

Department of Electrical and Computer Engineering

University of Colorado at Boulder

ECEN5623 - Real Time Embedded Systems



Homework 4

Submitted by

Parth Thakkar

Submitted on April 19, 2024

Contents

[List of Figures](#) 1

[List of Tables](#) 1

[1 Question 1](#) 2

[Answer](#) 2

[2 Question 2](#) 7

[Answer](#) 7

[3 Question 3](#) 11

[Answer](#) 12

[4 Referance](#) 13

[Appendices](#) 14

[A Excel Sheet](#) 14

[A.1 Q1](#) 14

List of Figures

List of Tables

1 Question 1

Q: Develop an example of a 32-bit Hamming encoded word (39 bits total) and show a correctable SBE scenario. Show the data word in a table like Figure 5.6 in the book.

Answer: For example, let's take a 32-bit integer:

- Binary: 0b10110011010100101100110101001011
- Hexadecimal: 0xB352CD4B
- Decimal: 3008548171

_ _ _ 1 _ 1 0 1 _ 0 0 1 0 1 0 1 _ 1 0 0 1 1 0 1 0 0 1 0 1 0 1 1 _ 0 0 1 1 0 1

I can't fit image in pdf but there is excel sheet in the same zip folder

Single Bit Error (Bit changed at 9)												
bit	D	0	1	2	3	4	5	6	7	8	9	10
1	p01		X		1		1	0	1	X	0	0
2	p02		X	X	1				1			
4	p03					X						
8	p04						1	0	1			
16	p05											
32	p06											
ED												

Single Bit Error (Bit changed at 9)												
SYN	In. Data	0	1	2	3	4	5	6	7	8	9	10
0b001001												
c01	1				1		1		1	1	1	
c02	0		1		1							
c03	0		1									
c04	1						1	0	1			
c05	0											
c06	0											
p01	1											
ED		0	1	0	1	0	1	0	1	1	0	0

With that number, we need to calculate all the parities:

$$\begin{aligned}
p1 &= \text{XOR}(d1, d2, d4, d5, d7, d9, d11, d12, d14, d16, d18, d20, d22, d24, d26, d27, d29, d31) \\
p2 &= \text{XOR}(d1, d3, d4, d6, d7, d10, d11, d13, d14, d17, d18, d21, d22, d25, d26, d28, d29, d32) \\
p3 &= \text{XOR}(d2, d3, d4, d8, d9, d10, d11, d15, d16, d17, d18, d23, d24, d25, d26, d30, d31, d32) \\
p4 &= \text{XOR}(d5, d6, d7, d8, d9, d10, d11, d19, d20, d21, d22, d23, d24, d25, d26) \\
p5 &= \text{XOR}(d12, d13, d14, d15, d16, d17, d18, d19, d20, d21, d22, d23, d24, d25, d26) \\
p6 &= \text{XOR}(d27, d28, d29, d30, d31, d32)
\end{aligned}$$

Here are the parity equations with the actual data bit values and the calculated parity bits: Given the data bits:

_ _ _ 1 _ 1 0 1 _ 0 0 1 0 1 0 1 _ 1 0 0 1 1 0 1 0 0 1 0 1 0 1 1 _ 0 0 1 1 0 1

The parity bits can be calculated as follows:

$$\begin{aligned}
p1 &= \text{XOR}(d1, d2, d4, d5, d7, d9, d11, d12, d14, d16, d18, d20, d22, d24, d26, d27, d29, d31) \\
&= \text{XOR}(1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 0, 0, 0, 1, 0, 1, 0) \\
&= 1 \\
p2 &= \text{XOR}(d1, d3, d4, d6, d7, d10, d11, d13, d14, d17, d18, d21, d22, d25, d26, d28, d29, d32) \\
&= \text{XOR}(1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 1, 1, 0, 1, 1, 0, 1, 1) \\
&= 0 \\
p3 &= \text{XOR}(d2, d3, d4, d8, d9, d10, d11, d15, d16, d17, d18, d23, d24, d25, d26, d30, d31, d32) \\
&= \text{XOR}(1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1) \\
&= 0 \\
p4 &= \text{XOR}(d5, d6, d7, d8, d9, d10, d11, d19, d20, d21, d22, d23, d24, d25, d26) \\
&= \text{XOR}(0, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1, 1) \\
&= 1 \\
p5 &= \text{XOR}(d12, d13, d14, d15, d16, d17, d18, d19, d20, d21, d22, d23, d24, d25, d26) \\
&= \text{XOR}(1, 0, 0, 1, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1, 1) \\
&= 0 \\
p6 &= \text{XOR}(d27, d28, d29, d30, d31, d32) \\
&= \text{XOR}(0, 0, 1, 1, 0, 1) \\
&= 1
\end{aligned}$$

