서버 시스템 내의 오류 정정 코드 분석에 관한 연구

A Study on Analysis of Error Correction Code in Server System

이 창 화*
Lee, Chang-Hwa

ABSTRACT

In this paper, a novel method is proposed how the ECC(Error Correction Code) in server system can be investigated and the robustness of each system against noisy environment and element failure in memory module has been verified. Chipset manufacturers have hided the algorithm of their Hamming code and the user has difficulty in verification of the robustness of each system. The proposed method is very simple, but the outputs of the experiment explain the core ability of error correction in server system and helps the detection of the failure element. On the basis of these results, we could expect the robustness of digitalized weapon system and the efficient design of our own error correction code.

주요기술용어(주제어) : Error Correction Code(오류 정정 코드, ECC), Chipset(칩셋), Memory module(메모리 모듈), Hamming code(해밍 코드)

1. 머리말

현대전에 쓰이는 첨단 정밀 무기 체계 및 무인 무기 체계가 점점 디지털화됨에 따라 작은 데이터 오류도 치명적인 작동오류로 이어질 수 있으므로 데이터의 무결성이 무기체계 개발에 있어서 그 어느 때보다 중요시되고 있다. 보통 디지털 방식은 아날로그 방식에 비하여 오류가 거의 없다고 생각한다. 예를 들어 디지털 방식은 복제를 거듭해도 질이 떨어지지 않는다든가, CD의 음질이 LP 레코드에 비해 더 깨끗하고 잡음이 적다고 생각한다.

그러나 실제로는 디지털 방식에서도 저장이나 전송 과정에서 오류가 발생할 가능성은 엄연히 존재하며 디지털 기기들이 점점 용량이 커지고 속도가 빨라짐에 따라서 완벽하게 동작하도록 하기는 그만큼 더 어려워진다고 할 수 있다. 이러한 문제를 해결하기 위해 디지털 방식에서는 오류정정코드(Error Correction Code)를 사용하여 오류가 있는지 없는지를 알아내고 그 다음으로 오류가 있으면 그것을 복구하고 있다.

오류가 있는지 없는지를 알아내는 방법에는 몇 가지가 있는데, 그것들 모두가 데이터에 오류 검출을 위한 정보를 덧붙여서 이를 확인하는 방식을 사용하며 이 중에서 가장 간단한 방법은 패리티 비트(parity bit)라 불리는 한개 비트를 원래 데이터에 붙여 전송함으로써 수신 측에서 전송 데이터 오류의 유무를 1의 개수(홀 또는 짝)로 판별하게 하는 것이다.

실제 상용되고 있는 서버 시스템은 해밍 코드(Hamming Code)를 사용하여 데이터 오류 검출 및

† 2005년 5월 27일 접수~2005년 9월 9일 게재승인
* 국방과학연구소(ADD)
주저자 이메일 : wha1973@hanafos.com

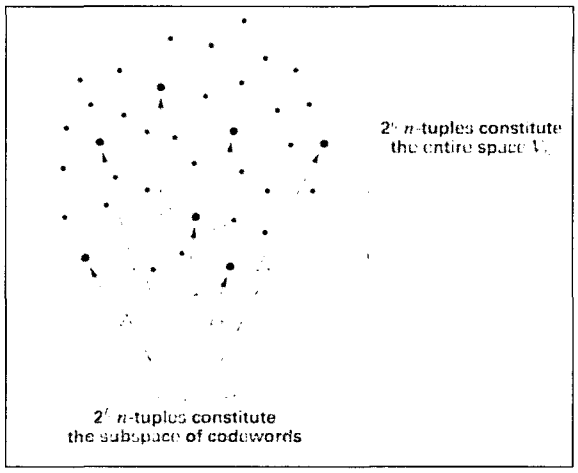
오류 정정을 하고 있다. 해밍 코드는 패리티를 응용한 코드로 데이터에 몇 개의 비트를 붙여 오류 검출 및 정정을 하고 있다. 따라서 이런 시스템을 동작시키기 위해 서버 칩셋(chipset)내에 제조회사는 고유의 해밍 코드를 설계하여 넣고 이 시스템에 쓰이는 메모리도 일반 PC에 쓰이는 64비트 외에 8비트를 추가한 메모리를 사용하여 72비트 시스템으로 구성하였다. 문제는 해밍 코드의 설계 방식에 따라 오류 검출 및 정정 능력이 다르며 설계 알고리즘을 공개하지 않기 때문에 서버 시스템의 데이터 복원 능력을 알기 어렵고 데이터 오류 시 표시하는 신드롬(Syndrome)으로는 어느 비트에 오류가 발생했는지 알 수 없다는 것이다.

본 연구에서는 서버 시스템의 칩셋 내에 설계된 해밍 코드를 추적하여 시스템의 오류 정정 알고리즘을 분석할 수 있는 간단하고 새로운 방법을 제시하였다. 제시된 방법을 이용하여 시스템의 오류 정정 알고리즘을 칩셋 제조사 별로 검증하였고 이로써 각 시스템의 노이즈 환경 및 시스템 내부 소자의 오작동에 견디는 시스템의 강인함을 분석하였다.

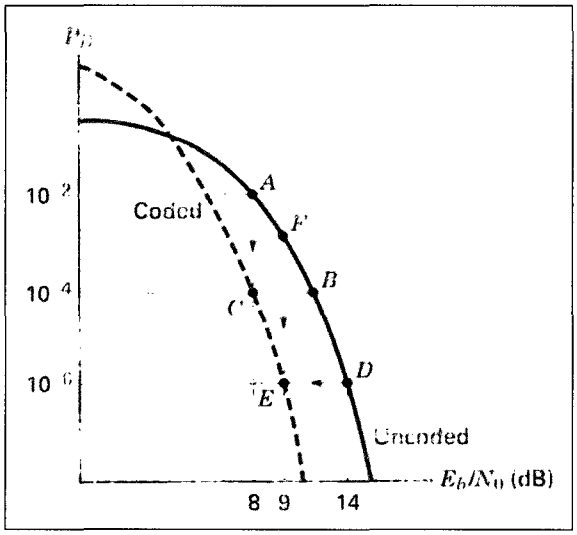
2. 오류 정정 코드

오류 정정 코드 원리는 그림 1에서와 같다[1]. 먼저 k 비트의 메시지 혹은 정보를 앤코더(encorder)가 n 비트의 코드워드(code word)로 만든다. 이때 만들어진 2^k개의 메시지를 2^n개의 전체 공간 V_n에 뿌리면 원래 해밍 거리가 1인 데이터 사이에 완충 공간을 넣어 주는 효과를 나타내어 데이터 전송 시 데이터의 오류를 정정할 가능성을 높이는 것이다.($2^k < 2^n$) 이러한 선형 블록 코드는 (n,k)로 나타낸다. 두 개의 코드워드 사이 거리는 서로 다른 비트 개수로 정의하며 해밍 거리는 모든 가능한 코드워드 조합 중 가장 가까운 거리로 정의된다. 당연히 해밍 거리가 멀면 멀수록 오류 정정 및 수정이 용이하므로 데이터 무결성이 보장되지만 데이터 대역너비(bandwidth)가 커지는 단점이 있다.

그림 2는 코드화된 데이터 오류율 대 코드화 안된 데이터 오류율을 보여주고 있다[1]. 전송파워를 일정하



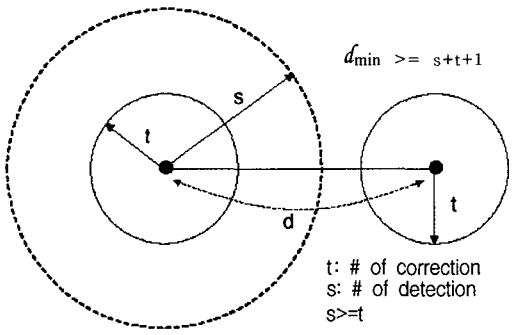
[그림 1] 선형 블록 코드 구조



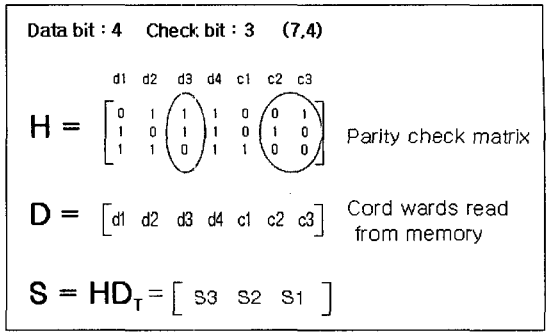
[그림 2] 코드화된 데이터 오류율 대 코드화 안 된 데이터 오류율

게 놓으면 코드화 된 데이터의 오류 확률이 낮은 것을 볼 수 있다. 그러나 낮은 전송파워에서는 코드화가 된 데이터의 대역너비가 크기 때문에 에러 확률이 커지므로 무작정 n을 늘리는 것이 좋은 것만은 아니다. 따라서 대역너비와 전송파워의 절충(trade-off)이 필요하다.

