



FPGA Básico

Parte 2

(Aplicação Prática)

Manoel Eusebio de Lima

Victor Medeiros

Abel Guilhermino Silva-Filho

(vwcm@cin.ufpe.br)



+ Agenda

- Problema: Multiplicação Matriz-Vetor
- Solução em alto nível (Linguagem C)
- Solução em FPGA
- Como aumentar o desempenho em FPGA?

+ Multiplicação Matriz-Vetor

1	1	1	1	1	1	1	...	1	1
2	2	2	2	2	2	2	...	2	2
3	3	3	3	3	3	3	...	3	3
4	4	4	4	4	4	4	...	4	4
5	5	5	5	5	5	5	...	5	5
6	6	6	6	6	6	6	...	6	6
7	7	7	7	7	7	7	...	7	7
8	8	8	8	8	8	8	...	8	8
9	9	9	9	9	9	9	...	9	9
10	10	10	10	10	10	10	...	10	10

matriz 10 X 100

*

1
1
1
1
1
1
1
...
1
1

vetor 100 X 1

=

100	0x64
200	0xC8
300	0x12C
400	0x190
500	0x1F4
600	0x258
700	0x2BC
800	0x320
900	0x384
1000	0x3E8

vetor resultante
10 X 1

+ Multiplicação Matriz-Vetor



1	1	1	1	1	1	1	...	1	1
2	2	2	2	2	2	2	...	2	2
3	3	3	3	3	3	3	...	3	3
4	4	4	4	4	4	4	...	4	4
5	5	5	5	5	5	5	...	5	5
6	6	6	6	6	6	6	...	6	6
7	7	7	7	7	7	7	...	7	7
8	8	8	8	8	8	8	...	8	8
9	9	9	9	9	9	9	...	9	9
10	10	10	10	10	10	10	...	10	10

matriz 10 X 100

*

1
1
1
1
1
1
1
...
1
1

vetor 100 X 1

=

100	0x64
200	0xC8
300	0x12C
400	0x190
500	0x1F4
600	0x258
700	0x2BC
800	0x320
900	0x384
1000	0x3E8

vetor resultante
10 X 1

+ Multiplicação Matriz-Vetor



1	1	1	1	1	1	1	...	1	1
2	2	2	2	2	2	2	...	2	2
3	3	3	3	3	3	3	...	3	3
4	4	4	4	4	4	4	...	4	4
5	5	5	5	5	5	5	...	5	5
6	6	6	6	6	6	6	...	6	6
7	7	7	7	7	7	7	...	7	7
8	8	8	8	8	8	8	...	8	8
9	9	9	9	9	9	9	...	9	9
10	10	10	10	10	10	10	...	10	10

matriz 10 X 100

*

1
1
1
1
1
1
1
...
1
1

vetor 100 X 1

=

100	0x64
200	0xC8
300	0x12C
400	0x190
500	0x1F4
600	0x258
700	0x2BC
800	0x320
900	0x384
1000	0x3E8

vetor resultante
10 X 1

+ Multiplicação Matriz-Vetor



1	1	1	1	1	1	1	...	1	1
2	2	2	2	2	2	2	...	2	2
3	3	3	3	3	3	3	...	3	3
4	4	4	4	4	4	4	...	4	4
5	5	5	5	5	5	5	...	5	5
6	6	6	6	6	6	6	...	6	6
7	7	7	7	7	7	7	...	7	7
8	8	8	8	8	8	8	...	8	8
9	9	9	9	9	9	9	...	9	9
10	10	10	10	10	10	10	...	10	10

matriz 10 X 100



1
1
1
1
1
1
1
...
1
1

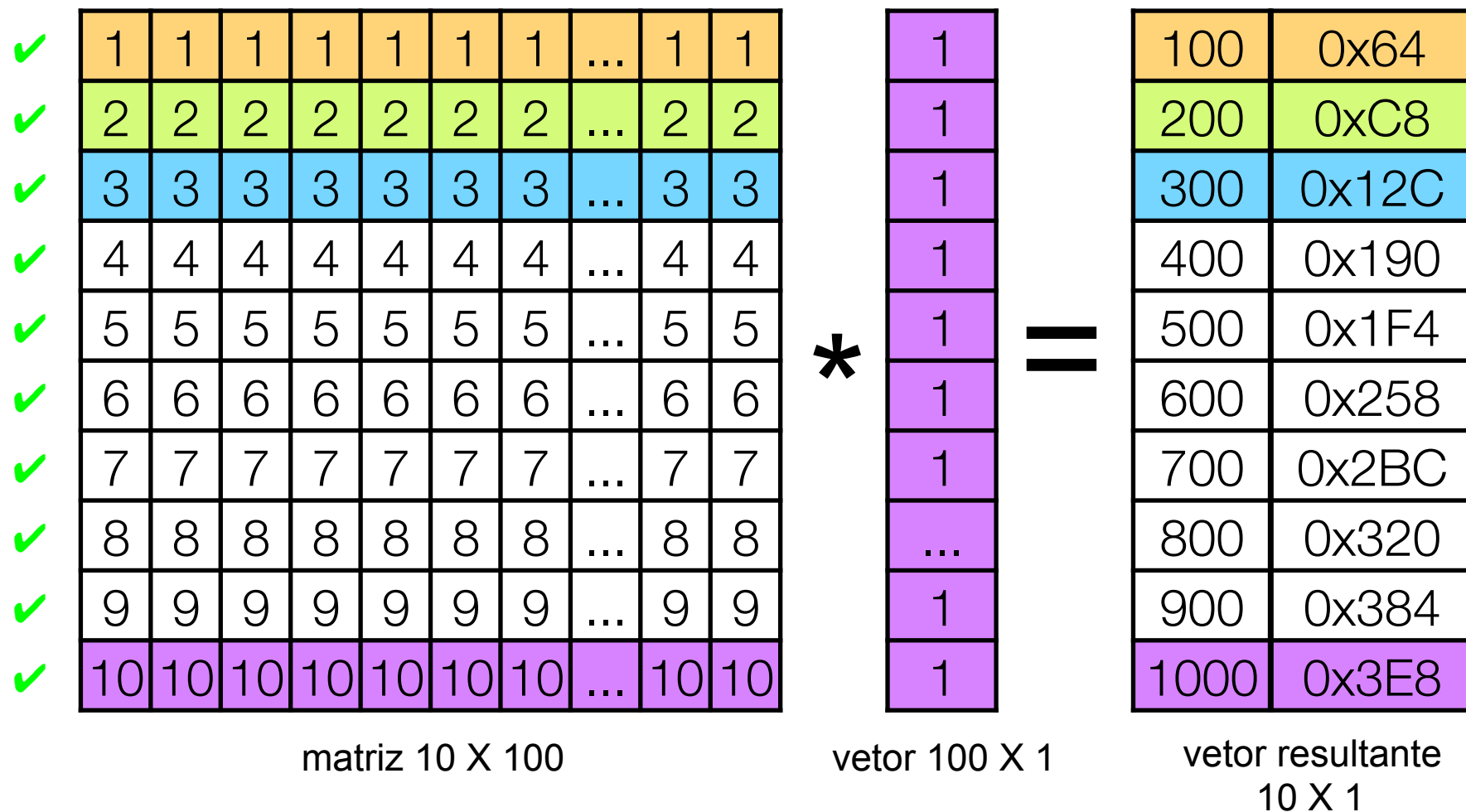
vetor 100 X 1



100	0x64
200	0xC8
300	0x12C
400	0x190
500	0x1F4
600	0x258
700	0x2BC
800	0x320
900	0x384
1000	0x3E8

vetor resultante
10 X 1

+ Multiplicação Matriz-Vetor



The diagram illustrates the multiplication of a 10x10 matrix by a 10x1 vector. The matrix is a 10x10 grid where each row contains the same value from 1 to 10. The vector is a 10x1 column of ones. The resulting vector is a 10x1 column where each element is the sum of its row in the matrix (100 to 1000) and its corresponding hex value (0x64 to 0x3E8).

✓	1	1	1	1	1	1	1	...	1	1
✓	2	2	2	2	2	2	2	...	2	2
✓	3	3	3	3	3	3	3	...	3	3
✓	4	4	4	4	4	4	4	...	4	4
✓	5	5	5	5	5	5	5	...	5	5
✓	6	6	6	6	6	6	6	...	6	6
✓	7	7	7	7	7	7	7	...	7	7
✓	8	8	8	8	8	8	8	...	8	8
✓	9	9	9	9	9	9	9	...	9	9
✓	10	10	10	10	10	10	10	...	10	10

matriz 10 X 100

$*$

vetor 100 X 1

$=$

100	0x64
200	0xC8
300	0x12C
400	0x190
500	0x1F4
600	0x258
700	0x2BC
800	0x320
900	0x384
1000	0x3E8

vetor resultante 10 X 1



Implementação em Alto Nível Linguagem C



```
void mult_mat_vec(int matrix[][], int vector[], int
    result_vector[], int numRows, int numCols) {

    int i, j, sum = 0;

    for(i = 0; i < numRows; i++) {
        for(j = 0; j < numCols; j++) {
            sum += matrix[i][j] * vector[j];
        }
        result_vector[i] = sum;
        sum = 0;
    }
}
```

+ Implementação em Alto Nível Linguagem C

```
void mult_mat_vec(int matrix[][], int vector[], int
    result_vector[], int numRows, int numCols) {

    int i, j, sum = 0;

    for(i = 0; i < numRows; i++)
        for(j = 0; j < numCols; j++)
            sum += matrix[i][j] * vector[j];
    result_vector[i] = sum;
}

}
```

parâmetros:

- matriz
- vetor
- vetor resultante
- número de linhas da matriz
- número de colunas da matriz que corresponde ao tamanho do vetor



Implementação em Linguagem C

```
void mult_mat_vec(int matrix[  
    result_vector[], int
```

```
int i, j, sum = 0;
```

```
for(i = 0; i < numRows; i++) {  
    for(j = 0; j < numCols; j++) {  
        sum += matrix[i][j] * vector[j];  
    }  
    result_vector[i] = sum;  
    sum = 0;
```

```
}
```

```
}
```

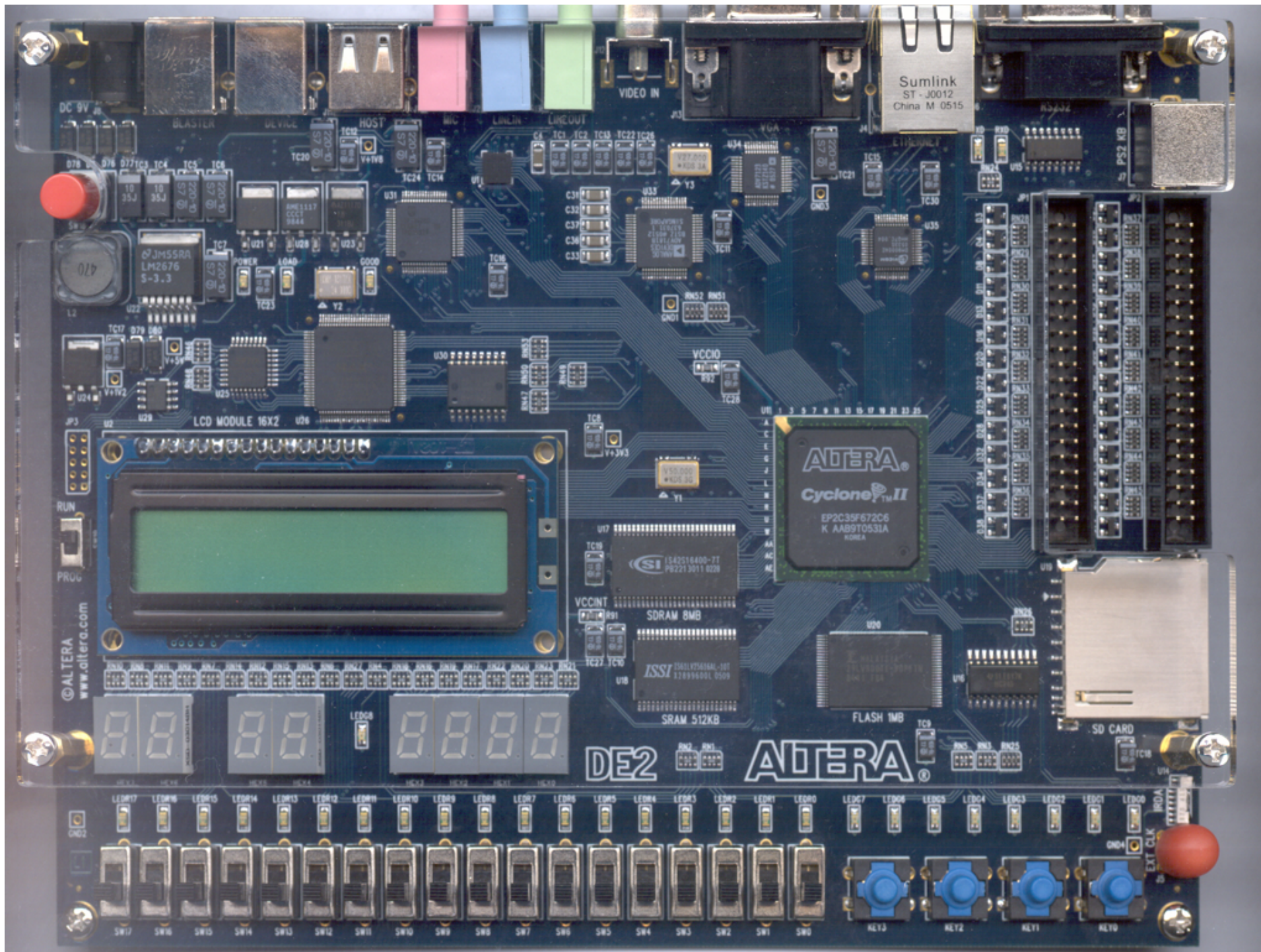
laço que controla a operação de multiplicação da matriz pelo vetor. A cada iteração um elemento da matriz é multiplicado por um elemento do vetor e a soma é acumulada. Ao fim de cada linha da matriz o valor acumulado é armazenado no vetor resultante.