The whole word parity bit (pW) can be calculated by XORing all the data bits and parity bits:

$$\begin{aligned}
pW &= \text{XOR}(p1, p2, d1, p3, d2, d3, d4, p4, d5, d6, d7, d8, d9, d10, d11, \\
&\quad p5, d12, d13, d14, d15, d16, d17, d18, d19, d20, d21, d22, d23, d24, d25, d26, \\
&\quad p6, d27, d28, d29, d30, d31, d32) \\
&= \text{XOR}(1, 0, 0, 1, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 1, 1, 0, 0, 1, 1, 0, 1, 0, 0, 1, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 1) \\
&= 0
\end{aligned}$$

Therefore, the parity bits and the whole word parity bit for the given data bits are:

pW	p1	p2	d1	p3	d2	d3	d4	p4	d5	d6	d7	d8	d9	d10	d11
0	1	0	1	0	1	0	1	1	0	0	1	0	1	0	1

p5	d12	d13	d14	d15	d16	d17	d18	d19	d20	d21	d22	d23	d24	d25	d26
0	1	0	0	1	1	0	1	0	0	1	0	1	1	0	

p6	d27	d28	d29	d30	d31	d32
1	0	0	1	1	0	1

and we are switching one bit at bit position d5(9th bit) so bit position d5(9th bit) is 0 in our case and we are flipping that so we would get

- Binary: 0b10110011010100101100110101011011
- Hexadecimal: 0xB352CD5B
- Decimal: 3008548187

-- -- -- 1 -- 1 0 1 -- 1 0 1 0 1 0 1 -- 1 0 0 1 1 0 1 0 0 1 0 1 0 1 1 -- 0 0 1 1 1 1

$$\begin{aligned}
p1 &= \text{XOR}(d1, d2, d4, d5, d7, d9, d11, d12, d14, d16, d18, d20, d22, d24, d26, d27, d29, d31) \\
&= \text{XOR}(1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 0, 0, 1, 0, 1, 0) \\
&= 0 \\
p2 &= \text{XOR}(d1, d3, d4, d6, d7, d10, d11, d13, d14, d17, d18, d21, d22, d25, d26, d28, d29, d32) \\
&= \text{XOR}(1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 1, 1, 0, 1, 1, 0, 1, 1) \\
&= 0 \\
p3 &= \text{XOR}(d2, d3, d4, d8, d9, d10, d11, d15, d16, d17, d18, d23, d24, d25, d26, d30, d31, d32) \\
&= \text{XOR}(1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1) \\
&= 0 \\
p4 &= \text{XOR}(d5, d6, d7, d8, d9, d10, d11, d19, d20, d21, d22, d23, d24, d25, d26) \\
&= \text{XOR}(1, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1, 1) \\
&= 0 \\
p5 &= \text{XOR}(d12, d13, d14, d15, d16, d17, d18, d19, d20, d21, d22, d23, d24, d25, d26) \\
&= \text{XOR}(1, 0, 0, 1, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1, 1) \\
&= 0 \\
p6 &= \text{XOR}(d27, d28, d29, d30, d31, d32) \\
&= \text{XOR}(0, 0, 1, 1, 0, 1) \\
&= 1
\end{aligned}$$

The whole word parity bit (pW) can be calculated by XORing all the data bits and parity bits:

$$\begin{aligned}
pW &= \text{XOR}(p1, p2, d1, p3, d2, d3, d4, p4, d5, d6, d7, d8, d9, d10, d11, \\
&\quad p5, d12, d13, d14, d15, d16, d17, d18, d19, d20, d21, d22, d23, d24, d25, d26, \\
&\quad p6, d27, d28, d29, d30, d31, d32) \\
&= \text{XOR}(0, 0, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 0, 1, 0, 0, 1, 0, 1, 1, 0, 1, 0, 0, 1, 1, 1, 1) \\
&= 1
\end{aligned}$$