해밍 거리에 따른 오류 정정 능력은 그림 3과 같다. d_{min}은 해밍 거리를 나타내며 정정가능 에러 비트수(t)와 탐지가능 에러 비트수(n)를 더한 값보다 1이



[그림 3] 해밍 거리에 따른 오류정정능력



[그림 4] 해밍거리 3의 예시

크다.

◆ 해밍거리가 1 일 때(+++++++)

모든 코드워드가 유효하므로(+는 유효, -는 에러) 오류 검출 및 정정 불가능하다.

◆ 해밍거리가 2 일 때(+-+-+-+)

대표적인 예로 패리티 체크 비트를 붙인 코드로 전송된 코드워드가 '-'일 때 한개 비트가 오류가 난 것이다. 만일 두개 비트가 동시에 오류가 발생하면 다음의 '+'로 전송되므로 시스템은 오류 발생을 감지하지 못한다.

◆ 해밍거리가 3 일 때(+--+--+)

2^C>=B+C+1 (B : 데이터 비트, C : 체크 비트) (1)

B+C=n (2)

B=k (3)

해밍거리를 3으로 확보하면 오류가 한개 비트 발생하면 가까운 코드워드로 보정하며(SEC : Single bit Error Correction) '-'로 전송된 코드워드가 가까운 '+'로 정정된다. 만일 두개 비트 오류가 동시에 발생하면 다음의 '+'로 잘못 인식된다. 해밍거리가 3일때 데이터 비트(B)와 체크 비트(C)의 관계는 식 (1)과 같고 코드 (n,k)는 식 (2), (3)와 같다. 그림 4는 해밍거리가 3일 때의 간단한 예시 (7,4)를 보여주고 있으며 H 매트릭스는 패리티 체크 매트리스로서 메모리로부터 들어온 코드워드 D와 연산하여 신드롬 S를 생성한다. 아래 식 (4), (5), (6)은 신드롬 생성 및 체크비트 생성식을 보여주고 있다.(앞으로 모든 '+'는 modulo-2 addition 임.)

S1=d1+d2+d4+c1 c1=d1+d2+d4 (4)

S2=d1+d3+d4+c2 c2=d1+d3+d4 (5)

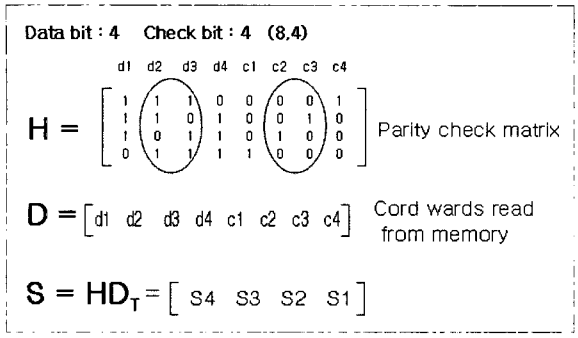
S3=d2+d3+d4+c3 c3=d2+d3+d4 (6)

그림 4의 D에서 c2와 c3에 동시에 오류가 나면 신드롬 값이 [011]이 생성되어 d3에서 오류가 나서 생긴 신드롬 [011]과 같아 잘못된 정정이 발생하며 이것이 해밍거리 3의 한계이다.

◆ 해밍거리가 4 일 때(+---+---)

해밍 거리를 4로 확보하면 오류가 한개 비트 발생하면 가까운 코드워드로 보정하고(SEC, Single bit Error Correction), 두개 비트 발생 시 전송된 코드워드는 가운데의 '-'로 전송되어 오류 정정은 불가능 하지만 오류 발생을 검출하게 된다(DED, Double bit Error Detection).

이 경우 H 매트릭스를 Odd-weight-column code로 하여 설계하면 된다[2]. 즉, H 매트릭스의 각 열에서 1의 개수를 홀수로 맞추어 설계한다. 해밍거리가 4일 때 데이터 비트(B)와 체크 비트(C)의 관계는 식 (7)과 같고 그림 5는 해밍거리가 4일 때의 간단한 예



[그림 5] 해밍거리 4의 예시

시 (8,4)를 보여주고 있다. 아래 식 (8), (9), (10), (11)은 신드롬 생성 및 체크비트 생성식을 보여주고 있다.

$$2^{C-1} >= B+C \quad (7)$$

$$S1=d2+d3+d4+c1 \qquad c1=d2+d3+d4 \quad (8)$$

$$S2=d1+d3+d4+c2 \qquad c2=d1+d3+d4 \quad (9)$$

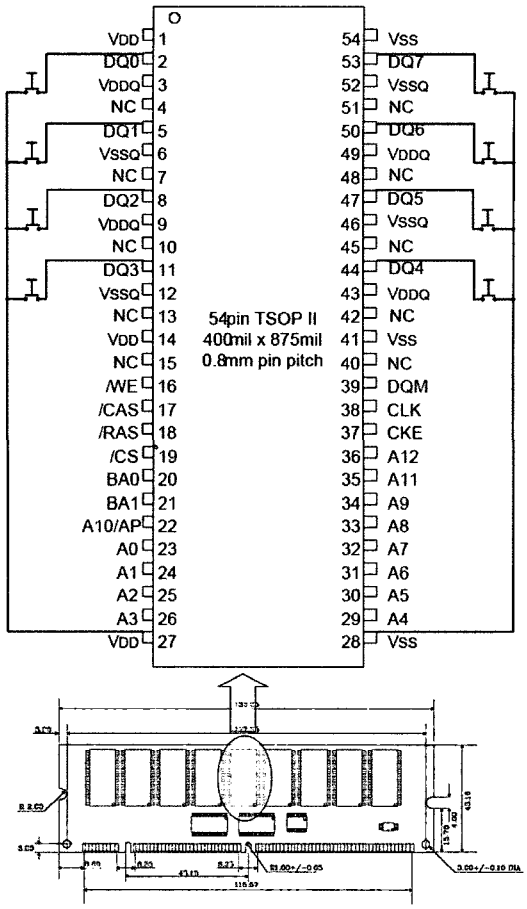
$$S3=d1+d2+d4+c3 \qquad c3=d1+d2+d4 \quad (10)$$

$$S3=d1+d2+d3+c4 \qquad c4=d1+d2+d3 \quad (11)$$

위의 H 매트릭스에서 d2, d3이 동시에 오류가 나면 그것의 신드롬은 [0110]으로 짝수의 1을 갖는 신드롬이 되므로 절대로 한개 비트 오류 시 나타나는 신드롬과 같을 수 없다. 하지만 이 경우 c2, c3이 동시에 오류가 났을 때의 신드롬과 같기 때문에 오류 정정은 불가능하고 단지 오류 발생만 검출할 수 있다.

3. 서버 오류정정 코드 분석 실험방법 제안

상용 서버 시스템 중 실험 대상으로 ServerWorks 사의 CNB30LE, CNB20HE와 컴팩사의 Profusion (PL8000) 칩셋을 조사하였다. 그림 6은 하이닉스 반도체의 메모리 모듈[3]을 이용하여 서버 시스템의 해



[그림 6] 제안된 시스템 분석 실험 구성도

밍거리 분석 방법을 제안한 실험 구성도이다. 신드롬 발생을 각 데이터 비트마다 발생시키기 위하여 각 데이터 비트(DQ)를 GND나 V_{DD}에 연결하여 시스템을 부팅시켰다. 이때 실험대상 비트는 연속하여 '1'혹은 '0'만 입력 또는 출력되기 때문에 서버 시스템에 오류가 발생한다. 서버 시스템은 적어도 한개 비트 오류는 자체적으로 오류 정정하여 동작하므로(SEC) 동작에는 이상이 없는 것으로 가정하였으며 혹시 발생될 수 있는 다른 비트의 오류를 감안하여 여러 회 반복 실험하여 결과를 확인하였다.

