Detecção e Correção de Erros

- Erros de leitura da mídia estão presentes na detecção
  - Ruídos aleatórios, pequenos defeitos na mídia, variação de distância mídia-cabeça
  - Taxa típica de erros nos HD's: 10^{-7} a 10^{-9}
- Códigos de correção de erros (ECC)
  - Inclusão de redundância na informação
  - Bits adicionais incluídos após os dados
  - Algorítmos de detecção e correção aplicados

Detecção e Correção de Erros

- Principais algoritmos:
  - Bit de paridade: 1 bit de paridade adicionado após cada byte
  - Reed Solomon: vários bits de paridade calculados a partir de um conjunto de bytes

Detecção e Correção de Erros

- Bit de paridade:
  - Paridade refere-se ao fato de um inteiro ser par ou ímpar
  - Para um número binário, o bit menos significativo indica a paridade
  - Técnica simples amplamente utilizada para verificação de erro nas memórias DRAM, interface RS232, etc.
  - Indica que há erro mas não informa a posição nem possibilita correção
  - Caso exista erro a leitura é refeita
  - Memória possui dados de 9 bits (*byte*) ou 18 (*word*)

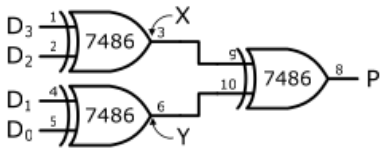
Detecção e Correção de Erros

- Bit de paridade:
  - Paridade par (even): bit de paridade=”0” quando soma é par e “1” quando soma é ímpar
  - Paridade ímpar (odd): bit de paridade=”1” quando soma é par e “0” quando soma é ímpar

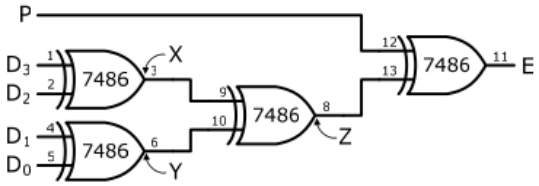
7 bits of data (number of 1s)	8 bits including parity	
	even	odd
0000000 (0)	**0**0000000	**1**0000000
1010001 (3)	**1**1010001	**0**1010001
1101001 (4)	**0**1101001	**1**1101001
1111111 (7)	**1**1111111	**0**1111111

Detecção e Correção de Erros

- Circuito para geração do bit de paridade par:



- Circuito para verificação do bit de paridade:
  - E=”0” => dados corretos; E=”1” => erro



Detecção e Correção de Erros

- Características:
  - Algorítmo e circuito equivalente simples e rápido
  - Capaz de identificar se existe erro em 1 byte da sequência de bits
  - Erros em 2 bytes são indicados errôneamente
  - Sujeito a erro no próprio bit de paridade
  - Não identifica a posição do erro

Detecção e Correção de Erros

- Algoritmo Reed-Solomon
  - Desenvolvido por Irving Reed and Gustave Solomon (1960)
  - Usado na detecção de erros de sinais digitais (armazenamento e transmissão)
  - Pode detectar e corrigir grande número de bits de forma simples e eficiente comparado a outros códigos
  - Necessita do menor número de bits extra para correção de um conjunto de bits de informação

Detecção e Correção de Erros

- Aplicações do Reed-Solomon
  - Memórias (HD, CD, DVD, BD, barcodes)
  - Comunicações sem fio
  - Televisão Digital
  - Comunicações via Satélite
  - Modem banda larga (ADSL, xDSL etc)
  - etc

Detecção e Correção de Erros

- Algoritmo Reed-Solomon
  - Teorema de álgebra linear: conjunto de *n* pontos distintos determinam polinômio de grau máximo *n-1*
  - Baseado na sobre-amostragem de um polinômio construído a partir dos dados binários
  - Livro-código construído a partir de:

$$\mathbf{C} = \{(f(x_1), f(x_2), ..., f(x_n)), f \in F[x], deg(f) < k\}$$

  - Onde:
    - *k*: número de dados da informação
    - *n*: número total de dados incluindo ECC