Therefore, the output with the parity bits for the modified data with the flipped bit at position 4 is:

pW	p1	p2	d1	p3	d2	d3	d4	p4	d5	d6	d7	d8	d9	d10	d11	p5	d12	d13	d14	d15
0	0	0	1	0	1	0	1	0	1	1	1	1	1	0	1	1	1	0	0	1

d16	d17	d18	d19	d20	d21	d22	d23	d24	d25	d26	p6	d27	d28	d29	d30	d31	d32
1	0	1	0	0	1	0	1	1	0	1	1	0	0	1	1	1	1

Comparing the recalculated parity bits with the received parity bits:

$p1' \neq p1$ (mismatches the received p1) $\Rightarrow c1 = 1$ $p2' = p2$ (matches the received p2) $\Rightarrow c2 = 0$ $p3' = p3$ (matches the received p3) $\Rightarrow c3 = 0$ $p4' \neq p4$ (mismatches the received p4) $\Rightarrow c4 = 1$ $p5' = p5$ (matches the received p5) $\Rightarrow c5 = 0$ $p6' = p6$ (matches the received p6) $\Rightarrow c6 = 0$
 and $pW' \neq pW$ that means there is single bit error and we can correct it
 arranging $c1, c2, c3, c4, c5, c6$ will give as 0b001001 which is basically 9 in binary so 9th bit(d5) is wrong
 in our case D5 is wrong. which proves the point

2 Question 2

Q: For the foregoing problem, now show an uncorrectable MBE scenario.

Answer:

Original Data

- Binary: 0b10110011010100101100110101001011
- Hexadecimal: 0xB352CD4B
- Decimal: 3008548171

pW	p1	p2	d1	p3	d2	d3	d4	p4	d5	d6	d7	d8	d9	d10	d11
0	1	0	1	0	1	0	1	1	0	0	1	0	1	0	1

p5	d12	d13	d14	d15	d16	d17	d18	d19	d20	d21	d22	d23	d24	d25	d26
0	1	0	0	1	1	0	1	0	0	1	0	1	1	1	0

p6	d27	d28	d29	d30	d31	d32
1	0	0	1	1	0	1

Data with two bit errors at positions d5 and d6:

- Binary: 0b10110011010100101100110101001011
- Hexadecimal: 0xD2ACC5B3
- Decimal: 3534536627

pW	p1	p2	d1	p3	d2	d3	d4	p4	d5	d6	d7	d8	d9	d10	d11
0	1	0	1	0	1	0	1	1	1	1	1	0	1	0	1

p5	d12	d13	d14	d15	d16	d17	d18	d19	d20	d21	d22	d23	d24	d25	d26
0	1	0	0	1	1	0	1	0	0	1	0	1	1	1	0

p6	d27	d28	d29	d30	d31	d32
1	0	0	1	1	0	1

I can't fit image in pdf but there is excel sheet in the same zip folder

Multi bit error which can be detected

	0	1	2	3	4	5	6	7	8	9	10	11	12	13
	pw	p01	p02	d01	p03	d02	d03	d04	p04	d05	d06	d07	d-8	d-9
D														
1	X	1	X	1		1	0	1	X	0	0	1		
2			0											
4					0		0						0	
8						1				0	0	1		
16									1					
32														
ED	0	1	0	1	0	1	0	1	1	0	0	1	0	

8

SYN	0	1	2	3	4	5	6	7	8	9	10	11	12	13
	pW	p01	p02	d01	p03	d02	d03	d04	p04	d05	d06	d07	d-8	d-9
c01	0	1	2	3	4	5	6	7	8	9	10	11	12	13
c02	1	0	0	1	0	1	0	1	1	1	1	1	0	0
c03	0	0	1	1	0	1	0	1	1	1	1	1	0	0
c04	1	1	0	1	0	0	0	1	1	1	1	1	0	0
c05	0	0	0	1	0	1	0	1	1	1	1	1	0	0
c06	0	0	0	1	0	1	0	1	1	1	1	1	0	0
pW2	0	1	0	1	0	1	0	1	1	1	1	1	0	0
ED	0	0	0	1	1	1	0	1	0	0	1	1	0	0

