2013년 8월 전자공학회 논문지 제 50 권 제 8 호
Journal of The Institute of Electronics Engineers of Korea Vol. 50, NO. 8, August 2013 http://dx.doi.org/10.5573/ieek.2013.50. 8.136

논문 2013-50-8-17

DDR4/GDDR5에서 고속동작을 위한 matrix형 CRC 및 XOR/XNOR

(Matrix type CRC and XOR/XNOR for high-speed operation in DDR4 and GDDR5)

이 중 호*

(JoongHo Lee ©)

요 약

DDR4와 같은 고속동작을 위한 메모리 제품에서, 데이타의 신뢰도 증가를 위해 CRC 기능이 추가되었다. 기존의 CRC 방식은 많은 부가회로 면적과 지연시간이 요구되기 때문에 고속동작의 메모리 제품에서 CRC 계산을 위한 내부 타이밍 마진의 부족현상이 증가한다. 따라서 본 논문에서는 이러한 문제를 해결할 수 있도록 matrix형 CRC 방법을 제시하고 CRC 계산을 빠르게 할 수 있는 XOR/XNOR 게이트를 제시하였다. matrix형 CRC는 모든 홀수 비트오류를 검출 가능하며, 4의 배수비트 오류를 제외한 짝수비트오류도 검출가능하다. 또한 단일오류(single error)에 대해서는 오류 정정이 가능하여 메모리 제품과 시스템간의 CRC 오류로 인한 데이터 재 전송의 부하를 감소시킬 수 있다. 또한 기존 방식대비 부가회로면적을 57% 개선할 수 있다. 제안한 XOR/XNOR는 6개의 TR.(트랜지스터)로 구성하였으며, 기존의 CRC 대비 35%의 면적 오버헤드를 감소시킬 수 있으며, 50%의 게이트 지연을 감소시킬 수 있다.

Abstract

CRC features have been added to increase the reliability of the data in memory products for high-speed operation, such as DDR4. High-speed memory products in a shortage of internal timing margin increases for the CRC calculation. Because the existing CRC requires many additional circuit area and delay time. In this paper, we show that the matrix-type CRC and a new XOR/XNOR gate could be improved the circuit area and delay time. Proposed matrix-type CRC can detect all odd-bit errors and can detect even number of bit errors, except for multiples of four bits. In addition, a single error in the error correction can reduce the burden of re-transmission of data between memory products and systems due to CRC errors. In addition, the additional circuit area, compared to existing methods can be improved by 57%. The proposed XOR gate which is consists of six transistors, it can reduce the area overhead of 35% compared to the existing CRC, 50% of the gate delay can be reduced.

Keywords : matrix type CRC, XOR/XNOR, DDR4, GDDR5, High-speed memory

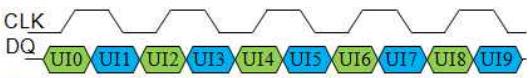
Ⅰ. 서 론

컴퓨터 시스템의 고속 동작에 따라 DRAM(Dynamic Random Access Memory)의 고속화 또한 지속적으로 진행되고 있고, 향후에도 지속적인 동작속도의 개선이

* 정회원, 용인대학교 컴퓨터과학과
(Department of Computer Science, Yongin University)
© Corresponding Author(E-mail: joongho65@yongin.ac.kr)
※ 본 연구는 용인대학교 학술연구 조성비에 의하여 진행된 것임.
접수일자: 2013년3월27일, 수정완료일: 2013년7월19일

요구될 것이다. JEDEC(Joint Electron for Devices Engineering Council)에 표준화된 DDR3 SDRAM 제품은 이미 상용화 되었고, 수년내에 DDR4 SDRAM이 선보일 것이다.[1] 뿐만 아니라 그래픽제품 전용의 고속동작을 위한 목적의 GDDR5도 고속동작 목적으로 개발된 제품이다. 컴퓨터 시스템이나 DRAM 제품이 고속화 될수록 시스템간의 전송된 데이터에서 오류가 발생할 확률이 증가한다. 이러한 문제를 해결하기위해 DDR4나 GDDR5 제품에서는 CRC(Cyclic Redundancy Check)와 DBI(Data Bus Inversion) 기능[2~3, 7~8]을 추가하여 데이터 전송 오류시 오류 검출 여부를 시스템에 전송하여 데이터를 다시 전송받을 수 있도록 하였다. CRC구현을 위해 DDR4와 GDDR5에서는 동일한 8비트-CRC 코드를 사용하며, ATM-8 HEC 코드로서 다항식 x^8+x^2+x+1을 사용한다.

