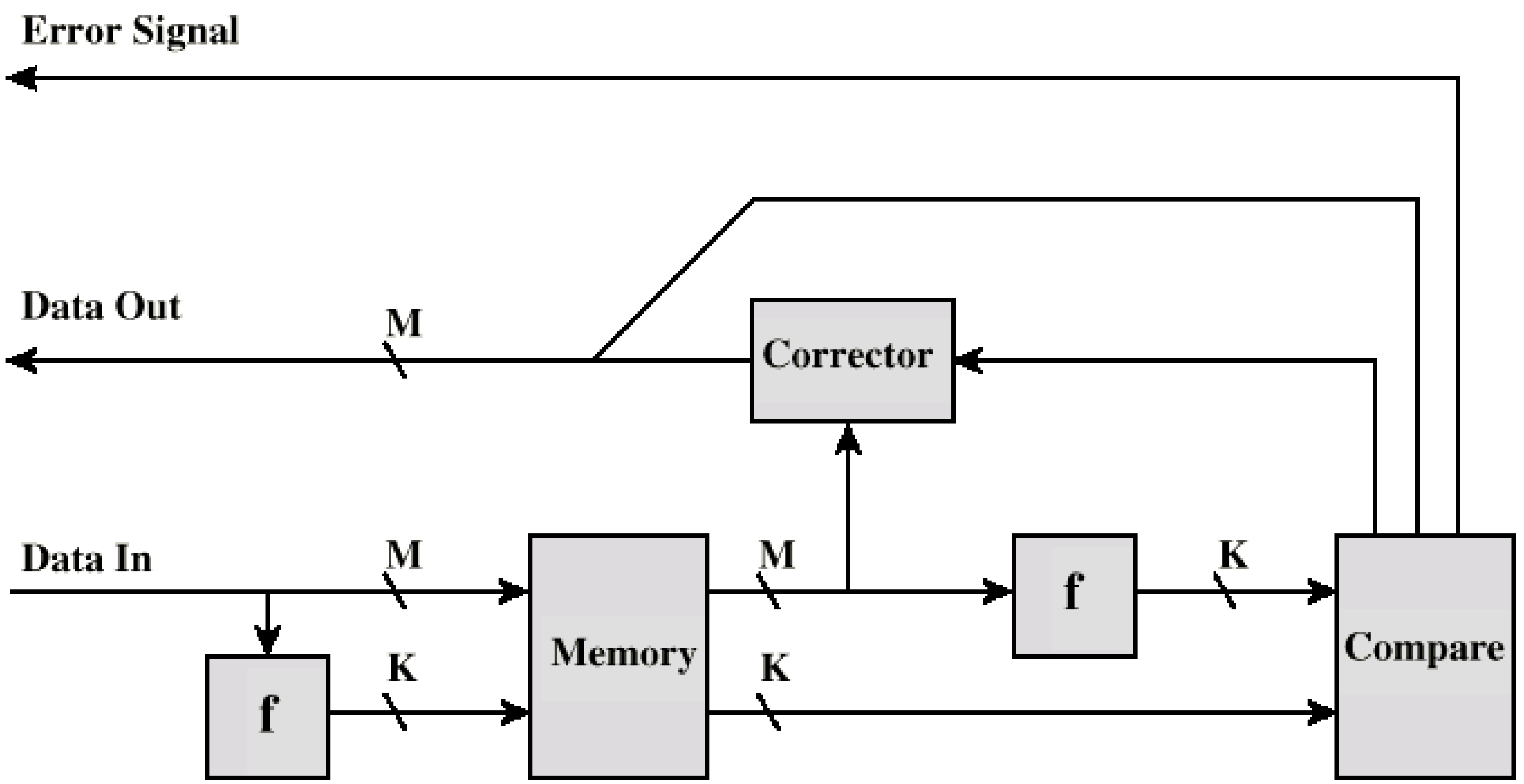
ERROR DETECTION AND CORRECTION

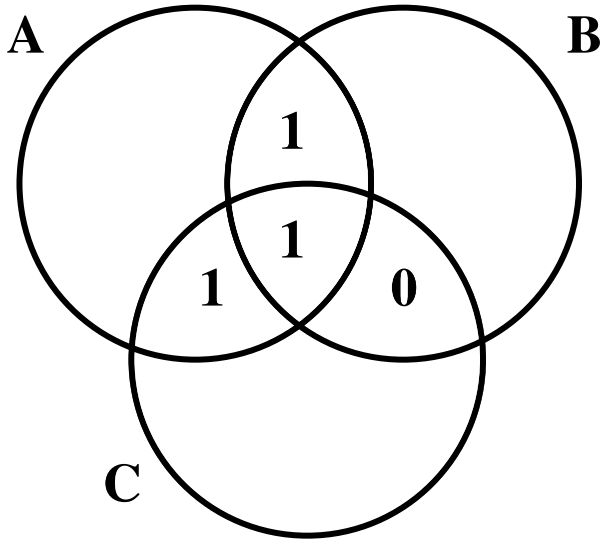
Error Correction

- A semiconductor memory system is subject to errors.
- Hard failures – permanent physical defects
  Environmental abuse, manufacturing defects, etc.
- Soft error
  Power supply problems etc.
- Need logic for detecting and correcting errors.
- Basic technique
  - Prior to storing data a code is generated from the bits in the word.
  - Code is stored along with the word in memory.
  - Code used to identify and correct errors.
  - When the word is fetched a new code is generated and compared to the stored code.
    - No error (normal case)
    - Correctable error is detected and corrected.
    - Non-fixable error is detected and reported.