그림 7은 제안한 실험 방법에 의해 메모리 데이터 비트에 고의로 오류를 발생시켜서 얻은 신드롬 테이블을 표로 정리해 놓은 것이다. 여기서는 각 DQ 및 CB 번호는 고의로 오류를 발생시킨 메모리 모듈의

SYNDROME	HEX	PL8000	CNB30LE	SYNDROME	HEX	PL8000	CNB30LE	SYNDROME	HEX	PL8000	CNB30LE	SYNDROME	HEX	PL8000	CNB30LE
0 0 0 0 0 0 0 0	0			0 1 0 0 0 0 0 0	40	CB6	CB6	1 0 0 0 0 0 0 0	80	CB7	CB7	1 1 0 0 0 0 0 0	C0	.	.
0 0 0 0 0 0 0 1	1	CB0	CB0	0 1 0 0 0 0 0 1	41	.	.	1 0 0 0 0 0 0 1	81	.	.	1 1 0 0 0 0 0 1	C1	DQ0	.
0 0 0 0 0 0 1 0	2	CB1	CB1	0 1 0 0 0 0 1 0	42	.	.	1 0 0 0 0 0 1 0	82	.	.	1 1 0 0 0 0 1 0	C2	DQ1	.
0 0 0 0 0 0 1 1	3	.	.	0 1 0 0 0 0 1 1	43	DQ54	.	1 0 0 0 0 0 1 1	83	DQ55	.	1 1 0 0 0 0 1 1	C3	.	.
0 0 0 0 0 1 0 0	4	CB2	CB2	0 1 0 0 0 1 0 0	44	.	.	1 0 0 0 0 1 0 0	84	.	.	1 1 0 0 0 1 0 0	C4	DQ2	.
0 0 0 0 0 1 0 1	5	.	.	0 1 0 0 0 1 0 1	45	DQ50	.	1 0 0 0 0 1 0 1	85	DQ51	.	1 1 0 0 0 1 0 1	C5	.	.
0 0 0 0 0 1 1 0	6	.	.	0 1 0 0 0 1 1 0	46	DQ46	.	1 0 0 0 0 1 1 0	86	DQ47	.	1 1 0 0 0 1 1 0	C6	.	.
0 0 0 0 0 1 1 1	7	DQ58	DQ5	0 1 0 0 0 1 1 1	47	.	DQ21	1 0 0 0 0 1 1 1	87	.	DQ37	1 1 0 0 0 1 1 1	C7	.	DQ53
0 0 0 0 1 0 0 0	8	CB3	CB3	0 1 0 0 1 0 0 0	48	.	.	1 0 0 0 1 0 0 0	88	.	.	1 1 0 0 1 0 0 0	C8	DQ3	.
0 0 0 0 1 0 0 1	9	.	.	0 1 0 0 1 0 0 1	49	DQ42	.	1 0 0 0 1 0 0 1	89	DQ43	.	1 1 0 0 1 0 0 1	C9	.	.
0 0 0 0 1 0 1 0	A	.	.	0 1 0 0 1 0 1 0	4A	DQ38	.	1 0 0 0 1 0 1 0	8A	DQ39	.	1 1 0 0 1 0 1 0	CA	.	.
0 0 0 0 1 0 1 1	B	DQ29	DQ6	0 1 0 0 1 0 1 1	4B	.	DQ22	1 0 0 0 1 0 1 1	8B	.	DQ38	1 1 0 0 1 0 1 1	CB	.	DQ54
0 0 0 0 1 1 0 0	C	.	.	0 1 0 0 1 1 0 0	4C	DQ34	.	1 0 0 0 1 1 0 0	8C	DQ35	.	1 1 0 0 1 1 0 0	CC	.	.
0 0 0 0 1 1 0 1	D	DQ59	DQ0	0 1 0 0 1 1 0 1	4D	.	DQ16	1 0 0 0 1 1 0 1	8D	.	DQ32	1 1 0 0 1 1 0 1	CD	.	DQ48
0 0 0 0 1 1 1 0	E	DQ28	DQ8	0 1 0 0 1 1 1 0	4E	.	DQ24	1 0 0 0 1 1 1 0	8E	.	DQ40	1 1 0 0 1 1 1 0	CE	.	DQ56
0 0 0 0 1 1 1 1	F	.	.	0 1 0 0 1 1 1 1	4F	DQ25	.	1 0 0 0 1 1 1 1	8F	DQ56	.	1 1 0 0 1 1 1 1	CF	.	.
0 0 0 1 0 0 0 0	10	CB4	CB4	0 1 0 1 0 0 0 0	50	.	.	1 0 0 1 0 0 0 0	90	.	.	1 1 0 1 0 0 0 0	D0	DQ27	.
0 0 0 1 0 0 0 1	11	.	.	0 1 0 1 0 0 0 1	51	DQ15	.	1 0 0 1 0 0 0 1	91	DQ8	.	1 1 0 1 0 0 0 1	D1	.	.
0 0 0 1 0 0 1 0	12	.	.	0 1 0 1 0 0 1 0	52	DQ17	.	1 0 0 1 0 0 1 0	92	DQ9	.	1 1 0 1 0 0 1 0	D2	.	.
0 0 0 1 0 0 1 1	13	DQ52	DQ7	0 1 0 1 0 0 1 1	53	.	DQ23	1 0 0 1 0 0 1 1	93	.	DQ39	1 1 0 1 0 0 1 1	D3	.	DQ55
0 0 0 1 0 1 0 0	14	.	.	0 1 0 1 0 1 0 0	54	DQ18	.	1 0 0 1 0 1 0 0	94	DQ10	.	1 1 0 1 0 1 0 0	D4	.	.
0 0 0 1 0 1 0 1	15	DQ48	DQ4	0 1 0 1 0 1 0 1	55	.	DQ20	1 0 0 1 0 1 0 1	95	.	DQ36	1 1 0 1 0 1 0 1	D5	.	DQ52
0 0 0 1 0 1 1 0	16	DQ44	.	0 1 0 1 0 1 1 0	56	.	.	1 0 0 1 0 1 1 0	96	.	.	1 1 0 1 0 1 1 0	D6	.	.
0 0 0 1 0 1 1 1	17	.	.	0 1 0 1 0 1 1 1	57	.	.	1 0 0 1 0 1 1 1	97	.	.	1 1 0 1 0 1 1 1	D7	.	.
0 0 0 1 1 0 0 0	18	.	.	0 1 0 1 1 0 0 0	58	DQ19	.	1 0 0 1 1 0 0 0	98	DQ11	.	1 1 0 1 1 0 0 0	D8	.	.
0 0 0 1 1 0 0 1	19	DQ40	DQ3	0 1 0 1 1 0 0 1	59	.	DQ19	1 0 0 1 1 0 0 1	99	.	DQ35	1 1 0 1 1 0 0 1	D9	.	DQ51
0 0 0 1 1 0 1 0	1A	DQ36	DQ11	0 1 0 1 1 0 1 0	5A	.	DQ27	1 0 0 1 1 0 1 0	9A	.	DQ43	1 1 0 1 1 0 1 0	DA	.	DQ59
0 0 0 1 1 0 1 1	1B	.	.	0 1 0 1 1 0 1 1	5B	.	.	1 0 0 1 1 0 1 1	9B	.	.	1 1 0 1 1 0 1 1	DB	..	.
0 0 0 1 1 1 0 0	1C	DQ32	.	0 1 0 1 1 1 0 0	5C	.	.	1 0 0 1 1 1 0 0	9C	.	.	1 1 0 1 1 1 0 0	DC	.	.
0 0 0 1 1 1 0 1	1D	.	.	0 1 0 1 1 1 0 1	5D	.	.	1 0 0 1 1 1 0 1	9D	.	.	1 1 0 1 1 1 0 1	DD	.	.
0 0 0 1 1 1 1 0	1E	.	.	0 1 0 1 1 1 1 0	5E	.	.	1 0 0 1 1 1 1 0	9E	.	.	1 1 0 1 1 1 1 0	DE	.	.
