Lecture 25 – Memory architecture 3

Dr. Aftab M. Hussain,

Assistant Professor, PATRIoT Lab, CVEST

Chapter 7

Error detection and correction

- The dynamic physical interaction of the electrical signals affecting the data path of a memory unit may cause occasional errors in storing and retrieving the binary information

- The reliability of a memory unit may be improved by employing error-detecting and error-correcting codes

- The most common error detection scheme is the parity bit

The parity bit

- A parity bit is generated and stored along with the data word in memory
- The parity of the word is checked after reading it from memory
- The data word is accepted if the parity of the bits read out is correct
- If the parity checked results in an inversion, an error is detected
- However, it cannot be corrected
- Error correction requires more complex mechanisms such as the Hamming code

The Hamming code

- The poison and the rats!
- You are throwing a party in an hour. You have 1000 wine bottles, and you come to know that one of them contains poison! You decide to use some rats to find out which bottle is poised. The rats will die one hour after consuming the poisoned wine. What is the minimum number of rats needed?

The Hamming code

- One of the most common error-correcting codes used in RAMs was devised by R. W. Hamming
- In the Hamming code, *k* parity bits are added to an *n* -bit data word, forming a new word of *n* + *k* bits
- The bit positions are numbered in sequence from 1 to *n* + *k*
- Those positions numbered as a power of 2 are reserved for the parity bits
- The code can be used with words of any length
- Consider, for example, the 8-bit data word 11000100
- We include 4 parity bits with the 8-bit word and make a 12 bit word

While storing:

Bit position:	1	2	3	4	5	6	7	8	9	10	11	12
	P_1	P_2	1	P_4	1	0	0	P_8	0	1	0	0

$$P_1 = \text{XOR of bits } (3, 5, 7, 9, 11) = 1 \oplus 1 \oplus 0 \oplus 0 \oplus 0 = 0$$

$$P_2 = \text{XOR of bits } (3, 5, 7, 10, 11) = 1 \oplus 0 \oplus 0 \oplus 1 \oplus 0 = 0$$

$$P_4 = \text{XOR of bits } (5, 6, 7, 12) = 1 \oplus 0 \oplus 0 \oplus 0 = 1$$

$$P_8 = \text{XOR of bits } (9, 10, 11, 12) = 0 \oplus 1 \oplus 0 \oplus 0 = 1$$

While reading:

	0	0	1	1	1	0	0	1	0	1	0	0
Bit position:	1	2	3	4	5	6	7	8	9	10	11	12

$$C_1 = \text{XOR of bits } (1, 3, 5, 7, 9, 11)$$

$$C_2 = \text{XOR of bits } (2, 3, 6, 7, 10, 11)$$

$$C_4 = \text{XOR of bits } (4, 5, 6, 7, 12)$$

$$C_8 = \text{XOR of bits } (8, 9, 10, 11, 12)$$

The Hamming code

- A 0 check bit designates even parity over the checked bits and a 1 designates odd parity
- Since the bits were stored with even parity, the result, $C = C_8C_4C_2C_1$ = 0000, indicates that no error has occurred
- Here is some magic: However, if $C \neq 0$, then the 4-bit binary number formed by the check bits gives the position of the erroneous bit!
- How? Check out the poison and rat puzzle

While storing:

Bit position:	1	2	3	4	5	6	7	8	9	10	11	12
	P_1	P_2	1	P_4	1	0	0	P_8	0	1	0	0

$$P_1 = \text{XOR of bits } (3, 5, 7, 9, 11) = 1 \oplus 1 \oplus 0 \oplus 0 \oplus 0 = 0$$
$$P_2 = \text{XOR of bits } (3, 5, 7, 10, 11) = 1 \oplus 0 \oplus 0 \oplus 1 \oplus 0 = 0$$
$$P_4 = \text{XOR of bits } (5, 6, 7, 12) = 1 \oplus 0 \oplus 0 \oplus 0 = 1$$
$$P_8 = \text{XOR of bits } (9, 10, 11, 12) = 0 \oplus 1 \oplus 0 \oplus 0 = 1$$

While reading:

	0	0	1	1	1	0	0	1	0	1	0	0
Bit position:	1	2	3	4	5	6	7	8	9	10	11	12

$$C_1 = \text{XOR of bits } (1, 3, 5, 7, 9, 11)$$
$$C_2 = \text{XOR of bits } (2, 3, 6, 7, 10, 11)$$
$$C_4 = \text{XOR of bits } (4, 5, 6, 7, 12)$$
$$C_8 = \text{XOR of bits } (8, 9, 10, 11, 12)$$

The Hamming code

While storing:

Bit position:	1	2	3	4	5	6	7	8	9	10	11	12
	P_1	P_2	1	P_4	1	0	0	P_8	0	1	0	0

$$P_1 = \text{XOR of bits } (3, 5, 7, 9, 11) = 1 \oplus 1 \oplus 0 \oplus 0 \oplus 0 = 0$$
$$P_2 = \text{XOR of bits } (3, 5, 7, 10, 11) = 1 \oplus 0 \oplus 0 \oplus 1 \oplus 0 = 0$$
$$P_4 = \text{XOR of bits } (5, 6, 7, 12) = 1 \oplus 0 \oplus 0 \oplus 0 = 1$$
$$P_8 = \text{XOR of bits } (9, 10, 11, 12) = 0 \oplus 1 \oplus 0 \oplus 0 = 1$$

While reading:

	0	0	1	1	1	0	0	1	0	1	0	0
Bit position:	1	2	3	4	5	6	7	8	9	10	11	12

$$C_1 = \text{XOR of bits } (1, 3, 5, 7, 9, 11)$$
$$C_2 = \text{XOR of bits } (2, 3, 6, 7, 10, 11)$$
$$C_4 = \text{XOR of bits } (4, 5, 6, 7, 12)$$
$$C_8 = \text{XOR of bits } (8, 9, 10, 11, 12)$$

Single correct, double detect

- The Hamming code can detect and correct only a single error
- By adding another parity bit to the coded word, the Hamming code can be used to correct a single error and detect double errors
- If we include this additional parity bit, then the previous 12-bit coded word becomes 001110010100P_{13}, where P_{13} is evaluated from the exclusive-OR of the other 12 bits
- This produces the 13-bit word 0011100101001 (even parity)
- When the 13-bit word is read from memory, the check bits are evaluated, as is the parity P over the entire 13 bits
- The following four cases can arise:

1. If $C = 0$ and $P = 0$, no error occurred
2. If $C = 0$ and $P = 1$, an error occurred in the P_{13} bit!
3. If $C \neq 0$ and $P = 1$, a single error occurred that can be corrected
4. If $C \neq 0$ and $P = 0$, a double error occurred, but that cannot be corrected