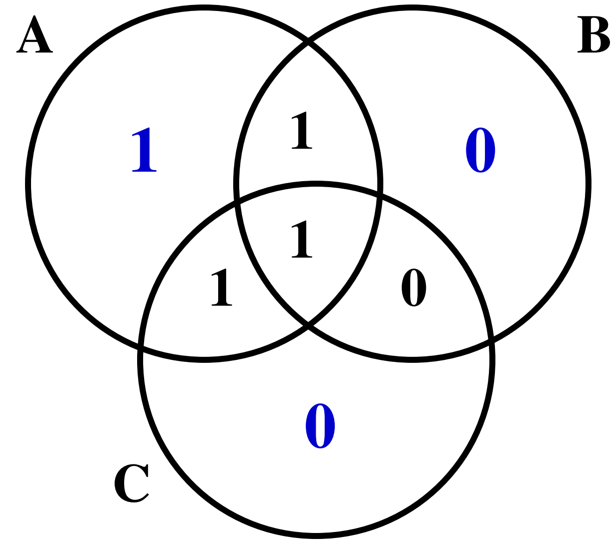
Error Correcting Code Function



Hamming Code



Assign data bits to the inner compartments.

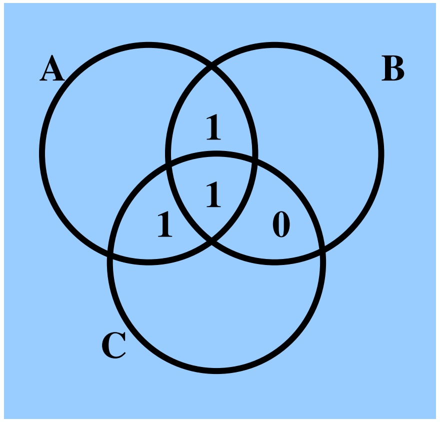


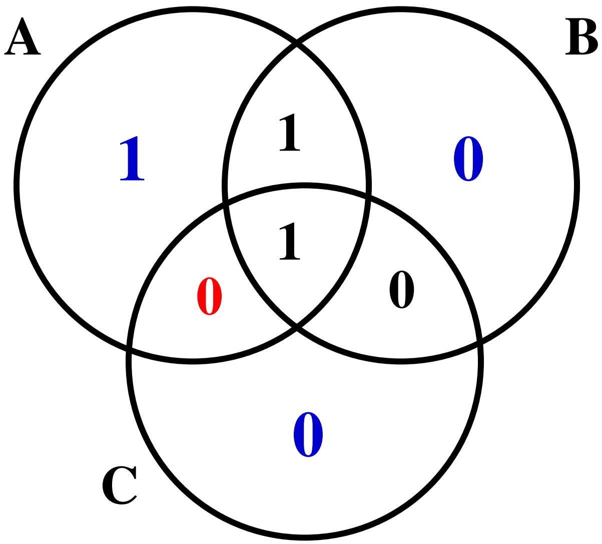
Fill the remaining compartments with parity bits.

The total number of bits in a circle must be even.

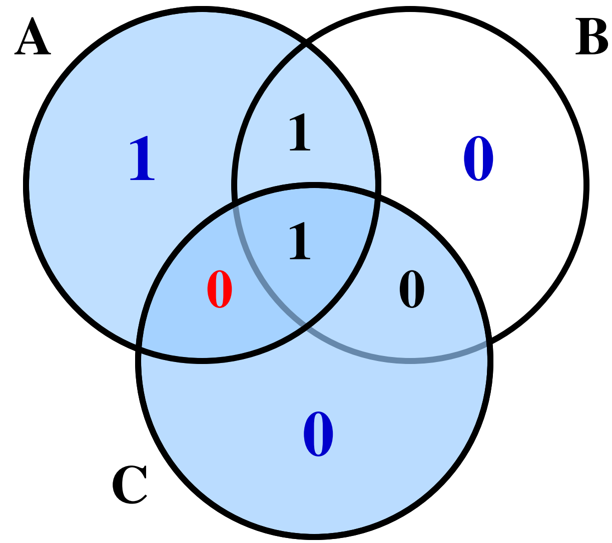
For example: The data bits in A = 1+1+1 = 3. This is odd – therefore add an additional 1.

Hamming Code





If a bit gets erroneously changed, the parity bits in that circle will no longer add up to 1.



Errors are found in A and C – and the shared bit in A and C is in error and can be fixed.