0 0 0 1 1 1 1 1	1F	DQ62	.	0 1 0 1 1 1 1 1	5F	.	.	1 0 0 1 1 1 1 1	9F	.	.	1 1 0 1 1 1 1 1	DF	.	.
0 0 1 0 0 0 0 0	20	CB5	CB5	0 1 1 0 0 0 0 0	60	.	.	1 0 1 0 0 0 0 0	A0	.	.	1 1 1 0 0 0 0 0	E0	DQ60	.
0 0 1 0 0 0 0 1	21	.	.	0 1 1 0 0 0 0 1	61	DQ12	.	1 0 1 0 0 0 0 1	A1	DQ4	.	1 1 1 0 0 0 0 1	E1	.	.
0 0 1 0 0 0 1 0	22	.	.	0 1 1 0 0 0 1 0	62	DQ13	.	1 0 1 0 0 0 1 0	A2	DQ5	.	1 1 1 0 0 0 1 0	E2	.	.
0 0 1 0 0 0 1 1	23	DQ53	.	0 1 1 0 0 0 1 1	63	.	.	1 0 1 0 0 0 1 1	A3	.	.	1 1 1 0 0 0 1 1	E3	.	.
0 0 1 0 0 1 0 0	24	.	.	0 1 1 0 0 1 0 0	64	DQ14	.	1 0 1 0 0 1 0 0	A4	DQ6	.	1 1 1 0 0 1 0 0	E4	.	.
0 0 1 0 0 1 0 1	25	DQ49	DQ2	0 1 1 0 0 1 0 1	65	.	DQ18	1 0 1 0 0 1 0 1	A5	.	DQ34	1 1 1 0 0 1 0 1	E5	.	DQ50
0 0 1 0 0 1 1 0	26	DQ45	DQ10	0 1 1 0 0 1 1 0	66	.	DQ26	1 0 1 0 0 1 1 0	A6	.	DQ42	1 1 1 0 0 1 1 0	E6	.	DQ58
0 0 1 0 0 1 1 1	27	.	.	0 1 1 0 0 1 1 1	67	.	.	1 0 1 0 0 1 1 1	A7	.	.	1 1 1 0 0 1 1 1	E7	.	.
0 0 1 0 1 0 0 0	28	.	.	0 1 1 0 1 0 0 0	68	DQ15	.	1 0 1 0 1 0 0 0	A8	DQ7	.	1 1 1 0 1 0 0 0	E8	.	.
0 0 1 0 1 0 0 1	29	DQ41	.	0 1 1 0 1 0 0 1	69	.	.	1 0 1 0 1 0 0 1	A9	.	.	1 1 1 0 1 0 0 1	E9	.	.
0 0 1 0 1 0 1 0	2A	DQ37	DQ12	0 1 1 0 1 0 1 0	6A	.	DQ28	1 0 1 0 1 0 1 0	AA	.	DQ44	1 1 1 0 1 0 1 0	EA	.	DQ60
0 0 1 0 1 0 1 1	2B	.	.	0 1 1 0 1 0 1 1	6B	.	.	1 0 1 0 1 0 1 1	AB	.	.	1 1 1 0 1 0 1 1	EB	.	.
0 0 1 0 1 1 0 0	2C	DQ33	DQ15	0 1 1 0 1 1 0 0	6C	.	DQ31	1 0 1 0 1 1 0 0	AC	.	DQ47	1 1 1 0 1 1 0 0	EC	.	DQ63
0 0 1 0 1 1 0 1	2D	.	.	0 1 1 0 1 1 0 1	6D	.	.	1 0 1 0 1 1 0 1	AD	.	.	1 1 1 0 1 1 0 1	ED	.	.
0 0 1 0 1 1 1 0	2E	.	.	0 1 1 0 1 1 1 0	6E	.	.	1 0 1 0 1 1 1 0	AE	.	.	1 1 1 0 1 1 1 0	EE	.	.
0 0 1 0 1 1 1 1	2F	DQ31	.	0 1 1 0 1 1 1 1	6F	.	.	1 0 1 0 1 1 1 1	AF	.	.	1 1 1 0 1 1 1 1	EF	.	.
0 0 1 1 0 0 0 0	30	.	.	0 1 1 1 0 0 0 0	70	DQ26	.	1 0 1 1 0 0 0 0	B0	DQ61	.	1 1 1 1 0 0 0 0	F0	.	.
0 0 1 1 0 0 0 1	31	DQ20	DQ1	0 1 1 1 0 0 0 1	71	.	DQ17	1 0 1 1 0 0 0 1	B1	.	DQ33	1 1 1 1 0 0 0 1	F1	DQ30	DQ49
0 0 1 1 0 0 1 0	32	DQ21	DQ9	0 1 1 1 0 0 1 0	72	.	DQ25	1 0 1 1 0 0 1 0	B2	.	DQ41	1 1 1 1 0 0 1 0	F2	DQ63	DQ57
0 0 1 1 0 0 1 1	33	.	.	0 1 1 1 0 0 1 1	73	.	.	1 0 1 1 0 0 1 1	B3	.	.	1 1 1 1 0 0 1 1	F3	.	.
0 0 1 1 0 1 0 0	34	DQ22	DQ14	0 1 1 1 0 1 0 0	74	.	DQ30	1 0 1 1 0 1 0 0	B4	.	DQ46	1 1 1 1 0 1 0 0	F4	DQ57	DQ62
0 0 1 1 0 1 0 1	35	.	.	0 1 1 1 0 1 0 1	75	.	.	1 0 1 1 0 1 0 1	B5	.	.	1 1 1 1 0 1 0 1	F5	.	.
0 0 1 1 0 1 1 0	36	.	.	0 1 1 1 0 1 1 0	76	.	.	1 0 1 1 0 1 1 0	B6	.	.	1 1 1 1 0 1 1 0	F6	.	.
0 0 1 1 0 1 1 1	37	.	.	0 1 1 1 0 1 1 1	77	.	.	1 0 1 1 0 1 1 1	B7	.	.	1 1 1 1 0 1 1 1	F7	.	.
0 0 1 1 1 0 0 0	38	DQ23	DQ13	0 1 1 1 1 0 0 0	78	.	DQ29	1 0 1 1 1 0 0 0	B8	.	DQ45	1 1 1 1 1 0 0 0	F8	DQ24	DQ61
0 0 1 1 1 0 0 1	39	.	.	0 1 1 1 1 0 0 1	79	.	.	1 0 1 1 1 0 0 1	B9	.	.	1 1 1 1 1 0 0 1	F9	.	.
0 0 1 1 1 0 1 0	3A	.	.	0 1 1 1 1 0 1 0	7A	.	.	1 0 1 1 1 0 1 0	BA	.	.	1 1 1 1 1 0 1 0	FA	.	.
0 0 1 1 1 0 1 1	3B	.	.	0 1 1 1 1 0 1 1	7B	.	.	1 0 1 1 1 0 1 1	BB	.	.	1 1 1 1 1 0 1 1	FB	.	.
0 0 1 1 1 1 0 0	3C	.	.	0 1 1 1 1 1 0 0	7C	.	.	1 0 1 1 1 1 0 0	BC	.	.	1 1 1 1 1 1 0 0	FC	.	.
0 0 1 1 1 1 0 1	3D	.	.	0 1 1 1 1 1 0 1	7D	.	.	1 0 1 1 1 1 0 1	BD	.	.	1 1 1 1 1 1 0 1	FD	.	.
0 0 1 1 1 1 1 0	3E	.	.	0 1 1 1 1 1 1 0	7E	.	.	1 0 1 1 1 1 1 0	BE	.	.	1 1 1 1 1 1 1 0	FE	.	.
0 0 1 1 1 1 1 1	3F	.	.	0 1 1 1 1 1 1 1	7F	.	.	1 0 1 1 1 1 1 1	BF	.	.	1 1 1 1 1 1 1 1	FF	.	.

[그림 7] 신드롬 테이블

데이터 비트 번호이고 왼쪽 숫자는 그때 발생된 신드롬의 2진수와 16진수를 나타내고 있다. 이 실험은 CNB30LE와 Profusion 칩셋을 대상으로 하였으며 다음은 각각의 칩셋에 대한 분석이다.

