ORGANIZAÇÃO E ARQUITETURA DE COMPUTADORES

Estrutura interna de computadores - Hamming

Profª. Fabiana F F Peres

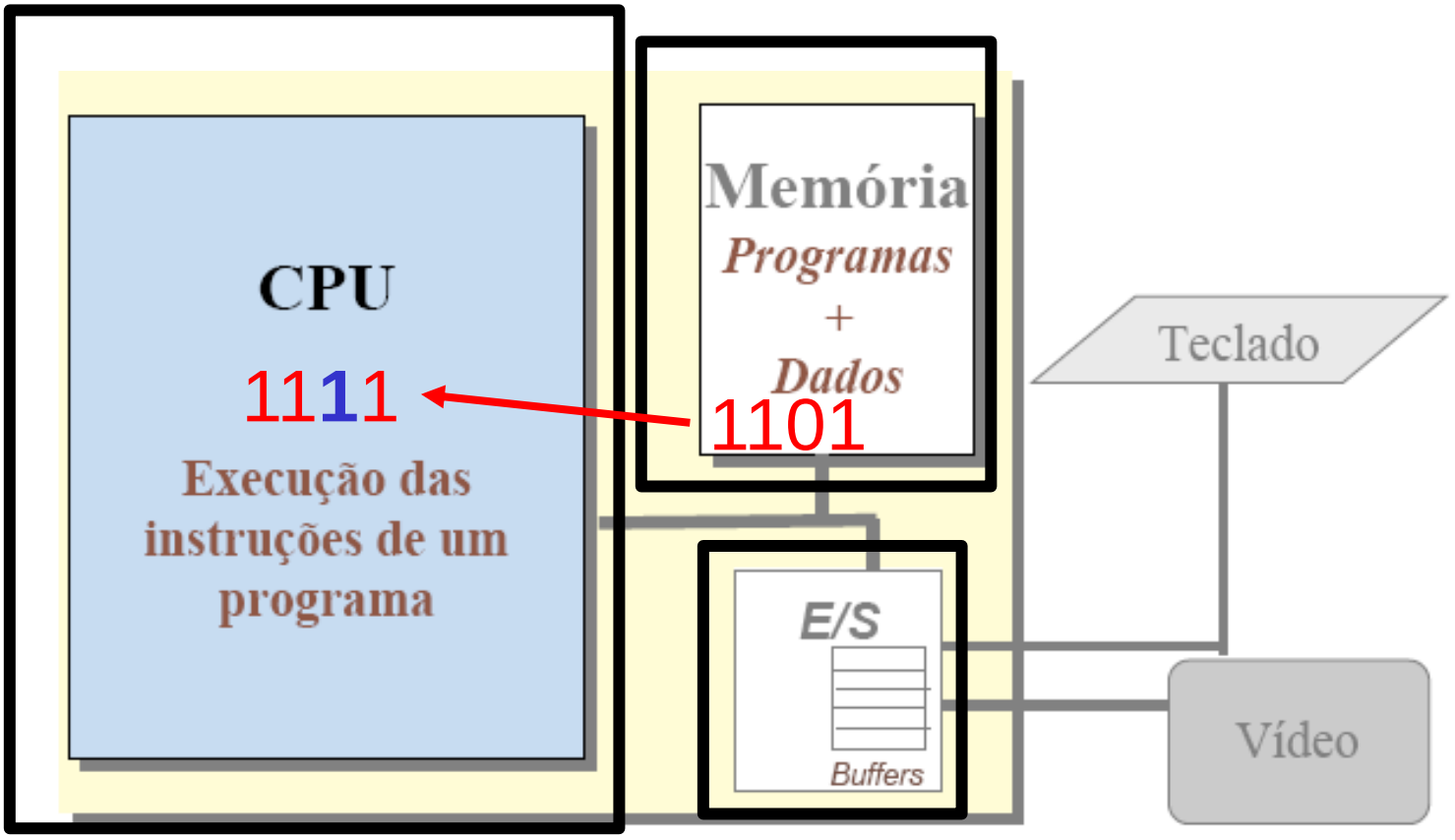
Apoio: Camile Bordini

Memória

- **Ordem dos bytes** ← Aula passada
  - Big endian
  - Little endian

- **Detecção/Correção de erros** ← Nesta aula
  - **Paridade**
  - **Palavra de memória** (armazenada na memória)
  - **Palavra de Código** (transmitida contendo bits de paridade para detector e corrigir erros)

Memória - Detecção e Correção de erros



Memória - Detecção e Correção de erros

- O dados armazenados em memória podem ocasionalmente ser alterados (ex: oscilações na tensão)

- Como prevenção, algumas memórias usam um **código** junto às informações que permite a correção ou detecção de erros
  - Acrescenta-se bits extras a cada palavra de memória
  - Os bits armazenados permitem verificar a ocorrência eventual de erros que tenham corrompido a informação

Memória - Detecção e Correção de erros

- Necessitam de placa mãe adequada para isso
- Necessitam de processadores adequados para isso
- Identificadas como Memórias ECC (*Error Correcting Codes*)
- Utilizadas em servidores

Paridade

- Usado para detectar erros
  - Adiciona-se **1 bit de paridade** na palavra
  - Pode-se usar paridade **par** ou paridade **ímpar**:
    - Paridade par: a quantidade de 1s na palavra é par;
    - Paridade ímpar: a quantidade de 1s na palavra é ímpar;

  - Exemplos: **Palavra (4 bits)**

	1	1	0	1

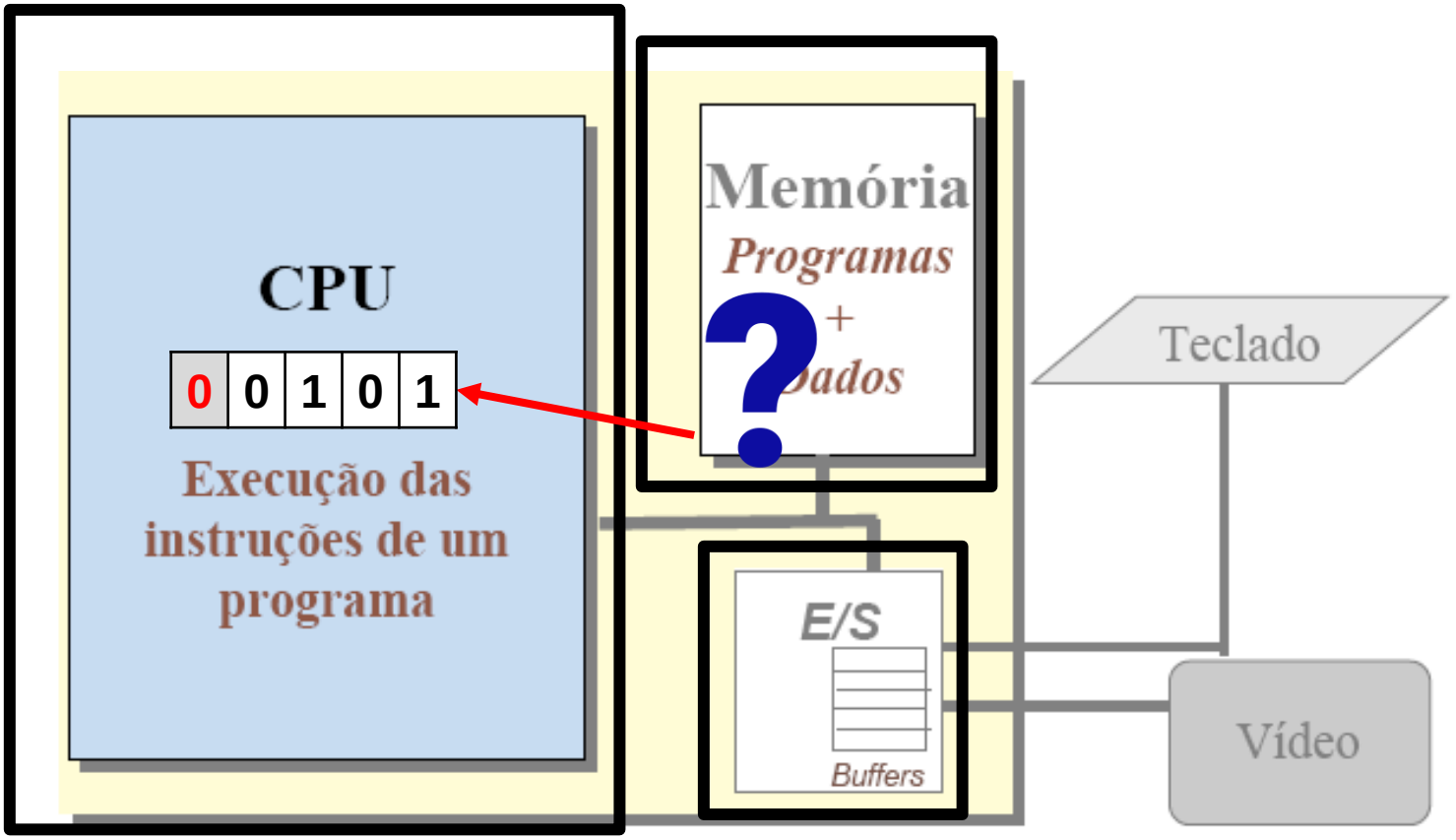
    - Paridade ímpar: **Palavra (4 bits)**