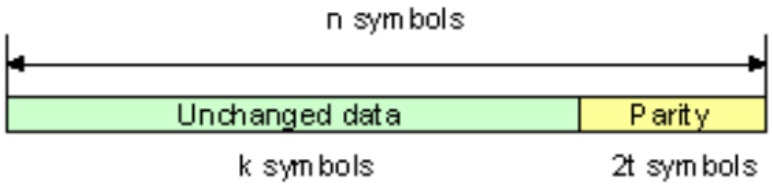
Detecção e Correção de Erros

- Algoritmo Reed-Solomon
  - Bytes de redundância: *m*= *n* - *k*
  - Capaz de detectar $\frac{n-k}{2}$ erros
  - Ex: Para cada setor em um HD padrão:
    - *k*=512 Bytes
    - *m*=48 Bytes
    - *n*=512+48 Bytes

    Detecção de até 24 bytes errôneos

Detecção e Correção de Erros

- Bits adicionais de ECC
  - Calculados pelo controlador de disco (matriz de Vandermonde)
  - Gravados após os dados em cada setor da mídia
  - Compromisso entre:
    - Bits usados para EEC em relação aos bits da informação
    - Capacidade de processamento do controlador



RS(n,k) code

Detecção e Correção de Erros

Análise e correção pelo controlador de disco:

- Detecção do erro
- Correção Reed-Solomon (Gaussian Elimination)
- Releitura do setor caso o erro persista

Detecção e Correção de Erros

Geração dos bits de ECC:

- matriz de Vandermonde:

$$c_i = F_i(d_1, d_2, \ldots, d_n) = \sum_{j=1}^{n} d_j f_{i,j}$$

$$FD = C.$$

$$\begin{bmatrix} f_{1,1} & f_{1,2} & \ldots & f_{1,n} \\ f_{2,1} & f_{2,2} & \ldots & f_{2,n} \\ \vdots & \vdots & & \vdots \\ f_{m,1} & f_{m,2} & \ldots & f_{m,n} \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_n \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & \ldots & 1 \\ 1 & 2 & 3 & \ldots & n \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & 2^{m-1} & 3^{m-1} & \ldots & n^{m-1} \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_n \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_m \end{bmatrix}$$

Detecção e Correção de Erros

Recuperação dos bits errôneos:

- Método de eliminação de Gauss:

$$A = \begin{bmatrix} I \\ F \end{bmatrix}$$

$$E = \begin{bmatrix} D \\ C \end{bmatrix}$$

$$AD = E$$

$$\begin{bmatrix} 1 & 0 & 0 & \ldots & 0 \\ 0 & 1 & 0 & \ldots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \ldots & 1 \\ 1 & 1 & 1 & \ldots & 1 \\ 1 & 2 & 3 & \ldots & n \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & 2^{m-1} & 3^{m-1} & \ldots & n^{m-1} \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_n \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_n \\ c_1 \\ c_2 \\ \vdots \\ c_m \end{bmatrix}$$

Detecção e Correção de Erros

- Aritmética baseada em “Galois Fields”:
  - Campo limitado de números
  - Qualquer operação entre estes números gera resultado dentro do próprio campo
  - Adaptada a operações binárias
  - Soma e subtração utilizam portas XOR (sem carry)
  - Ex: RS com símbolos de 8 bits usa um “Galois field” GF(2^8), contendo 256 símbolos
  -

Detecção e Correção de Erros

- Links externos:
  - http://www.eccpage.com/
  - http://www.highlandcomm.com/reed_solomon_codes.htm
  - http://zanotti.univ-tln.fr/enseignement/divers/chapter3.html
  - http://en.wikipedia.org/wiki/Cross-interleaved_Reed-Solomon_coding