가. ServerWorks CNB30LE

그림 8은 실험결과에 의해 CNB30LE가 사용하는 H 매트릭스를 보여준다. 비록 데이터 비트가 64개이고 체크 비트가 8개로 (72,64) 이지만 매트릭스 내부를 보면 '1'의 개수가 3개, 4개, 5개로 설계되어 있어 Odd-weight-column 코드로 작성되어 있지 못하다. 따라서 만일 그림 8의 좌측 타원인 두개 비트가 동시에 오류가 발생하면 각 신드롬을 연산한(modulo-2 addtion) 값이 한개 비트 오류 신드롬으로 나오는데 이 때 해밍 거리가 4가 되려면 이 값이 매트릭스 안에 없어야 한다.

$$00001101 + 10001101 = 10000000 \quad (12)$$

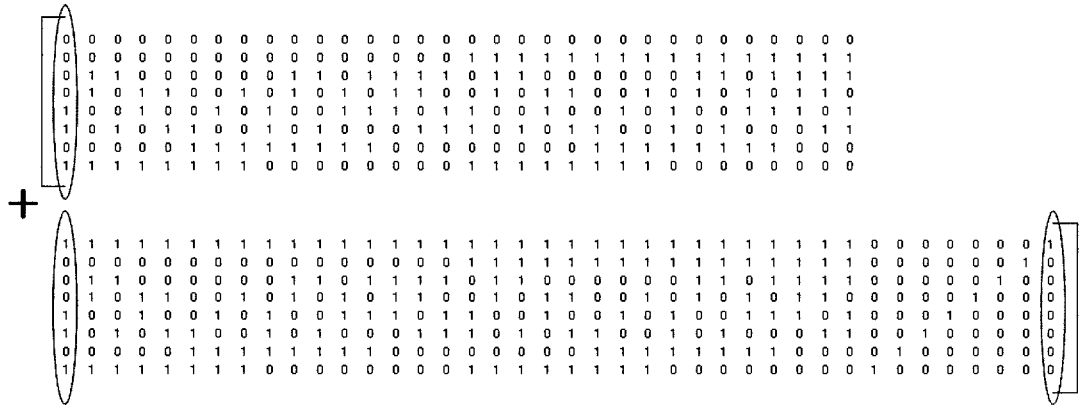
식 (12)에서처럼 두개 비트의 신드롬을 연산한 값이 매트릭스 내부에 있다. 따라서 이 시스템은 한 개 비트 오류 정정 능력만을 갖는 해밍 거리가 3인 시스

RCC(Reliance Computer Corp.) **CNB30LE**

Hamming distance = 3 ⟶ Single-bit-error correction (SEC) !

Data bit : 64 Check bit : 8 (72,64)

H =



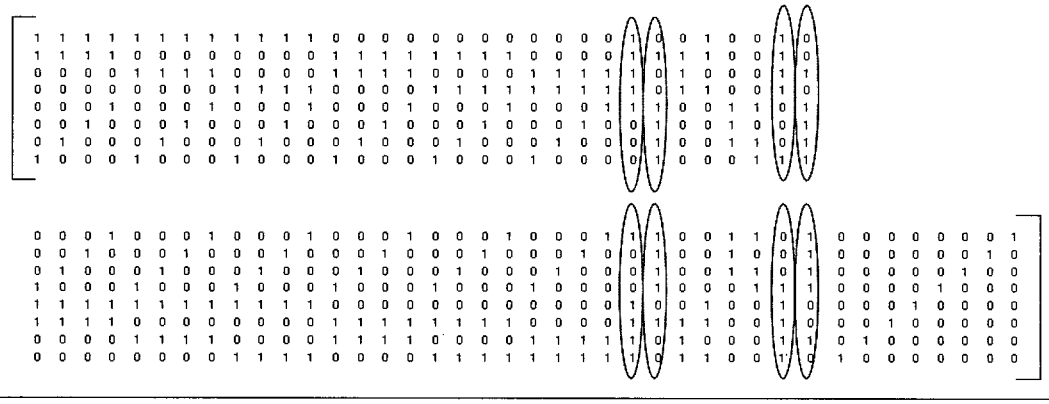
[그림 8] CNB30LE H 매트릭스

PL8000 PROFUSION

Hamming distance = 4 ⟶ Single-bit-error correction &
Double-bit error detection (SEC-DED) !

Data bit : 64 Check bit : 8 (72,64)

H =



[그림 9] Profusion H 매트릭스

템이다. 만일 이 시스템에서 두개 비트가 동시에 오류가 발생하면 한개 비트 오류 발생으로 잘못 인식하며 잘못된 정정을 행할 것이다. 따라서 높은 데이터 무결성을 요구하는 일에 있어서 본 칩셋을 사용하는 서버는 해밍 거리가 4인 서버에 비해 신뢰성이 떨어진다고 볼 수 있다.

나. 컴팩 PL8000 Profusion

컴팩 PL8000 서버의 Profusion 칩셋은 그림 9에서와 같은 H 매트릭스로 설계되어 있다. 이 시스템도 CNB30LE와 같이 (72,64)로 설계되어 있지만 매트릭스 내부가 Odd-weight-column 코드로 설계되어 있어 신드롬의 '1'의 개수가 홀수로 되어 있다

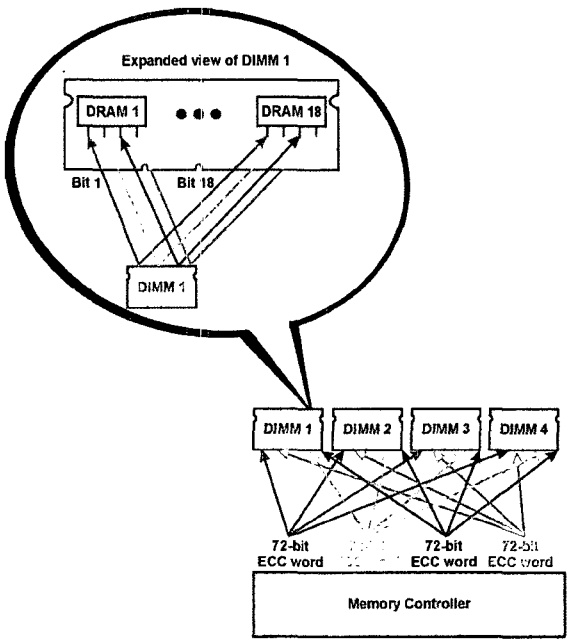
그림 9에서 '1'이 한개 들어간 8개 신드롬을 빼고 남은 64개 중에 '1'이 3개가 들어가는 조합의 개수는 ${}_8C_3$이고 남은 8개는 '1'의 개수가 5개인 신드롬이며 이것은 그림 9에서 타원으로 보이고 있다. 이 시스템은 해밍 거리가 4인 시스템으로 한개 비트 오류 정정 능력과 두개 비트 에러 검출능력이 가능하다. 따라서 Profusion 칩셋은 단일 비트 오류 발생 시 정상 동작이 가능하도록 자체 정정하여 동작하고 두개 비트 동시 오류 발생 시 두개 비트 오류 발생을 감지하여 알려줄 수 있다. 따라서 이 시스템은 CNB30LE과 같은 패리티 비트를 가지고 있으나 보다 강력한 오류 정정 및 검출 능력을 가지고 있음을 검증하였다.

4. 칩킬 보정(Chipkill Correct)

1990년 초에 비해 CPU 성능은 거의 20배 이상 향상되었으나 디스크 드라이브 속도는 약 5배 향상되어 이를 미처 따라가지 못하고 있다. 그래서 시스템 성능 향상을 위해 점차 메모리 요구량이 대용량화 되어 가고 있으며 메모리가 대용량화, 고집적화 되어감에 따라 여러 비트에서 동시에 발생하는 메모리 오류 확률이 증가되고 있는 실정이다. SEC-DED 코드 시스템에서는 한개 비트의 오류 정정 및 두개 비트의 오류 검출을 할 수 있기 때문에 동시에 여러 비트의 오류에는 시스템이 견디지 못한다. 이것은 대용량의 귀중한 정보의 손실을 의미하며 정밀 무기 체계 개발에 있어서도 매우 심각한 문제가 되고 있다. 이 문제를 해결하기 위하여 최근 서버 시스템에서는 칩킬(Chipkill) 보정이라는 방법이 사용된다.

