

2-12 CÓDIGOS DE DETECÇÃO E CORREÇÃO DE ERRO

Nesta seção, discutiremos dois métodos que acrescentam bits aos códigos com a finalidade de detectar erro num único bit ou detectar e corrigir erro num único bit. O método da paridade de detecção de erro é introduzido e o método Hamming de detecção e correção de erro num único bit é abordado com mais detalhes. Quando é identificado que um bit numa dada palavra de código está errado, ele pode ser corrigido fazendo simplesmente a inversão do bit.

Ao final do estudo desta seção você deverá ser capaz de:

- Determinar se existe um erro num código baseado no bit de paridade
- Associar a um código o bit de paridade apropriado
- Usar o código de Hamming para detecção e correção de erro num único bit
- Associar os bits de paridade adequados para correção de erro num único bit

Método da Paridade para Detecção de Erro

Muitos sistemas usam um bit de paridade como um meio de **detecção de erro** de bit. Qualquer grupo de bits possui um número de 1s par ou ímpar. Um bit de paridade é acrescentado a um grupo de bits para tornar o número de 1s no grupo sempre par ou sempre ímpar. Um bit de paridade par torna o número de 1s par e um bit de paridade ímpar torna ímpar o total de bits.

Um dado sistema pode operar com **paridade** par ou ímpar, porém não ambas. Por exemplo, se um sistema opera com paridade par, é feita uma verificação em cada grupo de bits recebido para certificar-se de que o número total de 1s no grupo seja par. Caso exista um número ímpar de 1s, ocorreu um erro.

Como uma ilustração da forma com que os bits de paridade são acrescentados a um código, a Tabela 2-10 apresenta uma lista dos bits de paridade para cada número BCD tanto para a paridade par quanto para a ímpar. O bit de paridade para cada número BCD está na coluna *P*.

Um bit de paridade diz se o número de 1s é ímpar ou par.

PARIDADE PAR		PARIDADE ÍMPAR	
<i>P</i>	BCD	<i>P</i>	BCD
0	0000	1	0000
1	0001	0	0001
1	0010	0	0010
0	0011	1	0011
1	0100	0	0100
0	0101	1	0101
0	0110	1	0110
1	0111	0	0111
1	1000	0	1000
0	1001	1	1001

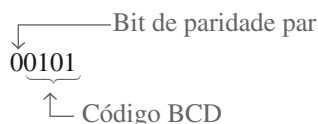
◀ TABELA 2-10

O código BCD com bits de paridade

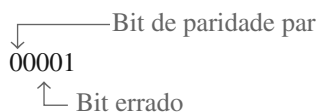
O bit de paridade pode ser acrescentado ao início ou ao final do código, dependendo do projeto do sistema. Observe que o número total de 1s, incluindo o bit de paridade, é sempre par para a paridade par e sempre ímpar para a paridade ímpar.

Detecção de um Erro Um bit de paridade provê a detecção de erro num único bit (ou qualquer número ímpar de erros, que é bem pouco provável) mas não pode verificar dois erros num grupo. Por exemplo, vamos admitir que desejamos transmitir o código BCD 0101. (A paridade pode ser

usada com qualquer número de bits; estamos usando quatro bits como ilustração.) O código total transmitido, incluindo o bit de paridade par, é:



Agora vamos admitir que ocorra um erro no terceiro bit a partir da esquerda (o 1 vira 0).



Quando esse código é recebido, o circuito de verificação de paridade determina que existe apenas um único 1 (paridade ímpar), quando deveria haver um número par de 1s. Devido ao número par de 1s não aparecer no código recebido, é indicado um erro.

Um bit de paridade ímpar também provê uma forma de detecção de erro num único bit num dado grupo de bits.

EXEMPLO 2-39

Associe o bit de paridade par apropriado para os seguintes grupos de códigos:

- (a) 1010 (b) 111000 (c) 101101
(d) 1000111001001 (e) 101101011111

Solução Faça o bit de paridade 0 ou 1 conforme necessário para tornar o número total de 1s par. O bit de paridade será o bit mais à esquerda (colorido).

- (a) 01010 (b) 1111000 (c) 0101101
(d) 0100011100101 (e) 1101101011111

Problema relacionado Acrescente um bit de paridade par ao código ASCII de 7 bits para a letra K.

EXEMPLO 2-40

Um sistema de paridade ímpar recebe os seguintes grupos de código: 10110, 11010, 110011, 110101110100 e 1100010101010. Determine quais grupos, se houver algum, estão com erro.

Solução Como é informado que a paridade é ímpar, qualquer grupo com um número par de 1s está incorreto. Os seguintes grupos estão com erro: **110011** e **1100010101010**.

