Question 1 : 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.

In this example, a 32-bit data word representing the decimal number 35 is encoded into a 39-bit Hamming code word to facilitate error detection and correction. The additional seven bits are parity bits strategically placed within the word to form a pattern that enables the identification and correction of a single-bit error (SBE). Upon simulating a bit flip at position 24, the Hamming code's error detection mechanism is triggered. The mismatch between the original parity word (pW) and the parity word after the flip (pW2)—highlighted by a non-zero check bit value—signifies an SBE has occurred. The error is located by analysing the disparity in parity checks, which are derived from the exclusive OR (XOR) operation between corresponding bits of the encoded word. The Hamming code's inherent design deduces the exact bit in error by comparing the expected and actual parity bit patterns, thus pinpointing the singular bit discrepancy and enabling its correction to restore the integrity of the original 32-bit word.

32 Bit Hamming ECC

Enter Decimal Number:	35
Binary:	100011

		0	1	10	11	100	101	110	111	1000	1001	1010	1011	1100	1101	1110	1111	10000	10001	10010	10011
Bit Positions		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
		pW	p01	p02	d01	p03	d02	d03	d04	p04	d05	d06	d07	d08	d09	d10	d11	p05	d12	d13	d14
Bit Data	D	X	X	X	1	X	1	0	0	X	0	1	0	0	0	0	0	X	0	0	0
1	p01		0		1		1		0		0		0		0		0		0		0
2	p02			0	1			0	0			1	0			0	0			0	0
4	p03					1	1	0	0					0	0	0	0				
8	p04									1	0	1	0	0	0	0	0				
16	p05																	0	0	0	0
32	p06																				
DO NOT EDIT THIS ROW	ED	1	0	0	1	1	1	0	0	1	0	1	0	0	0	0	0	0	0	0	0

		10100	10101	10110	10111	11000	11001	11010	11011	11100	11101	11110	11111	100000	100001	100010	100011	100100	100101	100110
Bit Positions		20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38
		d15	d16	d17	d18	d19	d20	d21	d22	d23	d24	d25	d26	p06	d27	d28	d29	d30	d31	d32
Bit Data	D	0	0	0	0	0	0	0	0	0	0	0	0	X	0	0	0	0	0	0
1	p01		0		0		0		0		0		0		0		0		0	
2	p02			0	0			0	0			0	0			0	0			0
4	p03	0	0	0	0					0	0	0	0					0	0	0
8	p04					0	0	0	0	0	0	0	0							
16	p05	0	0	0	0	0	0	0	0	0	0	0	0							
32	p06													0	0	0	0	0	0	0
DO NOT EDIT THIS ROW	ED	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

		0	1	10	11	100	101	110	111	1000	1001	1010	1011	1100	1101	1110	1111	10000	10001	10010	10011
Bit Positions		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
SYN		pW	p01	p02	d01	p03	d02	d03	d04	p04	d05	d06	d07	d08	d09	d10	d11	p05	d12	d13	d14
Check Bits	ED	1	0	0	1	1	1	0	0	1	0	1	0	0	0	0	0	0	0	0	0
c01	0		0		1		1		0		0		0		0		0		0		0
c02	0			0	1			0	0			1	0			0	0			0	0
c03	0					1	1	0	0					0	0	0	0				
c04	1									0	0	1	0	0	0	0	0				
c05	1																	1	0	0	0
c06	0																				
pW2		0	0	0	1	1	1	0	0	0	0	1	0	0	0	0	0	1	0	0	0

		10100	10101	10110	10111	11000	11001	11010	11011	11100	11101	11110	11111	100000	100001	100010	100011	100100	100101	100110
Bit Positions		20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38
SYN		d15	d16	d17	d18	d19	d20	d21	d22	d23	d24	d25	d26	p06	d27	d28	d29	d30	d31	d32
Check Bits	ED	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
c01	0		0		0		0		0		0		0		0		0		0	
c02	0			0	0			0	0			0	0			0	0			0
c03	0	0	0	0	0					0	0	0	0					0	0	0
c04	1					1	0	0	0	0	0	0	0							
c05	1	0	0	0	0	1	0	0	0	0	0	0	0							
c06	0													0	0	0	0	0	0	0
pW2		0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Is pW==pW2?	No
Check Bits Value:	24

Case	Yes/No
No Error	No
SBE, Can Correct	Yes
Double Bit Error	No
Parity Word Error	No