고속동작 시스템에서 CRC는 데이터의 신뢰도를 증가시키기 위해 필수적이지만 메모리 장치내에 CRC코드의 부호기 및 복호기(encoder and decoder)가 내장되어야 하므로 부가면적이 증가 할뿐만 아니라 코드 해석시 요구되는 부가적인 시간이 DRAM내부 동작의 타이밍 마진(margin)에 부담을 주고 있고, 제품이 고속화 할수록 타이밍 마진의 부담은 더욱 증가한다. ATM-8 HEC 코드는 x8 DQ 및 DBI 입력에 대해 burst length 8(8UI)의 데이터를 코드화하여 구성한다. DDR4의 경우 CRC 데이터 비트가 9번째 UI에 추가되어 실제로 9UI가 필요하지만, 하나의 클럭 사이클이 2UI로 구성되기 때문에 10번째 UI는 half 클럭 사이클을 사용할 수 없으므로 10번째 UI는 데이터 1로 고정하여 총 10UI로 데이터를 구성한다. 결론적으로 DQ 데이터는 5CLK 동안 burst length8 (BL8) + CRC비트로 구성되어 데이터 셋(set)이 완성되는데, 그림 1에 DDR4에 적용되고 있는 DQ data와 CRC 비트 구성 예를 나타내었다.



DQ	UI0	UI1	UI2	UI3	UI4	UI5	UI6	UI7	UI8	UI9
0	d0	d1	d2	d3	d4	d5	d6	d7	CRC0	1
1	d8	d9	d10	d11	d12	d13	d14	d15	CRC1	1
2	d16	d17	d18	d19	d20	d21	d22	d23	CRC2	1
3	d24	d25	d26	d27	d28	d29	d30	d31	CRC3	1
4	d32	d33	d34	d35	d36	d37	d38	d39	CRC4	1
5	d40	d41	d42	d43	d44	d45	d46	d47	CRC5	1
6	d48	d49	d50	d51	d52	d53	d54	d55	CRC6	1
7	d56	d57	d58	d59	d60	d61	d62	d63	CRC7	1
DBI	d64	d65	d67	d68	d69	d70	d71	d72	CRC8	1

그림 1. DDR4에 대한 8UI DQ 데이터와 CRC의 구성
Fig. 1. the configuration of 8UI DQ data and the CRC for DDR4.

그림 1의 72개(64 DQ + 8 DBI data)의 데이터요소에 대해 데이터와 CRC 패리티(parity)비트를 구성한다. 따라서 하나의 CRC비트를 위해 41개의 내부 GIO(Global I/O)데이터 입력과 이를 입력으로 하는 2-input XOR 게이트로 로직을 구성한다. 하나의 CRC 비트를 계산하기 위해 40개의 XOR 게이트로 구성되며, 그림 2에 기존의 DDR4 및 GDDR5에 적용된 CRC 로직 예를 나타내었으며, DQ0~DQ7에 대한 CRC0~CRC7로 구성되어있다. 그림 1에서 CRC 비트가 모두 9개 이므로 360개의 XOR 게이트가 소요되며, CRC 입력에서 최종 출력까지 6단의 XOR 게이트 delay가 소요된다. 또한 DBI를 위해서 4단의 XOR 게이트 delay가 소요된다. DDR4에서 데이터가 연속적으로 전송될 경우 CRC 동작 시간은 tCCD(CAS to CAS delay=5nCK)를 초과하지 않아야 하는데, 실제 CRC 계산 시간은 CRC용 비트 1CK를 제외한 4nCK보다 작아야 한다. 또한 DRAM 내부에서 각 요소간 데이터 전달 시간을 제외하면 4nCK(32Gbps 경우 1.25ns)보다 작은 값에 CRC 계산이 완료되어야 한다. CRC와 DBI가 동시에 enable될 경우 CRC와 DBI

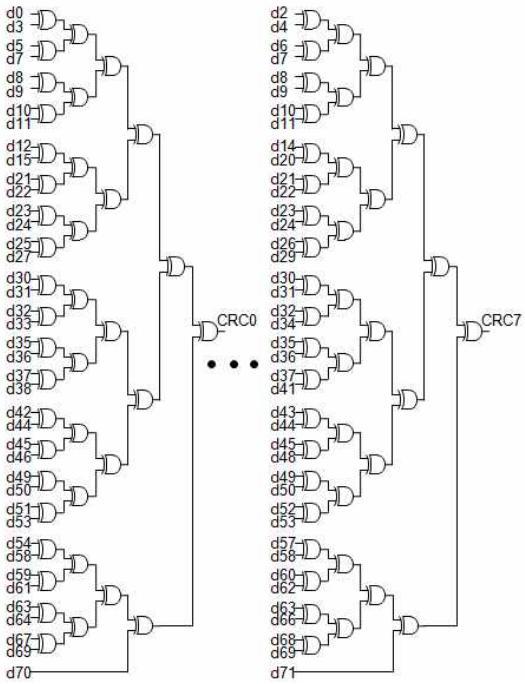


그림 2. DDR4 CRC 로직회로
Fig. 2. DDR4 CRC logic circuit.