Problema relacionado O seguinte caractere ASCII é recebido por um sistema de paridade ímpar: 00110111. Ele está correto?

O Código de Correção de Erro Hamming

Conforme estudado, um único bit de paridade permite a detecção de erro num único bit numa palavra de código. Um único bit de paridade pode indicar que existe um erro num certo grupo de bits. Para corrigir um erro detectado, mais informação é necessária porque a posição do bit errado tem que ser identificada antes que ele possa ser corrigido. Mais do que um bit de paridade tem que ser incluído no grupo de bits para tornar possível a correção do erro detectado. Em um código de 7 bits, existem sete possibilidades de erro num único bit. Nesse caso, três bits de paridade podem não apenas detectar um erro mas podem especificar a posição do bit errado. O **código Hamming** provê a correção de um único erro. A abordagem a seguir ilustra a construção de um código Hamming de 7 bits para a correção de um único erro.

Número de Bits de Paridade Se o número de bits de dados projetado for d , então o número de bits de paridade, p , é determinado pela seguinte relação:

$$2^p \geq d + p + 1$$

Por exemplo, se temos quatro bits de dados, então p é determinado por tentativa e erro por meio da Equação 2-1. Façamos $p = 2$. Então:

$$2^p = 2^2 = 4$$

e

$$d + p + 1 = 4 + 2 + 1 = 7$$

Como 2^p tem que ser igual ou maior a $d + p + 1$, a relação na Equação 2-1 não é satisfeita. Temos que tentar novamente. Façamos $p = 3$. Então:

$$2^p = 2^3 = 8$$

e

$$d + p + 1 = 4 + 3 + 1 = 8$$

Esse valor de p satisfaz a Equação 2-1, assim são necessários três bits de paridade para proporcionar a correção de um único erro para quatro bits de dados. Deve-se notar que a detecção e correção são proporcionadas por todos os bits, de paridade e de dados, no grupo de código; ou seja, os bits de paridade também são verificados.

Inserção de Bits de Paridade no Código Agora que sabemos determinar o número de bits de paridade necessários no nosso exemplo particular, temos que arranjar os bits adequadamente no código. Devemos saber que nesse exemplo o código é composto de quatro bits de dados e três bits de paridade. O bit mais à esquerda é designado como bit 1, o próximo bit é o 2 e assim por diante, conforme a seguir:

bit 1, bit 2, bit 3, bit 4, bit 5, bit 6, bit 7

Os bits de paridade estão localizados nas posições que são numeradas em correspondência às potências de dois ascendentes (1, 2, 4, 8,...), conforme indicado:

$P_1, P_2, D_1, P_3, D_2, D_3, D_4$

O símbolo P_n designa um bit de paridade em particular e D_n designa um bit de dado em particular.

Determinação dos Valores dos Bits de Paridade Finalmente, temos que designar adequadamente o valor 0 ou 1 a cada bit de paridade. Como cada bit de paridade provê uma verificação em outros determinados bits no código total, temos que saber o valor desses outros bits para determinar o valor do bit de paridade. Para determinar o valor do bit, primeiro numere cada posição de bit em binário, ou seja, escreva o número binário para cada número decimal da posição, conforme mostra a segunda e terceira linhas da Tabela 2-11. Em seguida, indique a localização dos bits de dados e de paridade, conforme mostra a primeira linha da Tabela 2-11. Observe que o número da posição em binário do bit de paridade P_1 tem um 1 no dígito mais à direita. *Esse bit de paridade verifica as posições de todos os bits, incluindo ele mesmo, que têm 1s na mesma posição nos números de posição em binário.* Portanto, o bit de paridade P_1 verifica as posições de bit 1, 3, 5 e 7.

▼ TABELA 2-11

Tabela de posicionamento dos bits para um código de correção de erro de 7 bits

DESIGNAÇÃO DOS BITS	P_1	P_2	D_1	P_3	D_2	D_3	D_4
POSIÇÃO DOS BITS	1	2	3	4	5	6	7
NÚMERO DA POS. EM BINÁRIO	001	010	011	100	101	110	111
Bits de dados (D_n)							
Bits de paridade (P_n)							

O número da posição em binário do bit de paridade P_2 tem um 1 no bit do meio. Ele verifica todas as posições de bit, incluindo ele mesmo, que têm 1 na mesma posição. Portanto, o bit de paridade P_2 verifica os bits das posições 2, 3, 6 e 7.

O número da posição em binário para o bit de paridade P_3 tem um 1 no bit mais à esquerda. Ele verifica todas as posições de bit, incluindo ele mesmo, que têm 1s na mesma posição. Portanto, o bit de paridade P_3 verifica as posições de bit 4, 5, 6 e 7.