0	1	1	0	1

    - Paridade par: **Palavra (4 bits)**

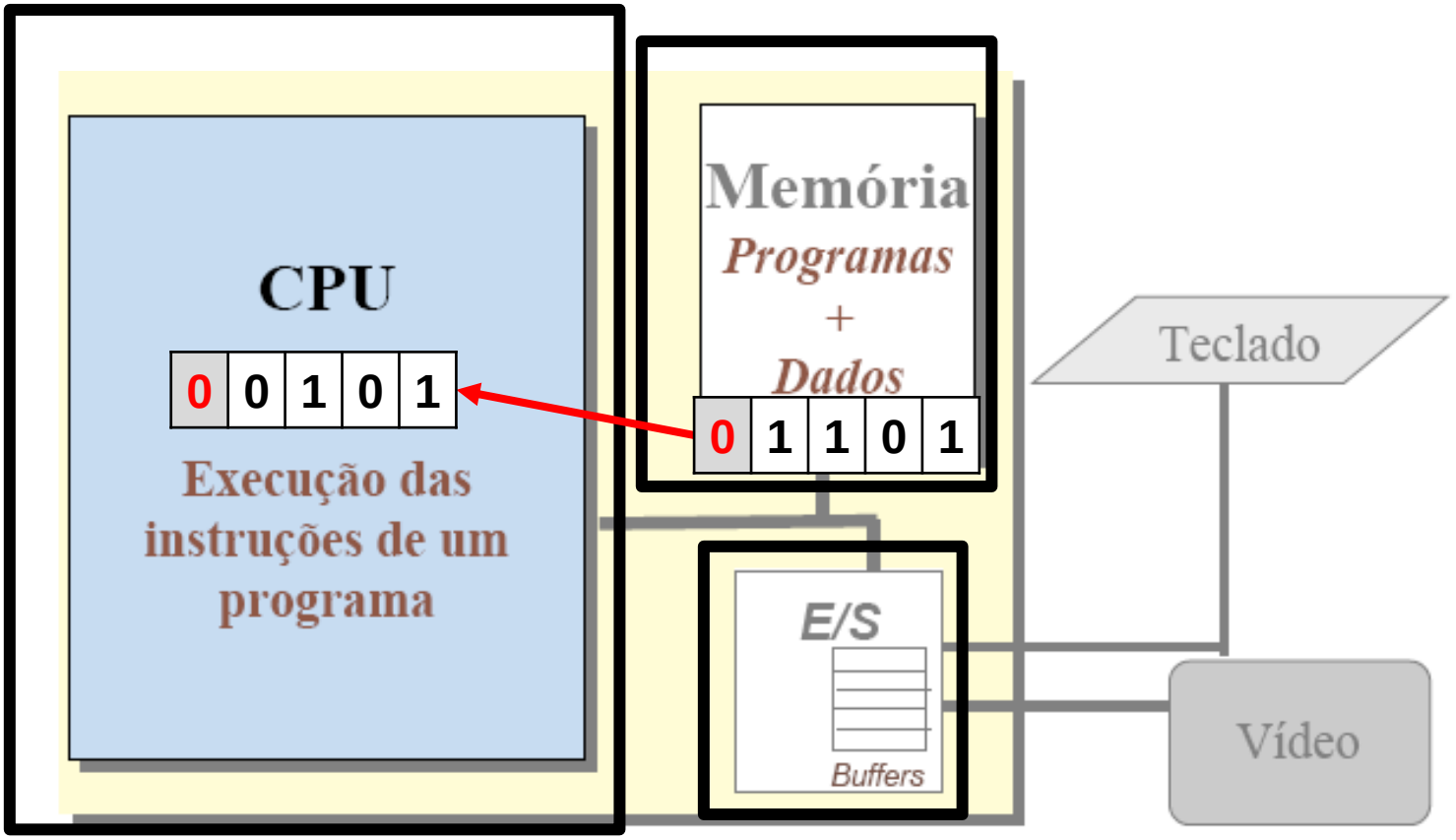
1	1	1	0	1

Ex1: Paridade ímpar



A palavra está correta?

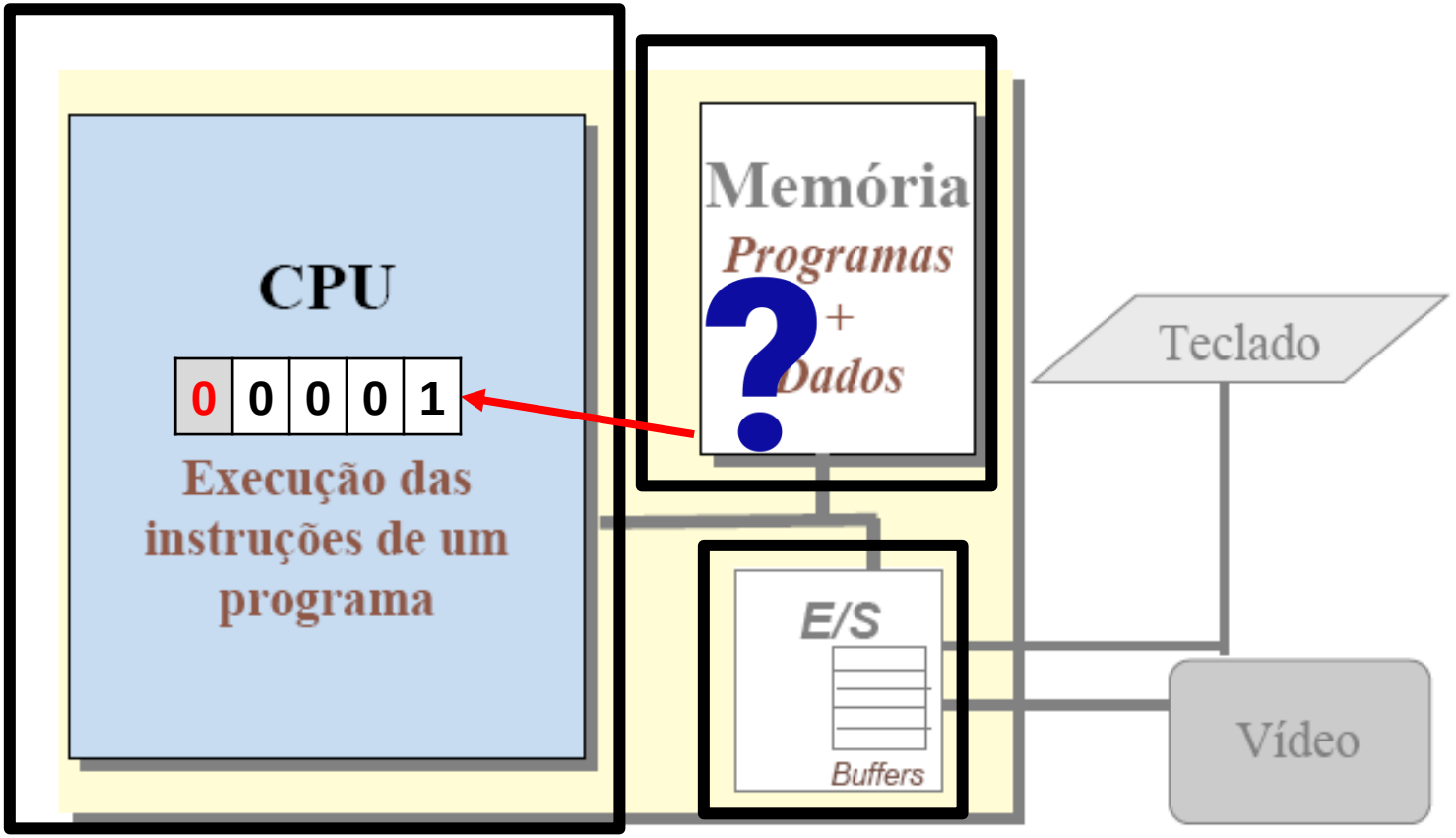
Ex1: Paridade ímpar



A palavra está correta?

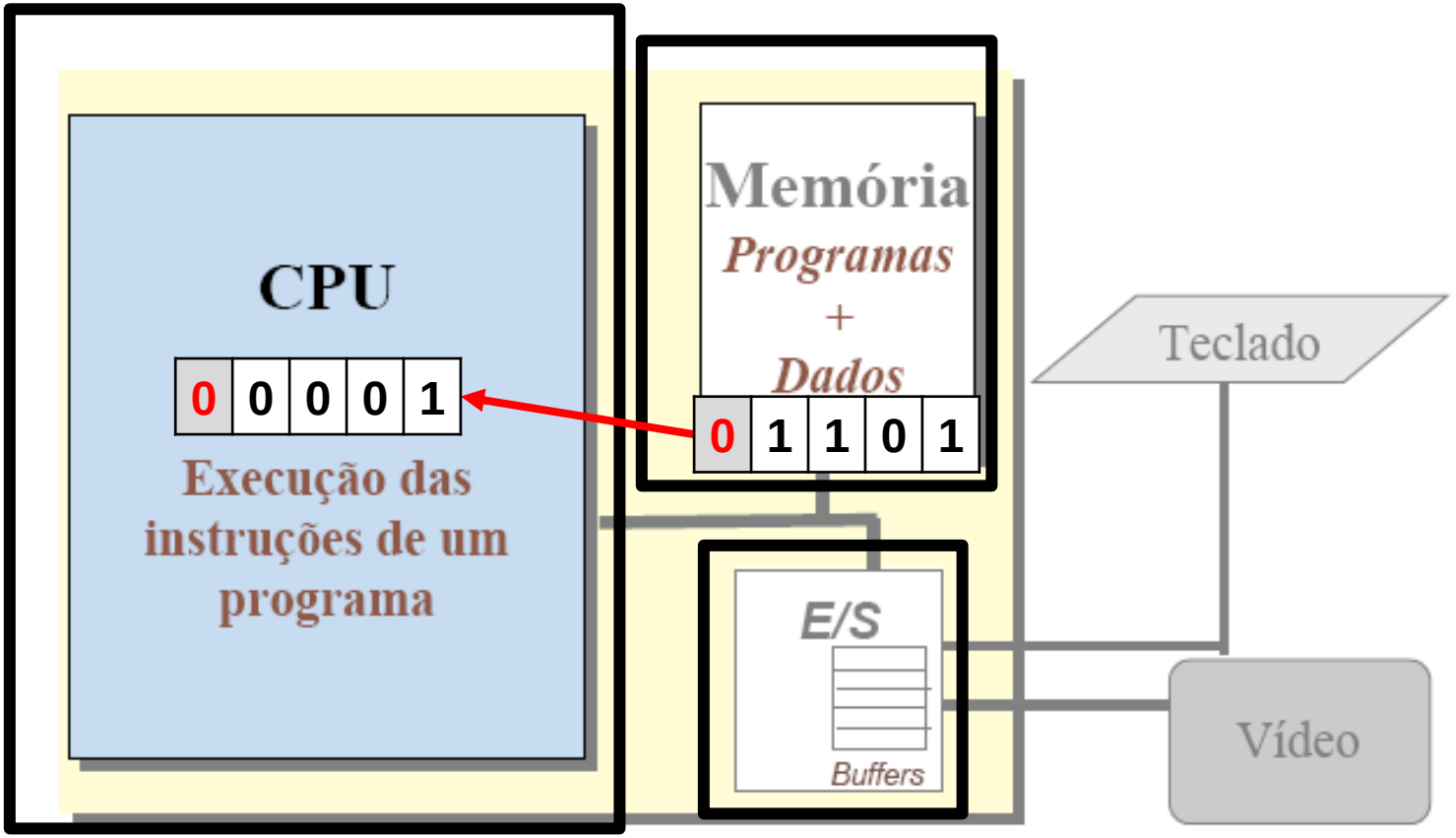
Não!

Ex2: Paridade ímpar



A palavra está correta?

Ex2: Paridade ímpar



A palavra está correta?

Também não!