계산을 위해서 10단의 XOR가 요구되며, 3.2Gbps 동작의 경우 이를 만족하기 위해 XOR 각 1단의 게이트 지연시간은 ~120ps이내를 만족해야한다. 이것은 slow PVT동작 조건을 만족하기에 더욱 열악한 상황에 놓이며, 게다가 향후 1V이하 동작조건의 제품에서는 더욱 동작마진을 확보하기가 어려워질 전망이다.

본 연구에서는 고속동작을 위한 CRC코드 구성이 간단하고 쉽게 구성 할 수 있도록 하는 matrix형 CRC 방식을 제안하였다. 제안한 방식은 부가회로 면적을 줄일 수 있고 CRC 코드의 해석 시간을 줄일 수 있도록 하여 CRC 구성이 효과적일 수 있도록 개선하였다. 또한 CRC 구현을 위한 부가회로 면적 축소를 위해 효과적인 XOR 로직 게이트를 제안하여 XOR 게이트의 부가면적을 감소시켰으며, 이로 인해 XOR 게이트의 시간지연을 감소할 수 있도록 하였다.

Ⅱ. CRC 구성

본 방식은 기존의 DDR4 SDRAM에서 CRC의 기본 구성을 유지한다. 즉, DQ 데이터는 CRC를 포함하여 10UI로 기존과 동일하게 구성하며, 기존에는 10번째 UI를 1로 고정하였으나 본 방식에서는 10번째 UI도 CRC 체크용 비트로 사용한다. write시 시스템에서 전송된 데이터에 대해 CRC 복호기에서 오류 발생여부를 검출하며, read,시 시스템에 전송할 데이터에 대해 CRC 부호기에서 코드워드(codeword)를 발생하여 전송한다. 본 방식은 CRC 구성시 matrix형으로 체크 비트를 구성함으로서 부가면적을 줄일 수 있도록 하여 CRC 계산시간을 감소시킬 수 있도록 하였다. 그림 3에 본 방식의 구성을 나타내었다.

CRC 구성을 위해 제안한 코드의 구성 예는 다음과 같다. DQ0에 8개의 UI0~UI7 데이터(d0, d1, d2, ~, d7)에 대한 패리티로 CRC0을 구성하며 로직 합이 0이 되도록 한다(그림 3의 DQ0에 대한 가로방향). 또한 UI0에 대한 세로 방향으로 8개 DQ 데이터(d0, d8, d16, ~, d64)에 대해 CRC9를 구성하여 로직 합이 0이 되도록 구성한다. 따라서 d0 데이터에 대해 CRC0과 CRC9에서 중복 체크가 가능하여 중복 비트오류(이중오류 이상)에 대해 검출 가능하다. 모든 DQ 입력의 CRC 구성은 그림 3의 가로 방향과 세로 방향에 대해 동일한 방식으로

DQ	UI0	UI1	UI2	UI3	UI4	UI5	UI6	UI7	UI8	UI9
0	d0	d1	d2	d3	d4	d5	d6	d7	CRC0	CRC9
1	d8	d9	d10	d11	d12	d13	d14	d15	CRC1	CRC10
2	d16	d17	d18	d19	d20	d21	d22	d23	CRC2	CRC11
3	d24	d25	d26	d27	d28	d29	d30	d31	CRC3	CRC12
4	d32	d33	d34	d35	d36	d37	d38	d39	CRC4	CRC13
5	d40	d41	d42	d43	d44	d45	d46	d47	CRC5	CRC14
6	d48	d49	d50	d51	d52	d53	d54	d55	CRC6	CRC15
7	d56	d57	d58	d59	d60	d61	d62	d63	CRC7	CRC16
DBI	d64	d65	d67	d68	d69	d70	d71	d72	CRC8	1

그림 3. matrix형 CRC의 구성
Fig. 3. the configuration of matrix-type CRC.

