

# Hadamard Codes

DIGITAL CIRCUIT IMPLEMENTATION



universidade  
de aveiro

João Torrinhas nº98435  
Diogo Torrinhas nº98440

# Decoder - Parallel Implementation

- O ponto de partida foram as expressões dadas pelo professor, onde o c11, c12, c13, c14 são as combinações para o m0 e assim sucessivamente.
- Através do mapa de karnaugh e tendo calculado os c's, para cada m, obtemos as seguintes fórmulas que nos ajudaram a obter o m'0, m'1 e m'2.

$$\begin{aligned}M\#mOne &= C3C2(C1 + C0) + C1C0(C3 + C2) \\M\#mZero &= \sim C3\sim C2(\sim C1 + \sim C2) + \sim C1\sim C0(\sim C3 + \sim C2) \\M\#Error &= \sim (M\#One + M\#Zero) \\Valid &= \sim (M\#0Error + M\#1Error + M\#2Error)\end{aligned}$$

$$\begin{aligned}M'0 &= m0One * Valid \\M'1 &= m1One * Valid \\M'2 &= m2One * Valid\end{aligned}$$

|                  |    | $c_{\#1}c_{\#0}$ |    |    |    |
|------------------|----|------------------|----|----|----|
|                  |    | 00               | 01 | 11 | 10 |
| $c_{\#3}c_{\#2}$ | 00 | 0                | 0  | E  | 0  |
|                  | 01 | 0                | E  | 1  | E  |
|                  | 11 | E                | 1  | 1  | 1  |
|                  | 10 | 0                | E  | 1  | E  |

$$\begin{aligned}c_{11} &= y_0 \oplus y_1 \\c_{12} &= y_2 \oplus y_3 \\c_{13} &= y_4 \oplus y_5 \\c_{14} &= y_6 \oplus y_7 \\c_{21} &= y_0 \oplus y_2 \\c_{22} &= y_1 \oplus y_3 \\c_{23} &= y_4 \oplus y_6 \\c_{24} &= y_5 \oplus y_7 \\c_{31} &= y_0 \oplus y_4 \\c_{32} &= y_1 \oplus y_5 \\c_{33} &= y_2 \oplus y_6 \\c_{34} &= y_3 \oplus y_7\end{aligned}$$

# Decoder - Parallel Implementation

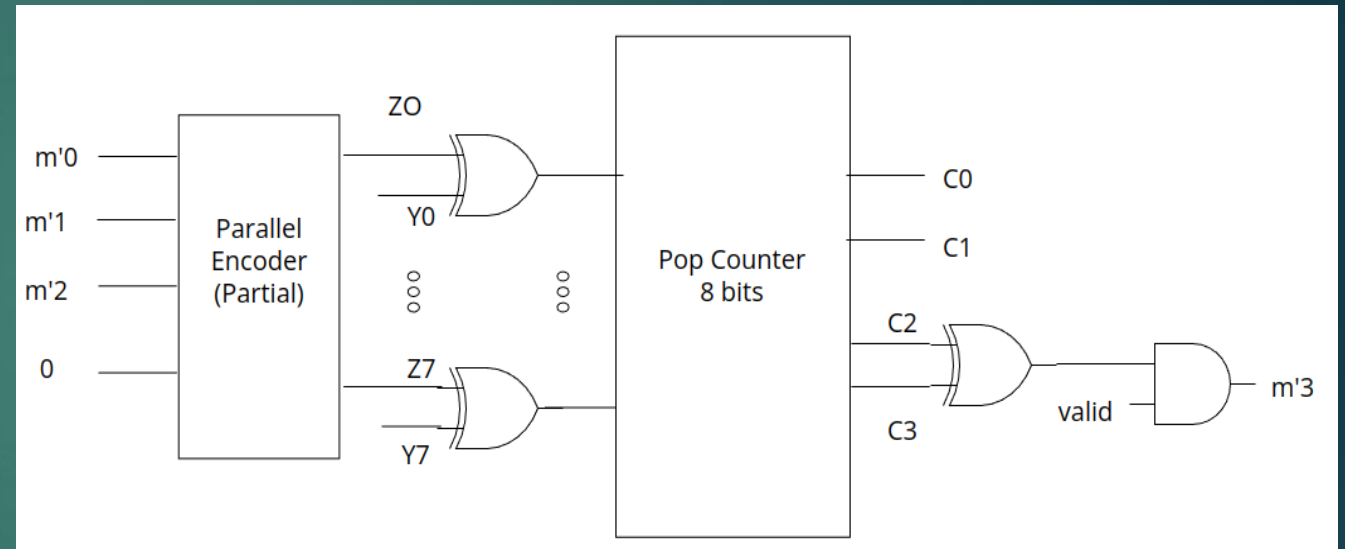
- Para calcular o  $m'3$  foi usado parallel encoder(Partial) e um popcounter de 8 bits.

► Implementation cost:

- 43 XOR
- 34 AND
- 13 OR
- 18 NOT

► Propagation delay:

- 7 XOR + 1 AND + 1 OR



# Encoder – Serial Implementation

- A ROM em cada posição tem 11 bits porque, os primeiros 8 bits são para guardar os valores do k para cada m e os restantes 3 bits são para guardar os resets, sets e clks.

- Por exemplo:

Na posição 01 está guardado o valor 2AF -> 01010101111 -> 01010101 111, ou seja, os primeiros 8 bits correspondem ao valor do k para m0 e os outros 3 bits são o valor do nRst, nSet0 e nEnClk.

- Na posição 01,02,03 e 04 são guardados os valores correspondentes ao m1, m2, m3 e m4 e os restantes é para o start e reset.

ROM contents

| Add[2..0] | CLines[10..0] |
|-----------|---------------|
| 00        | 5             |
| 01        | 2AF           |
| 02        | 19F           |
| 03        | 7F            |
| 04        | 7FE           |
| 05        | 6             |
| 06        | 1             |
| 07        | 7             |

# Encoder – Serial Implementation

- Implementation cost:

- 20 FlipFlops
- 10 AND
- 10 XOR
- 4 NAND
- 2 NOR