Código de Hamming

- Adiciona-se vários **bits de paridade** a **palavra de memória**

- Os bits da palavra são numerados começando de 1 da esquerda para a direita

- Todos os bits cuja numeração seja uma potência de 2 são bits de paridade

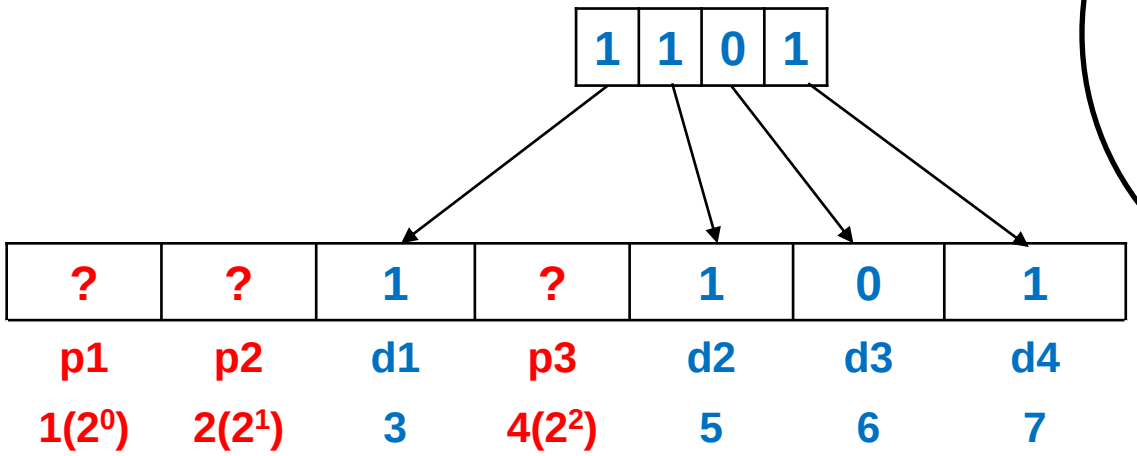
Nomenclatura:

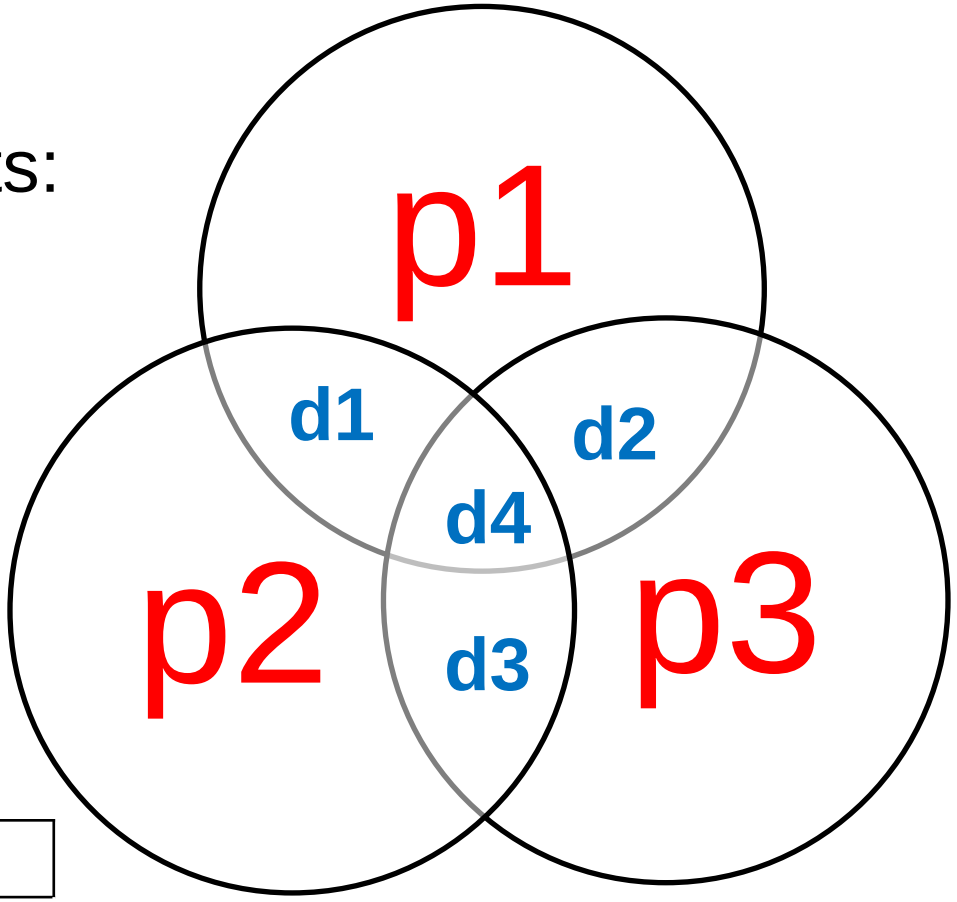
Hamming (nº de bits da palavra de código, nº de bits da palavra de memória)

- Exemplo: **Hamming (7,4)**
  - **Palavra de memória** de 4 bits:

Palavra da memória	1	1	0	1
Posição dos bits na palavra	1	2	3	4

  - **Palavra de código** de 7 bits





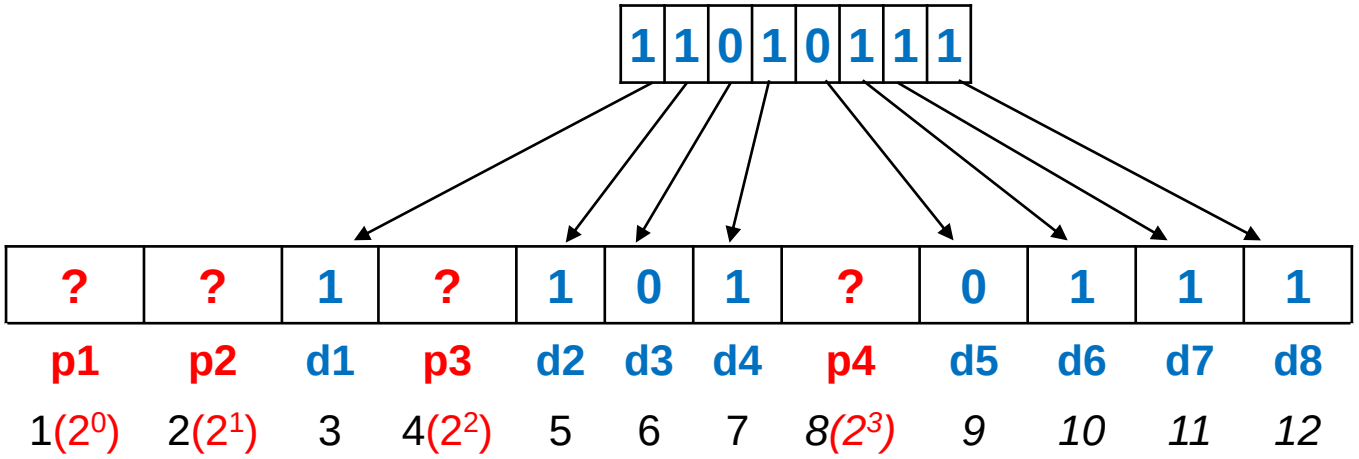
Código de Hamming

- Exemplo: **Hamming (12,8)**

  – **Palavra de memória** de 8 bits:

Palavra da memória	**1**	**1**	**0**	**1**	**0**	**1**	**1**	**1**
Posição dos bits na palavra	1	2	3	4	*5*	*6*	*7*	*8*

  – **Palavra de código** de 12 bits



Código de Hamming

- Exemplo: **Palavra da memória**

1	1	0	1	0	1	1	1
Posição dos bits na palavra	1	2	3	4	*5*	*6*	*7*	*8*

Palavra da código			**1**		**1**	**0**	**1**		**0**	**1**	**1**	**1**
	p1	**p2**	**d1**	**p3**	**d2**	**d3**	**d4**	**p4**	**d5**	**d6**	**d7**	**d8**
Posição dos bits na palavra	$1(2^0)$	$2(2^1)$	3	$4(2^2)$	5	6	7	$8(2^3)$	*9*	*10*	*11*	*12*

Código de Hamming

- Número de **bits de paridade**: **r**
- Tamanho da **Palavra de código** (bits de dados + bits de paridade): $\mathbf{2^r - 1}$
- Tamanho da **Palavra de memória** (bits de dados): $\mathbf{2^r - 1 - r}$

Código de Hamming

Nº de bits de paridade	Palavra de código MÁXIMA	Nº de bits de dados MÁXIMO	Nomenclatura
2	$2^2 - 1 = 3$	$2^2 - 2 - 1 = 1$	Hamming(3,1)
3	$2^3 - 1 = 7$	$2^3 - 3 - 1 = 4$	Hamming(7,4)
4	$2^4 - 1 = 15$	$2^4 - 4 - 1 = 11$	Hamming(15,11)
5	$2^5 - 1 = 31$	$2^5 - 5 - 1 = 26$	Hamming(31,26)
6	$2^6 - 1 = 63$	$2^6 - 6 - 1 = 57$	Hamming(63,57)
7	$2^7 - 1 = 127$	$2^7 - 7 - 1 = 120$	Hamming(127,120)
8	$2^8 - 1 = 255$	$2^8 - 8 - 1 = 247$	Hamming(255,247)
9	$2^9 - 1 = 511$	$2^9 - 9 - 1 = 502$	Hamming(511,502)
10	$2^{10} - 1 = 1023$	$2^{10} - 10 - 1 = 1013$	Hamming(1023,1013)

Código de Hamming

Nº de bits de dados	Nº de bits de paridade	Tamanho da palavra de código	Nomenclatura
1	2	3	**Hamming(3,1)**
2	3	5	**Hamming(5,2)**
3	3	6	**Hamming(6,3)**
4	3	7	**Hamming(7,4)**
5	4	9	**Hamming(9,5)**
6	4	10	**Hamming(10,6)**
7	4	11	**Hamming(11,7)**
8	4	12	**Hamming(12,8)**
9	4	13	**Hamming(13,9)**
10	4	14	**Hamming(14,10)**
11	4	15	**Hamming(15,11)**
12	5	17	**Hamming(17,12)**
13	5	18	**Hamming(18,13)**
14	5	19	**Hamming(19,14)**
15	5	20	**Hamming(20,15)**

← Menor código de Hamming

Código de Hamming

Exercício 1: Para a **palavra de memória** com os seguintes bits de dados **0001**