+ Implementação em FPGA

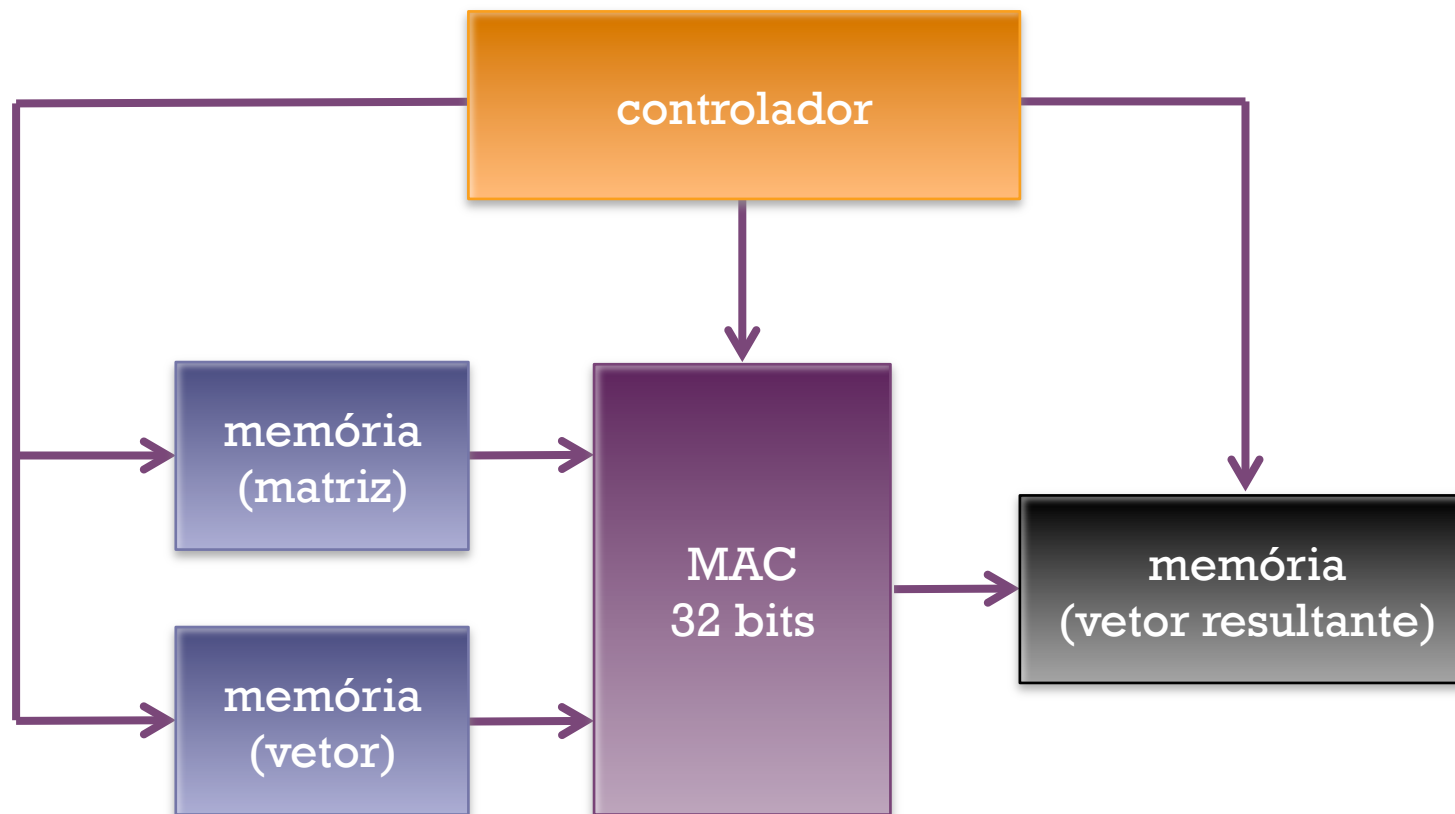
Linguagem VHDL

- Como descreveríamos a operação em VHDL?
- Qual a arquitetura do multiplicador de matrizes por vetores?
- Que módulos devem ser definidos?
- Como estes módulos interagem?

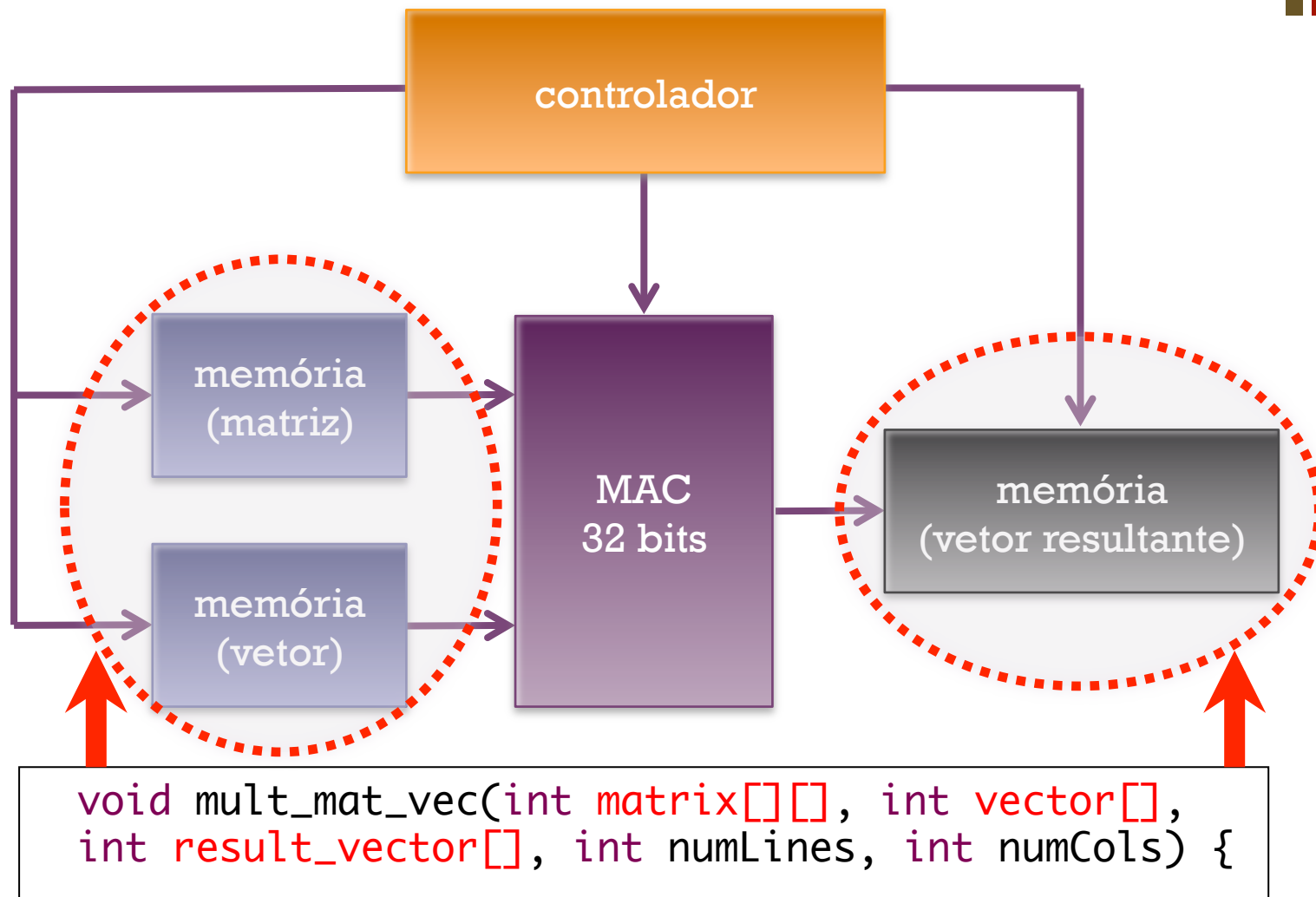




+ Arquitetura da solução proposta



+ Arquitetura da solução proposta



+ Memórias



- As memórias utilizadas no exemplo são internas ao FPGA.
- A ferramenta **MegaWizard** gera os arquivos necessários para que instancie estas memórias no projeto de acordo com nossas necessidades.
- Outros fabricantes como a Xilinx fornecem ferramentas semelhantes (**CoreGenerator**).
- A seguir iremos ver os procedimentos para geração dessas memórias.

+ Criação dos arquivos de conteúdo .MIF

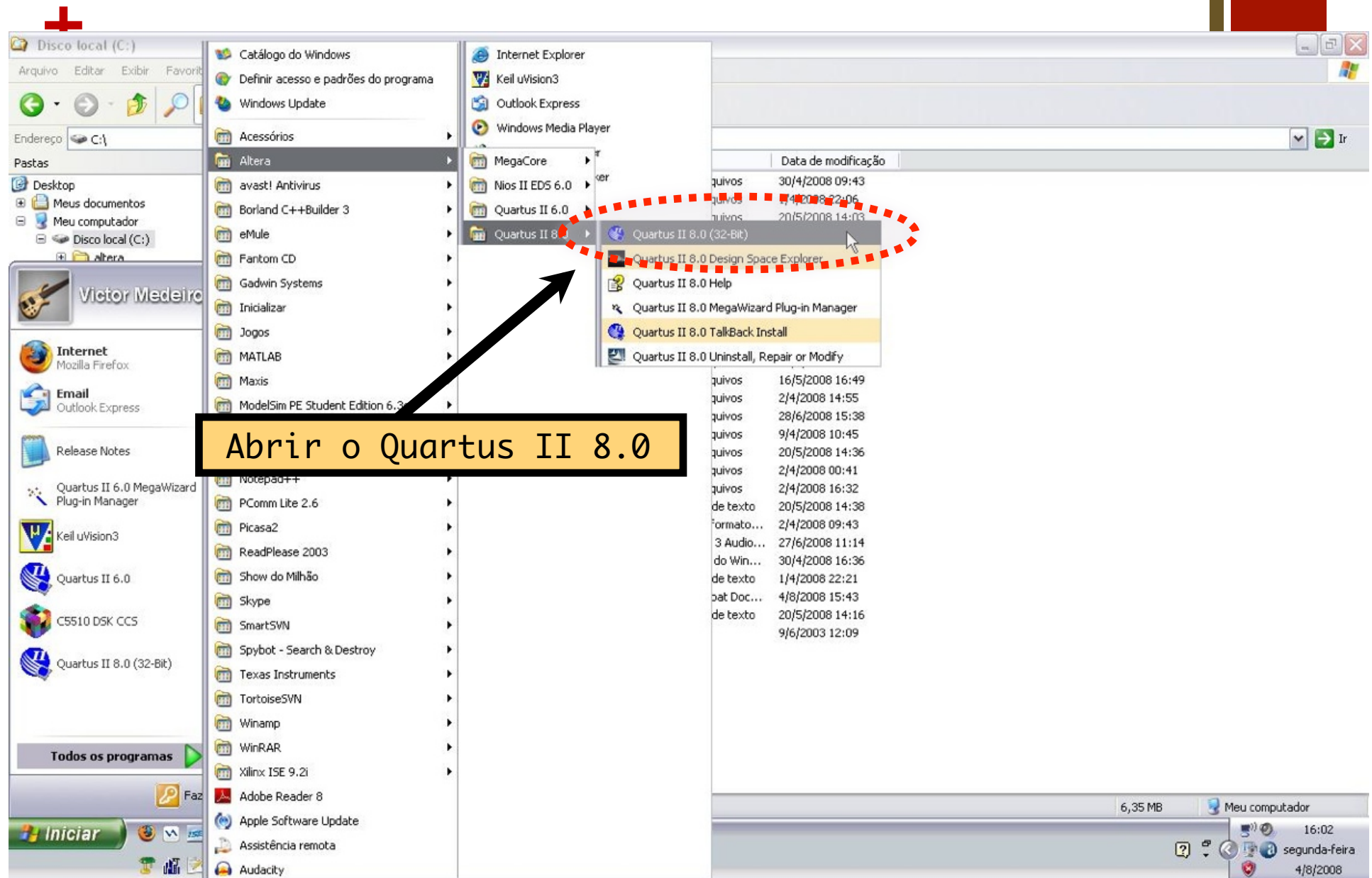
- Os arquivos MIF são capazes de definir o conteúdo inicial das memórias geradas. Iremos definir o conteúdo das memórias de acordo com o exemplo do início da apresentação.

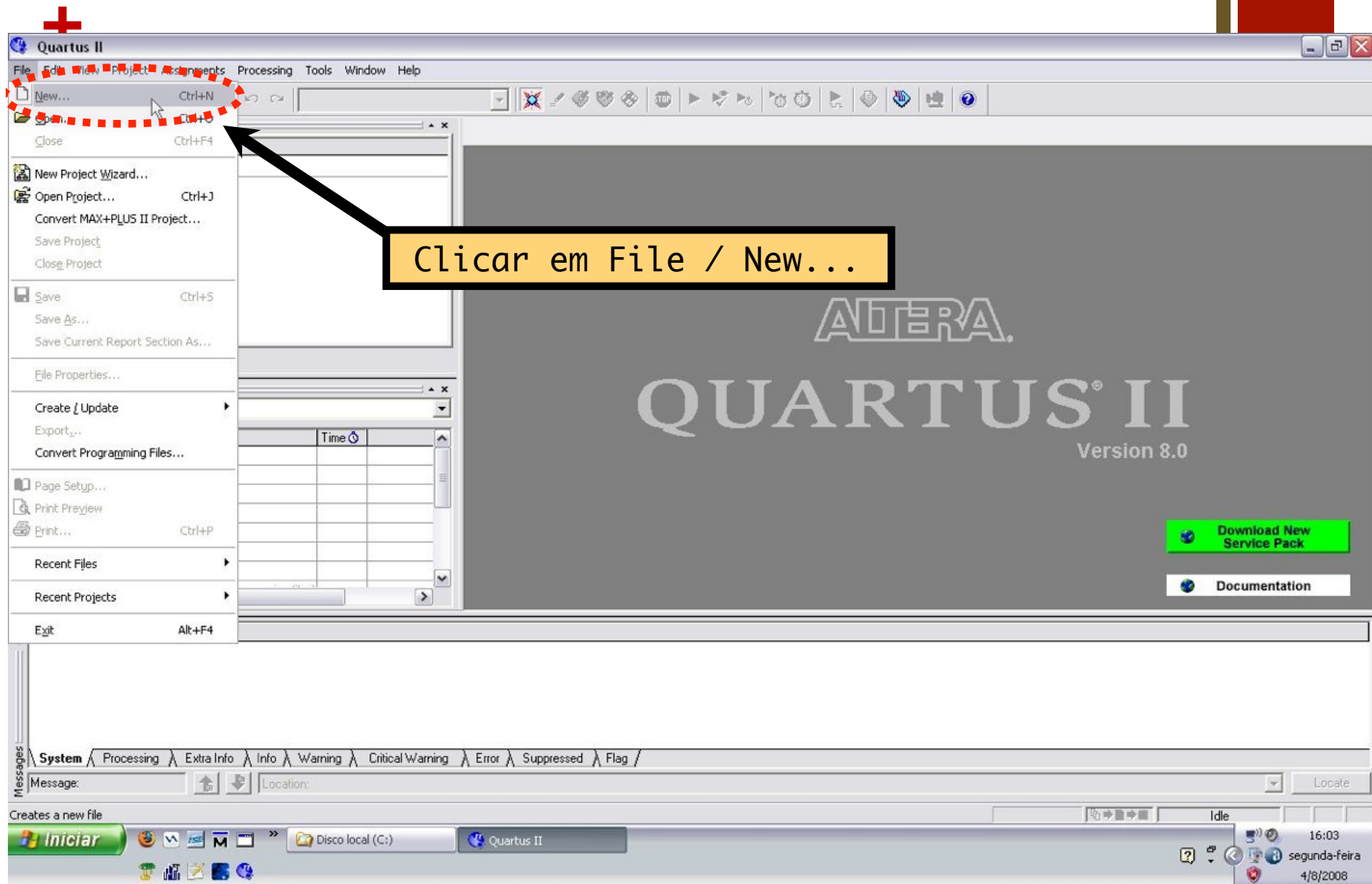
1	1	1	1	1	1	1	...	1	1
2	2	2	2	2	2	2	...	2	2
3	3	3	3	3	3	3	...	3	3
4	4	4	4	4	4	4	...	4	4
5	5	5	5	5	5	5	...	5	5
6	6	6	6	6	6	6	...	6	6
7	7	7	7	7	7	7	...	7	7
8	8	8	8	8	8	8	...	8	8
9	9	9	9	9	9	9	...	9	9
10	10	10	10	10	10	10	...	10	10