칩킬 보정이란 그림 10에서처럼 메모리 한 개에서 단 하나의 데이터만을 사용하여 한개 모듈에 18개의 비트를 4개의 모듈에서 72개의 비트를 모아 사용하는 방식이다[4]. 이렇게 4개의 모듈에 흩어져서 데이터를 운영하므로 만일 한 개의 메모리가 통째로 작동을 못하던지 혹은 메모리 소자 내부의 블록이 오류가 나서 4비트가 동시에 오류가 생기더라도 각각의 SEC-DED 코드가 한 개 비트씩을 정정하게 된다.

따라서 앞에 설명한 단순 SEC-DED 코드로 설계된 서버에 비해 4개까지의 오류를 견디는 시스템을 만들 수 있다. 칩킬 보정이 가능한 칩셋인 CNB20HE (HP ProLiant ML330, Dell Power Edge 6400)를 실험하여 그림 11에 결과를 정리하였다. 이 서버는 4개의 모듈(DIMM)을 동시에 장착해야 동작하며 한 개의 코드워드는 (72,64)이고 매트릭스 내부는Odd-weight-column code로 설계되어 있으므로 해밍거리가 4인 시스템으로 판단할 수 있다. 진하게 표시된 곳은 '1'의 개수가 5개인 신드롬이다. 따라서 사용된



[그림 10] 델 서버에서 칩킬 보정 방법

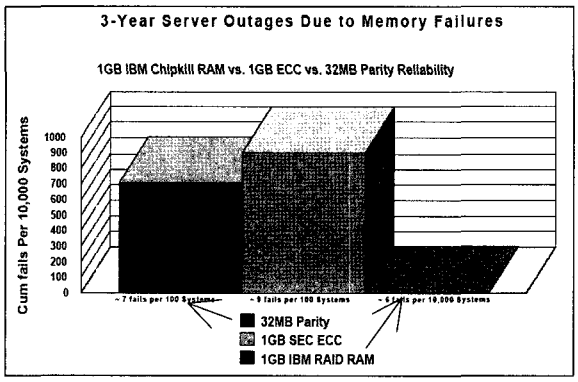
SYNDROME	HEX	DIMM1	DIMM2	DIMM3	DIMM4
0 0 0 0 0 0 0 0	0				
0 0 0 0 0 0 0 1	1		CB	DQ34, 35	.
0 0 0 0 0 0 1 0	2	DQ34, 35	DQ62, 63	.	.
0 0 0 0 0 0 1 1	3	.	.	.	.
0 0 0 0 0 1 0 0	4	.	.	DQ2, 3	CB
0 0 0 0 0 1 0 1	5	.	.	.	.
0 0 0 0 0 1 1 0	6	.	.	.	.
0 0 0 0 0 1 1 1	7	.	.	DQ8, 9	DQ20, 21
0 0 0 0 1 0 0 0	8	DQ2, 3	.	.	DQ30, 31
0 0 0 0 1 0 0 1	9	.	.	.	.
0 0 0 0 1 0 1 0	A	.	.	.	.
0 0 0 0 1 0 1 1	B	.	DQ32, 33	DQ10, 11	.
0 0 0 0 1 1 0 0	C	.	.	.	.
0 0 0 0 1 1 0 1	D	DQ8, 9	.	.	DQ6, 7
0 0 0 0 1 1 1 0	E	DQ14, 15	DQ40, 41	.	.
0 0 0 0 1 1 1 1	F	.	.	.	.
0 0 0 1 0 0 0 0	10	.	.	CB	DQ2, 3
0 0 0 1 0 0 0 1	11	.	.	.	.
0 0 0 1 0 0 1 0	12	.	.	.	.
0 0 0 1 0 0 1 1	13	.	.	DQ54, 55	DQ60, 61
0 0 0 1 0 1 0 0	14	.	.	.	.
0 0 0 1 0 1 0 1	15	.	DQ38, 39	DQ30, 31	.
0 0 0 1 0 1 1 0	16	.	.	DQ36, 37	DQ42, 43
0 0 0 1 0 1 1 1	17	.	.	.	.
0 0 0 1 1 0 0 0	18	.	.	.	.
0 0 0 1 1 0 0 1	19	.	.	DQ44, 45	DQ56, 57
0 0 0 1 1 0 1 0	1A	DQ26, 27	DQ24, 25	.	.
0 0 0 1 1 0 1 1	1B	.	.	.	.
0 0 0 1 1 1 0 0	1C	DQ62, 63	DQ28, 29	.	.
0 0 0 1 1 1 0 1	1D	.	.	.	.
0 0 0 1 1 1 1 0	1E	.	.	.	.
0 0 0 1 1 1 1 1	1F	DQ32, 33	DQ46, 47	.	.
0 0 1 0 0 0 0 0	20	CB	DQ2, 3	.	.
0 0 1 0 0 0 0 1	21	.	.	.	.
0 0 1 0 0 0 1 0	22	.	.	.	.
0 0 1 0 0 0 1 1	23	DQ22, 23	.	.	DQ24, 25
0 0 1 0 0 1 0 0	24	.	.	.	.
0 0 1 0 0 1 0 1	25	DQ30, 31	.	.	CB
0 0 1 0 0 1 1 0	26	.	DQ6, 7	DQ4, 5	.
0 0 1 0 0 1 1 1	27	.	.	.	.
0 0 1 0 1 0 0 0	28	.	.	.	.
0 0 1 0 1 0 0 1	29	.	.	DQ12, 13	DQ10, 11
0 0 1 0 1 0 1 0	2A	.	DQ48, 49	DQ22, 23	.
0 0 1 0 1 0 1 1	2B	.	.	.	.
0 0 1 0 1 1 0 0	2C	.	.	DQ62, 63	DQ26, 29
0 0 1 0 1 1 0 1	2D	.	.	.	.
0 0 1 0 1 1 1 0	2E	.	.	.	.
0 0 1 0 1 1 1 1	2F	DQ6, 7	DQ, 0 1	.	.
0 0 1 1 0 0 0 0	30	.	.	.	.
0 0 1 1 0 0 0 1	31	.	DQ54, 55	DQ56, 57	.
0 0 1 1 0 0 1 0	32	.	.	DQ52, 53	DQ26, 27
0 0 1 1 0 0 1 1	33	.	.	.	.
0 0 1 1 0 1 0 0	34	.	DQ48, 49	DQ18, 19	.
0 0 1 1 0 1 0 1	35	.	.	.	.
0 0 1 1 0 1 1 0	36	.	.	.	.
0 0 1 1 0 1 1 1	37	.	.	.	.
0 0 1 1 1 0 0 0	38	DQ52, 53	.	.	DQ50, 51
0 0 1 1 1 0 0 1	39	.	.	.	.
0 0 1 1 1 0 1 0	3A	.	.	.	.
0 0 1 1 1 0 1 1	3B	.	.	.	.
0 0 1 1 1 1 0 0	3C	.	.	.	.
0 0 1 1 1 1 0 1	3D	.	.	.	.
0 0 1 1 1 1 1 0	3E	.	.	.	.
0 0 1 1 1 1 1 1	3F	.	.	.	.

