~~OR OFFICIAL USE ONLY~~

13 May 59

HARVEST

MEMO FOR THE RECORD

SUBJECT: Trip to IBM, 6 May 1959

NSA:	IBM:
T. Lane	J. Pomerene
W. Chadwell	M. Every
	L. Ulfsparre

<u>High Speed Memory features</u>

The high speed memory feature status was reviewed by Mr. Every. It is his contention that he can produce a high speed memory capable of performing an 8 bit count in Memory in less than 0.75 u-seconds provided the check bit updating is simple parity. If simple parity updating for features is used there are two ways to implement parity for the high speed memory. These are:

1) Simple parity all the time

   a) There will be no single error correction on data coming from high speed memory

   b) The C/G will not be utilized as extensively so Store conflicts on the bus might be reduced.

2) Simple parity when features are used and ECC parity when in the normal mode.

While the second method is very desirable from an operational viewpoint it implies extensive control problems for both the memory bus and the high speed memory area. This method does not seem practical at this time.

Simple parity as a substitute for ECC parity in the high speed memory was accepted on the assurance of Mr. Every that he will produce a 0.75 high speed memory (includes at least one 8 bit count operation). He further contends that the requirement of ECC in the high speed memory area would lead to a slower memory access time while in the feature mode by a minimum of 0.25 u-seconds. In the event that the drift of the matrix core during READOUT does require an additional read-write cycle one can expect a memory access time of 1.25 u-seconds or 0.50 additional delay if ECC parity is used. Mr. Every stated that he and his group had been doing their work under the assumption of simple parity updating in the high speed memory. It was not established that previous work would be invalidated if the ECC were in fact required.

[It might be desirable to include an 8 bit parity check circuit for the data field that is to be updated. This was impossible with an ECC parity because it required too much hardware. If all the words are read out to the bus there would be no need for this check, but if any problems can be visualized that requires no readout the check could be very desirable..] ????

The simple parity on high speed memory should save approximately 2000 transistors on the fast memory frame. Space in this area is critical because in order to keep the delays in the system to a minimum the whole package must be kept small.

~~FOR OFFICIAL USE ONLY~~

W. Chadwell

DRAFT

MEMO FOR THE RECORD

SUBJECT: Removal of Error Checking and Correction circuits from the High Speed Memory in the HARVEST system.

On May 6, 1959 representatives of ~~Aneq~~ ANEQ 1 on the Harvest contract met with IBM representatives at Poughkeepsie N.Y. The object of the meeting was to reach agreement on the special Memory Features to be included in the High Speed Memory for HARVEST. During the conversation the ANEQ representative agreed to drop the ECC requirement on this memory. The purpose of ~~the~~ this Memo is to cite the basis for the MPRO-35 representatives' objection to this decision.

Mr. M.~~A~~Every of IBM is the engineer in charge of the production of the H.S. memory at Poughkeepsie. He presented the following argument for justifying the removal of the ECC requirement.

1.) He cannot build a H.S. memory which includes the count in memory feature and ECC in the .75 usec cycle time required by NSA. (The point is valid only if he restricts himself to the standard HARVEST Logic circuits and IBM drift transistors.)

2.) He can give us a H.S. Memory in which "count ...." is available within the .75~~USEC~~usec cycle time.

3.) He can include a simple parity checking feature in the memory, effective on an 8 bit byte width, which will not extend the memory cycle time.

4.) He contends that the value of E66 is doubtful anyway when the Memory Features are being used. You do not perform ECC on the word extracted to count into, you could have an error here. You count into the word and change the data section of the word without changing the parity bits so parity doesnt match data anymore. You finally extract the count and use it thus doing ECC and correcting the data to match the parity thus possibly losing the count. The addition of 150-250 musecs required to perform the ECC logic effectively during this chain of events would stretch the memory cycle time beyond the point acceptable to NSA,i.e. .75 usecs.

My objections to these aggguments follow.

1.) There is nothing sacred about IBM transistors and HARVEST Logic circuits. The company has research experience which tells them that there are faster transistors available and the gain in speed these devices gives them could make ECC in H.S.Memory possible.

2.) If he can build a memory with "Count" available in the .75 USEC then a memory with "Count" and ECC in 1.0 usec is possible and a memory with ECC and no "Count" in .75 usec is possible. (Here I can add that the Memory Buss Slot time now pegged at .25 usecs ~~will~~ MAY eventually be .3 usecs. If this happens an additional 150 musecs is available to perform ECC and we would have a memory with a cycle time of .9 usecs and both "Count and ECC. A thorough investigation of faster, non-Harvest circuitry for the logic of memory would prove or disprove this point.)

3.) The inclusion of Simple Parity in the H.S.Memory while better than no checking doesn't help much f~~or~~rom the operational view. An error will be detected but the vital importance of the H.S.Memory in programs on HARVEST will make the lack of the H.S.Memory very painful and will severly inhibit the choice of alternate programs by the supervisory program. In short a program on HARVEST which doesn't use the H.S.Memory will be rare. At present the "Count" feature is only available in H.S.Memory although ~~Aneq~~ ANEQ 1 has proposed that it be made available for Main Memory also. Therefore with automatic single error correction a consistent single error would be corrected and the program would continue to run. This would prevent the loss of machine time expended on the program until the error occured. RE-RUN time could be decreased.

Servicing of this memory will not be easy and ECC would make it possible to ~~plan~~ PLAN the maintenance effort much more efficiently so that the servicing or repair could be carried out at the best time, when engineers and special equipment ~~is~~ ARE available. Simple parity simply says an error is occuring and needs to be fixed right now before the opertions program can proceed, and if no program using main memory exclusively is available then operations must stop until repairs to fast memory are complete.

4.) His contention that the value of ECC when features are being used is doubtful, is well taken, but only if you accept his contention that HARVEST logic circuits and IBM drift transistors are the fastest available. With faster circuits it is entirely possible to do an ECC check/generate at each step in the count preceedure. In this way all single errors would be corrected (at a cost of ~~more~~ ADDITIONAL time) and all double errors would be detected more reliably than simple partity can guarantee.

In conclusion I would like to say that I feel that the removal of the ECC requirement at this time was premature. The agency is paying for developmemt costs on the memory features. The cost is in addition to the fixed cost of the memory. The advantages of EDD in this memory are abvious from the above. The extremely difficult engineering job that the building of this memory represents should not inhibit the agency from insisting on the insurance that ECC gives us. The memory promises to be the most critical segment of the system and therefore ECC on the memory would be most valuable. Lack of ECC on the memory will prove a costly compromise both during acceptanc periods and in the long run costs of operations. The overal reliability of the system is reduced by it absence.