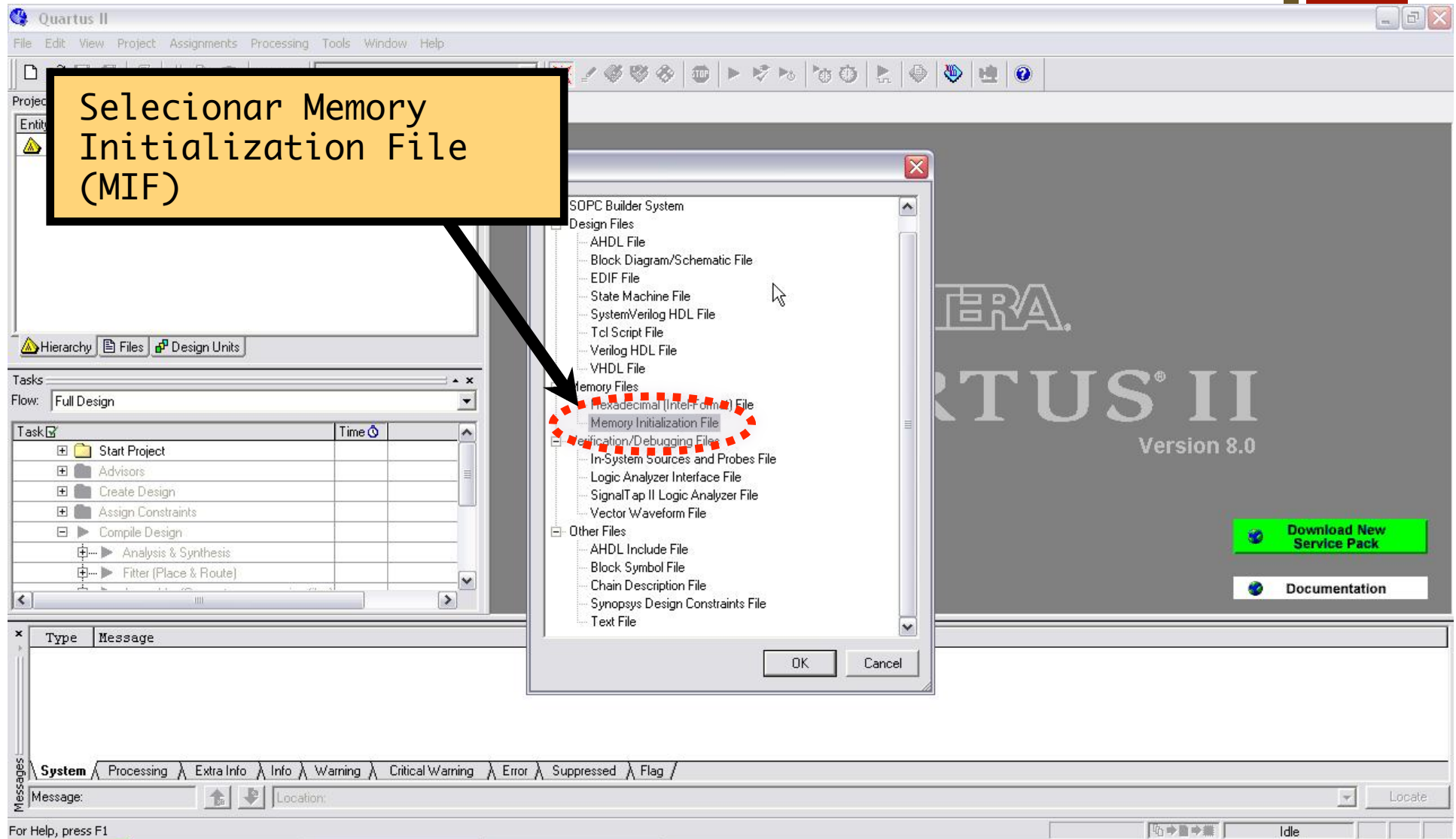
1
1
1
1
1
1
1
...
1
1

matriz 10 X 100

vetor 100 X 1









Definir o número de palavras como 1000 e o tamanho da palavra como 32 bits

Number of Words & Word Size

Number of words: 1000

Word size: 32

OK Cancel



Quartus II - [Mif1.mif]

File Edit View Project Assignments Processing Tools Window Help

Project Navigator

Entity

- Compilation Hierarchy

Hierarchy Files Design Units

Tasks

Flow: Full Design

Task

- Start Project
- Advisors
- Create Design
- Assign Constraints
- Compile Design
 - Analysis & Synthesis
 - Filter (Place & Route)

Mif1.mif

Addr	+0	+1	+2	+3	+4	+5	+6	+7
0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0
32	0	0	0	0	0	0	0	0
40	0	0	0	0	0	0	0	0
48	0	0	0	0	0	0	0	0
56	0	0	0	0	0	0	0	0
64	0	0	0	0	0	0	0	0
72	0	0	0	0	0	0	0	0
80	0	0	0	0	0	0	0	0
88	0	0	0	0	0	0	0	0
96	0	0	0	0	0	0	0	0
104	0	0	0	0	0	0	0	0
112	0	0	0	0	0	0	0	0
120	0	0	0	0	0	0	0	0
128	0	0	0	0	0	0	0	0
136	0	0	0	0	0	0	0	0
144	0	0	0	0	0	0	0	0
152	0	0	0	0	0	0	0	0
160	0	0	0	0	0	0	0	0
168	0	0	0	0	0	0	0	0
176	0	0	0	0	0	0	0	0
184	0	0	0	0	0	0	0	0
192	0	0	0	0	0	0	0	0
200	0	0	0	0	0	0	0	0

Clicar com o botão direito na área cinza a direita e selecionar a opção Custom Fill Cells...

Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V
Paste Insert
Insert Cells
Delete Del
Fill Cells with 0's
Fill Cells with 1's
Custom Fill Cells...
Reverse Address Contents
Cells Per Row / AutoFit

Messages

System Processing Extra Info Info Warning Critical Warning Error Suppressed Flag

Message: Location: Locate

For Help, press F1

Idle



Quartus II - [Mif1.mif]

File Edit View Project Assignments Processing Tools Window Help

Project Navigator

Entity

Compilation Hierarchy

Hierarchy Files Design Units

Tasks

Flow: Full Design

Task

Start Project

Advisors

Create Design

Assign Constraints

Compile Design

Analysis & Synthesis

Filter (Place & Route)

Mif1.mif

Addr	+0	+1	+2	+3	+4	+5	+6	+7
0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0
32	0	0	0	0	0	0	0	0
40	0	0	0	0	0	0	0	0
48	0	0	0	0	0	0	0	0
56	0	0	0	0	0	0	0	0
64	0	0	0	0	0	0	0	0
72	0	0	0	0	0	0	0	0
80	0	0	0	0	0	0	0	0
88	0	0	0	0	0	0	0	0
96	0	0	0	0	0	0	0	0
104	0	0	0	0	0	0	0	0
112	0	0	0	0	0	0	0	0
120	0	0	0	0	0	0	0	0
128	0	0	0	0	0	0	0	0
136	0	0	0	0	0	0	0	0
144	0	0	0	0	0	0	0	0
152	0	0	0	0	0	0	0	0
160	0	0	0	0	0	0	0	0
168	0	0	0	0	0	0	0	0
176	0	0	0	0	0	0	0	0
184	0	0	0	0	0	0	0	0
192	0	0	0	0	0	0	0	0
200	0	0	0	0	0	0	0	0

Address range

The current address radix is: unsigned decimal

Starting address: 0 Ending address: 99

Custom value(s)

The current memory radix is: unsigned decimal

Repeating sequence (numbers can be delimited by either a space or a decimal)

1

Incrementing / decrementing

Starting value:

Increment by

OK Cancel

System Processing Extra Info Info Warning Critical Warning Error Suppressed Flag

Message:

For Help, press F1

Idle



Definir as posições de 100 a 199 com o valor 2 e assim sucessivamente até a posição 999



Quartus II - [Mif1.mif]

File Edit View Project Assignments Processing Tools Window Help

New... Ctrl+N
Open... Ctrl+O
Close Ctrl+F4
New Project Wizard...
Open Project... Ctrl+J
Convert MAX+PLUS II Project...
Save Project
Close Project
Save Ctrl+S
Save As...
Save Current File as AS...
File Properties...
Create / Update
Export...
Convert Programming Files...
Page Setup...
Print Preview
Print... Ctrl+P
Recent Files
Recent Projects
Exit Alt+F4

Mif1.mif

Addr	+0	+1	+2	+3	+4	+5	+6	+7
0	1	1	1	1	1	1	1	1
8	1	1	1	1	1	1	1	1
						1	1	1
						1	1	1
48	1	1	1	1	1	1	1	1
56	1	1	1	1	1	1	1	1
64	1	1	1	1	1	1	1	1
72	1	1	1	1	1	1	1	1
80	1	1	1	1	1	1	1	1
88	1	1	1	1	1	1	1	1
96	1	1	1	1	2	2	2	2
104	2	2	2	2	2	2	2	2
112	2	2	2	2	2	2	2	2
120	2	2	2	2	2	2	2	2
128	2	2	2	2	2	2	2	2
136	2	2	2	2	2	2	2	2
144	2	2	2	2	2	2	2	2
152	2	2	2	2	2	2	2	2
160	2	2	2	2	2	2	2	2
168	2	2	2	2	2	2	2	2
176	2	2	2	2	2	2	2	2
184	2	2	2	2	2	2	2	2
192	2	2	2	2	2	2	2	2
200	3	3	3	3	3	3	3	3

Salvar o arquivo MIF

System Processing Extra Info Info Warning Critical Warning Error Suppressed Flag

Message: Location: Locate

Saves the current file with a new name

Idle



Quartus II - [Mif1.mif]

