

# ECC 기능을 갖춘 IEEE-796 버스용 메모리 시스템 설계에 관한 연구

## (The Study on a Main Memory System with ECC Capability Based on IEEE-796 Bus)

朴炳寬\*, 方勝楊\*\*

(Byung Kwan Park and Sung Yang Bang)

### 要約

본 논문은 작년도 국책 과제의 하나인 "16 bit 마이크로컴퓨터 개발"에서 개발된 16 bit 마이크로컴퓨터를 위한 주기억 시스템 개발에 관하여 기술한다. 이 주기억 시스템은 ECC 기능을 갖고 있으며 IEEE-796 bus를 이용했다. 설계과정에서는 top-down 방식을 따랐으며 시험과정에서는 bottom-up 방식을 따랐다. 이렇게 해서 개발된 기억 보오드는 미국에서 시판되고 있는 것들보다도 좋은 성능을 발휘했으며 가격도 저렴하다.

### Abstract

The paper reports the development of a main memory system with ECC capability based on IEEE 796 bus for a 16 bit microcomputer the development of which was one of the last year's National projects. The top-down approach of the design and the bottom-up approach of the testing resulted in a memory board which demonstrates a better performance and a less expensive cost than those available in the U.S. market.

### I. 序論

컴퓨터의 메인 메모리는 코어 메모리를 시작으로 오늘날에는 반도체 기술의 발전으로 chip 1개당 64K bit의 기억용량을 갖기에 이르렀다. 메인 메모리의 특성은 적은 용량(수십 K byte)을 갖고 액세스 타임이 빠르다.

최근들어 메모리의 값(bit/cost)이 내려가고 사용자의 메모리의 요구가 늘어감에 따라 작은 시스템에도

큰 메인 메모리를 사용하는 추세이다. 특히 마이크로컴퓨터에서 증가추세가 현저한데 IEEE-796 bus를 사용하는 마이크로컴퓨터의 경우, 512K byte 이상을 사용하는 경우가 보통이다. 메모리 용량의 증가와 칩내부의 집적도가 높아짐에 따라 메모리 에러의 발생율이 증가하여 error protection이 심각하게 고려되기 시작했다.

메모리 에러는 크게 하드 에러와 소프트 에러로 구별된다. 하드 에러는 메모리 칩내의 셀의 고장으로 인한 영구적인 에러를 말하며 소프트 에러는 외부의 전기적 혹은 자기적인 간섭에 의하여 발생하기도 하며, 또 칩집적도가 증가함에 따라 하나의 셀이 포함하는 전하수가 적어져서  $\alpha$ -particle에 의한 전하의 유도때문에 발생한다. 메모리 에러의 protection을 위하여 parity와 hamming의 ECC(Error Check & Correct)

\*準會員, \*\*正會員, 韓國電子技術研究所  
(Korea Institute of Electronics Technology)

接受日次: 1983年 6月 18日

(※本 연구는 科學技術處 '82年度 국책과제 CN8252  
로 이루어졌음.)

코드를 사용한다.<sup>[1],[10]</sup>

각 방법을 비교한 것은 표1과 같다.

집적도가 높은 RAM을 사용하고 저장용량을 크게 하기 위해 많은 RAM을 사용하게 됨에 따라 single bit error는 정정할 수 있는 ECC 회로를 대형 컴퓨터뿐만 아니라 마이크로컴퓨터의 메인 메모리에도 최근들어 많이 이용하기 시작했다.

본 논문에서는 IEEE-796 bus용 512K byte ECC 메모리 시스템 설계에 관한 연구에 대해 기술한다. 연구과정은 국책 과제중의 하나인 "16 bit 컴퓨터 시스템 개발" 프로젝트중에 수행되었으며 설계 결과 만들어진 메모리 시스템은 16 bit 컴퓨터의 메인 메모리에 사용되었다.

표 1. Error protection 방법의 비교

Table 1. Comparison of error protections.

	1 bit error	2 bit error	2이상 error
Error protection 이 없는 경우	system crash	system crash	system crash
Parity 방법	system halt	system crash	system crash
ECC (single bit 정정과 double bit 검출)	No effect	system halt	system crash

표 2. 설계사양

Table 2. Design specification.

