Parity bit

From Wikipedia, the free encyclopedia

Jump to: navigation, search

7 bits of data	byte with parity bit	
	even	odd
0000000	0000000**0**	0000000**1**
1010001	1010001**1**	1010001**0**
1101001	1101001**0**	1101001**1**
1111111	1111111**1**	1111111**0**

A **parity bit** is a binary digit that indicates whether the number of bits with value of one in a given set of bits is even or odd. Parity bits are used as the simplest error detecting code.

There are two types of parity bits: **even parity bit** and **odd parity bit**. An even parity bit is set to 1 if the number of ones in a given set of bits is odd (making the number of ones even). An odd parity bit is set to 1 if the number of ones in a given set of bits is even (making the number of ones odd). Even parity is actually a special case of a cyclic redundancy check (CRC), where the 1-bit CRC is generated by the polynomial x+1.

Error detection

If an odd number of bits (including the parity bit) is changed in transmission of a set of bits then parity bit will be incorrect and will thus indicate that an error in transition has occurred. Therefore, parity bit is an error detecting code, but is not an error correcting code as there is no way to determine which particular bit is corrupted. The data must be discarded entirely, and re-transmitted from scratch. On a noisy transmission medium a successful transmission could take a long time, or even never occur. Parity does have the advantage, however, that it is about the best possible code that uses only a single bit of space and it requires only a number of XOR gates to generate. See Hamming code for other error-correction codes.

There is a limitation to parity schemes. A parity bit is only guaranteed to detect an odd number of bit errors (one, three, five, and so on). If an even number of bits (two, four, six and so on) have an error, the parity bit records the correct number of ones, even though the data is corrupt. (See also error detection and correction.)

Usage

Because of its simplicity, parity is used in many hardware applications where an operation can be repeated in case of difficulty, or where simply detecting the error is helpful. For example, the SCSI bus uses parity to detect transmission errors, and many microprocessor instruction caches include parity protection. Because the I-cache data is just a copy of main memory, it can be thrown away and re-fetched if it is found to be corrupted.

In serial data transmission, a common format is 7 data bits, an even parity bit, and one or two stop bits. This format neatly accommodates all the 7-bit ASCII characters in a convenient 8-bit byte. Other formats are possible; 8 bits of data plus a parity bit can convey all 8-bit byte values.

In serial communication contexts, parity is usually generated and checked by interface hardware (e.g., a UART) and, on reception, the result made available to the CPU (and so to, for instance, the operating system) via a status bit in a register in the interface hardware. Recovery from the error is usually done by retransmitting the data, the details of which are usually handled by software (e.g., the operating system I/O routines).

Parity block

A parity block is used by certain RAID levels, redundancy is achieved by the use of parity blocks. If a single drive in the array fails, data blocks and a parity block from the working drives can be combined to reconstruct the missing data.

Given the diagram below, where each column is a disk, assume A1 = 00000111, A2 = 00000101, and A3 = 0000000. Ap, generated by XORing A1, A2, and A3, will then equal 00000010. If the second drive fails, A2 will no longer be accessible, but can be reconstructed by XORing A1, A3, and Ap:

A1 XOR A3 XOR Ap = 00000101

```
       RAID Array
 A1         A2         A3
 Ap         B1         B2
 Bp         C1         C2
 C3         C4         Cp
Note: Data blocks are in the format A#, parity blocks Ap.
```