File Edit View Project Assignments Processing Tools Window Help

Project Navigator

Entity

Compilation Hierarchy

Mif1.mif

Addr	+0	+1	+2	+3	+4	+5	+6	+7
0	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1	1
3	1	1	1	1	1	1	1	1
4	1	1	1	1	1	1	1	1
5	1	1	1	1	1	1	1	1
6	1	1	1	1	1	1	1	1
7	1	1	1	1	1	1	1	1
8	1	1	1	1	1	1	1	1
9	1	1	1	1	1	1	1	1
10	1	1	1	1	1	1	1	1
11	1	1	1	1	1	1	1	1
12	1	1	1	1	1	1	1	1
13	1	1	1	1	1	1	1	1
14	1	1	1	1	1	1	1	1
15	1	1	1	1	1	1	1	1
16	1	1	1	1	1	1	1	1
17	1	1	1	1	1	1	1	1
18	1	1	1	1	1	1	1	1
19	1	1	1	1	1	1	1	1
20	1	1	1	1	1	1	1	1
21	1	1	1	1	1	1	1	1
22	1	1	1	1	1	1	1	1
23	1	1	1	1	1	1	1	1
24	1	1	1	1	1	1	1	1
25	1	1	1	1	1	1	1	1
26	1	1	1	1	1	1	1	1
27	1	1	1	1	1	1	1	1
28	1	1	1	1	1	1	1	1
29	1	1	1	1	1	1	1	1
30	1	1	1	1	1	1	1	1
31	1	1	1	1	1	1	1	1
32	1	1	1	1	1	1	1	1
33	1	1	1	1	1	1	1	1
34	1	1	1	1	1	1	1	1
35	1	1	1	1	1	1	1	1
36	1	1	1	1	1	1	1	1
37	1	1	1	1	1	1	1	1
38	1	1	1	1	1	1	1	1
39	1	1	1	1	1	1	1	1
40	1	1	1	1	1	1	1	1
41	1	1	1	1	1	1	1	1
42	1	1	1	1	1	1	1	1
43	1	1	1	1	1	1	1	1
44	1	1	1	1	1	1	1	1
45	1	1	1	1	1	1	1	1
46	1	1	1	1	1	1	1	1
47	1	1	1	1	1	1	1	1
48	1	1	1	1	1	1	1	1
49	1	1	1	1	1	1	1	1
50	1	1	1	1	1	1	1	1
51	1	1	1	1	1	1	1	1
52	1	1	1	1	1	1	1	1
53	1	1	1	1	1	1	1	1
54	1	1	1	1	1	1	1	1
55	1	1	1	1	1	1	1	1
56	1	1	1	1	1	1	1	1
57	1	1	1	1	1	1	1	1
58	1	1	1	1	1	1	1	1
59	1	1	1	1	1	1	1	1
60	1	1	1	1	1	1	1	1
61	1	1	1	1	1	1	1	1
62	1	1	1	1	1	1	1	1
63	1	1	1	1	1	1	1	1
64	1	1	1	1	1	1	1	1
65	1	1	1	1	1	1	1	1
66	1	1	1	1	1	1	1	1
67	1	1	1	1	1	1	1	1
68	1	1	1	1	1	1	1	1
69	1	1	1	1	1	1	1	1
70	1	1	1	1	1	1	1	1
71	1	1	1	1	1	1	1	1
72	1	1	1	1	1	1	1	1
73	1	1	1	1	1	1	1	1
74	1	1	1	1	1	1	1	1
75	1	1	1	1	1	1	1	1
76	1	1	1	1	1	1	1	1
77	1	1	1	1	1	1	1	1
78	1	1	1	1	1	1	1	1
79	1	1	1	1	1	1	1	1
80	1	1	1	1	1	1	1	1
81	1	1	1	1	1	1	1	1
82	1	1	1	1	1	1	1	1
83	1	1	1	1	1	1	1	1
84	1	1	1	1	1	1	1	1
85	1	1	1	1	1	1	1	1
86	1	1	1	1	1	1	1	1
87	1	1	1	1	1	1	1	1
88	1	1	1	1	1	1	1	1
89	1	1	1	1	1	1	1	1
90	1	1	1	1	1	1	1	1
91	1	1	1	1	1	1	1	1
92	1	1	1	1	1	1	1	1
93	1	1	1	1	1	1	1	1
94	1	1	1	1	1	1	1	1
95	1	1	1	1	1	1	1	1
96	1	1	1	1	1	1	1	1
97	1	1	1	1	1	1	1	1
98	1	1	1	1	1	1	1	1
99	1	1	1	1	1	1	1	1
100	1	1	1	1	1	1	1	1
101	1	1	1	1	1	1	1	1
102	1	1	1	1	1	1	1	1
103	1	1	1	1	1	1	1	1
104	1	1	1	1	1	1	1	1
105	1	1	1	1	1	1	1	1
106	1	1	1	1	1	1	1	1
107	1	1	1	1	1	1	1	1
108	1	1	1	1	1	1	1	1
109	1	1	1	1	1	1	1	1
110	1	1	1	1	1	1	1	1
111	1	1	1	1	1	1	1	1
112	1	1	1	1	1	1	1	1
113	1	1	1	1	1	1	1	1
114	1	1	1	1	1	1	1	1
115	1	1	1	1	1	1	1	1
116	1	1	1	1	1	1	1	1
117	1	1	1	1	1	1	1	1
118	1	1	1	1	1	1	1	1
119	1	1	1	1	1	1	1	1
120	1	1	1	1	1	1	1	1
121	1	1	1	1	1	1	1	1
122	1	1	1	1	1	1	1	1
123	1	1	1	1	1	1	1	1
124	1	1	1	1	1	1	1	1
125	1	1	1	1	1	1	1	1
126	1	1	1	1	1	1	1	1
127	1	1	1	1	1	1	1	1
128	1	1	1	1	1	1	1	1
129	1	1	1	1	1	1	1	1
130	1	1	1	1	1	1	1	1
131	1	1	1	1	1	1	1	1
132	1	1	1	1	1	1	1	1
133	1	1	1	1	1	1	1	1
134	1	1	1	1	1	1	1	1
135	1	1	1	1	1	1	1	1
136	1	1	1	1	1	1	1	1
137	1	1	1	1	1	1	1	1
138	1	1	1	1	1	1	1	1
139	1	1	1	1	1	1	1	1
140	1	1	1	1	1	1	1	1
141	1	1	1	1	1	1	1	1
142	1	1	1	1	1	1	1	1
143	1	1	1	1	1	1	1	1
144	1	1	1	1	1	1	1	1
145	1	1	1	1	1	1	1	1
146	1	1	1	1	1	1	1	1
147	1	1	1	1	1	1	1	1
148	1	1	1	1	1	1	1	1
149	1	1	1	1	1	1	1	1
150	1	1	1	1	1	1	1	1
151	1	1	1	1	1	1	1	1
152	1	1	1	1	1	1	1	1
153	1	1	1	1	1	1	1	1
154	1	1	1	1	1	1	1	1
155	1	1	1	1	1	1	1	1
156	1	1	1	1	1	1	1	1
157	1	1	1	1	1	1	1	1
158	1	1	1	1	1	1	1	1
159	1	1	1	1	1	1	1	1
160	1	1	1	1	1	1	1	1
161	1	1	1	1	1	1	1	1
162	1	1	1	1	1	1	1	1
163	1	1	1	1	1	1	1	1
164	1	1	1	1	1	1	1	1
165	1	1	1	1	1	1	1	1
166	1	1	1	1	1	1	1	1
167	1	1	1	1	1	1	1	1
168	1	1	1	1	1	1	1	1
169	1	1	1	1	1	1	1	1
170	1	1	1	1	1	1	1	1
171	1	1	1	1	1	1	1	1
172	1	1	1	1	1	1	1	1
173	1	1	1	1	1	1	1	1
174	1	1	1	1	1	1	1	1
175	1	1	1	1	1	1	1	1
176	1	1	1	1	1	1	1	1
177	1	1	1	1	1	1	1	1
178	1	1	1	1	1	1	1	1
179	1	1	1	1	1	1	1	1
180	1	1	1	1	1	1	1	1
181	1	1	1	1	1	1	1	1
182	1	1	1	1	1	1	1	1
183	1	1	1	1	1	1	1	1
184	1	1	1	1	1	1	1	1
185	1	1	1	1	1	1	1	1
186	1	1	1	1	1	1	1	1
187	1	1	1	1	1	1	1	1
188	1	1	1	1	1	1	1	1
189	1	1	1	1	1	1	1	1
190	1	1	1	1	1	1	1	1
191	1	1	1	1	1	1	1	1
192	1	1	1	1	1	1	1	1
193	1	1	1	1	1	1	1	1
194	1	1	1	1	1	1	1	1
195	1	1	1	1	1	1	1	1
196	1	1	1	1	1	1	1	1
197	1	1	1	1	1	1	1	1
198	1	1	1	1	1	1	1	1
199	1	1	1	1	1	1	1	1
200	1	1	1	1	1	1	1	1
201	1	1	1	1	1	1	1	1
202	1	1	1	1	1	1	1	1
203	1	1	1	1	1	1	1	1
204	1	1	1					

+ Criação das Memórias na Ferramenta

- Para a criação das memórias, utilizaremos a ferramenta **MegaWizard**.
- Iremos criar três memórias dual-port com palavras de 32 bits do seguinte tamanho:
 - Memória Matriz (10 x 100) - 1000 palavras.
 - Memória Vetor (100 x 1) - 100 palavras.
 - Memória Vetor Resultante (10 x 1) - 10 palavras.

Quartus II - C:/projetos/...

File Edit View Project Ass...

Project Navigator

Entity

- Cyclone II: EP2C35F672C6
- mult_mat_vector

Victor Medeiros

Internet

- Mozilla Firefox

Email

- Outlook Express

Release Notes

- Quartus II 6.0 MegaWizard Plug-in Manager

- Keil uVision3
- Quartus II 6.0
- C5510 DSK CCS
- Quartus II 8.0 (32-Bit)

Todos os programas

Iniciar

Catálogo do Windows

- Definir acesso e padrões do programa
- Windows Update
- Acessórios
- Altera
- avast! Antivirus
- Borland C++Builder 3
- eMule
- Fantom CD
- Gadwin Systems
- Inicializar
- Jogos
- MATLAB
- Maxis
- ModelSim PE Student Edition 6.3g
- ModelSim XE III 6.2g
- Mozilla Firefox
- Notepad++
- PComm Lite 2.6
- Picasa2
- ReadPlease 2003
- Show do Milhão
- Skype
- SmartSVN
- Spybot - Search & Destroy
- Texas Instruments
- TortoiseSVN
- Winamp
- WinRAR
- Xilinx ISE 9.2i
- Adobe Reader 8
- Apple Software Update
- Assistência remota
- Audacity

Internet Explorer

- Keil uVision3
- Outlook Express
- Windows Media Player
- MegaCore
- Nios II EDS 6.0
- Quartus II 6.0
- Quartus II 8.0

Quartus II 8.0 (32-Bit)

- Quartus II 8.0 Design Space Explorer
- Quartus II 8.0 MegaWizard Plug-in Manager
- Quartus II 8.0 Uninstall, Repair or Modify

matriz_vector - multiplicador_matriz_vector - [Compilation Repo]

mem_control.vhd

Summary

Successful - Mon Aug 04 15:53:07 2008

8.0 Build 215 05/29/2008 SJ Full Version

multiplicador_matriz_vector

mult_mat_vector

Cyclone II

EP2C35F672C6

Final

Yes

Requirements

Total logic elements	134 / 33,216 (< 1 %)
Total combinational functions	134 / 33,216 (< 1 %)
Dedicated logic registers	82 / 33,216 (< 1 %)
Total registers	82
Total pins	33 / 475 (7 %)
Total virtual pins	0
Total memory bits	17,760 / 483,840 (4 %)
Embedded Multiplier 9-bit elements	2 / 70 (3 %)
PLLs	0 / 4 (0 %)

lock pin = "Pclk") is 6.441 ns

"memory_out_2p:mem_out|altsyncram:altsyncram_component|altsyncram_dall:auto_generated

in = "Pclk") is -3.522 ns

Flag /

Locate

Idle

15:54

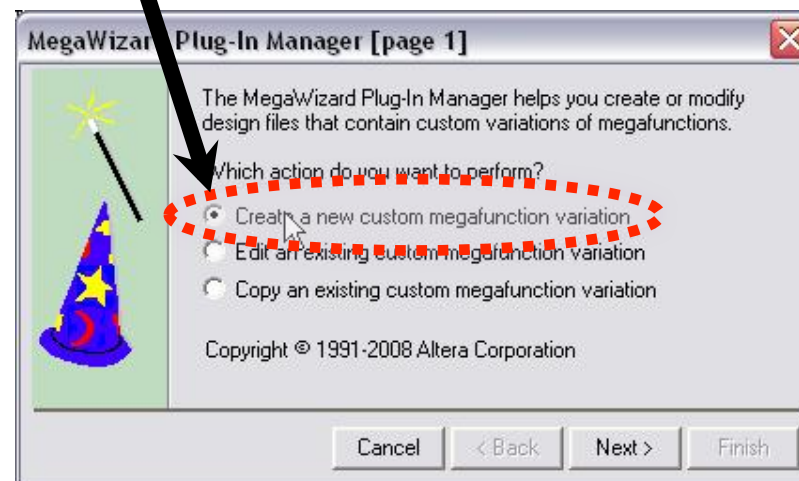
segunda-feira

4/8/2008

Abrir o MegaWizard



Selecionar a primeira opção



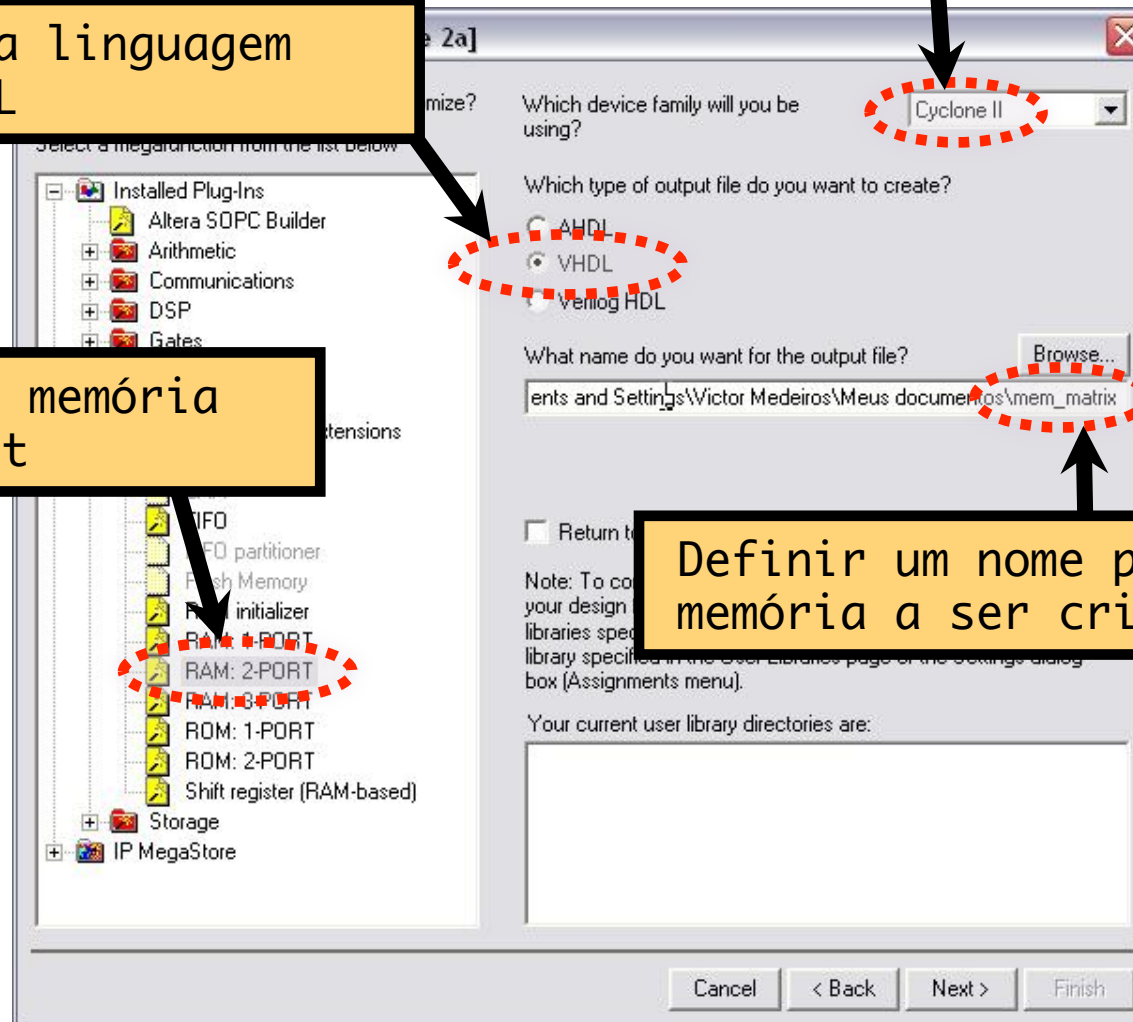


Definir a linguagem
como VHDL

Definir o FPGA como
Cyclone II

Definir como memória
RAM Dual-port

Definir um nome para
memória a ser criada





Nesta tela não é necessário modificar nenhum parâmetro

MegaWizard Plug-In Manager - RAM: 2-PORT [page 3 of 10]

RAM: 2-PORT

About Documentation

1 Parameter Settings 2 EDA 3 Summary

General Widths/Blk Type Clks/Rd, Byte En Regs/Clocks/Aclrs Output1 Mem Init

mem_matrix

data[7..0]
✕
waddress[4..0]
✕
wren
✕
rdaddress[4..0]
✕
clock
✕

32 Word(s) RAM

q[7..0]
✕

Block Type: AUTO

Resource Usage

1 M4K

Currently selected device family: Cyclone II

☒ Match project/default

How will you be using the dual port ram?