Design specification
- Bus : IEEE-796 bus
- 기억용량 : 512K byte
- Access time :
read 나 byte write 일 경우 : 300-350 ns
word write 일 경우 : 150-200 ns
- Word size : 16 bit & 8 bit
- 64K bit 150nsec dynamic RAM 사용

표 3. 체크 비트 인코딩 식

Table 3. Check bit encoding equation.

$$\begin{aligned}
 C0 &= D1 \oplus D2 \oplus D3 \oplus D5 \oplus D8 \oplus D9 \oplus D11 \oplus D14 \\
 C1 &= D0 \oplus D1 \oplus D2 \oplus D4 \oplus D6 \oplus D8 \oplus D10 \oplus D12 \\
 C2 &= D0 \oplus D3 \oplus D4 \oplus D7 \oplus D9 \oplus D10 \oplus D13 \oplus D15 \\
 C3 &= D0 \oplus D1 \oplus D5 \oplus D6 \oplus D7 \oplus D11 \oplus D12 \oplus D13 \\
 C4 &= D2 \oplus D3 \oplus D4 \oplus D5 \oplus D6 \oplus D7 \oplus D14 \oplus D15 \\
 C5 &= D8 \oplus D9 \oplus D10 \oplus D11 \oplus D12 \oplus D13 \oplus D14 \oplus D15
 \end{aligned}$$

표 4. 신드롬 디코딩 표

Table 4. Syndrome decode to bit-in-error.

Syndrome	S5	0	1	0	1	0	1	0	1	
Bits	S4	0	0	1	1	0	0	1	1	
S0	S1	S2	S3	0	0	0	0	1	1	1
0	0	0		* C5	C4	T	C2	T	T	M
0	0	1		C2	T	T	15	T	13	7
0	1	0		C1	T	T	M	T	12	6
0	1	1		T	10	4	T	0	T	T
1	0	0		C0	T	T	14	T	11	5
1	0	1		T	9	3	T	M	T	T
1	1	0		T	8	2	T	1	T	T
1	1	1		M	T	T	M	T	M	T

\* Error 가 없을 때

숫자 : Data error의 위치

T : Double bit error

M : 셋이상의 error

C숫자 : Check bit error의 위치

## II. 설계 및 시험

설계는 기본적으로 top-down 방법으로 진행했다. 보오드의 기능을 분석하여 설계 specification을 만들었으며 specification을 기초로 하여 기능별로 tree 구조의 블럭들을 만들었다. 가능한한 블럭들을 게이트 레벨로 변환할 수 있을 정도로 세분화하여 가장 밑의 레벨부터 회로를 만들어 갔다. 각 블럭의 회로가 완성 되면 블럭간의 인터페이스를 고려, 종합하여 전체의 회로도들을 만들었다. 설계 단계에서 생길 수 있는 bug는 review 과정\*을 거쳐서 찾아 내었다.

설계하는 메모리 시스템은 16 bit 컴퓨터 시스템에 메인 메모리로 사용되어야 하기 때문에 다음의 요구조건을 만족해야 한다.

- 시스템 bus인 IEEE-796 bus용
- Multi-user 시스템 (UNIX) 이기 때문에 보오드 기억용량은 가능한한 큰 용량(512K byte)
- 68000의 wait cycle을 줄이기 위한 fast access time : 500 ns 미만

위의 조건들을 만족시키기 위하여 설계 단계에서 고려된 사항은 다음과 같다.

- IEEE-796 bus specification<sup>21</sup>
- 기억용량을 크게 하기 위하여 density가 가장 높은 64K bit dynamic RAM을 사용하여 512K byte로 하였다.

- 기억용량이 커짐에 따라 메모리 에러의 protection을 위해 ECC 회로사용
- 보오드내의 칩의 수효가 많음에 따라 가능한한 LSI와 PLA(programmable logic array) 등을 사용했다.
- 다이내믹 RAM을 사용하므로 리후레쉬등 다이내믹 RAM을 위한 로직이 필요했다.

회로의 설계를 완성한 후, 회로를 증명하기 위하여 P. C. B(printed circuit board)를 만들었다. 조립과 시험은 설계때와 반대로 bottom-up approach를 이용했다.