Em cada caso, ao bit de paridade é designado um valor que torna a quantidade de 1s, no conjunto de bits que ele verifica, par ou ímpar, dependendo do que for especificado. Os exemplos a seguir devem tornar esse procedimento mais claro.

EXEMPLO 2-41

Determine o código de Hamming para o número BCD 1001 (bits de dados), usando paridade par.

Solução **Passo 1** Determine o número de bits de paridade necessários. Façamos $p = 3$. então:

$$2^p = 2^3 = 8$$

$$d + p + 1 = 4 + 3 + 1 = 8$$

Três bits de paridade são suficientes.

$$\text{Total de bits do código} = 4 + 3 = 7$$

Passo 2 Construa uma tabela de posições de bits, conforme mostra a Tabela 2-12 e insira os bits de dados. Os bits de paridade são determinados nos passos a seguir.

▼ TABELA 2-12

DESIGNAÇÃO DOS BITS	P_1	P_2	D_1	P_3	D_2	D_3	D_4
POSIÇÃO DOS BITS	1	2	3	4	5	6	7
NÚMERO DA POS. EM BINÁRIO	001	010	011	100	101	110	111
Bits de dados			1		0	0	1
Bits de paridade	0	0		1			

Passo 3 Determine os bits de paridade como a seguir:

O bit P_1 verifica os bits das posições 1, 3, 5 e 7 e tem que ser 0 para que o número de 1s (2) seja par nesse grupo.

O bit P_2 verifica os bits das posições 2, 3, 6 e 7 e tem que ser 0 para que o número de 1s (2) seja par nesse grupo.

O bit P_3 verifica os bits das posições 4, 5, 6 e 7 e tem que ser 1 para que o número de 1s (2) seja par nesse grupo.

Passo 4 Esses bits de paridade são inseridos na Tabela 2-12 e o código combinado resultante é 0011001.

Problema relacionado Determine o código Hamming para o número BCD 1000 usando a paridade par.

EXEMPLO 2-42

Determine o código Hamming para os bits de dados 10110 usando a paridade ímpar.

Solução **Passo 1** Determine o número de bits de paridade necessários. Nesse caso o número de bits de dados, d , é cinco. A partir do exemplo anterior sabemos que fazendo $p = 3$ não adianta. Experimente fazer $p = 4$:

$$2^p = 2^4 = 16$$

$$d + p + 1 = 5 + 4 + 1 = 10$$

Quatro bits de paridade são suficientes.

$$\text{Total de bits do código} = 5 + 4 = 9$$

Passo 2 Construa uma tabela de posições de bits, Tabela 2-13, e insira os bits de dados. Os bits de paridade são determinados de acordo com os passos a seguir. Observe que P_4 está na posição do bit 8.

Passo 3 Determine os bits de paridade como a seguir:

▼ TABELA 2-13

DESIGNAÇÃO DOS BITS	P_1	P_2	D_1	P_3	D_2	D_3	D_4	P_4	D_5
POSIÇÃO DOS BITS	1	2	3	4	5	6	7	8	9
NÚMERO DA POS. EM BINÁRIO	0001	0010	0011	0100	0101	0110	0111	1000	1001
Bits de dados			1		0	1	1		0
Bits de paridade	1	0		1				1	

O bit P_1 verifica os bits das posições 1, 3, 5, 7 e 9 e tem que ser 1 para que o número de 1s (3) seja ímpar nesse grupo.

O bit P_2 verifica os bits das posições 2, 3, 6 e 7 e tem que ser 0 para que o número de 1s (3) seja ímpar nesse grupo.

O bit P_3 verifica os bits das posições 4, 5, 6 e 7 e tem que ser 1 para que o número de 1s (3) seja ímpar nesse grupo.

O bit P_4 verifica os bits das posições 8 e 9 e tem que ser 1 para que o número de 1s (1) seja ímpar nesse grupo.

Passo 4: Esses bits de paridade são inseridos na Tabela 2-13 e o código combinado resultante é 101101110.

Problema relacionado Determine o código Hamming para 11001 usando paridade ímpar.