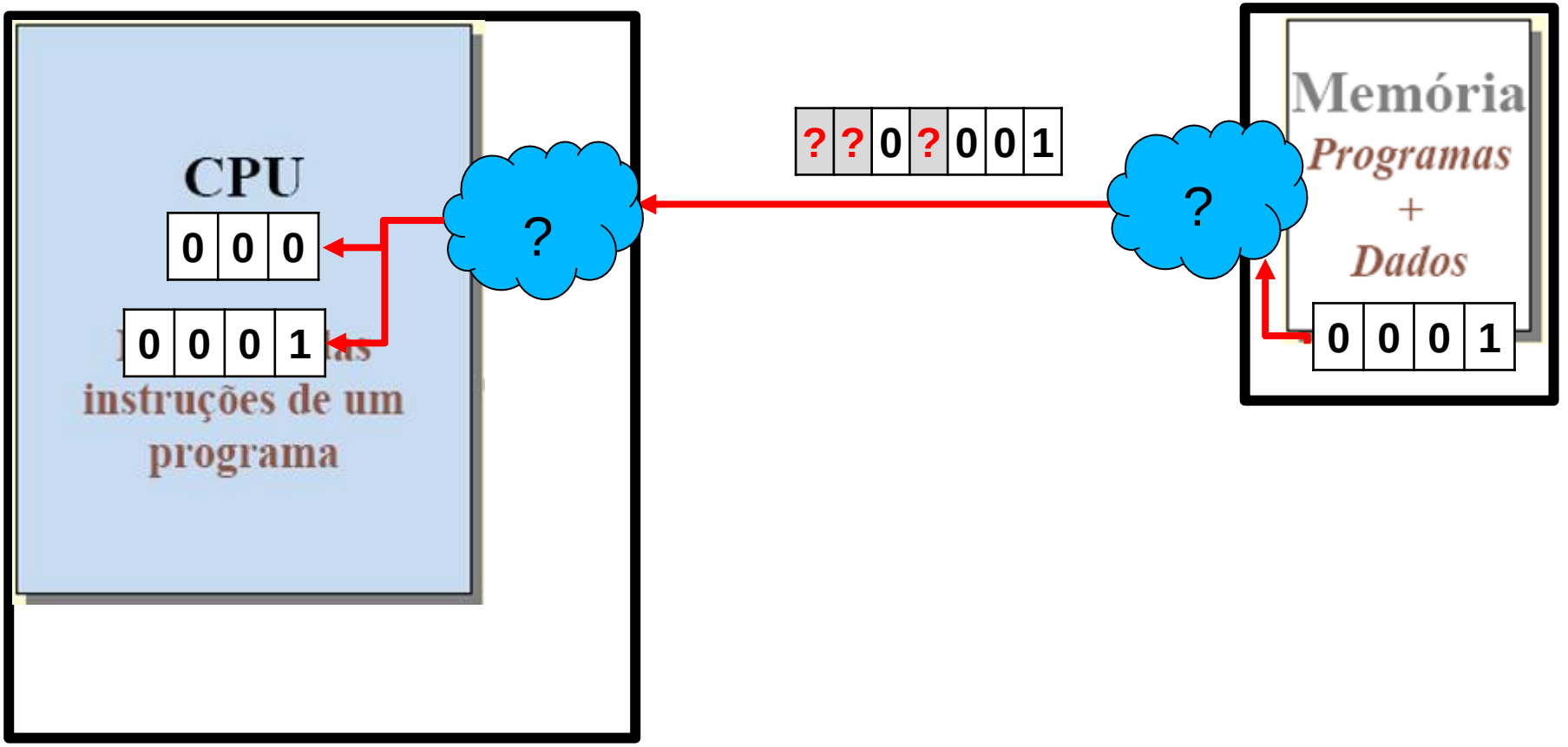
a) qual a quantidade de bits de paridade necessária?

b) qual será o tamanho da palavra de código total?

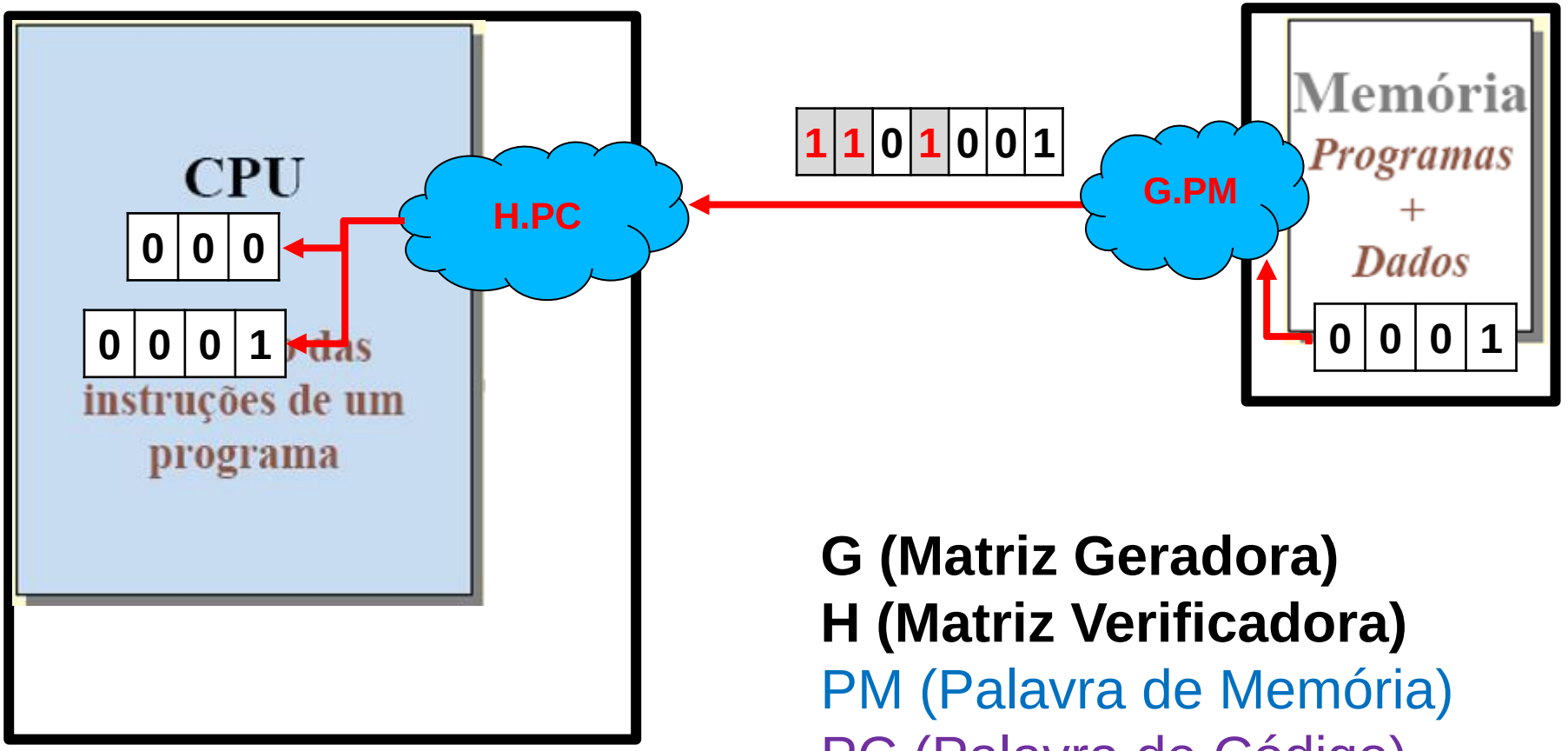
c) qual a sequência de bits na palavra de código resultante?

Exercício 2: faça o mesmo para a palavra **00011**

Código de Hamming



Código de Hamming



G (Matriz Geradora)
H (Matriz Verificadora)
PM (Palavra de Memória)
PC (Palavra de Código)

Matriz Geradora

Matriz Geradora

- Exemplo 1:
  - Palavra de memória de **4** bits (**0001**)
  - Bits de paridade: **3**
  - Palavra de código: **7**

- Portanto, a **matriz geradora** terá dimensões 7x4 e será preenchida conforme a seguir

- *Obs: para todos os exemplos, vamos utilizar sempre a paridade par*

Exemplo 1 - Matriz Geradora

Palavra da código			0		0	0	1
	001	010	011	100	101	110	111
	p1	p2	d1	p3	d2	d3	d4

		1		1	0	1
001	010	011	100	101	110	111
p1	p2	d1	p3	d2	d3	d4

		1		1	0	1
001	010	011	100	101	110	111
p1	p2	d1	p3	d2	d3	d4

		1		1	0	1
001	010	011	100	101	110	111
p1	p2	d1	p3	d2	d3	d4

	d1	d2	d3	d4
p1	1	1	0	1
p2	1	0	1	1
d1	1	0	0	0
p3	?	?	?	?
d2	?	?	?	?
d3	?	?	?	?
d4	?	?	?	?

Exemplo 1 – Matriz Geradora

Palavra da código			0		0	0	1
	001	010	011	100	101	110	111
	p1	p2	d1	p3	d2	d3	d4

		1		1	0	1
001	010	011	100	101	110	111
p1	p2	d1	p3	d2	d3	d4

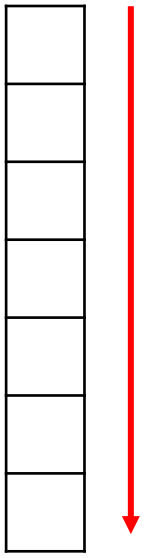
		1		1	0	1
001	010	011	100	101	110	111
p1	p2	d1	p3	d2	d3	d4

		1		1	0	1
001	010	011	100	101	110	111
p1	p2	d1	p3	d2	d3	d4

	d1	d2	d3	d4
p1	1	1	0	1
p2	1	0	1	1
d1	1	0	0	0
p3	0	1	1	1
d2	0	1	0	0
d3	0	0	1	0
d4	0	0	0	1

Exemplo 1 – Matriz Geradora

$$\begin{vmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \cdot \begin{vmatrix} 0 \\ 0 \\ 0 \\ 1 \end{vmatrix} =$$



Exemplo 1 – Matriz Geradora

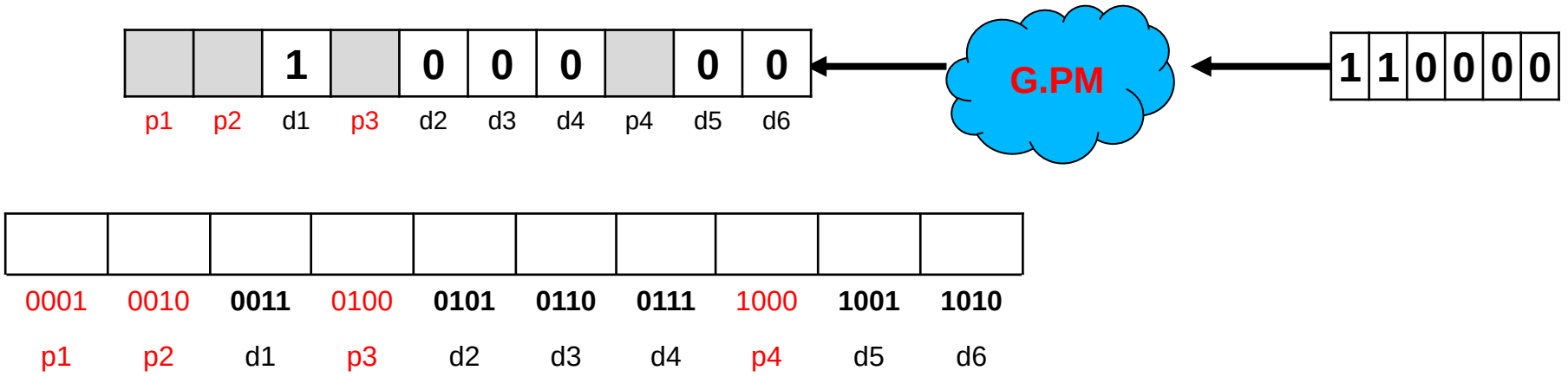
$$\begin{vmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \cdot \begin{vmatrix} 0 \\ 0 \\ 0 \\ 1 \end{vmatrix} = \begin{vmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \end{vmatrix}$$