☒ With one read port and one write port

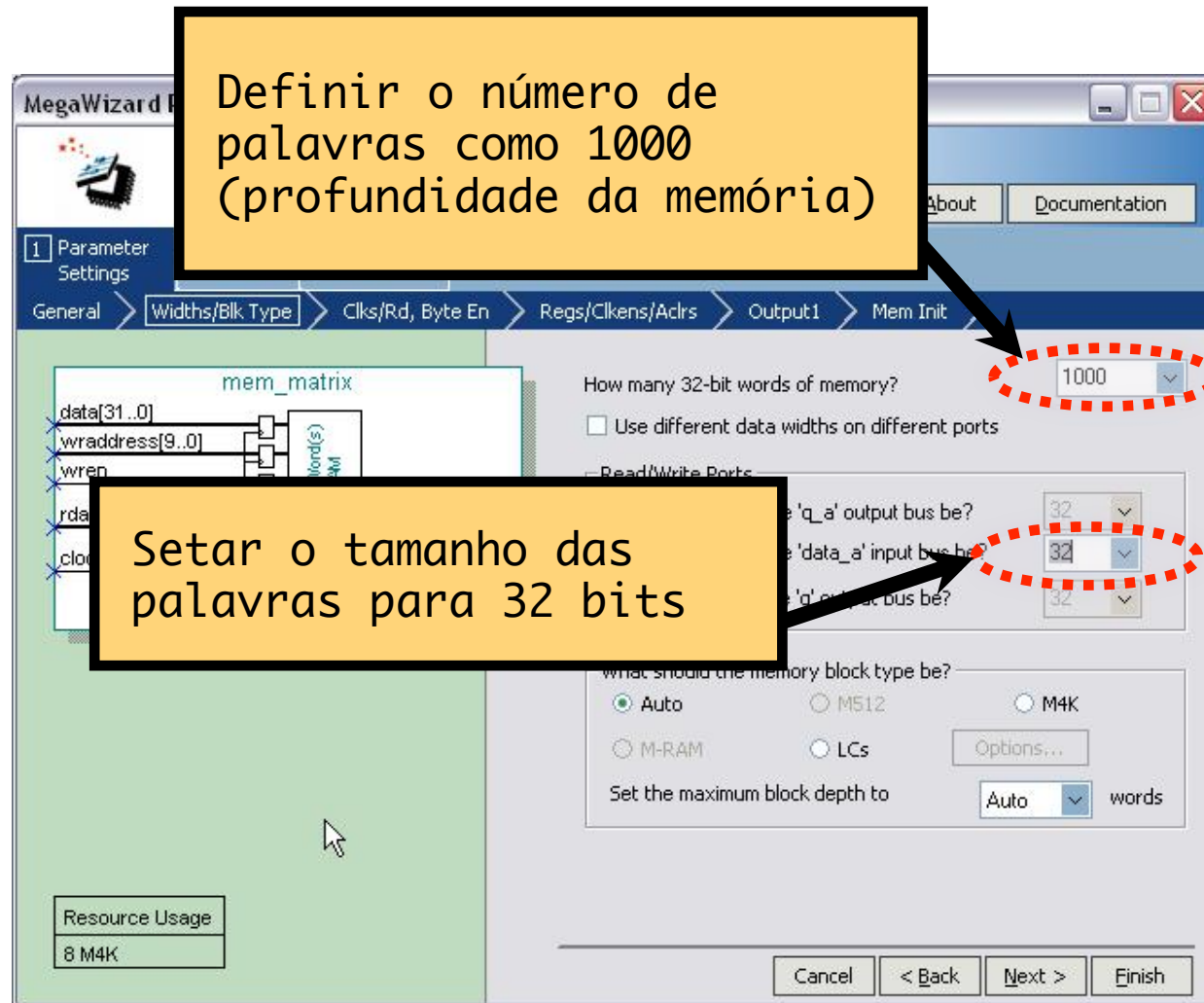
☐ With two read/write ports

How do you want to specify the memory size?

☒ As a number of words

☐ As a number of bits

Cancel < Back Next > Finish





Nesta tela não é
necessário modificar
nenhum parâmetro

RAM: 2-PORT

About Documentation

1 Parameter Settings 2 EDA 3 Summary

General Widths/Blk Type Clks/Rd, Byte En Regs/Clkens/Aclrs Output1 Mem Init

mem_matrix

data[31..0]
waddress[9..0]
wren
rdaddress[9..0]
clock

1000 Word(s) RAM

q[31..0]

Block Type: AUTO

Which clocking method do you want to use?

- ☒ Single clock
- ☐ Dual clock: use separate 'read' and 'write' clocks
- ☐ Dual clock: use separate 'input' and 'output' clocks
- ☐ No clock (fully asynchronous)
- ☐ Customize clocks for A and B ports

☐ Create an 'rden' read enable signal

Byte Enable Ports

- ☐ Create byte enable for port A
- ☐ Create byte enable for port B

What is the width of a byte for byte enables? 8 bits

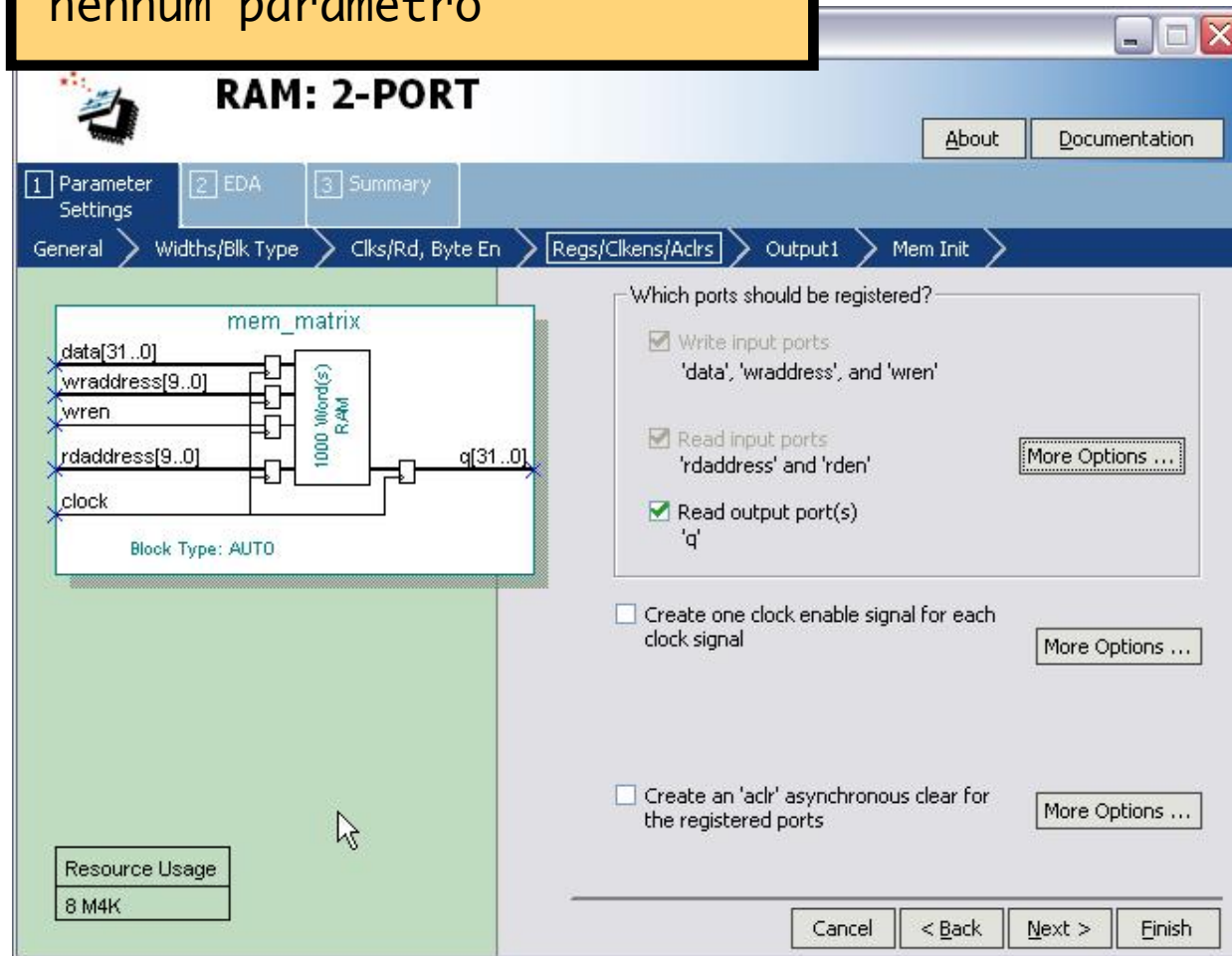
Resource Usage

8 M4K

Cancel < Back Next > Finish



Nesta tela não é
necessário modificar
nenhum parâmetro





Nesta tela não é
necessário modificar
nenhum parâmetro

RAM: 2-PORT

About Documentation

1 Parameter Settings 2 EDA 3 Summary

General Widths/Blk Type Clks/Rd, Byte En Regs/Clkens/Aclrs **Output1** Mem Init

mem_matrix

data[31..0]
wraddress[9..0]
wren
rdaddress[9..0]
clock

1000 Word(s) RAM

q[31..0]

Block Type: AUTO

Resource Usage
8 M4K

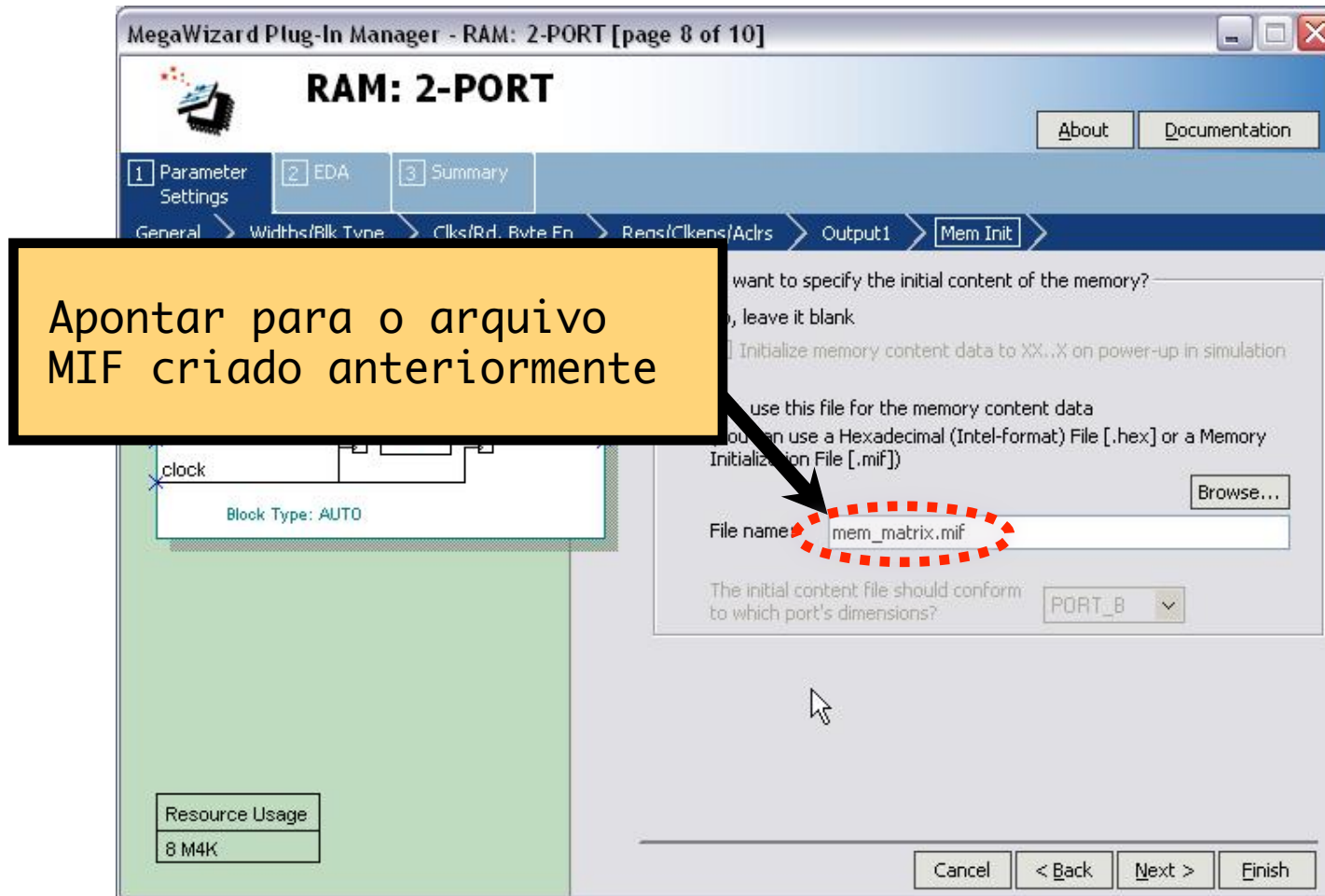
Mixed Port Read-During-Write for Single Input Clock RAM

How should the q output behave when reading a memory location that is being written from the other port?

☐ Old memory contents appear
Note: M-RAM cannot be used with this behavior

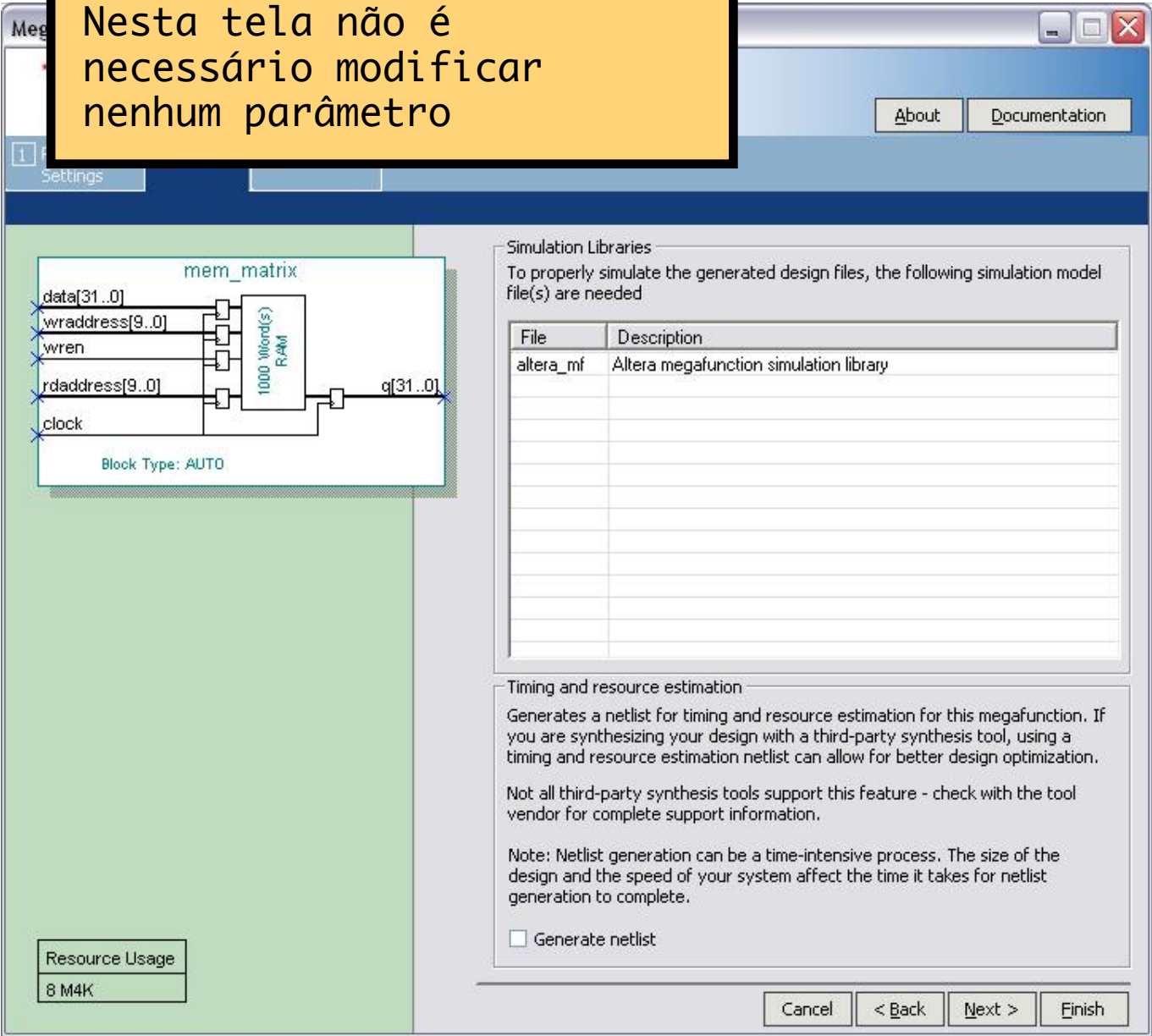
☒ I don't care

Cancel < Back Next > Finish





Nesta tela não é necessário modificar nenhum parâmetro





MegaWizard Plug-In Manager - RAM: 2-PORT [page 10 of 10] -- Summary

RAM: 2-PORT

1 Parameter Settings 2 EDA 3 Summary

About Documentation

mem_matrix

data[31..0]
waddress[9..0]
wren
rdaddress[9..0]
clock

1000 Word(s) RAM

q[31..0]