Detecção e Correção de Erro com o Código de Hamming

Agora que o método Hamming para construção de um código de erro foi abordado, como o usamos para localizar e corrigir um erro? Cada bit de paridade, ao longo dos seu grupos de bits correspondentes, tem que ser verificado para a paridade adequada. Caso existam três bits de paridade na palavra de código, são geradas três verificações. Caso existam quatro bits de paridade, são geradas quatro verificações, e assim por diante. Cada verificação de paridade apresenta um resul-

tado bom ou ruim. O resultado total de todas as verificações de paridade indica o bit, se houver algum, que está errado, como a seguir:

- Passo 1** Comece com o grupo verificado por P_1 .
- Passo 2** Verifique o grupo quanto a paridade correta. Um 0 representa uma verificação de paridade correta e um 1 representa uma verificação incorreta.
- Passo 3** Repita o passo 2 para cada grupo de paridade.
- Passo 4** O número binário formado pelo resultado de todas as verificações de paridade determina a posição do bit do código que está errado. Esse é o *código de posição de erro*. A primeira verificação de paridade gera o bit menos significativo (LSB). Se todas as verificações forem corretas, não há erro.

EXEMPLO 2-43

Considere que a palavra de código dada no Exemplo 2-41 (0011001) seja transmitida e que 0010001 seja recebida. O receptor não “sabe” o que foi transmitido e tem que testar as paridades para determinar se o código está correto. Determine qualquer erro que tenha ocorrido na transmissão se a paridade usada foi a par.

Solução Primeiro, faça uma tabela de posição de bit, conforme indicado na Tabela 2-14.

▼ TABELA 2-14

DESIGNAÇÃO DOS BITS	P_1	P_2	D_1	P_3	D_2	D_3	D_4
POSIÇÃO DOS BITS	1	2	3	4	5	6	7
NÚMERO DA POSIÇÃO EM BINÁRIO	001	010	011	100	101	110	111
Código recebido	0	0	1	0	0	0	1

Primeira verificação de paridade:

O bit P_1 verifica as posições 1, 3, 5 e 7.

Existem dois 1s nesse grupo.

A verificação de paridade é correta. → 0 (LSB)

Segunda verificação de paridade:

O bit P_2 verifica as posições 2, 3, 6 e 7.

Existem dois 1s nesse grupo.

A verificação de paridade é correta. → 0

Terceira verificação de paridade:

O bit P_3 verifica as posições 4, 5, 6 e 7.

Existe um 1 nesse grupo.

A verificação de paridade é incorreta. → 1 (MSB)

Resultado:

O código de posição de erro é 100 (binário quatro). Isso diz que o bit na posição 4 está errado. Ele é 0 e deveria ser 1. O código corrigido é 0011001, que está de acordo com o código transmitido.

Problema relacionado Repita o processo ilustrado nesse exemplo se o código recebido for 0111001.

EXEMPLO 2-44

O código 101101010 é recebido. Corrija qualquer erro. Existem quatro bits de paridade, sendo que a paridade usada é a ímpar.

Solução Primeiro, faça uma tabela de posição de bit como a Tabela 2-15.

▼ **TABELA 2-15**

DESIGNAÇÃO DOS BITS	P_1	P_2	D_1	P_3	D_2	D_3	D_4	P_4	D_5
POSIÇÃO DOS BITS	1	2	3	4	5	6	7	8	9
NÚMERO DA POS. EM BINÁRIO	0001	0010	0011	0100	0101	0110	0111	1000	1001
Código recebido	1	0	1	1	0	1	0	1	0

Primeira verificação de paridade:

O bit P_1 verifica as posições 1, 3, 5, 7 e 9.

Existem dois 1s nesse grupo.

A verificação de paridade é incorreta. —————> 1 (LSB)

Segunda verificação de paridade:

O bit P_2 verifica as posições 2, 3, 6 e 7.

Existem dois 1s nesse grupo.

A verificação de paridade é incorreta. —————> 1

Terceira verificação de paridade:

O bit P_3 verifica as posições 4, 5, 6 e 7.

Existem dois 1s nesse grupo.

A verificação de paridade é incorreta. —————> 1

Quarta verificação de paridade:

O bit P_4 verifica as posições 8 e 9.

Existe um 1 nesse grupo.

A verificação de paridade é correta. —————> 0 (MSB)

Resultado:

O código de posição de erro é 0111 (binário sete). Isso diz que o bit na posição 7 está errado. O código correto é portanto 101101110.

Problema relacionado O código 101111001 é recebido. Corrija qualquer erro se a paridade ímpar foi usada.

SEÇÃO 2-12
REVISÃO

- Qual código de paridade ímpar está errado?
(a) 1011 (b) 1110 (c) 0101 (d) 1000
- Qual código de paridade par está errado?
(a) 11000110 (b) 00101000 (c) 10101010 (d) 11111011
- Acrescente um bit de paridade par no final de cada um dos seguintes códigos:
(a) 1010100 (b) 0100000 (c) 1110111 (d) 10001100
- Quantos bits de paridade são necessários para os bits de dados 11010 usando o código de Hamming?
- Crie o código de Hamming para os bits de dados 0011 usando a paridade par.