Código de Hamming

Nº de bits de dados	Nº de bits de paridade	Tamanho da palavra de código	Nomenclatura
1	2	3	**Hamming(3,1)**
2	3	5	**Hamming(5,2)**
3	3	6	**Hamming(6,3)**
4	3	7	**Hamming(7,4)**
5	4	9	**Hamming(9,5)**
6	4	10	**Hamming(10,6)**
7	4	11	**Hamming(11,7)**
8	4	12	**Hamming(12,8)**
9	4	13	**Hamming(13,9)**
10	4	14	**Hamming(14,10)**
11	4	15	**Hamming(15,11)**
12	5	17	**Hamming(17,12)**
13	5	18	**Hamming(18,13)**
14	5	19	**Hamming(19,14)**
15	5	20	**Hamming(20,15)**

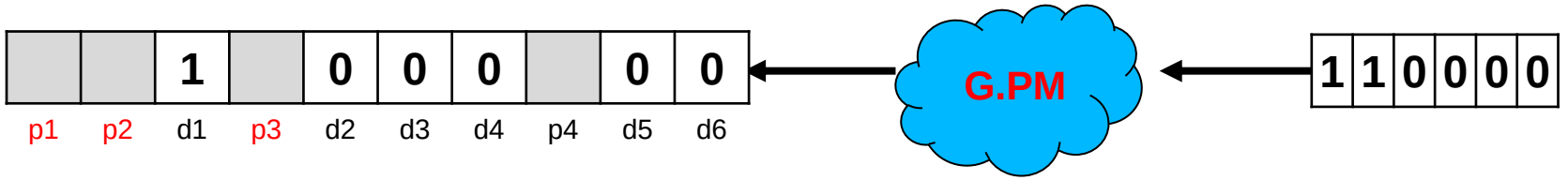
← Menor código de Hamming

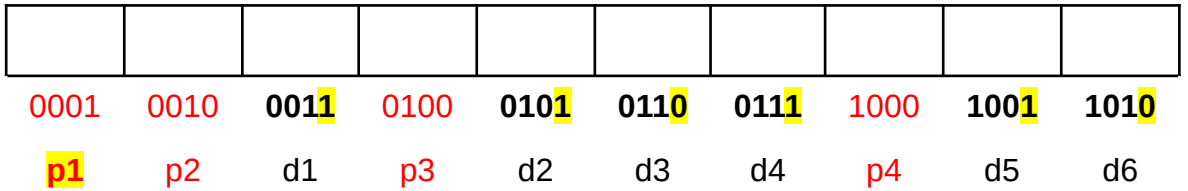
Exemplo 2 - Matriz Geradora

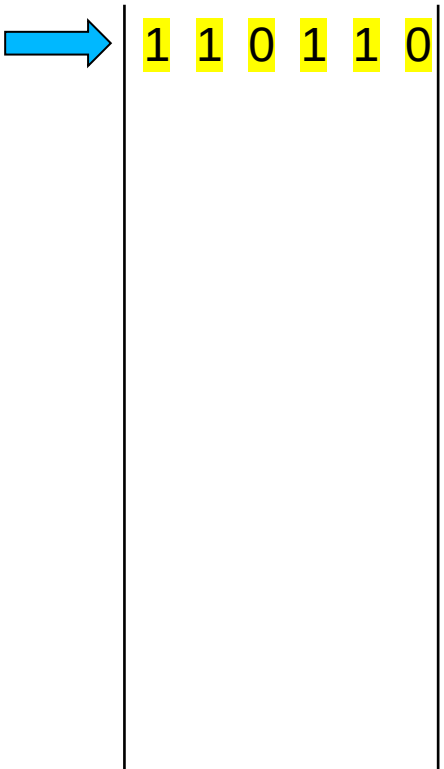


- Como será montada a matriz neste exemplo?