Single Bit Errors in 8-bit words

- 8 data bits
- $2^K - 1 \geq M + K$ is used to find the value of K (number of check bits).
- One error bit => error occurred in one of the check bits. No action.
- More than one bit set to '1' => the numerical value of the syndrome indicates the position of the data bit in error.
- $2^K - 1 \geq 8 + K \Rightarrow K = 4$

Single Bit Errors in 8-bit words

- Data and check bits arranged into a 12-bit word.
- Bit positions numbered from 1 to 12.
- Bit positions representing position numbers that are powers of 2 are designated as check bits.
- Check bits calculated as follows:

$C1 = D1 \oplus D2 \oplus D4 \oplus D5 \oplus D7$
$C2 = D1 \oplus D3 \oplus D4 \oplus D6 \oplus D7$
$C4 = D2 \oplus D3 \oplus D4 \oplus D8$
$C8 = D5 \oplus D6 \oplus D7 \oplus D8$

- Data and check bits arranged into a 12 bit syndrome word:

8 data bits	4 check bits

Calculating check bits

Bit Position	Binary	Type		
1	0001	C1		$\oplus$All D's with a 1 in bit 1
2	0010	C2		$\oplus$All D's with a 1 in bit 2
3	0011		D1	
4	0100	C4		$\oplus$All D's with a 1 in bit 3
5	0101		D2	
6	0110		D3	
7	0111		D4	
8	1000	C8		$\oplus$ All D's with a 1 in bit 4
9	1001		D5	
10	1010		D6	
11	1011		D7	
12	1100		D8	

$C1 = D1 \oplus D2 \oplus D4 \oplus D5 \oplus D7$

Each check bit works on every data bit who shares the same bit position

Example

- Input word: 00111001
  Data bit D1 in rightmost position
- Calculate check bits:
  $C1 = 1 \oplus 0 \oplus 1 \oplus 1 \oplus 0 = 1$
  $C2 = 1 \oplus 0 \oplus 1 \oplus 1 \oplus 0 = 1$
  $C3 = 0 \oplus 0 \oplus 1 \oplus 0 = 1$
  $C4 = 1 \oplus 1 \oplus 0 \oplus 0 = 0$
  Stored word = 001101001111
- If data bit 3 sustains an error (001101101111)
  $C1 = 1 \oplus 0 \oplus 1 \oplus 1 \oplus 0 = 1$
  $C2 = 1 \oplus 1 \oplus 1 \oplus 1 \oplus 0 = 0$
  $C3 = 0 \oplus 1 \oplus 1 \oplus 0 = 0$
  $C4 = 1 \oplus 1 \oplus 0 \oplus 0 = 0$
- Calculate syndrome word:
  0110 = bit position 6.
- D3 resides in bit position 6.

$$
\begin{array}{ccccc}
 & C & C & C & C \\
 & 8 & 4 & 2 & 1 \\
 & 0 & 1 & 1 & 1 \\
\oplus & 0 & 0 & 0 & 1 \\
\hline
 & 0 & 1 & 1 & 0
\end{array}
$$

Problems

- Suppose an 8-bit data word stored in memory is 11000010. Using the Hamming algorithm, determine what check bits would be stored in memory with the data word. Show how you got your answer.

Problems

- Suppose an 8-bit data word stored in memory is 11000010. Using the Hamming algorithm, determine what check bits would be stored in memory with the data word. Show how you got your answer.

- C1 = X-OR[1 2 4 5 7] = 0 1 0 0 1 =0
- C2 = X-OR[1 3 4 6 7] = 0 0 0 0 1 =1
- C4 = X-OR[2 3 4 8] = 1 0 0 1 =0
- C8 = X-OR[5 6 7 8] = 0 0 1 1 =0

- 12 bit word is: 110000010010

- At the receiver side,
- If the fetched data word is,

110001010010

→extract the 8bit data

11001010

→calculate the check bits:

- C1 = X-OR[1 2 4 5 7] = 0 1 1 0 1 =1
- C2 = X-OR[1 3 4 6 7] = 0 0 1 0 1 =0
- C4 = X-OR[2 3 4 8] = 1 0 1 1 =1
- C8 = X-OR[5 6 7 8] = 0 0 1 1 =0

- Now check the error
- Check bits at sender side is: 0010
- Check bits at sender side is: 0101
- E-OR both

$$
\begin{array}{r}
0\ 0\ 1\ 0 \\
0\ 1\ 0\ 1 \\
\hline
0\ 1\ 1\ 1
\end{array}
= 7
$$

; if this is 0, no error else,

Error is in the seventh position of 12 bit data.

Problems

- For the 8-bit word 00111001, the check bits stored with it would be 0111. Suppose when the word is read from memory, the check bits are calculated to be 1101. What is the data word that was read from memory?
- How many check bits are needed if the hamming error correction code is used to detect single bit errors in 1024 bit data word
- Generate the code for the data word 0101000000111001. show that the code will correctly identify an error in data bit 5.

References

- William Stallings "Computer Organization and architecture" Prentice Hall, 7th edition, 2006
- Internet: PPT slides, Author: Jane and Harold Huang