SYNDROME	HEX	DIMM1	DIMM2	DIMM3	DIMM4
0 1 0 0 0 0 0 0	40		DQ10, 11	CB	.
0 1 0 0 0 0 0 1	41	.	.	.	.
0 1 0 0 0 0 1 0	42	.	.	.	.
0 1 0 0 0 0 1 1	43	.	CB	DQ18, 19	.
0 1 0 0 0 1 0 0	44	.	.	.	.
0 1 0 0 0 1 0 1	45	.	CB	DQ26, 27	.
0 1 0 0 0 1 1 0	46	.	.	DQ32, 33	DQ34, 35
0 1 0 0 0 1 1 1	47	.	.	.	.
0 1 0 0 1 0 0 0	48	.	.	.	.
0 1 0 0 1 0 0 1	49	.	DQ52, 53	DQ40, 41	.
0 1 0 0 1 0 1 0	4A	.	DQ56, 57	DQ50, 51	.
0 1 0 0 1 0 1 1	4B	.	.	.	.
0 1 0 0 1 1 0 0	4C	.	.	DQ58, 59	CB
0 1 0 0 1 1 0 1	4D	.	.	.	.
0 1 0 0 1 1 1 0	4E	.	.	.	.
0 1 0 0 1 1 1 1	4F	.	DQ44, 45	DQ16, 17	.
0 1 0 1 0 0 0 0	50	.	.	.	.
0 1 0 1 0 0 0 1	51	CB	.	.	CB
0 1 0 1 0 0 1 0	52	.	.	DQ28, 29	DQ62, 63
0 1 0 1 0 0 1 1	53	.	.	.	.
0 1 0 1 0 1 0 0	54	.	DQ58, 59	DQ24, 25	.
0 1 0 1 0 1 0 1	55	.	.	.	.
0 1 0 1 0 1 1 0	56	.	.	.	.
0 1 0 1 0 1 1 1	57	.	.	.	.
0 1 0 1 1 0 0 0	58	DQ60, 61	.	.	DQ54, 55
0 1 0 1 1 0 0 1	59	.	.	.	.
0 1 0 1 1 0 1 0	5A	.	.	.	.
0 1 0 1 1 0 1 1	5B	.	.	.	.
0 1 0 1 1 1 0 0	5C	.	.	.	.
0 1 0 1 1 1 0 1	5D	.	.	.	.
0 1 0 1 1 1 1 0	5E	.	.	.	.
0 1 0 1 1 1 1 1	5F	.	.	.	.
0 1 1 0 0 0 0 0	60	.	.	.	.
0 1 1 0 0 0 0 1	61	.	.	DQ14, 15	DQ40, 41
0 1 1 0 0 0 1 0	62	DQ36, 39	DQ4, 5	.	.
0 1 1 0 0 0 1 1	63	.	.	.	.
0 1 1 0 0 1 0 0	64	.	.	.	DQ4, 5
0 1 1 0 0 1 0 1	65	.	.	.	.
0 1 1 0 0 1 1 0	66	.	.	.	.
0 1 1 0 0 1 1 1	67	.	.	.	.
0 1 1 0 1 0 0 0	68	.	.	DQ6, 7	DQ0, 1
0 1 1 0 1 0 0 1	69	.	.	.	.
0 1 1 0 1 0 1 0	6A	.	.	.	.
0 1 1 0 1 0 1 1	6B	.	.	.	.
0 1 1 0 1 1 0 0	6C	.	.	.	.
0 1 1 0 1 1 0 1	6D	.	.	.	.
0 1 1 0 1 1 1 0	6E	.	.	.	.
0 1 1 0 1 1 1 1	6F	.	.	.	.
0 1 1 1 0 0 0 0	70	DQ16, 17	DQ36, 37	.	.
0 1 1 1 0 0 0 1	71	.	.	.	.
0 1 1 1 0 0 1 0	72	.	.	.	.
0 1 1 1 0 0 1 1	73	.	.	.	.
0 1 1 1 0 1 0 0	74	.	.	.	.
0 1 1 1 0 1 0 1	75	.	.	.	.
0 1 1 1 0 1 1 0	76	.	.	.	.
0 1 1 1 0 1 1 1	77	.	.	.	.
0 1 1 1 1 0 0 0	78	.	.	.	.
0 1 1 1 1 0 0 1	79	.	.	.	.
0 1 1 1 1 0 1 0	7A	.	.	.	.
0 1 1 1 1 0 1 1	7B	.	.	.	.
0 1 1 1 1 1 0 0	7C	.	.	.	.
0 1 1 1 1 1 0 1	7D	.	.	.	.
0 1 1 1 1 1 1 0	7E	.	.	.	.
0 1 1 1 1 1 1 1	7F	.	.	.	.

SYNDROME	HEX	DIMM1	DIMM2	DIMM3	DIMM4
1 0 0 0 0 0 0 0	80	CB	DQ34, 35	.	.
1 0 0 0 0 0 0 1	81	.	.	.	.
1 0 0 0 0 0 1 0	82	.	.	.	.
1 0 0 0 0 0 1 1	83	DQ18, 19	.	.	DQ46, 49
1 0 0 0 0 1 0 0	84	.	.	.	.
1 0 0 0 0 1 0 1	85	DQ54, 55	DQ60, 61	.	.
1 0 0 0 0 1 1 0	86	.	.	DQ0, 1	DQ38, 39
1 0 0 0 0 1 1 1	87	.	.	.	.
1 0 0 0 1 0 0 0	88	.	.	.	.
1 0 0 0 1 0 0 1	89	DQ40, 41	.	.	DQ16, 17
1 0 0 0 1 0 1 0	8A	DQ50, 51	DQ20, 21	.	.
1 0 0 0 1 0 1 1	8B	.	.	.	.
1 0 0 0 1 1 0 0	8C	DQ58, 59	.	.	DQ52, 53
1 0 0 0 1 1 0 1	8D	.	.	.	.
1 0 0 0 1 1 1 0	8E	.	.	.	.
1 0 0 0 1 1 1 1	8F	DQ44, 45	.	.	DQ46, 47
1 0 0 1 0 0 0 0	90	.	.	.	.
1 0 0 1 0 0 0 1	91	.	.	DQ20, 21	DQ12, 13
1 0 0 1 0 0 1 0	92	DQ48, 49	.	.	DQ44, 45
1 0 0 1 0 0 1 1	93	.	.	.	.
1 0 0 1 0 1 0 0	94	DQ46, 47	.	.	DQ36, 37
1 0 0 1 0 1 0 1	95	.	.	.	.
1 0 0 1 0 1 1 0	96	.	.	.	.
1 0 0 1 0 1 1 1	97	.	.	.	.
1 0 0 1 1 0 0 0	98	.	.	DQ42, 43	DQ8, 9
1 0 0 1 1 0 0 1	99	.	.	.	.
1 0 0 1 1 0 1 0	9A	.	.	.	.
1 0 0 1 1 0 1 1	9B	.	.	.	.
1 0 0 1 1 1 0 0	9C	.	.	.	.
1 0 0 1 1 1 0 1	9D	.	.	.	.
1 0 0 1 1 1 1 0	9E	.	.	.	.
1 0 0 1 1 1 1 1	9F	.	.	.	.
1 0 1 0 0 0 0 0	A0	.	.	.	.
1 0 1 0 0 0 0 1	A1	DQ28, 29	.	.	DQ22, 23
1 0 1 0 0 0 1 0	A2	DQ24, 25	DQ18, 19	.	.
1 0 1 0 0 0 1 1	A3	.	.	.	.
1 0 1 0 0 1 0 0	A4	DQ56, 57	DQ50, 51	.	.
1 0 1 0 0 1 0 1	A5	.	.	.	.
1 0 1 0 0 1 1 0	A6	.	.	.	.
1 0 1 0 0 1 1 1	A7	.	.	.	.
1 0 1 0 1 0 0 0	A8	DQ20, 21	DQ22, 23	.	.
1 0 1 0 1 0 0 1	A9	.	.	.	.
1 0 1 0 1 0 1 0	AA	.	.	.	.
1 0 1 0 1 0 1 1	AB	.	.	.	.
1 0 1 0 1 1 0 0	AC	.	.	.	.
1 0 1 0 1 1 0 1	AD	.	.	.	.
1 0 1 0 1 1 1 0	AE	.	.	.	.
1 0 1 0 1 1 1 1	AF	.	.	.	.
1 0 1 1 0 0 0 0	B0	DQ4, 5	DQ14, 15	.	.
1 0 1 1 0 0 0 1	B1	.	.	.	.
1 0 1 1 0 0 1 0	B2	.	.	.	.
1 0 1 1 0 0 1 1	B3	.	.	.	.
1 0 1 1 0 1 0 0	B4	.	.	.	.
1 0 1 1 0 1 0 1	B5	.	.	.	.
1 0 1 1 0 1 1 0	B6	.	.	.	.
1 0 1 1 0 1 1 1	B7	.	.	.	.
1 0 1 1 1 0 0 0	B8	.	.	.	.
1 0 1 1 1 0 0 1	B9	.	.	.	.
1 0 1 1 1 0 1 0	BA	.	.	.	.
1 0 1 1 1 0 1 1	BB	.	.	.	.
1 0 1 1 1 1 0 0	BC	.	.	.	.
1 0 1 1 1 1 0 1	BD	.	.	.	.
1 0 1 1 1 1 1 0	BE	.	.	.	.
1 0 1 1 1 1 1 1	BF	.	.	.	.