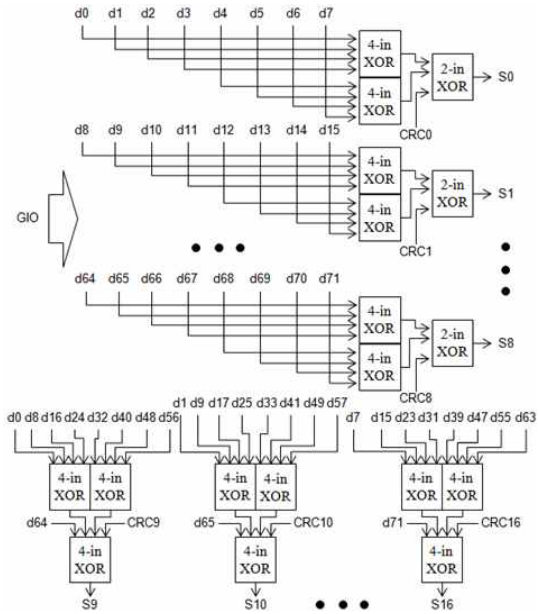


그림 4. 제안한 CRC 복호기
Fig. 4. proposed CRC decoder.

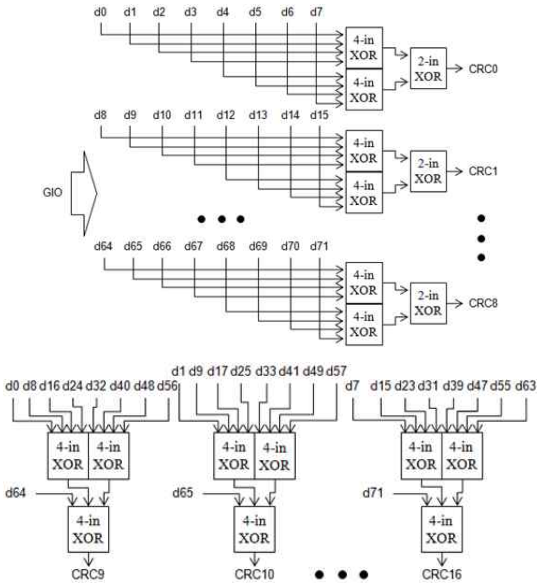


그림 5. 제안한 CRC 부호기
Fig. 5. proposed CRC encoder.

구성된다. 이때 DBI 입력도 CRC 구성에 포함하며, DBI의 9번째 UI는 로직 1로 고정한다. 결론적으로 모든 데이터 비트에 대해 가로 방향과 세로 방향에 각각의 CRC 체크 비트가 존재하여 DQ 및 DBI 데이터에 대해 행과 열에 존재하는 홀수 및 짝수 중복오류를 검출 할 수 있다. 각 코드의 구성에 대한 syndrome 구성은 다음과 같으며, 각 syndrome(S0, S1, ... , S16)은 로직 합이 0인 값을 가지도록 한다. 제안한 CRC 구성 방식에 대한 복호기와 부호기를 그림 4와 5에 나타내었다. syndrom값 S0, S1, ... , S16중 하나 이상 로직 1값이 발생시 오류가 발생했음을 알 수 있고, 이에 따라 시스템에 오류 여부를 알려주는(DDR4에서 Alert신호발생) 등의 후속 조치가 발생된다.

codeword(9비트) =
data bit(8비트) + parity bit(CRC 1비트) (1)

$$
\begin{aligned}
&d0 + d8 + d16 + d24 + d32 + d40 + d48 + d56 + d64 + CRC9 = S9\\
&d1 + d9 + d17 + d25 + d33 + d41 + d49 + d57 + d65 + CRC10 = S10\\
&d2 + d10 + d18 + d26 + d34 + d42 + d50 + d58 + d66 + CRC11 = S11\\
&d3 + d11 + d19 + d27 + d35 + d43 + d51 + d59 + d67 + CRC12 = S12\\
&d4 + d12 + d20 + d28 + d36 + d44 + d52 + d60 + d68 + CRC13 = S13\\
&d5 + d13 + d21 + d29 + d37 + d45 + d53 + d61 + d69 + CRC14 = S14\\
&d6 + d14 + d22 + d30 + d38 + d46 + d54 + d62 + d70 + CRC15 = S15\\
&d7 + d15 + d23 + d31 + d39 + d47 + d55 + d63 + d71 + CRC16 = S16\\
&d0 + d1 + d2 + d3 + d4 + d5 + d6 + d7 + CRC0 = S0\\
&d8 + d9 + d10 + d11 + d12 + d13 + d14 + d15 + CRC1 = S1\\
&d16 + d17 + d18 + d19 + d20 + d21 + d22 + d23 + CRC2 = S2\\
&d24 + d25 + d26 + d27 + d28 + d29 + d30 + d31 + CRC3 = S3\\
&d32 + d33 + d34 + d35 + d36 + d37 + d38 + d39 + CRC4 = S4\\
&d40 + d41 + d42 + d43 + d44 + d45 + d46 + d47 + CRC5 = S5\\
&d48 + d49 + d50 + d51 + d52 + d53 + d54 + d55 + CRC6 = S6\\
&d56 + d57 + d58 + d59 + d60 + d61 + d62 + d63 + CRC7 = S7\\
&d64 + d65 + d66 + d67 + d68 + d69 + d70 + d71 + CRC8 = S8
\end{aligned}
$$