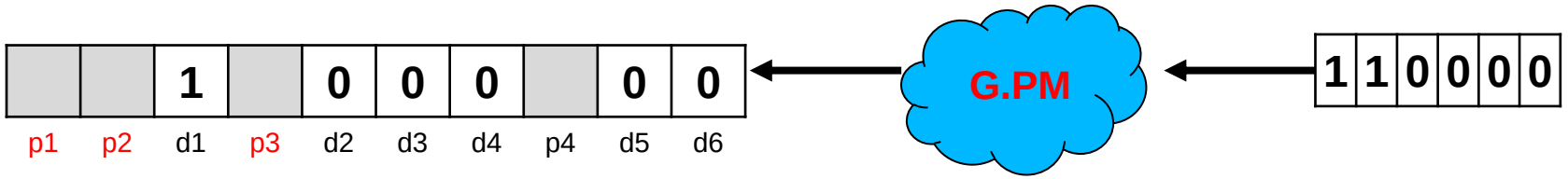
Exemplo 2 - Matriz Geradora

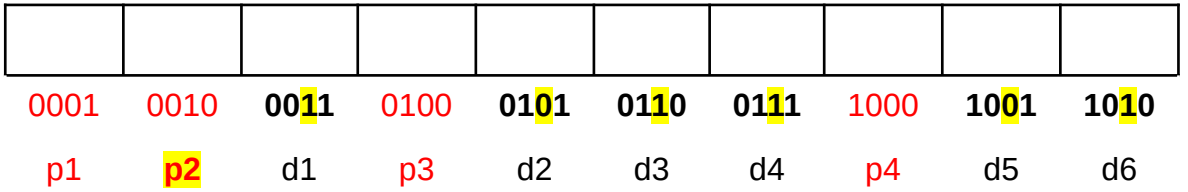


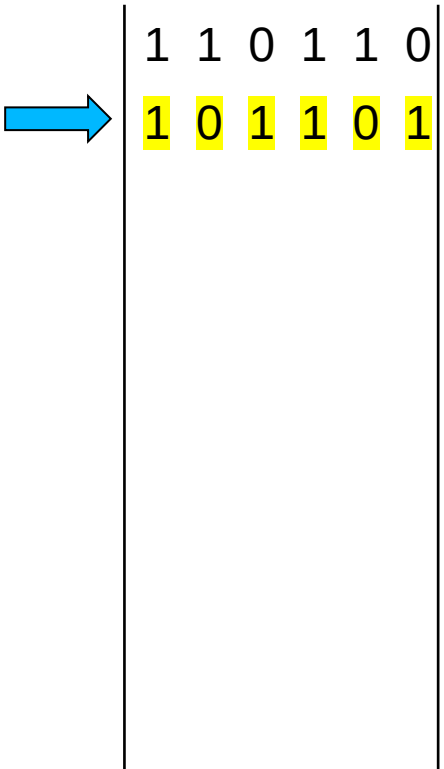




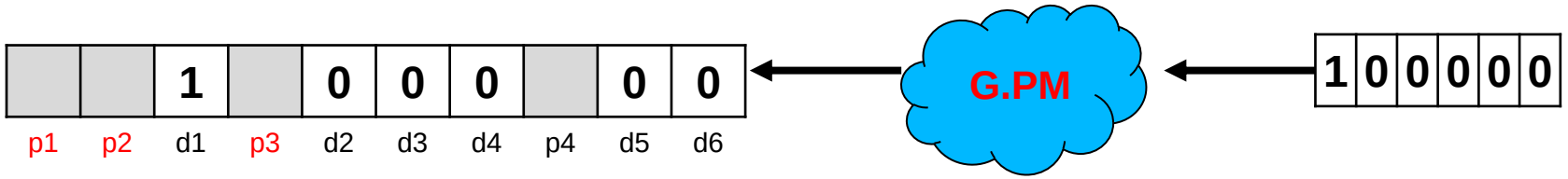
Exemplo 2 - Matriz Geradora

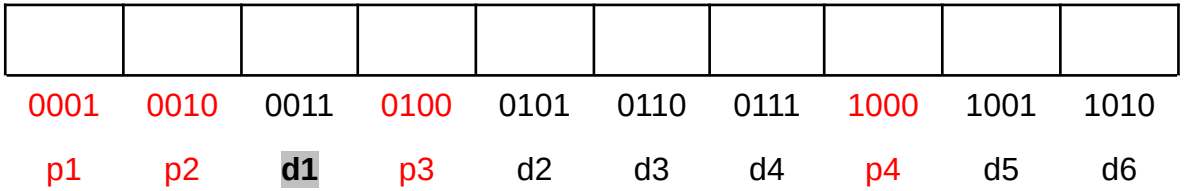


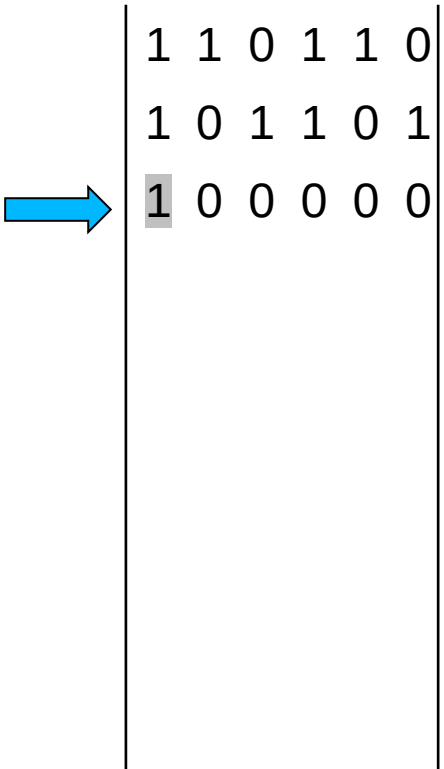




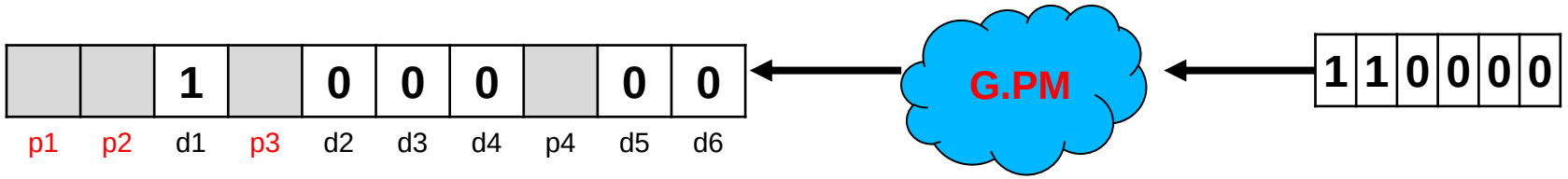
Exemplo 2 - Matriz Geradora

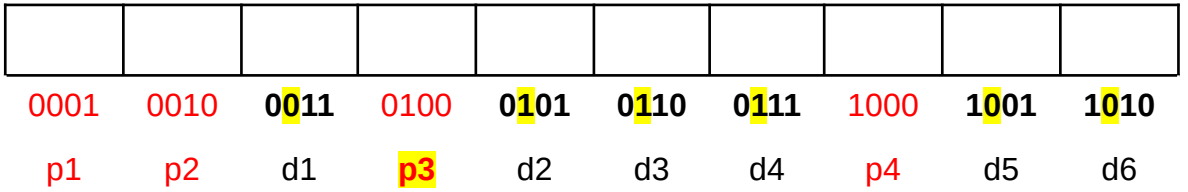


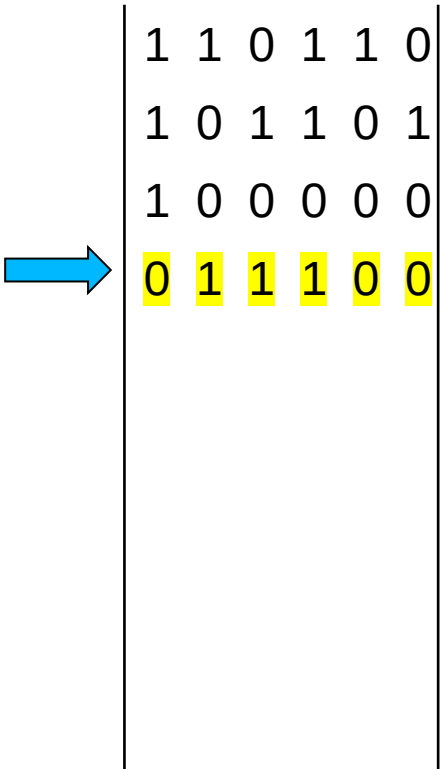




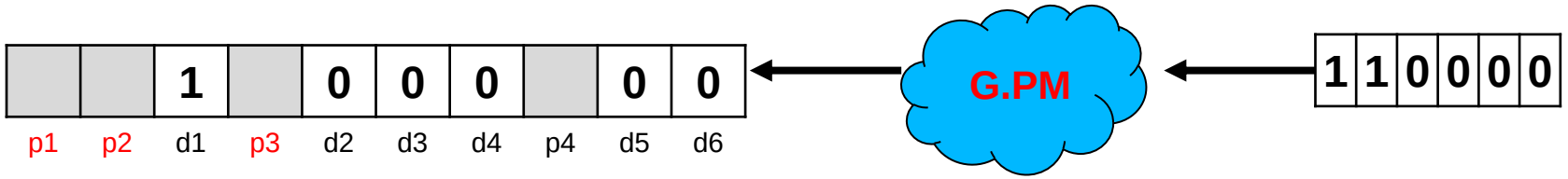
Exemplo 2 - Matriz Geradora

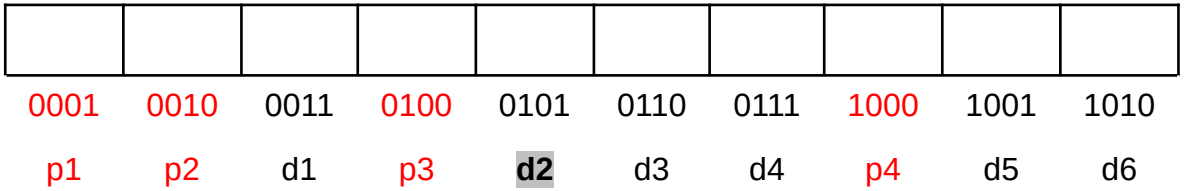


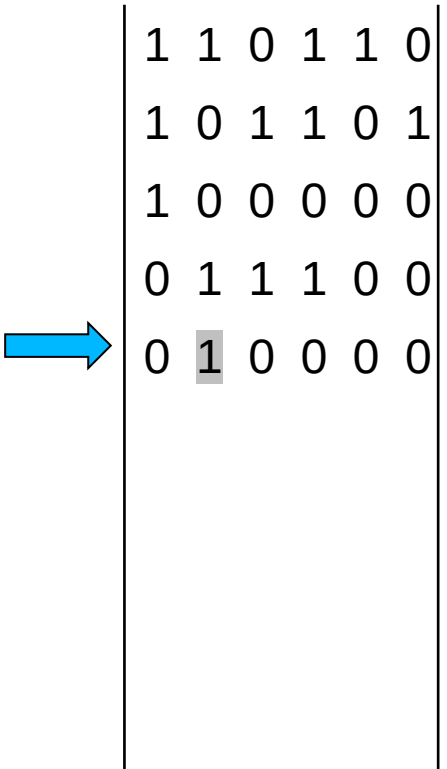




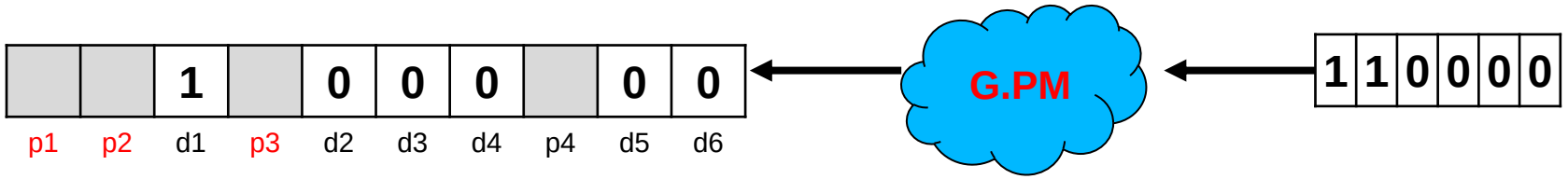
Exemplo 2 - Matriz Geradora

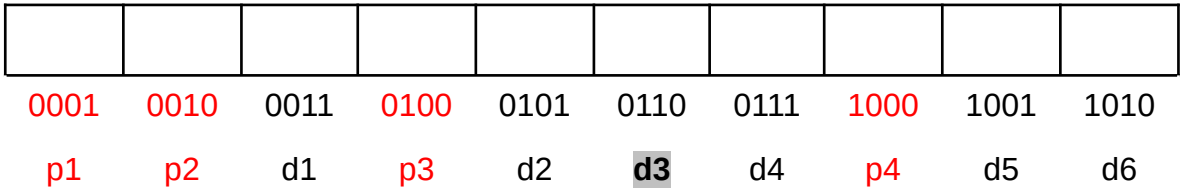


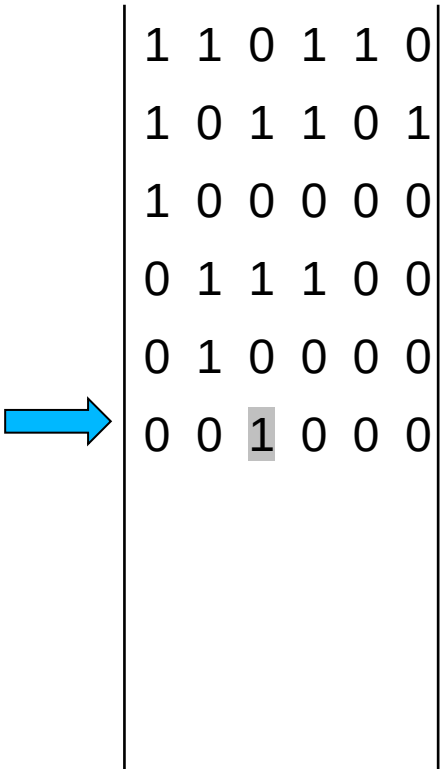




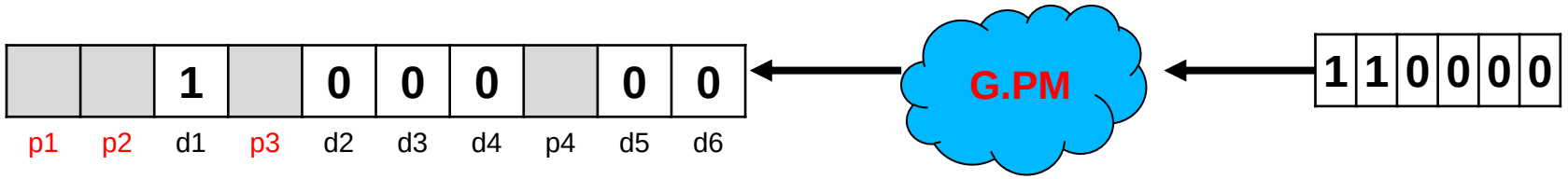
Exemplo 2 - Matriz Geradora

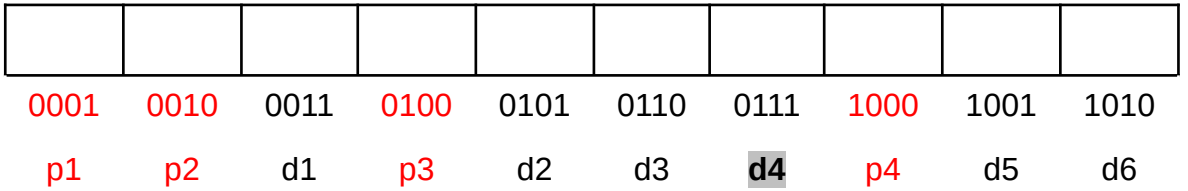


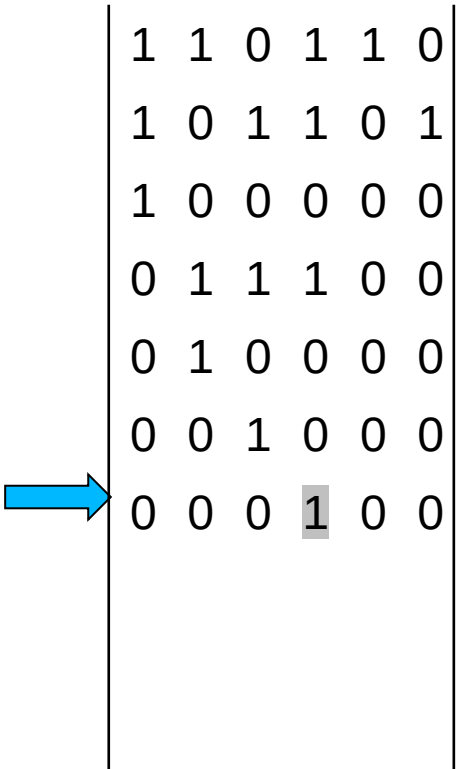




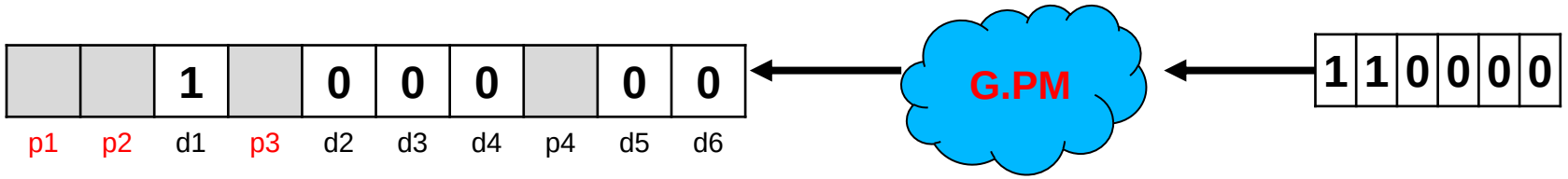