ECEN 5623 **Homework4** **Kiran Jojare**

Question 2 : For the foregoing problem, now show an uncorrectable MBE scenario

In the provided scenario, the Hamming code's limitation is demonstrated when faced with a multiple bit error (MBE) situation. Specifically, the 39-bit encoded word, originally representing the decimal value 35 (or 100011 in binary), encounters simultaneous errors at bit positions 14 and 26. The Hamming code, while adept at detecting single bit errors (SBE), cannot correct multiple bit errors. In this case, the parity checks before (pW) and after the error (pW2) remain consistent with each other, indicating no change and hence a failure to detect a discrepancy, which would be expected in an SBE scenario. However, the non-zero check bit value signals an anomaly. This incongruence—where the check bits indicate an error but the parity consistency suggests otherwise—points to an MBE. It confirms that while the Hamming code can detect the occurrence of an MBE, it lacks the ability to correct it, because the pattern of errors does not alter the parity bits in a way that is identifiable to the Hamming algorithm designed for single bit error correction. Thus, the system accurately detects the presence of an MBE, fulfilling the conditions Check-Bits != 0 and parity-encoded-word = parity-read-word, yet it is unable to isolate and correct the errors due to the limitations of the single-error-correction design.

32 Bit Hamming ECC

Enter Decimal Number:	35
Binary:	100011

		0	1	10	11	100	101	110	111	1000	1001	1010	1011	1100	1101	1110	1111	10000	10001	10010	10011	10100	10101	10110	10111	11000	11001	11010	11011	11100	11101	11110	11111	100000	100001	100010	100011	100100	100101	100110
Bit Positions		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38
		pW	p01	p02	d01	p03	d02	d03	d04	p04	d05	d06	d07	d08	d09	d10	d11	p05	d12	d13	d14	d15	d16	d17	d18	d19	d20	d21	d22	d23	d24	d25	d26	p06	d27	d28	d29	d30	d31	d32
Bit Data	D	X	X	X	1	X	1	0	0	X	0	1	0	0	0	0	0	X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	X	0	0	0	0	0	0
1	p01		0		1		1		0		0		0		0		0		0		0		0		0		0		0		0		0		0		0		0	
2	p02			0	1			0	0			1	0			0	0			0	0			0	0			0	0			0	0			0	0			0
4	p03					1	1	0	0					0	0	0	0					0	0	0	0					0	0	0	0					0	0	0
8	p04									1	0	1	0	0	0	0	0									0	0	0	0	0	0	0	0							
16	p05																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0							
32	p06																																	0	0	0	0	0	0	0
DO NOT EDIT THIS ROW	ED	1	0	0	1	1	1	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

		0	1	10	11	100	101	110	111	1000	1001	1010	1011	1100	1101	1110	1111	10000	10001	10010	10011	10100	10101	10110	10111	11000	11001	11010	11011	11100	11101	11110	11111	100000	100001	100010	100011	100100	100101	100110
Bit Positions		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38
SYN		pW	p01	p02	d01	p03	d02	d03	d04	p04	d05	d06	d07	d08	d09	d10	d11	p05	d12	d13	d14	d15	d16	d17	d18	d19	d20	d21	d22	d23	d24	d25	d26	p06	d27	d28	d29	d30	d31	d32
Check Bits	ED	1	0	0	1	1	1	0	0	1	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
c01	0		0		1		1		0		0		0		0		0		0		0		0		0		0		0		0		0		0		0		0	
c02	0			0	1			0	0			1	0			1	0			0	0			0	0			1	0			0	0			0	0			0
c03	1					0	1	0	0					0	0	1	0					0	0	0	0					0	0	0	0					0	0	0
c04	0									1	0	1	0	0	0	1	0									0	0	1	0	0	0	0	0							
c05	1																	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0							
c06	0																																	0	0	0	0	0	0	0
pW2		1	0	0	1	0	1	0	0	1	0	1	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0

Is pW==pW2?	Yes
Check Bits Value:	20

Case	Yes/No
No Error	No
SBE, Can Correct	No
Double Bit Error	Yes
Parity Word Error	No

ECEN 5623 **Homework4** **Kiran Jojare**

Question 3: 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

		#1 - Start	#2	#3	#4	#5	#6	#7
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	

		#8	#9	#10	#11	#12	#13	#14
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 ____________________ **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

ECEN 5623 **Homework4** **Kiran Jojare**

#7 – WRITE ____________**LB 0,1,2** to S0

Write Amplification = ________________________________**11/10 = 1.1**

#8 - Read LB 1, 3, Modify and Cache

#9 – Erase S1

#10 – WRITE ____**LB 1,3**

#11 – Read LB 0, 2, Modify, WRITE ___________________**LB 0,2**

#12 – Read LB 1, 3, Modify and Cache

#13 – Erase S0

#14 – WRITE ____**LB 1,3**

Write Amplification = ______________________________**1.06**

Total sector erases for both S0 and S1 = ____________**5**