Block Type: AUTO

Resource Usage
8 M4K

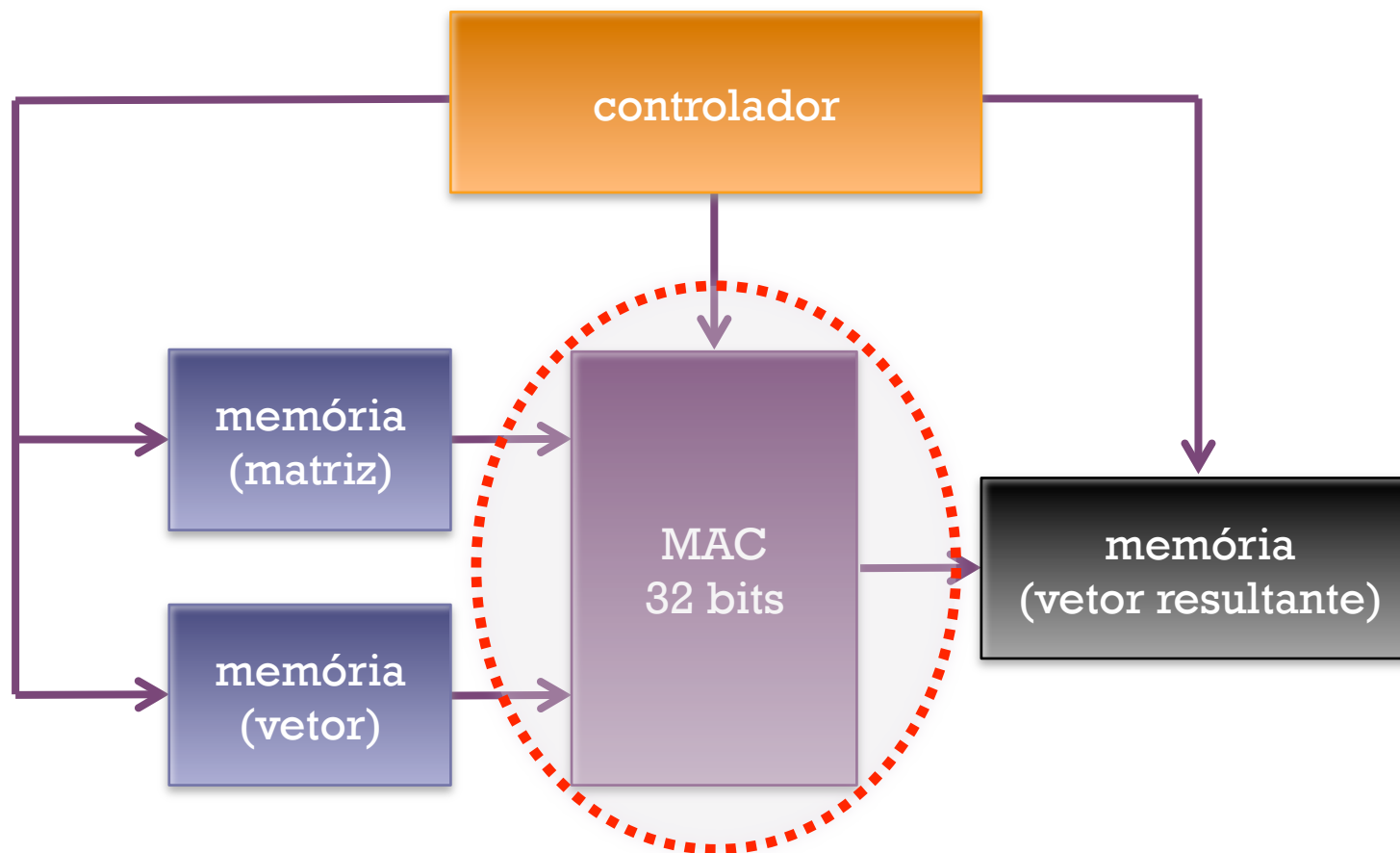
Selecção dos arquivos que serão gerados

directory:
C:\Documents and Settings\Victor Medeiros\Meus documentos\

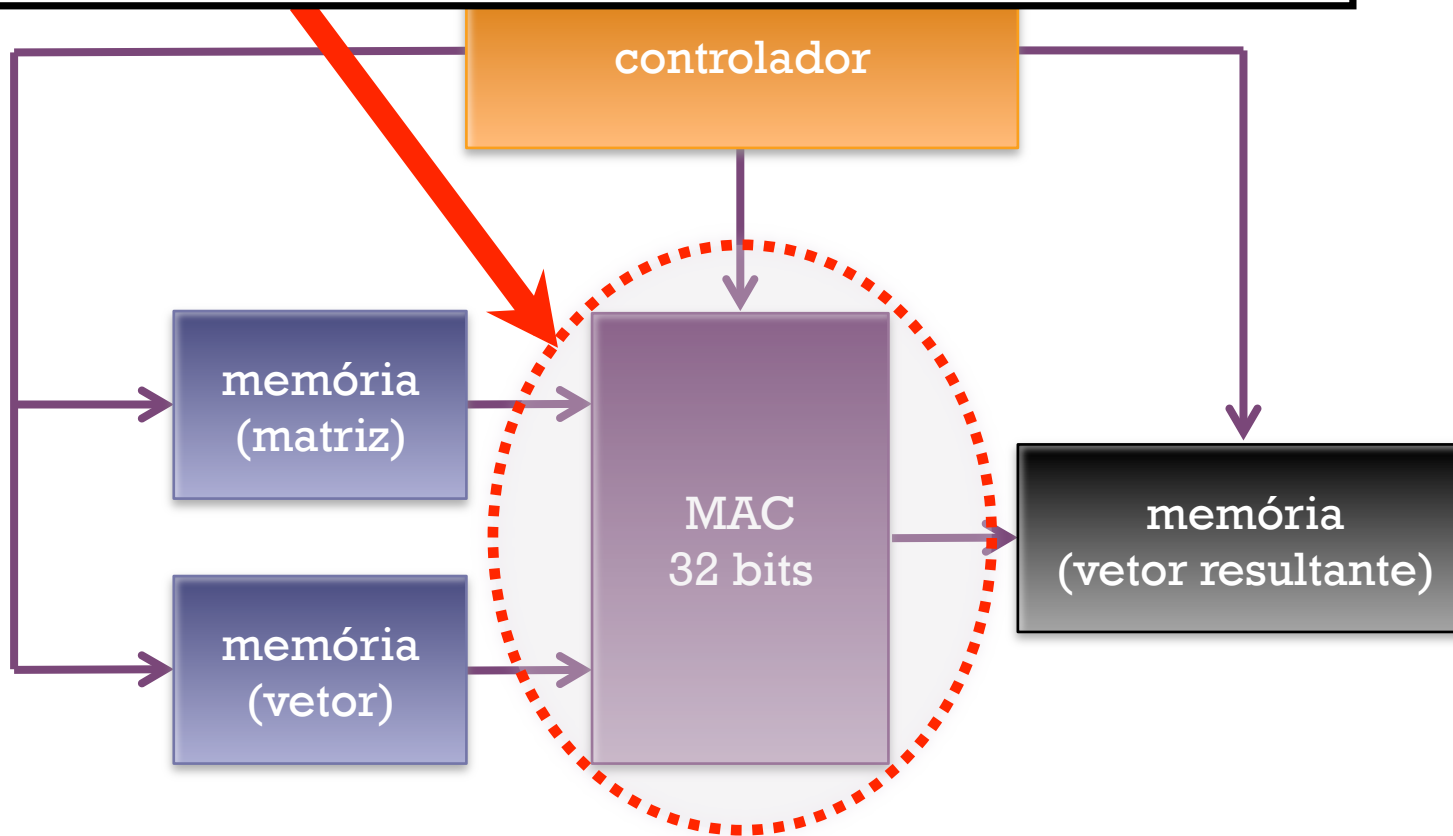
File	Description
<input checked="" type="checkbox"/> mem_matrix.vhd	Variation file
<input type="checkbox"/> mem_matrix.inc	AHDL Include file
<input checked="" type="checkbox"/> mem_matrix.cmp	VHDL component declaration file
<input type="checkbox"/> mem_matrix.bsf	Quartus II symbol file
<input checked="" type="checkbox"/> mem_matrix_inst.vhd	Instantiation template file
<input checked="" type="checkbox"/> mem_matrix_waveforms.html	Sample waveforms in summary
<input checked="" type="checkbox"/> mem_matrix_wave*.jpg	Sample waveform file(s)

Cancel < Back Next > Finish

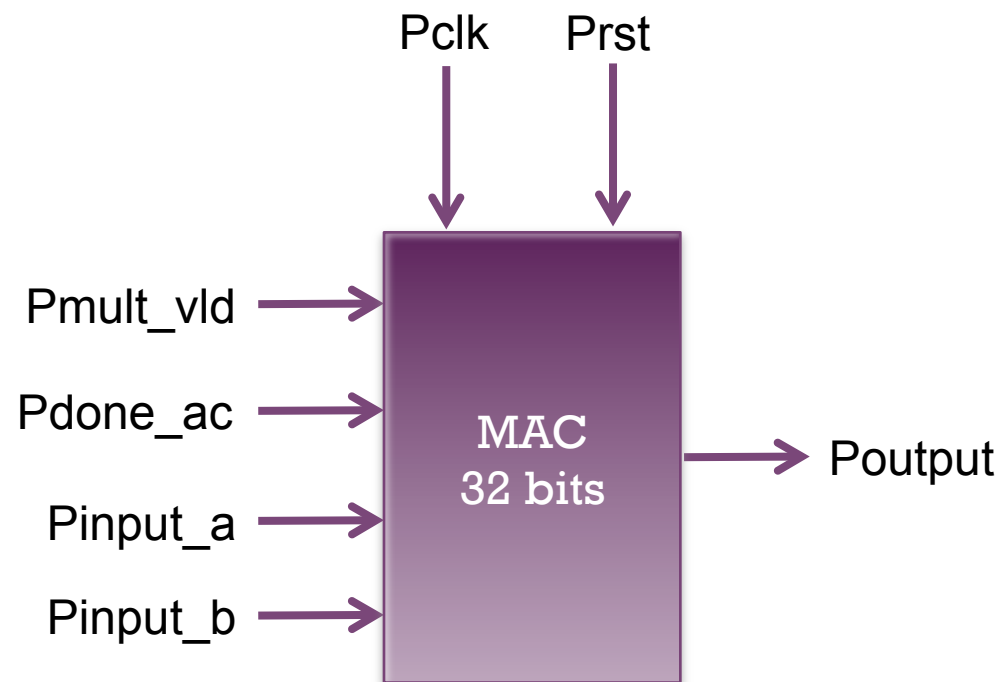
+ MAC 32 bits



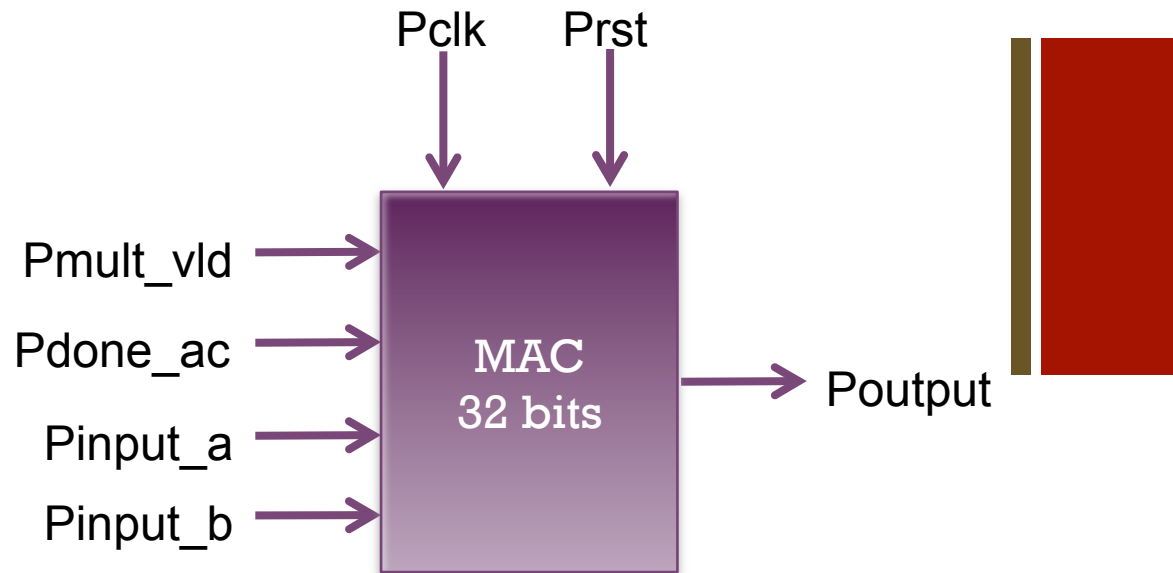
```
for(i = 0; i < numLines; i++) {  
    for(j = 0; j < numCols; j++) {  
        sum += matrix[i*numLines+j] * vector[j];  
    }  
    result_vector[i] = sum;  
    sum = 0;  
}
```