Exemplo 2 - Matriz Geradora

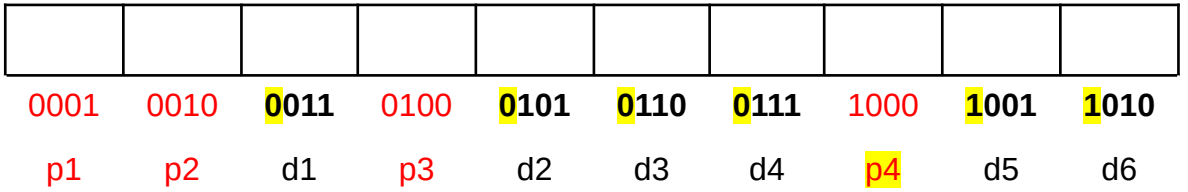


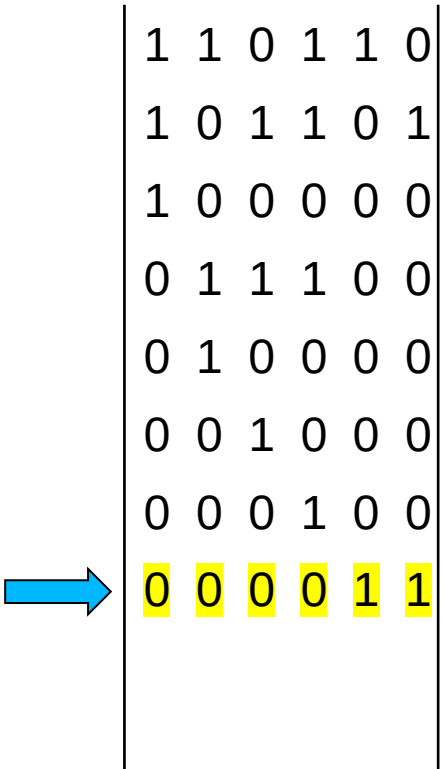




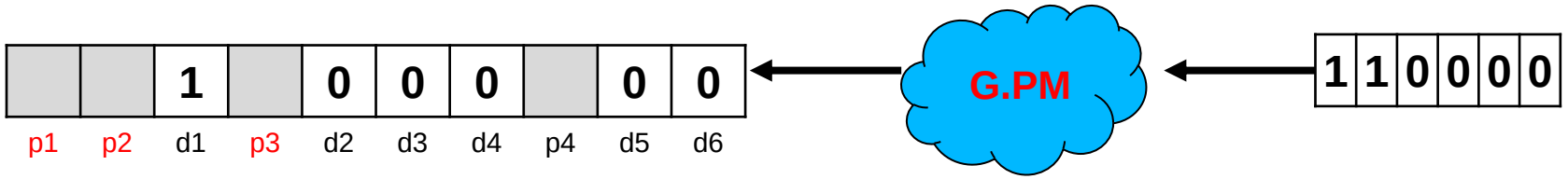
Exemplo 2 - Matriz Geradora

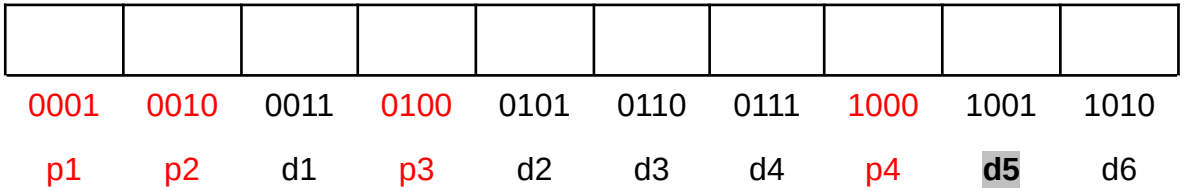






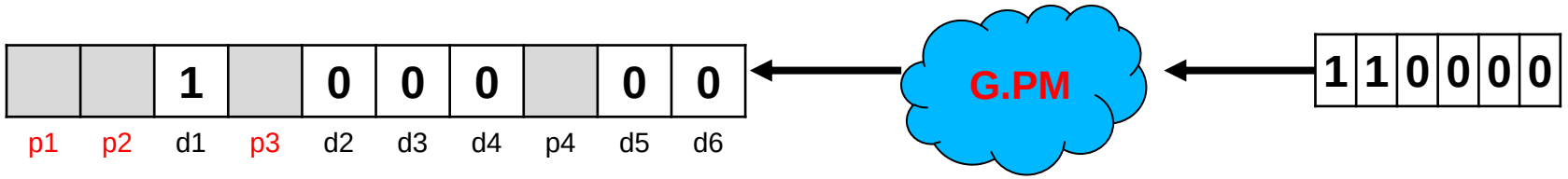
Exemplo 2 - Matriz Geradora

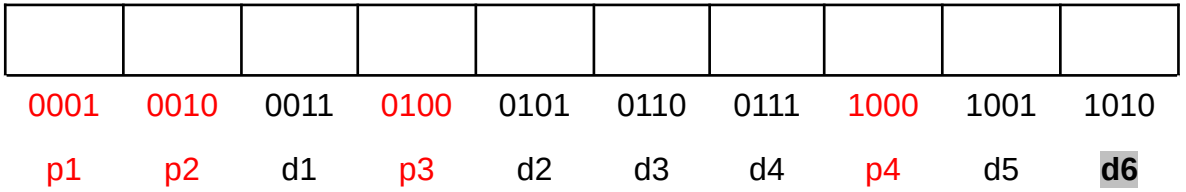




$$
\begin{array}{|cccccc|}
1 & 1 & 0 & 1 & 1 & 0 \\
1 & 0 & 1 & 1 & 0 & 1 \\
1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 1 & 0 \\
\end{array}
$$

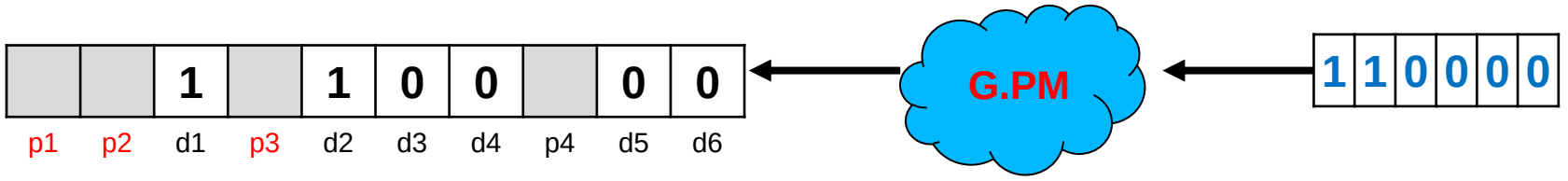
Exemplo 2 - Matriz Geradora





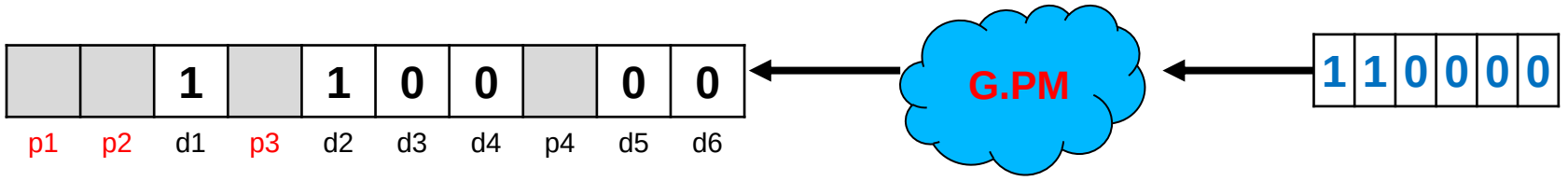
$$\begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Exemplo 2 - Matriz Geradora



$$\begin{vmatrix} 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{vmatrix} \cdot \begin{vmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{vmatrix} = \begin{vmatrix} \\ \\ \\ \\ \\ \\ \\ \\ \\ \end{vmatrix} \begin{vmatrix} \\ \\ \\ \\ \\ \\ \\ \\ \\ \end{vmatrix}$$

