is still running when the test is done. To check the parity of an isolated reference, the following works:

```
FETCH
RØ→RØ   (or some other instruction which will
         stop execution until the memory stops)
NOP     (the parity error latch will be set at
         the end of this instruction.)
GO TO ZILCH IF BAD PARITY
```

Odd address parity is generated over 17 bits of the RØ register (6-22). In most cases, the address resides in RØ for 1 cycle before being sent to the memory (while the microprocessor is making a request to the MPMBM), but in the case of the CPU, this is not necessarily the case. Address parity will be stable 60 nsec after the data in RØ is stable. This implies that in the worst case, address parity will arrive at the memory 60 nsec after the address and the request field arrive.