설계때 만들어진 각 블록을 밑에서 부터 조립하여 각각의 기능시험을 하고 같은 레벨의 블록들을 연결하여 그위 레벨의 블록을 완성, 기능시험을 했다. 이와 같은 방법으로 전체 보오드를 완성했다. 71

III. 설계 결과

설계 결과 만들어진 메모리 시스템의 블록 다이어그램, operational flowchart와 design specification은 그림 1, 2, 표 2와 같다.

IEEE-796 bus의 메모리 오퍼레이션 사이클은 버스 마스터가 어세스하고자 하는 slave(memory)의 어드레스를 버스상에 보내고 커맨드 신호(MRDC/or MWTC/)를 주면 slave에서 XACK/로 동작의 완료를 알리는 것으로 끝이 난다. 버스 마스터의 결정은 priority가 정해진 마스터들간에 arbitration 과정에서 정해지며 버스 마스터가 버스를 제어하게 된다.

메모리는 버스에서 자기 어드레스를 받게 되면 코맨드에 따라 read와 write 기능을 수행하게 되는데 코맨드부터 XACK/까지의 시간을 어세스 타임이라 한다.

ECC 회로의 동작은 각 메모리 사이클마다 이루어지는데 write cycle때 16 bit의 데이터당 hgmming code에 의하여 6 bit의 체크 비트를 만들어 이것을 데이터와 같이 기억시킴으로써 read cycle때 검출 및 수정기능을 하게 된다.

Read cycle 읽어 낸 데이터와 체크 비트를 이용하여 syndrome bit를 만들어 에러의 상태를 알게 된다. Check bit encoding식과 syndrome bit decoding 표는 표 3, 4와 같다. 16 bit를 기본단위로 check bit를 만들기 때문에 8-bit(byte)를 쓸 경우, 플로우 차트에서 저럼 쓰고자 하는 곳의 word를 먼저 읽어 내어 새로운 byte를 옛것과 합쳐서 새로운 word를 만든 후, 그것에 따라 체크 비트를 만들어 쓰게 된다.

기억소자로서 다이내믹 RAM을 사용하고 있기 때문에 refresh, row/column address multiplexing 그리

고 드라이버가 필요하게 되는데 LSI를 이용하여 처리하였다. 이외 블록 다이어그램에 나타낸 블록에 관해 간단히 설명하면 각 버스쪽에 신호에는 버퍼를 갖추고 있으며 byte access할 경우 IEEE-796 bus에 맞게 low byte 쪽으로 swap해주는 기능, 어드레스 일치를 알리는 디코더, 더블-에러가 발생할 때 인터럽트를 보내는 기능등이 있다.

메모리 시스템의 performance는 크게 어세스 타임에 의존하는데 설계한 것은 16 bit write의 경우 150-200 nsec, read나 byte write의 경우 300-350 ns이다.

어세스 타임에 여유가 있는 것은 리후레쉬 요청과 메

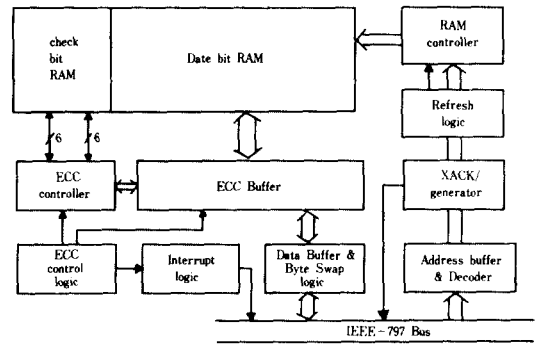


그림 1. 블록 다이어그램  
Fig. 1. Block diagram.