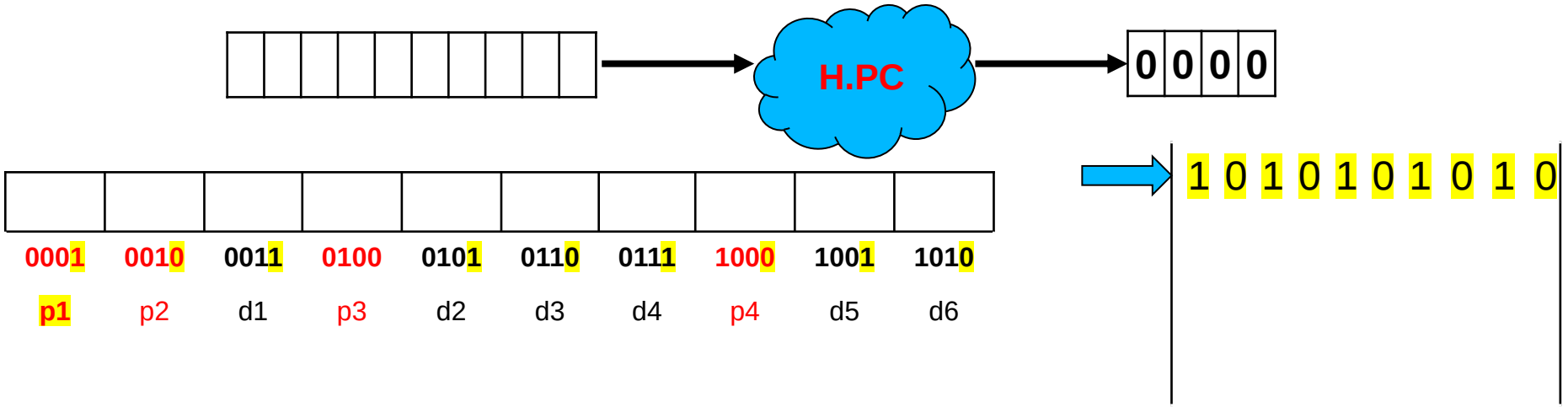
Exemplo 2 - Matriz Geradora



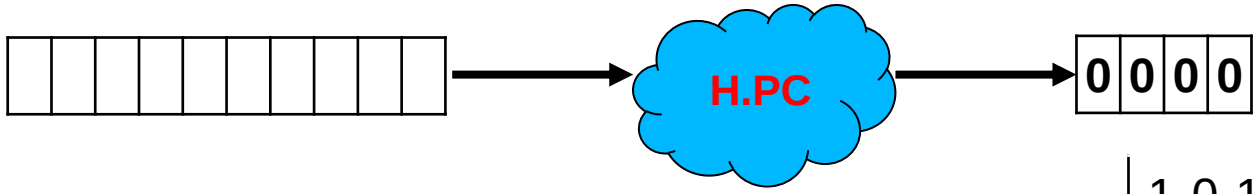
$$\begin{vmatrix} 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{vmatrix} \cdot \begin{vmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{vmatrix} = \begin{vmatrix} 0 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{vmatrix}$$

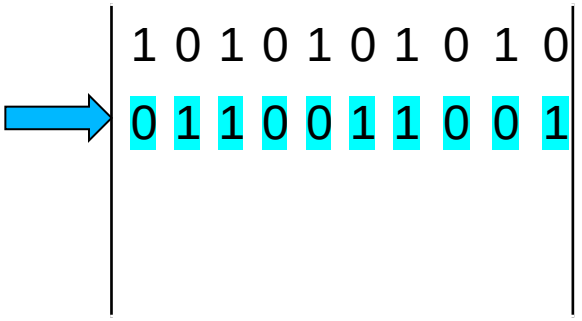
Matriz Verificadora

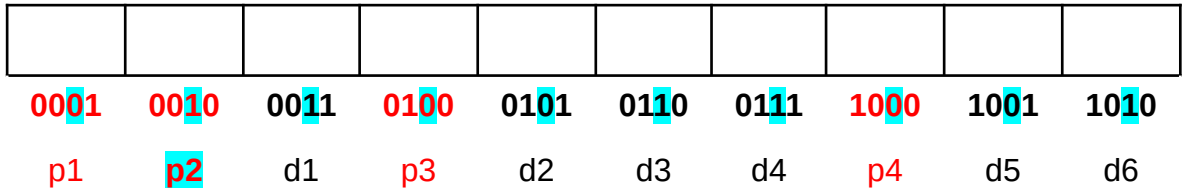
Matriz Verificadora



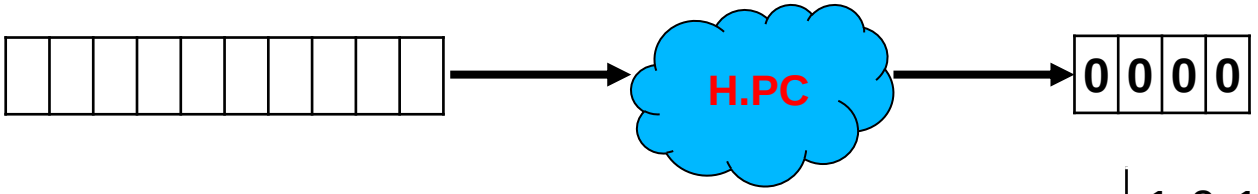
Matriz Verificadora

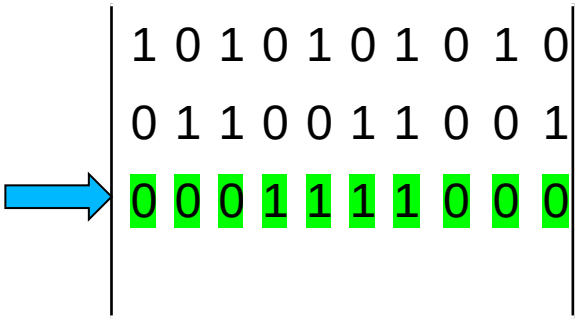


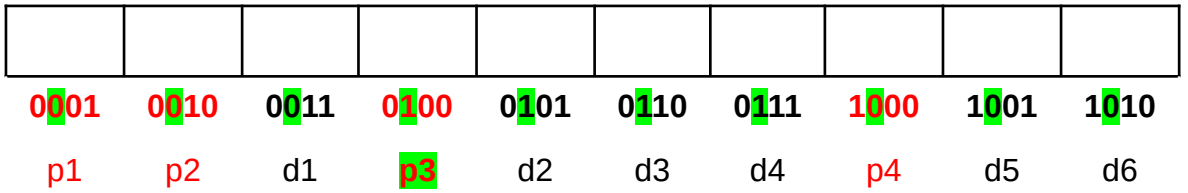




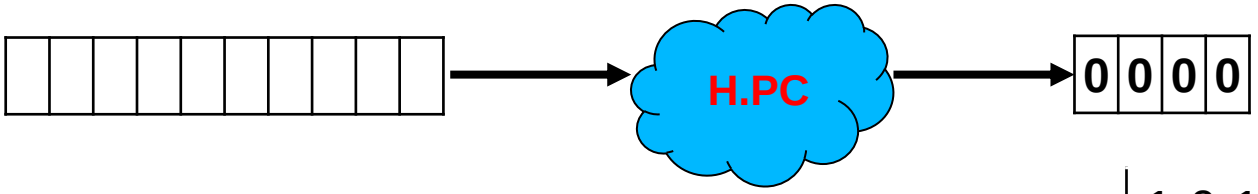
Matriz Verificadora







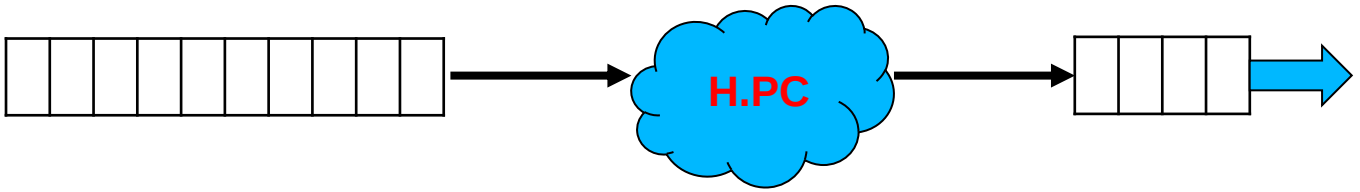
Matriz Verificadora



$$\begin{array}{|cccccccccc} 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{array}$$

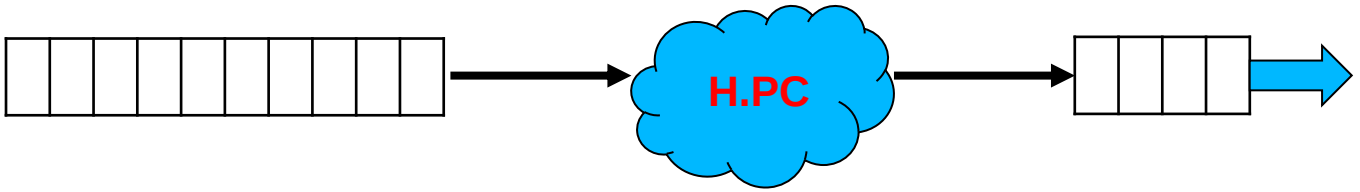
0001	0010	0011	0100	0101	0110	0111	1000	1001	1010
p1	p2	d1	p3	d2	d3	d4	p4	d5	d6

Matriz Verificadora



$$\begin{vmatrix} 1&0&1&0&1&0&1&0&1&0 \\ 0&1&1&0&0&1&1&0&0&1 \\ 0&0&0&1&1&1&1&0&0&0 \\ 0&0&0&0&0&0&0&1&1&1 \end{vmatrix} \cdot \begin{vmatrix} 0 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{vmatrix} = \begin{vmatrix} ? \\ ? \\ ? \\ ? \end{vmatrix} \begin{matrix} \text{1 (p1)} \\ \text{2 (p2)} \\ \text{4 (p3)} \\ \text{8 (p4)} \end{matrix}$$

Matriz Verificadora

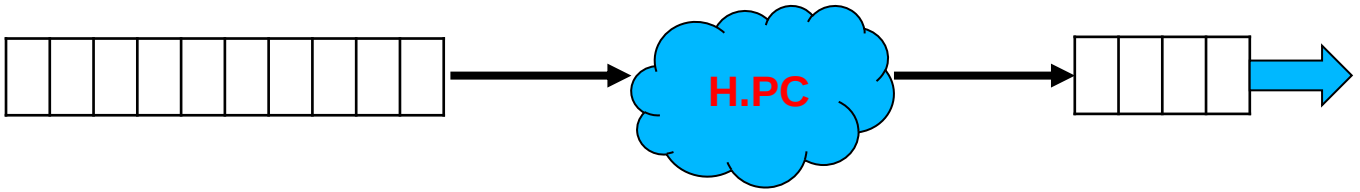


$$\begin{vmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{vmatrix} \cdot \begin{vmatrix} 0 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{vmatrix} = \begin{vmatrix} 0 \\ 0 \\ 0 \\ 0 \end{vmatrix}$$

1 (p1)
2 (p2)
4 (p3)
8 (p4)

- 0000 (se não ocorreu erro)
- Se ocorreu erro: consterá o valor 1 no(s) bit(s) de paridade correspondentes ao bit onde ocorreu o erro

Matriz Verificadora



$$\begin{vmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{vmatrix} \cdot \begin{vmatrix} 0 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{vmatrix} = \begin{vmatrix} 1 \\ 1 \\ 1 \\ 0 \end{vmatrix} \quad \begin{matrix} 1\ (p1) \\ 2\ (p2) \\ 4\ (p3) \\ 8\ (p4) \end{matrix}$$

Referências Bibliograficas

Tanenbaum, A S “**Organização Estruturada de Computadores”** – Prentice Hall do Brasil 5ª edição, 2006; capitulo 2(Tanenbaum): 2.1→ 2.1.1, 2.1.2 (pg 29.); 2.2 → 2.2.1, 2.2.2, 2.2.3 e 2.2.4

Video aula da USP: http://eaulas.usp.br/portal/video.action?idItem=7727