1. 오류 검출(error detect)

본 matrix형 CRC구성방식의 오류검출 능력은 다음과 같으며, 오류검출 범위에 대한 예를 그림 6에 나타내었다.

(1) 단일오류 검출(single error detect)

그림 6의 위치 ❶에서 d0 데이터에 오류가 발생한 경우의 예를 나타내었으며, CRC0과 CRC9에 의해 S0, S9가 로직 1값을 가져 오류 검출이 가능하다. 동일한 방식으로 모든 단일 데이터 비트에 대해 단일오류에 대한 검출이 가능하다.

(2) 이중오류 검출(double error detect)

그림 6의 위치 ❷에서 d18와 d19 데이터에 오류가 발생한 경우이며, CRC2에서는 검출이 불가하나, CRC11과 CRC12에 의해 S11과 S12가 로직 1값을 가져 오류 검출이 가능하다. 동일한 방식으로 모든 데이터 비트에 대해 이중오류 검출이 가능하다.

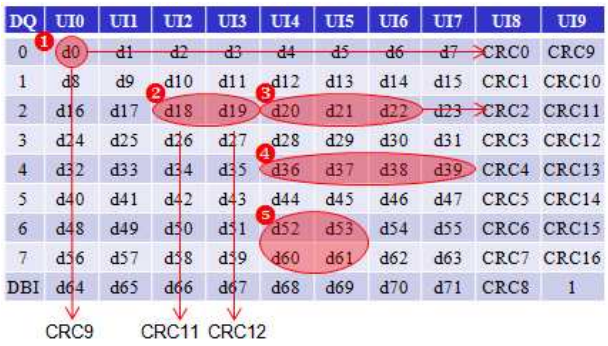


그림 6. 오류 검출의 예
Fig. 6. example of the error detection.

(3) 삼중오류 검출(triple error detect)

그림 6의 위치 ❸에서 d20, d21 및 d22 데이터에 오류가 발생한 예이며, CRC2, CRC13, CRC14 및 CRC15에 의해 S2, S13, S14 및 S15가 로직 1값을 가져 오류 검출이 가능하다. 동일한 방식으로 모든 데이트 비트에 대해 삼중오류 검출이 가능하다.

(4) 사중오류 검출(quadruple error detect)

그림 6의 위치 ❹에서 d36, d37, d38 및 d39 데이터에 오류가 발생한 예이며, CRC13~CRC16에 의해 오류 검출이 가능하다. 그러나 위치 ❺(d52, d53, d60, d61)와 같은 형태의 오류에 대해서는 S6, S7, S15 및 S16이 모두 로직 0이 되어 오류 검출이 불가능하다. 앞의 예와 같이 이중오류가 중복되어 발생한 사중오류에 대해서는 오류검출이 불가하여 전체 사중고장중 부분 검출이 가능하다. 그림 7에 사중오류의 유형을 나타내었으며, 유형 중 (C), (d)와 (g)는 오류 검출이 불가능한 예이다. 그림에서 보다시피 행과 열 방향으로 홀수개의 오류비트 조합이 존재하면 오류 검출이 가능하다. 단, 행과 열을 교환하여 같은 패턴은 동일한 유형으로 본다.

결론적으로 제안한 matrix형 CRC방식은 모든 홀수 비트에 대해 그림 3의 구성에서 보다시피 x축과 y축의 어떤 방향이든지 홀수비트의 구성이 존재하기 때문에 오류검출이 가능하다. 또한 모든 짝수 비트중 4의 배수 비트오류에 대해서는 부분적으로 오류검출이 불가능하며, 사중오류 경우 그림 7의 유형으로부터 모든 사중오

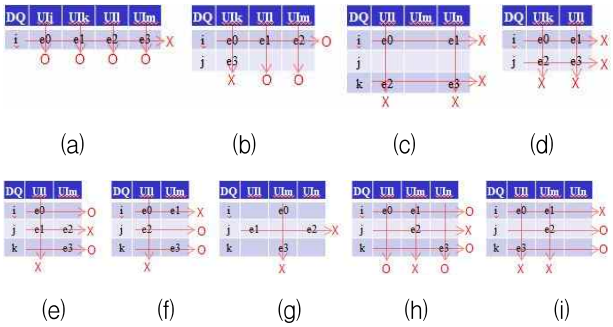


그림 7. 사중오류의 유형
Fig. 7. type of quadruple error

표 1. 제안한 Matrix type CRC의 오류검출 능력
Table 1. error detection capability of Matrix type CRC.