With that number, we need to calculate all the parities:

$$\begin{aligned}
p1 &= \text{XOR}(d1, d2, d4, d5, d7, d9, d11, d12, d14, d16, d18, d20, d22, d24, d26, d27, d29, d31) \\
p2 &= \text{XOR}(d1, d3, d4, d6, d7, d10, d11, d13, d14, d17, d18, d21, d22, d25, d26, d28, d29, d32) \\
p3 &= \text{XOR}(d2, d3, d4, d8, d9, d10, d11, d15, d16, d17, d18, d23, d24, d25, d26, d30, d31, d32) \\
p4 &= \text{XOR}(d5, d6, d7, d8, d9, d10, d11, d19, d20, d21, d22, d23, d24, d25, d26) \\
p5 &= \text{XOR}(d12, d13, d14, d15, d16, d17, d18, d19, d20, d21, d22, d23, d24, d25, d26) \\
p6 &= \text{XOR}(d27, d28, d29, d30, d31, d32)
\end{aligned}$$

The parity bits can be calculated as follows:

$$\begin{aligned}
p1 &= \text{XOR}(d1, d2, d4, d5, d7, d9, d11, d12, d14, d16, d18, d20, d22, d24, d26, d27, d29, d31) \\
&= \text{XOR}(1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 0, 0, 0, 1, 0, 1, 0) \\
&= 0 \\
p2 &= \text{XOR}(d1, d3, d4, d6, d7, d10, d11, d13, d14, d17, d18, d21, d22, d25, d26, d28, d29, d32) \\
&= \text{XOR}(1, 0, 1, 0, 1, 1, 1, 0, 0, 0, 1, 1, 0, 1, 1, 0, 1, 1) \\
&= 1 \\
p3 &= \text{XOR}(d2, d3, d4, d8, d9, d10, d11, d15, d16, d17, d18, d23, d24, d25, d26, d30, d31, d32) \\
&= \text{XOR}(1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1) \\
&= 0 \\
p4 &= \text{XOR}(d5, d6, d7, d8, d9, d10, d11, d19, d20, d21, d22, d23, d24, d25, d26) \\
&= \text{XOR}(1, 1, 1, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1, 1) \\
&= 0 \\
p5 &= \text{XOR}(d12, d13, d14, d15, d16, d17, d18, d19, d20, d21, d22, d23, d24, d25, d26) \\
&= \text{XOR}(1, 0, 0, 1, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1, 1) \\
&= 0 \\
p6 &= \text{XOR}(d27, d28, d29, d30, d31, d32) \\
&= \text{XOR}(0, 0, 1, 1, 0, 1) \\
&= 1
\end{aligned}$$

The whole word parity bit (pW) can be calculated by XORing all the data bits and parity bits:

$$\begin{aligned}
pW &= \text{XOR}(p1, p2, d1, p3, d2, d3, d4, p4, d5, d6, d7, d8, d9, d10, d11, \\
&\quad p5, d12, d13, d14, d15, d16, d17, d18, d19, d20, d21, d22, d23, d24, d25, d26, \\
&\quad p6, d27, d28, d29, d30, d31, d32) \\
&= \text{XOR}(1, 0, 0, 1, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1) \\
&= 1
\end{aligned}$$

Therefore, the parity bits and the whole word parity bit for the given data bits are:

pW	p1	p2	d1	p3	d2	d3	d4	p4	d5	d6	d7	d8	d9	d10	d11
1	0	1	1	0	1	0	1	0	1	1	1	0	1	0	1

p5	d12	d13	d14	d15	d16	d17	d18	d19	d20	d21	d22	d23	d24	d25	d26
0	1	0	0	1	1	0	1	0	0	1	0	1	1	0	

p6	d27	d28	d29	d30	d31	d32
1	0	0	1	1	0	1

Comparing the recalculated parity bits with the received parity bits:

$p1' \neq p1$ (mismatches the received $p1$) $\Rightarrow c1 = 1$ $p2' \neq p2$ (matches the received $p2$) $\Rightarrow c2 = 0$ $p3' = p3$ (matches the received $p3$) $\Rightarrow c3 = 0$ $p4' = p4$ (mismatches the received $p4$) $\Rightarrow c4 = 1$ $p5' = p5$ (matches the received $p5$) $\Rightarrow c5 = 0$ $p6' = p6$ (matches the received $p6$) $\Rightarrow c6 = 0$
and parity word $pW' = pW$ so there is multibit error but we can't correct that

3 Question 3

Q: For the following Nand flash block update history for 2 sectors that contain 4 blocks each (e.g. 16K sectors, with 4K blocks), fill in the missing WRITE operations as needed and compute write- amplification.