+ MAC 32 bits



+ MAC 32 bits



```
-- definicao da entidade
entity mac32bits is port (
    Prst      : in std_logic;
    Pclk      : in std_logic;
    Pmult_vld : in std_logic;
    Pdone_ac  : in std_logic;
    Pinput_a  : in std_logic_vector(31 downto 0);
    Pinput_b  : in std_logic_vector(31 downto 0);
    Poutput   : out std_logic_vector(31 downto 0)
);
end mac32bits;
```

```

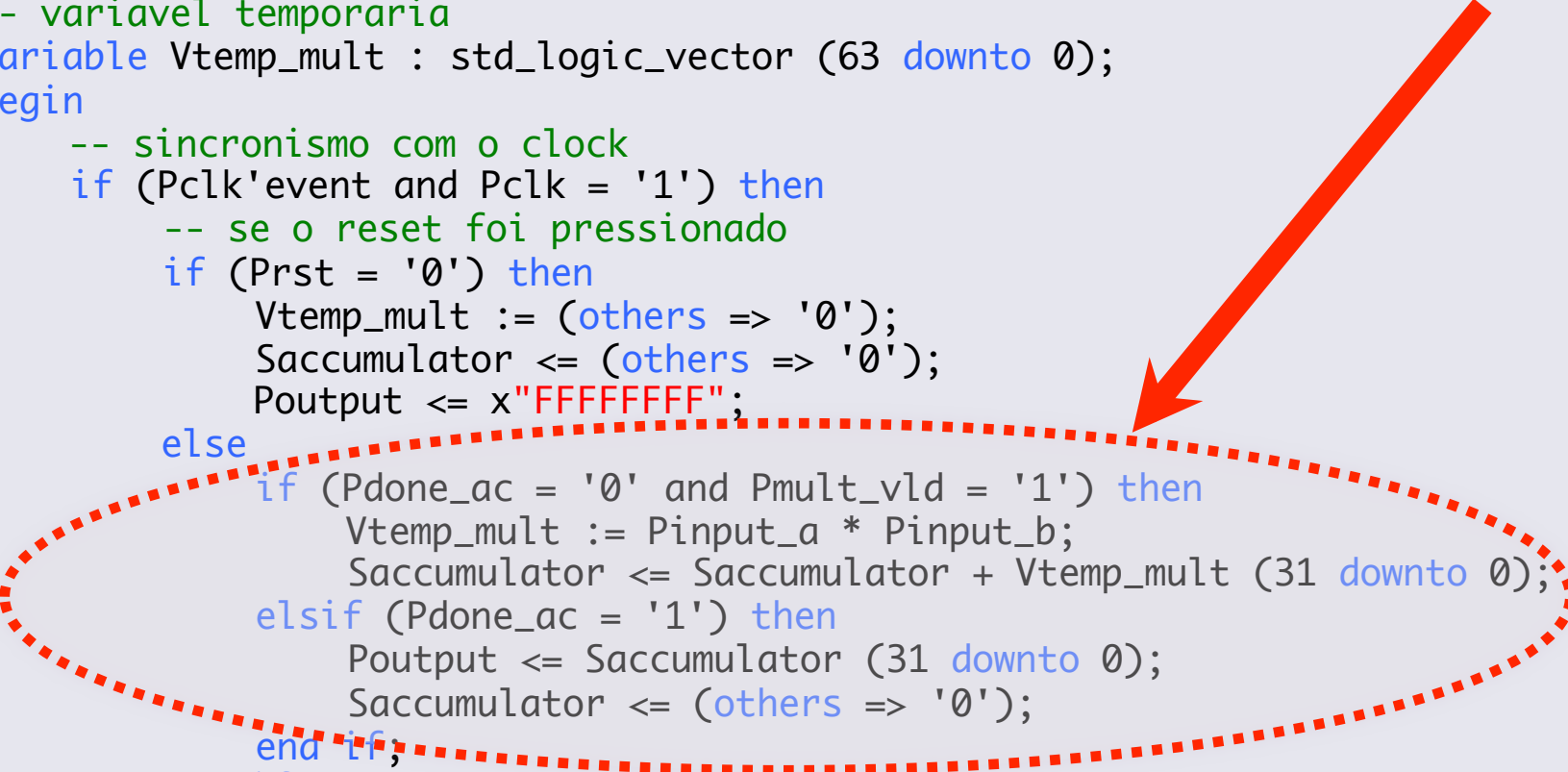
-- descricao da arquitetura comportamental da entidade
architecture arq_mac32bits of mac32bits is
-- definicao do registrador acumulador
signal Saccumulator : std_logic_vector (31 downto 0);
begin
    process (Pclk, Prst, Pmult_vld, Pdone_ac)
        -- declaracao das variaveis do processo
        -- variavel temporaria
        variable Vtemp_mult : std_logic_vector (63 downto 0);
        begin
            -- sincronismo com o clock
            if (Pclk'event and Pclk = '1') then
                -- se o reset foi pressionado
                if (Prst = '0') then
                    Vtemp_mult := (others => '0');
                    Saccumulator <= (others => '0');
                    Poutput <= x"FFFFFFFF";
                else
                    if (Pdone_ac = '0' and Pmult_vld = '1') then
                        Vtemp_mult := Pinput_a * Pinput_b;
                        Saccumulator <= Saccumulator + Vtemp_mult (31 downto 0);
                    elsif (Pdone_ac = '1') then
                        Poutput <= Saccumulator (31 downto 0);
                        Saccumulator <= (others => '0');
                    end if;
                end if;
            end if;
        end process;
    end arq_mac32bits;

```

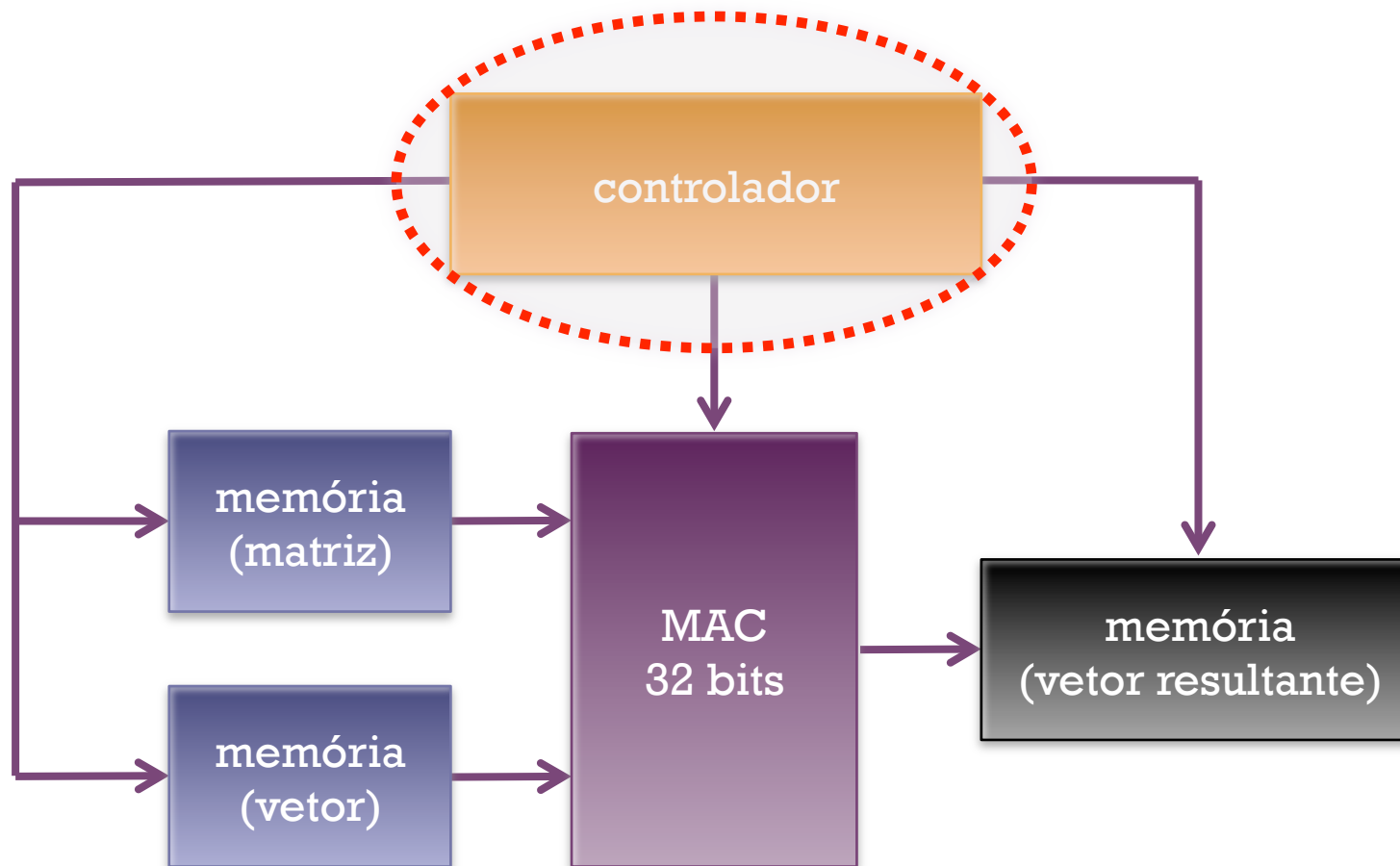
```

-- descricao da arquitetura comportamental da entidade
architecture arq_mac32bits of mac32bits is
-- definicao do registrador acumulador
signal Saccumulator : std_logic_vector (31 downto 0);
begin
    process (Pclk, Prst, Pmult_vld, Pdone_ac)
        -- declaracao das variaveis do processo
        -- variavel temporaria
        variable Vtemp_mult : std_logic_vector (63 downto 0);
        begin
            -- sincronismo com o clock
            if (Pclk'event and Pclk = '1') then
                -- se o reset foi pressionado
                if (Prst = '0') then
                    Vtemp_mult := (others => '0');
                    Saccumulator <= (others => '0');
                    Poutput <= x"FFFFFFFF";
                else
                    if (Pdone_ac = '0' and Pmult_vld = '1') then
                        Vtemp_mult := Pinput_a * Pinput_b;
                        Saccumulator <= Saccumulator + Vtemp_mult (31 downto 0);
                    elsif (Pdone_ac = '1') then
                        Poutput <= Saccumulator (31 downto 0);
                        Saccumulator <= (others => '0');
                    end if;
                end if;
            end if;
        end process;
    end arq_mac32bits;

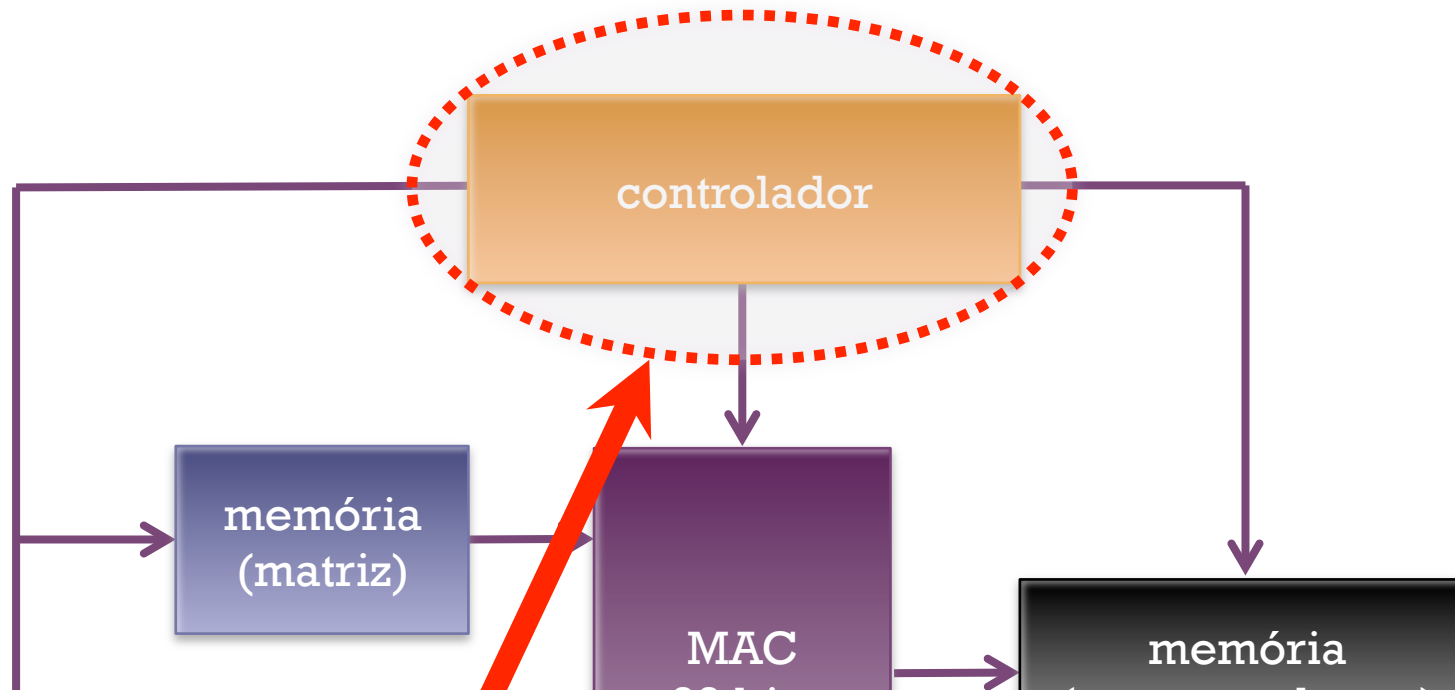
```



