universidade de aveiro

Departamento de Eletrónica,
Telecomunicações e
Informática

Hadamard codes$[8,4,4]_2$

Digital Circuit implementation

Assignment 1 - Nov. 2022

Arquiteturas de Alto Desempenho
Prof. António Rui Borges

André Clérigo 98485
Pedro Rocha 98256
Turma 1 - Grupo 7

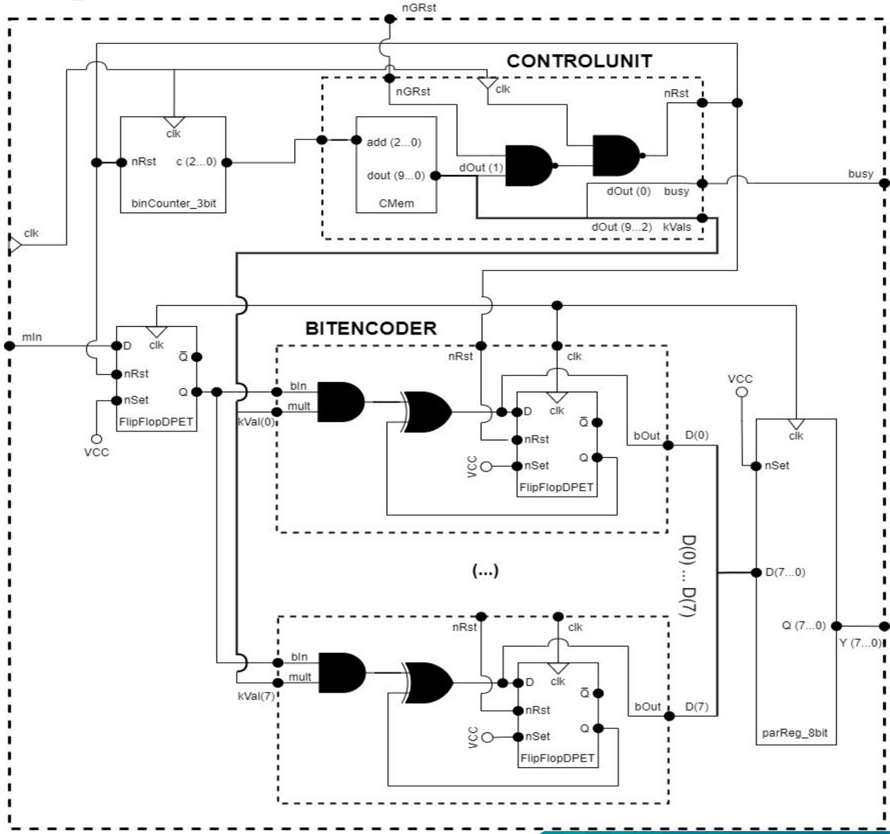
Encoder's bit serial implementation

While the busy flag has the value 0, the Y value does not have a meaning.
It takes 5 clock cycles to process the 4 bits and show the encode the message.

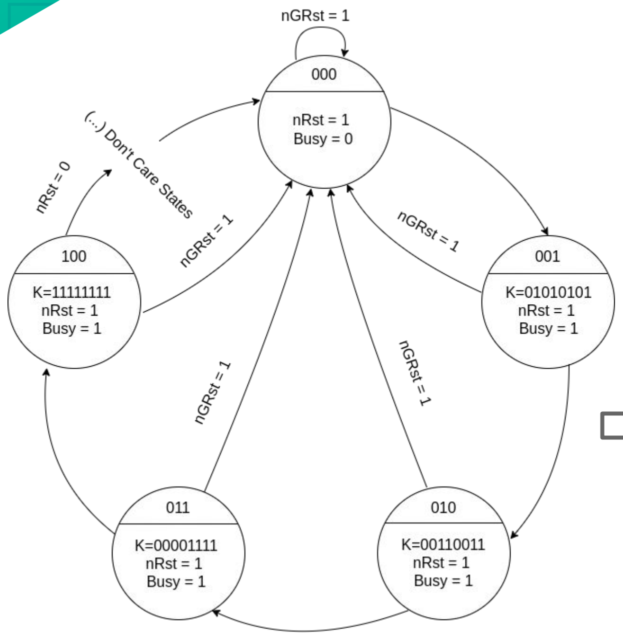
Implementation cost:

BINCOUNTER_3BIT
1 AND + 2 XOR + 3 D-type Flip Flop
BITENCODER
1 AND + 1 XOR + 1 D-type FlipFlop
Total cost of 8x: 8 AND + 8 XOR + 8 D-type FlipFlop
PARALLEL REGISTER_8BIT
1 AND + 1 XOR + 1 D-type FlipFlop

Total ENCODER Cost: 10 AND + 11 XOR + 13 D-type FlipFlop



Encoder’s bit serial implementation



Encoder straightforward implementation

$x_0 = m_3$
$x_1 = m_0 \oplus m_3$
$x_2 = m_1 \oplus m_3$
$x_3 = m_0 \oplus m_1 \oplus m_3$
$x_4 = m_2 \oplus m_3$
$x_5 = m_0 \oplus m_2 \oplus m_3$
$x_6 = m_1 \oplus m_2 \oplus m_3$
$x_7 = m_0 \oplus m_1 \oplus m_2 \oplus m_3$

Regular expressions

$x_0 = k_{00}m_0 \oplus k_{01}m_1 \oplus k_{02}m_2 \oplus k_{03}m_3$
…
$x_7 = k_{70}m_0 \oplus k_{71}m_1 \oplus k_{72}m_2 \oplus k_{73}m_3$

Algorithm

```
for (int i = 0; i < 4; i++) {
    for (int j = 0; j < 8; j++) {
        y_j = y_j XOR (k_ji AND m_i);
    }
}
```

CMem content

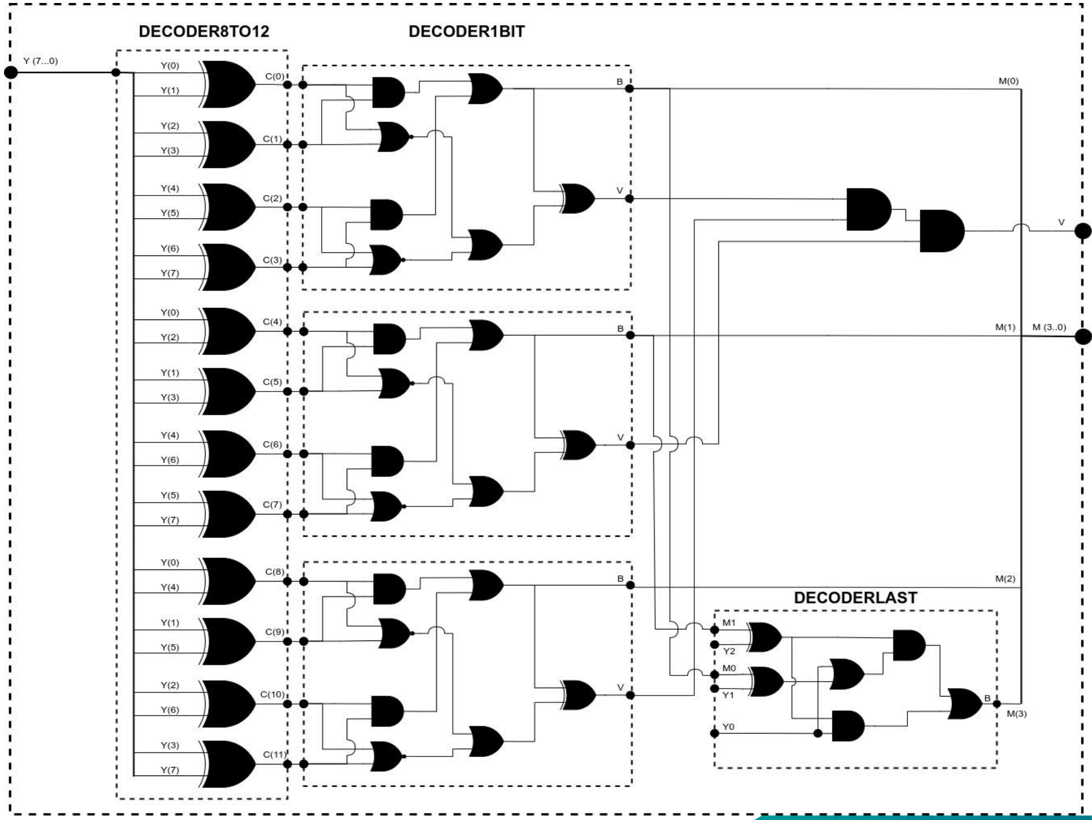
```
("0000000010",     -- Initial State   nRst = 1 busy = 0
 "0101010111",     -- K0: 01010101    nRst = 1 busy = 1
 "0011001111",     -- K1: 00110011    nRst = 1 busy = 1
 "0000111111",     -- K2: 00001111    nRst = 1 busy = 1
 "1111111111",     -- K3: 11111111    nRst = 1 busy = 1
 "0000000001",     -- Reset           nRst = 0 busy = 1
 "0000000011",     -- Dont' care
 "0000000011");    -- Don't care
```