Sector Erased (S0, S1)		0,0	1,1	1,1	1,1	1,1	2,1	2,1
S1								
	PB7	FREE	FREE	FREE	LB3	LB3	LB3	LB3
	PB6	FREE	FREE	LB2	LB2	INVLD	INVLD	INVLD
	PB5	FREE	LB3	LB3	INVLD	INVLD	INVLD	INVLD
	PB4	FREE	LB2	INVLD	INVLD	INVLD	INVLD	INVLD
S0								
	PB3	FREE	FREE	FREE	LB1	LB1	FREE	LB1
	PB2	FREE	FREE	LB0	LB0	INVLD	FREE	FREE
	PB1	FREE	LB1	LB1	INVLD	INVLD	FREE	LB2
	PB0	FREE	LB0	INVLD	INVLD	INVLD	FREE	LB0
FS LBs Updated			0,1,2,3	0,2	1,3	0,2	0,2	0,2
FS LBs Cached						0,2	0,2	
Sector LBs Buffered							1	
		#1	#2	#3	#4	#5	#6	#7
Sectors Erased (S0, S1)	2,1	2,1	2,2	2,2	2,2	2,2	3,2	3,2
S1								
	LB3	INVLD	FREE	FREE	LB2	LB2	LB2	LB2
	INVLD	INVLD	FREE	FREE	LB0	LB0	LB0	LB0
	INVLD	INVLD	FREE	LB3	LB3	INVLD	INVLD	INVLD
	INVLD	INVLD	FREE	LB1	LB1	INVLD	INVLD	INVLD
S0								
	LB1	INVLD	INVLD	INVLD	INVLD	INVLD	FREE	FREE
	FREE	FREE	FREE	FREE	FREE	FREE	FREE	FREE
	LB2	LB2	LB2	LB2	INVLD	INVLD	FREE	LB3
	LB0	LB0	LB0	LB0	INVLD	INVLD	FREE	LB1
FS LBs Updated	0,2	1,3	1,3	1,3	0,2	1,3	1,3	1,3
FS LBs Cached		1,3	1,3			1,3	1,3	
Sector LBs Buffered								

- #1 - All blocks FREE
- #2 - Erase S0 & S1, WRITE _____
- #3 - Read LB 0, 2, Modify, WRITE _____
- #4 - Read LB 1, 3, Modify, WRITE _____
- #5 - Read LB 0, 2, Modify and Cache
- #6 - Buffer LB 0, 1, 2, Erase S0
- #7 - WRITE _____ to S0 Write Amplification = _____
- #8 - Read LB 1, 3, Modify and Cache
- #9 - Erase S1
- #10 - WRITE _____
- #11 - Read LB 0, 2, Modify, WRITE _____
- #12 - Read LB 1, 3, Modify and Cache
- #13 - Erase S0

#14 - WRITE _____ Write Amplification = _____
Total sector erases for both S0 and S1 = _____

Answer:

Answer is :

#1 - All blocks FREE
#2 - Erase S0 & S1, Write LB 0, 1, 2, 3
#3 - Read LB 0, 2, Modify, Write LB 0, 2
#4 - Read LB 1, 3, Modify, Write LB 1, 3
#5 - Read LB 0, 2, Modify and Cache
#6 - Buffer LB 0, 1, 2, Erase S0
#7 - Write-back LB 0, 1, 2 to S0
11 Writes, 3 Sector Erases
Write Amplification = $11 / 10 = 1.1$

#0 - Start State from End State Above
#1 - Read LB 1, 3, Modify and Cache
#2 - Erase S1
#3 - Write-back LB 1, 3 to S1
#4 - Read LB 0, 2, Modify, Write LB 0, 2
#5 - Read LB 1, 3, Modify and Cache
#6 - Erase S0
#7 - Write-back LB 1, 3
6 Writes, 2 Sector Erases
Write Amplification = $17 / 16 = 1.0625$
Total sector erases for both cases S0 and S1 = $2 + 3 = 5$

Explanation: Write amplification is a measure of the efficiency of a storage system, particularly in the context of solid-state drives (SSDs) and other flash memory-based storage devices. It represents the ratio of the amount of data written to the storage device to the amount of data requested to be written by the host system.