SYNDROME	HEX	DIMM1	DIMM2	DIMM3	DIMM4
1 1 0 0 0 0 0 0	C0	.	.	.	.
1 1 0 0 0 0 0 1	C1	.	CB	CB	.
1 1 0 0 0 0 1 0	C2	.	DQ30, 31	CB	.
1 1 0 0 0 0 1 1	C3	.	.	.	.
1 1 0 0 0 1 0 0	C4	CB	.	.	DQ56, 59
1 1 0 0 0 1 0 1	C5	.	.	.	.
1 1 0 0 0 1 1 0	C6	.	.	.	.
1 1 0 0 0 1 1 1	C7	.	.	.	.
1 1 0 0 1 0 0 0	C8	.	DQ26, 27	DQ60, 61	.
1 1 0 0 1 0 0 1	C9	.	.	.	.
1 1 0 0 1 0 1 0	CA	.	.	.	.
1 1 0 0 1 1 0 0	CB	.	.	.	.
1 1 0 0 1 1 0 1	CC	.	.	.	.
1 1 0 0 1 1 1 0	CD	.	.	.	.
1 1 0 0 1 1 1 1	CE	.	.	.	.
1 1 0 1 0 0 0 0	CF	.	.	.	.
1 1 0 1 0 0 0 1	D0	DQ42, 43	DQ8, 9	.	.
1 1 0 1 0 0 1 0	D1	.	.	.	.
1 1 0 1 0 0 1 1	D2	.	.	.	.
1 1 0 1 0 1 0 0	D3	.	.	.	.
1 1 0 1 0 1 0 1	D4	.	.	.	.
1 1 0 1 0 1 1 0	D5	.	.	.	.
1 1 0 1 0 1 1 1	D6	.	.	.	.
1 1 0 1 1 0 0 0	D7	.	.	.	.
1 1 0 1 1 0 0 1	D8	.	.	.	.
1 1 0 1 1 0 1 0	D9	.	.	.	.
1 1 0 1 1 0 1 1	DA	.	.	.	.
1 1 0 1 1 1 0 0	DB	.	.	.	.
1 1 0 1 1 1 0 1	DC	.	.	.	.
1 1 0 1 1 1 1 0	DD	.	.	.	.
1 1 0 1 1 1 1 1	DE	.	.	.	.
1 1 1 0 0 0 0 0	DF	.	.	.	.
1 1 1 0 0 0 0 1	E0	DQ36, 37	DQ42, 43	.	.
1 1 1 0 0 0 1 0	E1	.	.	.	.
1 1 1 0 0 0 1 1	E2	.	.	.	.
1 1 1 0 0 1 0 0	E3	.	.	.	.
1 1 1 0 0 1 0 1	E4	.	.	.	.
1 1 1 0 0 1 1 0	E5	.	.	.	.
1 1 1 0 0 1 1 1	E6	.	.	.	.
1 1 1 0 1 0 0 0	E7	.	.	.	.
1 1 1 0 1 0 0 1	E8	.	.	.	.
1 1 1 0 1 0 1 0	E9	.	.	.	.
1 1 1 0 1 0 1 1	EA	.	.	.	.
1 1 1 0 1 1 0 0	EB	.	.	.	.
1 1 1 0 1 1 0 1	EC	.	.	.	.
1 1 1 0 1 1 1 0	ED	.	.	.	.
1 1 1 0 1 1 1 1	EE	.	.	.	.
1 1 1 1 0 0 0 0	EF	.	.	.	.
1 1 1 1 0 0 0 1	F0	.	.	.	.
1 1 1 1 0 0 1 0	F1	.	.	DQ38, 39	DQ32, 33
1 1 1 1 0 0 1 1	F2	DQ0, 1	.	.	DQ14, 15
1 1 1 1 0 1 0 0	F3	.	.	.	.
1 1 1 1 0 1 0 1	F4	DQ12, 13	DQ16, 17	.	.
1 1 1 1 0 1 1 0	F5	.	.	.	.
1 1 1 1 0 1 1 1	F6	.	.	.	.
1 1 1 1 1 0 0 0	F7	.	.	.	.
1 1 1 1 1 0 0 1	F8	.	DQ12, 13	DQ46, 47	.
1 1 1 1 1 0 1 0	F9	.	.	.	.
1 1 1 1 1 0 1 1	FA	.	.	.	.
1 1 1 1 1 1 0 0	FB	.	.	.	.
1 1 1 1 1 1 0 1	FC	.	.	.	.
1 1 1 1 1 1 1 0	FD	.	.	.	.
1 1 1 1 1 1 1 1	FE	.	.	.	.

[그림 11] CNB20HE 신드롬 테이블



[그림 12] 메모리 오류로 인한 서버 오작동

코드는 SEC-DED가 가능한 코드임을 확인하였다. 따라서 일반 SEC-DED 시스템에 비해 칩킬 보정 능력까지 갖추고 있으므로 매우 강력한 오류 정정 능력을 가지고 있으며 실험으로 검증한 서버 중 가장 신뢰할 수 있는 서버 시스템이다.

그림 12는 3년 동안 메모리 오류로 인한 서버의 오작동을 나타낸 것이며 그래프와 같이 서버에서 사용하는 메모리 용량이 1GB에 이르면 패리티 비트를 사용하는 32MB 서버보다 오작동 확률이 더 커진다.(100개 시스템 중 9개 오작동) 그러나, 칩킬 보정을 사용하는 서버는 10,000개 시스템에서 겨우 6개의 오작동을 기록하므로 기존의 시스템에 비해 매우 큰 오류 정정 능력을 보인다[5].

5. 맺음말

점차 무기체계가 정밀화 및 무인화 되어감에 따라 시스템의 데이터 오류가 치명적인 오작동으로 연결될 수 있다. 따라서 무기 체계에 사용되는 시스템의 오

류 데이터 처리능력의 검증이 중요하며 본 논문에서는 서버 시스템의 칩셋 내에 설계된 해밍 코드를 추적하여 시스템에 설계된 오류 정정 알고리즘을 분석할 수 있는 간단하고 새로운 방법을 제시하였다. 제시된 방법을 이용하여 시스템의 오류 정정 알고리즘을 칩셋 제조사 별로 검증하였고 이로써 각 시스템의 노이즈 환경 및 시스템 내부 소자의 오작동에 견디는 시스템의 강인함(robustness)을 분석하였다.

실험에서 한개 비트를 완전히 GND 및 V_{DD}로 연결하여도 실제 서버가 동작함을 확인하여 한개 비트 오류 시 시스템 스스로 정정하여 동작함(SEC)을 검증하였으며 같은 (72,64) 코드를 갖더라도 제조사 별로 오류 정정 및 검출 능력이 다름을 검증하였다.또한 칩킬 보정 방법을 사용하는 시스템은 메모리 소자 한 개를 제거하여도(4비트 동시 제거) 동작할 정도로 발생되는 오류에 강장 강력한 정정 능력이 있음을 검증하였다. 표 1은 실험한 3개 칩셋의 오류정정능력을 정리한 것이다.

[표 1] 칩셋별 오류정정 능력

칩셋	오류정정능력
CNB30LE	SEC
Profusion	SEC-DED
CNB20HE	SEC-DED, 칩킬 보정

향후 체계 시스템에 적용된 서버의 오류 발생 비트의 추적 및 독자적인 오류 정정 시스템 설계에 본 연구 결과가 참조될 수 있을 것으로 판단된다.

참 고 문 헌

[1] Bernard Sklar "Digital Communications", Prentice Hall PTR, 2001, pp.318~356.

[2] MY Hsiao "A Class of Optimal Minimum Odd-weight-column SEC-DED Codes", IBMJ. RES. DEVELOP. July 1970, pp.395~401.

[3] hynix technical data sheet "HYM71V32C735HCT4".

[4] David Locklear "Chipkill Correct Memory Architecture", Dell Enterprise Systems Group, technology brief, 2000.

[5] Timothy J. Dell, "A White Paper on the Benefits of Chipkill-Correct ECC for PC Server Main Memory", IBM Microelectronics Division, July 1997.