CRC Size(bit)	CRC	Bits Error에 대한 검출 불가능 경우의 수					
		1bits	2bits	3bits	4bits	5bits	6bits
16	CCITT-16	0	0	0	84	0	2430
15	CAN	0	0	0	0	0	4314
12	CRC-12	0	0	0	575	0	28809
8	DARC-8	0	66	0	2039	13122	124248
8	CRC-8	0	0	0	2984	0	253084
16	Matrix type	0	0	0	1461	0	0
7	CRC-7	0	0	216	2690	27051	226856

류의 데이터비트 구성가능 조합의 경우의 수로부터 검출 불가능한 경우의 수는 1461가지이다. ATM-8 HEC 방식은 표 1의 DARC-8과 CRC-8의 중간 특성을 가지는데,[9] 제안한 Matrix type이 가장 뛰어난 오류 검출능력을 가진다.

2. 오류 정정(error correct)

본 논문의 CRC 방식은 단일오류에 대해 오류 비트 정정기능을 가진다. matrix형식으로 각 DQ데이타에 대해 행과 열에 존재하는 체크비트 때문에 오류 정정이 가능하다. 오류 정정의 순서는 먼저 오류 비트 위치를 찾아내고 찾아낸 오류 비트를 정정한다. 그림 8에 오류 정정 회로를 나타내었다.

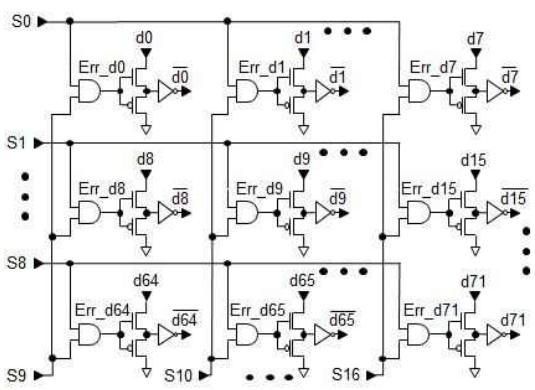


그림 8. 오류 정정 회로부
Fig. 8. error correction circuitry.

(1) 단일오류 정정(single error correct)

그림 6의 위치 ❶의 오류에 대해 S0과 S9에 의해 Err_d0가 로직 1이 되어 오류 위치를 찾아낸다. 또한 transfer Tr.을 turn on 시켜 d0를 통과 시킨후 d0의 보수를 취해서 d0를 GIO에 실어주면 데이터 오류를 정정할 수 있다. 동일한 방법으로 모든 단일 오류 비트에 대해 오류 정정이 가능하다. 동일한 방식으로 모든 단일 데이터 비트에 대해 단일오류에 대한 정정이 가능하다.

(2) 이중오류 정정(double error correct)

그림 6의 위치 ❷의 오류에 대해 S11과 S12가 로직 1 값을 가지나 S2는 로직 0이므로 오류위치를 찾을 수 없다. 따라서 이중오류에 대해서는 오류정정이 불가하다.

(3) 삼중오류 정정(triple error correct)

그림 6의 위치 ❸의 오류에 대해 S2, S13, S14 및 S15에 의해 Err_d20, Err_d21 및 Err_d22가 로직 1이 되어 오류 위치를 찾아낼 수 있으므로 단일 오류 정정과 동일한 방법으로 오류 정정이 가능하다. 그림 9는 삼중오류의 유형으로 (a)의 경우 오류정정이 가능하나, (b)의 경우 e2 비트에 대해서만 오류정정이 가능하며, (c)의 경우 e1, e3 비트에 대해서만 오류정정이 가능하여 전체 삼중 오류에 대한 정정이 불가능하다.

결론적으로 단일오류에 대해서는 모두 오류 정정이 가능하고 이중오류 이상에 대해서는 부분적인 정정만 가능하다. 따라서 단일 고장에 대해서만 정정기능을 수행한다.