In the given example, the write amplification is calculated as $11/10$, which means that for every 10 units of data requested to be written by the host, the storage device actually writes 11 units of data. This additional writing overhead is due to various factors, such as garbage collection, wear leveling, and maintaining data integrity.

Let's break down the given scenario to understand the write amplification of $11/10$:

The host system requested to write 10 logical blocks (LB0, LB1, LB2, LB3) to the storage device. To accommodate these writes, the storage device performed the following operations: Erased S0 and S1 (2 sector erases) Wrote LB0, LB1, LB2, LB3 to S0 (4 block writes) Wrote modified LB0, LB2 to S1 (2 block writes) Wrote modified LB1, LB3 to S1 (2 block writes) Erased S0 (1 sector erase) Wrote buffered LB0, LB1, LB2 to S0 (3 block writes)

in the next case :

Write Amplification = $(\text{Previous total writes} + \text{Current total writes}) / (\text{Previous requested writes} + \text{Current requested writes})$
Write Amplification = $(11 + 6) / (10 + 6) = 17 / 16 = 1.0625$

This means that for every 16 units of data requested to be written by the host system, the storage device actually wrote 17 units of data. The write amplification is lower than in the previous scenario (1.0625 vs. 1.1), indicating an improvement in efficiency.

4 Reference

1. REAL-TIME EMBEDDED COMPONENTS AND SYSTEMS with LINUX and RTOS by Sam Siewert
John Pratt

Appendices

A Excel Sheet

A.1 Q1

Sheet1

		0	1	2	3	4
		pW	p01	p02	d01	p03
bit	D	X	X	X	1	X
1	p01		1		1	
2	p02			0	1	
4	p03					0
8	p04					
16	p05					
32	p06					
	ED	0	1	0	1	0

		0	1	2	3	4
		pW	p01	p02	d01	p03
SYN	In. Data	0	1	0	1	0
0b001001			0		1	
c01	1			0	1	
c02	0				1	
c03	0					0
c04	1					
c05	0					
c06	0					
pW2	1	0	1	0	1	0
	ED	0	0	0	1	1

		0	1	2	3	4
		pW	p01	p02	d01	p03
bit	D	X	X	X	1	X
1	p01		1		1	
2	p02			0	1	
4	p03					0
8	p04					
16	p05					
32	p06					
	ED	0	1	0	1	0

		0	1	2	3	4
		pW	p01	p02	d01	p03
SYN	In. Data	0	1	0	1	0
c01	0		0		1	
c02	1			1	1	
c03	0					0
c04	1					

Sheet1

c05	0						
c06	0						
pW2	0	0	1	0	1	0	
	ED	0	0	0	1	1	

Single Bit Error (B

5	6	7	8	9	10	11
d02	d03	d04	p04	d05	d06	d07
1	0	1	X	0	0	1
1		1		0		1
	0	1			0	1
1	0	1				
			1	0	0	1
1	0	1	1	0	0	1

5	6	7	8	9	10	11
d02	d03	d04	p04	d05	d06	d07
1	0	1	1	1	0	1
1		1		1		1
	0	1			0	1
1	0	1				
			0	1	0	1
1	0	1	1	0	0	1
1	0	1	0	1	0	1

Multi bit error which ca

5	6	7	8	9	10	11
d02	d03	d04	p04	d05	d06	d07
1	0	1	X	0	0	1
1		1		0		1
	0	1			0	1
1	0	1				
			1	0	0	1
1	0	1	1	0	0	1

5	6	7	8	9	10	11
d02	d03	d04	p04	d05	d06	d07
1	0	1	1	1	1	1
1		1		1		1
	0	1			1	1
1	0	1				
			1	1	1	1

Sheet1

1	0	1	1	0	0	1
1	0	1	0	1	1	1

it changed at 9th position D5)

12	13	14	15	16	17	18
d-8	d09	d10	d11	p05	d12	d13
0	1	0	1	X	1	0
	1		1		1	
		0	1			0
0	1	0	1			
0	1	0	1			
				0	1	0
0	1	0	1	0	1	0