Decoder's parallel implementation

Implementation cost:

DECODER8TO12:
Cost: 12 XOR
Worst case propagation delay: 1 XOR
DECODER1BIT:
Cost: 2 AND, 2 NOR, 2 OR, 1 XOR
Worst case propagation delay: 1 NOR + 1 OR + 1 XOR
Total Cost of 3x: 6 AND, 6 NOR, 6 OR, 3 XOR
DECODERLAST:
Cost: 2 XOR, 2 AND, 2 OR
Worst case propagation delay: 1 XOR + 1 OR + 1 AND + 1 OR

Total DECODER Cost: 17 XOR, 10 AND, 6 NOR, 8 OR
Parallel DECODER worst case propagation delay:
(1 XOR) + (1 NOR + 1 OR + 1 XOR) + (1 XOR + 1 OR + 1 AND + 1 OR)
= 3 XOR + 1 NOR + 3 OR + 1 AND



Decoder's parallel implementation

Application of the property of local decodability for m'_0 m'_1 m'_2

$$\left.\begin{array}{l} c11 = y_0 \oplus y_1 \\ c12 = y_2 \oplus y_3 \\ c13 = y_4 \oplus y_5 \\ c14 = y_6 \oplus y_7 \end{array}\right\} m'_0$$

$$\left.\begin{array}{l} c21 = y_0 \oplus y_2 \\ c22 = y_1 \oplus y_3 \\ c23 = y_4 \oplus y_6 \\ c24 = y_5 \oplus y_7 \end{array}\right\} m'_1$$

$$\left.\begin{array}{l} c31 = y_0 \oplus y_4 \\ c32 = y_1 \oplus y_5 \\ c33 = y_2 \oplus y_6 \\ c34 = y_3 \oplus y_7 \end{array}\right\} m'_2$$

T - Value of bit if 1

T		C1C0			
		00	01	11	10
C3 C2	00	0	0	E	0
	01	0	E	1	E
	11	E	1	1	1
	10	0	E	1	E

F - Value of bit if 0

F		C1C0			
		00	01	11	10
C3 C2	00	1	1	E	1
	01	1	E	0	E
	11	E	0	0	0
	10	1	E	0	E

E - Error 2 bits equal

E		C1C0			
		00	01	11	10
C3 C2	00	0	0	E	0
	01	0	E	0	E
	11	E	0	0	0
	10	0	E	0	E

Extracting SOP from Karnaugh Maps

T = (C3 ∧ C2) ∨ (C1 ∧ C0)
F = (¬C3 ∧ ¬C2) ∨ (¬C1 ∧ ¬C0) = ¬(C3 ∨ C2) ∨ ¬(C1 ∨ C0)

Green - If T and F are both 1
Red - If T and F are both 0

E = T ⊕ F

Decoder's parallel implementation

Application of the property of local decodability for m'$_3$

Abstracting $a = x_0$,$b = x_1 \oplus m'_0$,$c = x_2 \oplus m'_1$

m'$_3$		BA			
		00	01	11	10
C	0	0	0	1	0
	1	0	1	1	1

Extracting SOP from Karnaugh Map

$m'_3 = (A \wedge B) \vee (A \wedge C) \vee (B \wedge C)$

With the Karnaugh Map expression and knowing that $x_\#$ on the encoder is $y_\#$ on the decoder

$m'_3 = (y_2 \oplus m'_1) \wedge (y_0 \vee (y_1 \oplus m'_0)) \vee (y_0 \wedge (y_1 \oplus m'_0))$

Having $m'_0\, m'_1\, m'_2$ only 3 equations are needed local decodability

Encoder straightforward implementation

$x_0 = m'_3$
$x_1 = m'_0 \oplus m'_3$
$x_2 = m'_1 \oplus m'_3$

In order of m_3

$m'_3 = x_0$
$m'_3 = x_1 \oplus m'_0$
$m'_3 = x_2 \oplus m'_1$