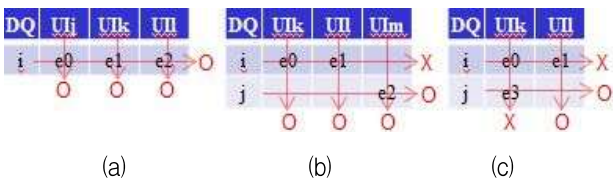
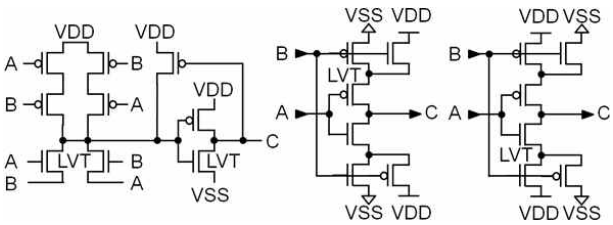


그림 9. 삼중오류 유형
Fig. 9. type of triple error.

Ⅲ. XOR/XNOR 구성

앞에서도 언급하였듯이 CRC와 DBI의 계산은 DRAM이 고속화 될수록 내부 동작 타이밍 마진 확보가 어려워진다. 이러한 문제를 개선하기 위해 CRC나 DBI 계산에서 기본소자인 XOR의 게이트 단을 줄여 게이트 지연을 개선하도록 제안하였다. DDR4에서 read/write CRC 및 DBI를 위해 약 700 XOR 게이트를 사용하는데, 기존의 9개 Tr.(트랜지스트)를 사용할 경우 부가회로 면적에 많은 부담을 준다.[4~6] 본 논문에서 제안한 방식은 6개의 Tr.를 사용하여 2-input XOR 및 XNOR 게이트를 효과적으로 구현하였으며, 그림 10에 나타내었다. 그림 10의 a)는 기존의 XOR 구성방식이며 SK-hynix GDDR5(VDD=1.5V, 4Gbps)에 채용된 방식이다.[2] 이 방식은 XOR 로직구현을 위해 2단의 게이트를 거친다. 제안한 XOR는 기존의 XOR 방식들에 비해 가장 작은 부가회로 면적을 가지는 방식중의 하나이며, 게이트 단수를 줄일 수 있는 구성방식이어서 동작속도 또한 개선할 수 있도록 하였다. 그림 11에 4-input XOR 게이트 구성을 나타내었다. XOR에서 PMOS Tr.을 통과할 때 신호감쇄를 보완하기 위하여 신호 A, B를 게이트 입력으로 하는 직렬로 연결된 PMOS 소자에 대해 Low-threshold Voltage 소자(LVT PMOS)를 사용하여 개선하도록 하였다. XNOR의 경우 직렬로 연결된 NMOS Tr.에 대해 LVT NMOS을 소자를 사용한다. 기존의 XOR게이트 로직(그림 10. a)에서도 LVT NMOS를 사용하여 A,B=1 일때 전압강하를 방지하도록 하였으며 1V이하 동작전압에서는 추가적인 개선이 요구된다. 그림 11에 전압별 지연시간의 결과를 나타내었다. 제안한 방식은 기존의 XOR 게이트 대비 면적 오버헤드를 35% 개선할 수 있으며, 출력단의 Inverter를 제거함으로써 기존의 2단 지연회로에서 1단 지연회로 방식으로 개선하여 CRC 계산을 위한 타이밍 마진을 개선할 수 있다.



(a)기존 2-input XOR (b)제안한 2-input XOR/XNOR

그림 10. 2-input XOR/XNOR 게이트
Fig. 10. 2-input XOR/XNOR gate.

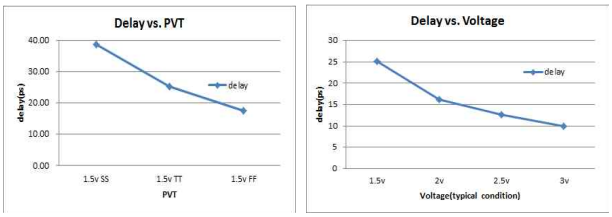


그림 11. 2-input XOR 게이트의 전압별 지연시간특성
Fig. 11. 2-input XOR gate delay characteristics vs. voltage.