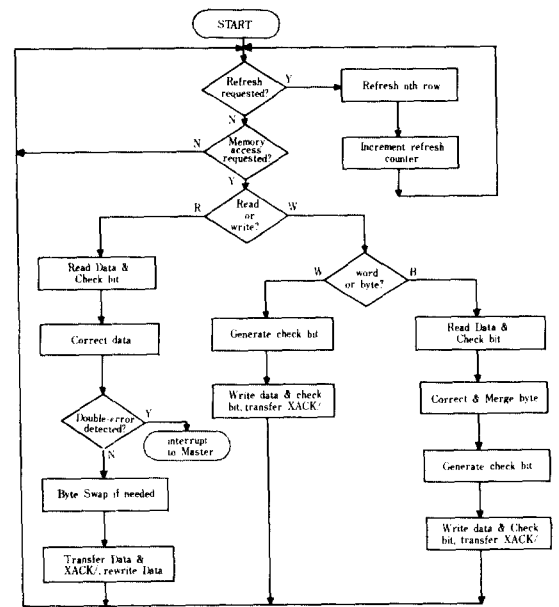


그림 2. 동작 플로우 차트  
Fig. 2. Operational flowchart.

메모리 액세스 요청을 받아주는 기능에 20MHz 클럭을 사용했기 때문이며 측정값은 logic analyzer의 클럭을 100MHz로 하여 얻은 값이므로 10ns의 오차가 있을 수 있다.

ECC 회로의 사용에 따라 여러 발생율은 ECC 회로가 없는 경우보다 60배정도 줄일 수 있고 double bit error의 발생을 최대한 억제키 위해 매 메모리 액세스마다 rewrite 과정을 넣었다.

#### IV. 결과의 검토

설계된 메모리 시스템을 계속 read와 write를 번갈아 진행시켜 볼 때 평균 액세스 타임은 약 250ns이다. 이 값은 같은 RAM을 사용하고 ECC 회로를 사용하지 않은 메모리보다 50ns이상 늦지 않으며 현재 미국에서 사용되는 비슷한 수준의 메모리에 비해 50ns 이상 빠르다. 이같은 결과는 ECC 회로에서 드는 시간은 최대한 다른 동작들과 동시에 일어날 수 있게 설계했으며 동작의 완료를 알리는 XACK/를 짧은 사이클과 긴 사이클을 구분함으로써 긴 사이클에 맞추어서 낭비했던 wait 상태를 줄였기 때문이다. 매 메모리 액세스(read or write)마다 rewrite 과정은 수행함에 따라 single bit error의 축적을 방지, 멀티 에러의 발생 가능성을 최대한 줄이었다. 그러나 시스템 안에서 환경시험, MTBF(mean time between failure) 등 신뢰도에 관한 시험은 장비 및 방법의 준비가 없어서 하지 못하였다.

가격면에서 외국에서 판매되고 있는 같은 수준의 모오드의 값보다 50~60% 정도로 낮출 수 있다.

#### V. 結 論

10개월간의 설계 및 제작, 시험과정을 통하여 메모리 시스템의 기능 및 구조파악, hamming 코드의 응용에 의한 ECC 회로설계, 버스 인터페이스 방법의 확인을 할 수 있었다. 종래에 우리나라에서 해 왔던 reference를 이용한 보오드 및 시스템 디자인 방법에서 벗어나 scratch design 함으로써 scratch design 방법의 파악, tool setup이 의의라 할 수 있다. 그러나 짧은 기간안에 수행됨에 따라 환경에 대한 성능실험, 신뢰도에 대한 실험등을 고려하지 못한점을 문제점으로 지적할 수 있다.

앞으로 버스를 사용하는 시스템이 안고 있는 버스 오버헤드 문제를 해결하기 위한 page(block) transfer 방법이나 다른 fast access 방법을 연구하고 있으며 diagnostic 기능등을 추가할 예정이다.

#### 參 考 文 獻

- [1] Richard W. Hamming, *Coding and Information Theory*. Prentice-Hall Inc., Chapter 2, Chapter 3, 1980.
- [2] Intel, *Multi-Bus Specification*. Intel, 1978.
- [3] AMD, *Bipolar Microprocessor Logic and Interface Data Book*. AMD, 1982.
- [4] National Semiconductor, *DP8409 Advanced Information*. National, 1982.
- [5] Drew Lucy, *Choose the Right Level of Memory-Error Protection Electronic Design*. Feb. 1982.
- [6] 박명관, *Preparations for Memory Board Circuit Reviewers*. KIET, 1982.
- [7] *Memory Board Test*. KIET, 1982.
- [8] Gregory J. Walker, Birch Acres, *Error Checking and Correcting for Your Computer*. BYTE, May 1980.