12	13	14	15	16	17	18
d-8	d09	d10	d11	p05	d12	d13
0	1	0	1	0	1	0
	1		1		1	
		0	1			0
0	1	0	1			
0	1	0	1			
				0	1	0
0	1	0	1	0	1	0
0	1	0	1	0	1	0

n be detected but not correctable

12	13	14	15	16	17	18
d-8	d09	d10	d11	p05	d12	d13
0	1	0	1	X	1	0
	1		1		1	
		0	1			0
0	1	0	1			
0	1	0	1			
				0	1	0
0	1	0	1	0	1	0

12	13	14	15	16	17	18
d-8	d09	d10	d11	p05	d12	d13
0	1	0	1	0	1	0
	1		1		1	
		0	1			0
0	1	0	1			
0	1	0	1			

Sheet1

				0	1	0
0	1	0	1	0	1	0
0	1	0	1	0	1	0

Sheet1

19 d14	20 d15	21 d16	21 d17	23 D18	24 D19	25 D20
0	1	1	0	1	0	0
0		1		1		0
0			0	1		
	1	1	0	1		
					0	0
0	1	1	0	1	0	0
0	1	1	0	1	0	0

19 d14	20 d15	21 d16	21 d17	23 D18	24 D19	25 D20
0	1	1	0	1	0	0
0		1		1		0
0			0	1		
	1	1	0	1		
					0	0
0	1	1	0	1	0	0
0	1	1	0	1	0	0

19 d14	20 d15	21 d16	21 d17	23 D18	24 D19	25 D20
0	1	1	0	1	0	0
0		1		1		0
0			0	1		
	1	1	0	1		
					0	0
0	1	1	0	1	0	0
0	1	1	0	1	0	0

19 d14	20 d15	21 d16	21 d17	23 D18	24 D19	25 D20
0	1	1	0	1	0	0
0		1		1		0
0			0	1		
	1	1	0	1		
					0	0

Sheet1

0	1	1	0	1	0	0
0	1	1	0	1	0	0
0	1	1	0	1	0	0

Sheet1

26 d21	27 d22	28 d23	29 d24	30 d25	31 d26	32 p06
1	0	1	0	1	1	X
	0		0		1	
1	0			1	1	
		1	0	1	1	
1	0	1	0	1	1	
1	0	1	0	1	1	
						1
1	0	1	0	1	2	1

26 d21	27 d22	28 d23	29 d24	30 d25	31 d26	32 p06
1	0	1	0	1	1	1
	0		0		1	
1	0			1	1	
		1	0	1	1	
1	0	1	0	1	1	
1	0	1	0	1	1	
						1
1	0	1	0	1	1	1
1	0	1	0	1	2	1

26 d21	27 d22	28 d23	29 d24	30 d25	31 d26	32 p06
1	0	1	0	1	1	X
	0		0		1	
1	0			1	1	
		1	0	1	1	
1	0	1	0	1	1	
1	0	1	0	1	1	
						1
1	0	1	0	1	2	1

26 d21	27 d22	28 d23	29 d24	30 d25	31 d26	32 p06
1	0	1	0	1	1	1
	0		0		1	
1	0			1	1	
		1	0	1	1	
1	0	1	0	1	1	

Sheet1

1	0	1	0	1	1	
1	0	1	0	1	1	1
1	0	1	0	1	2	1

Sheet1

33 d27	34 d28	35 d29	36 d30	37 d31	38 d32
0	0	1	1	0	1
0		1		0	
	0	1			1
			1	0	1

0	0	1	1	0	1
0	0	1	1	0	1

33 d27	34 d28	35 d29	36 d30	37 d31	38 d32
0	0	1	1	0	1
0		1		0	
	0	1			1
			1	0	1

0	0	1	1	0	1
0	0	1	1	0	1

33 d27	34 d28	35 d29	36 d30	37 d31	38 d32
0	0	1	1	0	1
0		1		0	
	0	1			1
			1	0	1

0	0	1	1	0	1
0	0	1	1	0	1

33 d27	34 d28	35 d29	36 d30	37 d31	38 d32
0	0	1	1	0	1
0		1		0	
	0	1			1
			1	0	1

Sheet1

0	0	1	1	0	1
0	0	1	1	0	1
0	0	1	1	0	1