Ⅳ. 결 론

기존의 DDR4에서 read/write CRC 및 DBI를 위해 약 700 XOR 게이트를 사용하고 있다. 제안한 matrix-type CRC 방식에서는 read/write CRC 및 DBI를 위해 300 XOR 게이트만 사용하여 부가회로 면적을 기존대비 57% 개선하였다. 제안한 방식은 CRC를 효과적으로 수행하면서 모든 홀수비트 오류에 대해 검출이 가능하고, 모든 짝수 비트 중 4의 배수비트오류(4,8,12,16,...비트 오류)에 대해서는 부분적으로 오류검출이 불가능하며, 사중오류(quadruple error)에 대해 1461가지의 경우에 대해 검출 불가능하다. 표 1에 데이터 비트에 대해 비트 오류의 모든 가능한 조합의 경우의 수로부터 검출 불가능한 경우의 수를 기존의 CRC 방식과 비교하였다. 또한 단일오류(single error)에 대한 정정(correction)이 가능하도록 하여, CRC 단일오류시 메모리 동작을 정지시키고 시스템에 오류여부를 알려주는 “Alert” 동작을 제거할 수 있어서 시스템과 메모리 제품간의 데이터 재전송의 부하를 개선할 수 있다. 제안한 방식에 대해 기존 방식과 CRC 구현시 장단점을 표 2에 나타내었다. CRC의 빠른 계산을 위한 새로운 XOR 게이트를 제안하여, XOR 게이트 부가회로 면적을 기존의 방식(SK Hynix GDDR5) 대비 35% 축소하였고,

표 2. 제안한 Matrix type과 기존의 CRC방식의 비교
Table 2. comparison of Matrix type and existing CRC.

구분		CRC		XOR type			
		CRC 방식	CRC size	size	전압특성	전류특성	속도
기존	DDR4	ATM-8 HEC	700 gate	6	좋음	좋음	좋음
	GDDR5	ATM-8 HEC	700 gate	9	나쁨	좋음	나쁨
제안		Matrix type	300 gate	6	나쁨	나쁨	좋음

XOR 단을 1단으로 구성하여 기존의 2단 지연 XOR 대비 타이밍 마진을 절감할 수 있도록 하였으나 전류소모 및 전압 특성개선에 대한 추가적인 개선이 필요하다.

REFERENCES

[1] D. Graham-Smith, “IDF: DDR3 won’t catch up with DDR2 during 2009,” in PC Pro, Aug. 2008.

[2] S. Yoon, B. Kim, Y. Kim, B. Chung, “A Fast GDDR5 Read CRC Calculation Circuit with Read DBI Operation,” IEEE Asian Solid-Sate Circuits Conference, pp. 249-252, November, 2008.

[3] Seung-Jun Bae, Kwang-Il Park, “An 80 nm 4 Gb/s/pin 32 bit 512 Mb GDDR4 Graphics DRAM With Low Power and Low Noise Data Bus Inversion,” IEEE Journal of Solid-State Circuits, Vol.43, pp. 121-131, January, 2008.

[4] K. Lin, C. Wu, “A Low-cost Realization of Multiple-input Exclusive-OR gates,” ASIC Conference and Exhibit, Proceedings of the Eighth Annual IEEE International, pp. 307 - 10, September. 1995.

[5] H. Bui, Y. Wang, and Y. Jiang, “Design and Analysis of Low-Power 10-Transistor Full Adders Using Novel XOR-XNOR Gates,” IEEE Transactions on Circuits and Systems II, Vol. 49, No. 1, pp. 25 - 0, January, 2002.

[6] H. Bui, Y. Wang, and Y. Jiang, “New 4-Transistor XOR and XNOR Designs,” in Proc. 2nd IEEE Asia Pacific Conf. ASICs, pp.25 - 8, August, 2000.

[7] Kibong Koo, et al., “A 1.2V 38nm 2.4Gb/s/pin 2Gb DDR4 SDRAM with Bank Group and x4 Half-Page Architecture” IEEE International Solid State Circuits Conference, pp. 40 - 41, Feb. 2012.

[8] Kyomin Sohn, et al., “A 1.2V 30nm 3.2Gb/s/pin 4Gb DDR4 SDRAM With Dual-Error Detection and PVT-Tolerant Data-Fetch Scheme” IEEE Journal of Solid State Circuits, Vol.48, pp. 168-177, Jan. 2013.

[9] Koopman, P. and Chakravarty T. “Cyclic Redundancy Code(CRC) Polynomial Selection For Embedded Networks” 2004 International Conference on Dependable Systems and Networks, pp. 145-154 28 June-1 July 2004.

저 자 소 개

이 중 호(정회원)

1988년 울산대학교 전자및전산기공학과 학사 졸업.

1990년 울산대학교 전자및전산기공학과 석사 졸업.

1994년 울산대학교 전자및전산기공학과 박사 졸업.

1994년～2012년 하이닉스 반도체 수석연구원

2012년～현재 용인대학교 컴퓨터과학과 조교수

<주관심분야 : 반도체 설계 및 테스트, 컴퓨터 디지털 통신, 음성인식, 인공지능>