+ Controlador do multiplicador

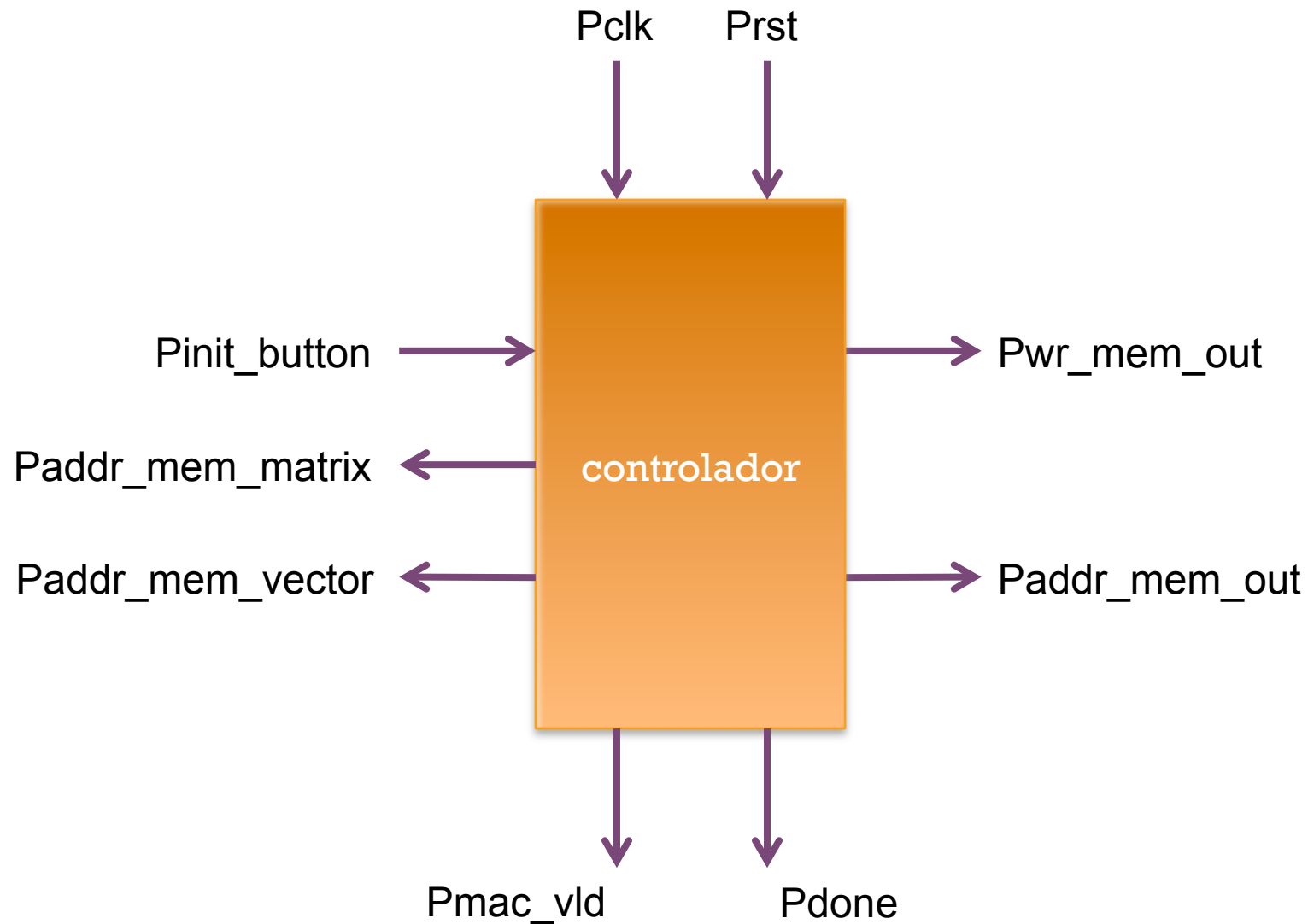


+ Controlador do multiplicador



```
for(i = 0; i < numLines; i++) {  
    for(j = 0; j < numCols; j++) {  
        sum += matrix[i*numLines+j] * vector[j];  
    }  
    result_vector[i] = sum;  
    sum = 0;  
}
```

+ Controlador do multiplicador

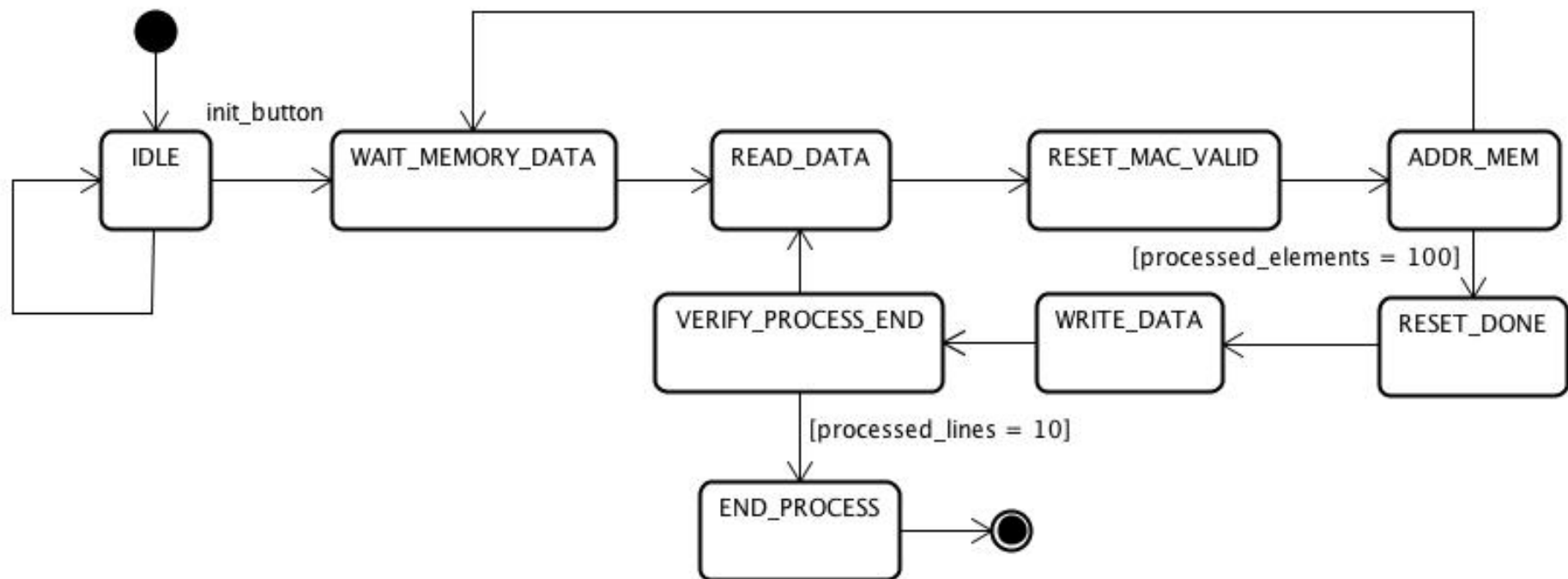


```

-- definicao da entidade
entity controlador is port (
    Pclk          : in std_logic;
    Prst          : in std_logic;
    -- portas de conexao com as memorias
    -- porta que endereca a memoria que contem a matriz
    Paddr_mem_matrix : out std_logic_vector(9 downto 0);
    -- porta que endereca a memoria que contem o vetor
    Paddr_mem_vector  : out std_logic_vector(6 downto 0);
    -- porta que endereca a memoria que contem o vetor de saida
    Paddr_mem_out     : out std_logic_vector(3 downto 0);
    -- porta que aciona a escrita na memoria de saida
    Pwr_mem_out       : out std_logic;
    -- sinais de controle
    -- porta que indica que o botao de init foi pressionado
    Pinit_button      : in std_logic;
    -- porta que indica para o MAC que ja existem dados validos
    Pmac_vld          : out std_logic;
    -- sinaliza para o MAC que todos os produtos ja foram realizados
    Pdone             : out std_logic
);
end controlador;

```

+ Máquina de Estados do Controlador (FSM)



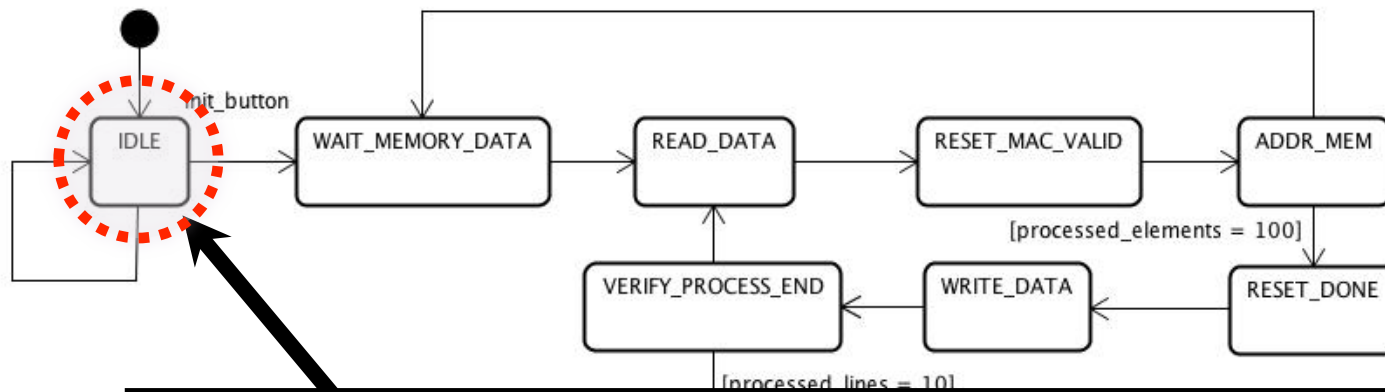
```

-- processo que implementa o comportamento do controlador
process(Pclk, Prst, Pinit_button)
    -- variaveis utilizadas no processo
    -- variavel que conta numero de operacoes do MAC
    variable Vop_counter    : integer range 0 to 100;
    -- variavel que endereca a memoria de saida
    variable Vaddr_mem_out : std_logic_vector(3 downto 0);

begin
    -- sincronismo com o clock
    if (Pclk'event and Pclk = '1') then
        -- se o reset foi pressionado
        if (Prst = '0') then
            Saddr_mem_matrix    <= "00000000000";
            Saddr_mem_vector    <= "00000000";
            Pmac_vld             <= '0';
            Vop_counter          := 0;
            Sctrl_mem_state      <= INIT;
            Pwr_mem_out          <= '1';
            Saddr_mem_out        <= "0000";
            Pdone                <= '0';

        else
            case Sctrl_mem_state is
                -- estado de inicializacao
                when IDLE =>
                    ...
            end case;
        end if;
    end if;
end process;

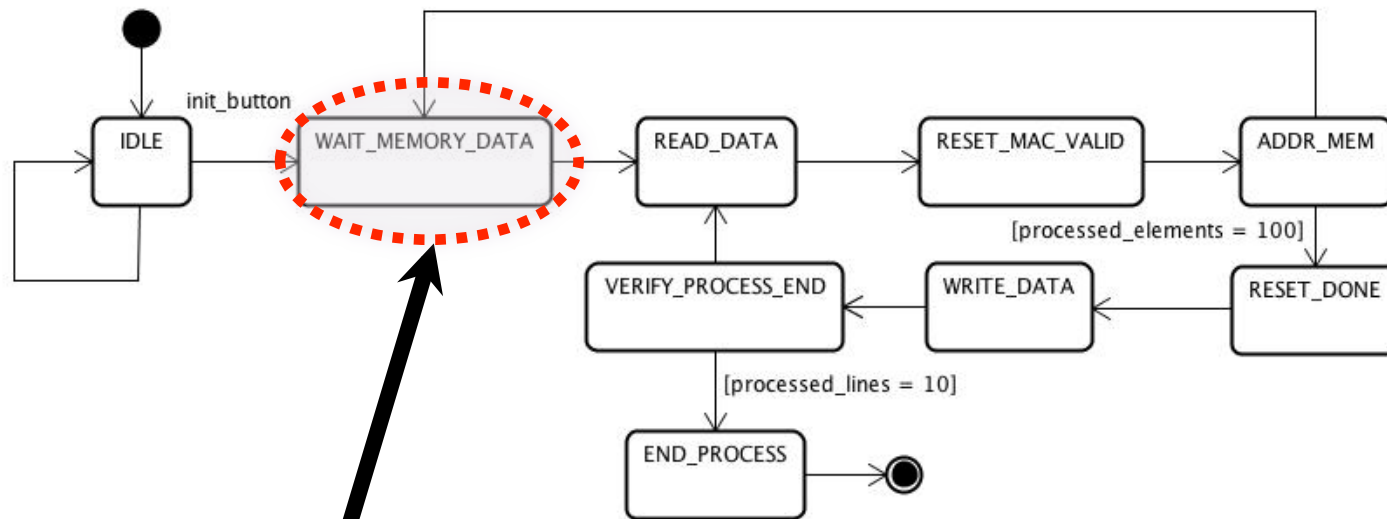
```



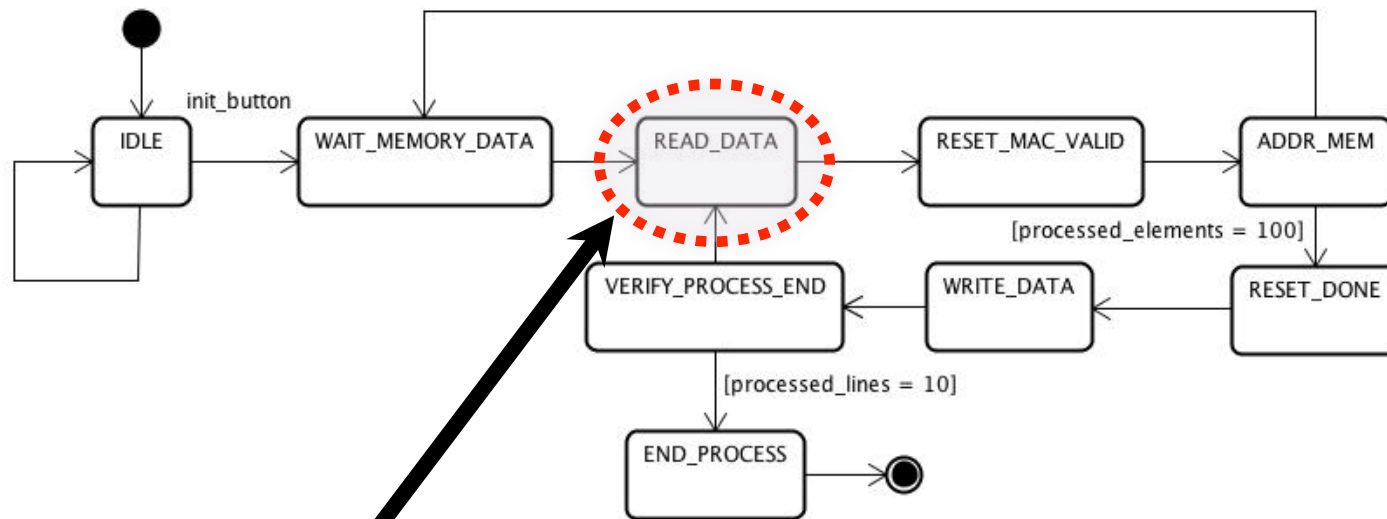
```

when IDLE =>
  Pwr_mem_out <= '0';
  -- caso o botao init tenha sido pressionado
  -- inicializa as portas de saida e dispara a execucao
  if (Pinit_button = '1') then
    Saddr_mem_matrix <= "0000000000";
    Saddr_mem_vector <= "00000000";
    Pmac_vld <= '0';
    Vop_counter := 0;
    Pwr_mem_out <= '0';
    Saddr_mem_out <= "0000";
    Pdone <= '0';
    Sctrl_mem_state <= WAIT_MEMORY_DATA;
  else
    Sctrl_mem_state <= IDLE;
  end if;

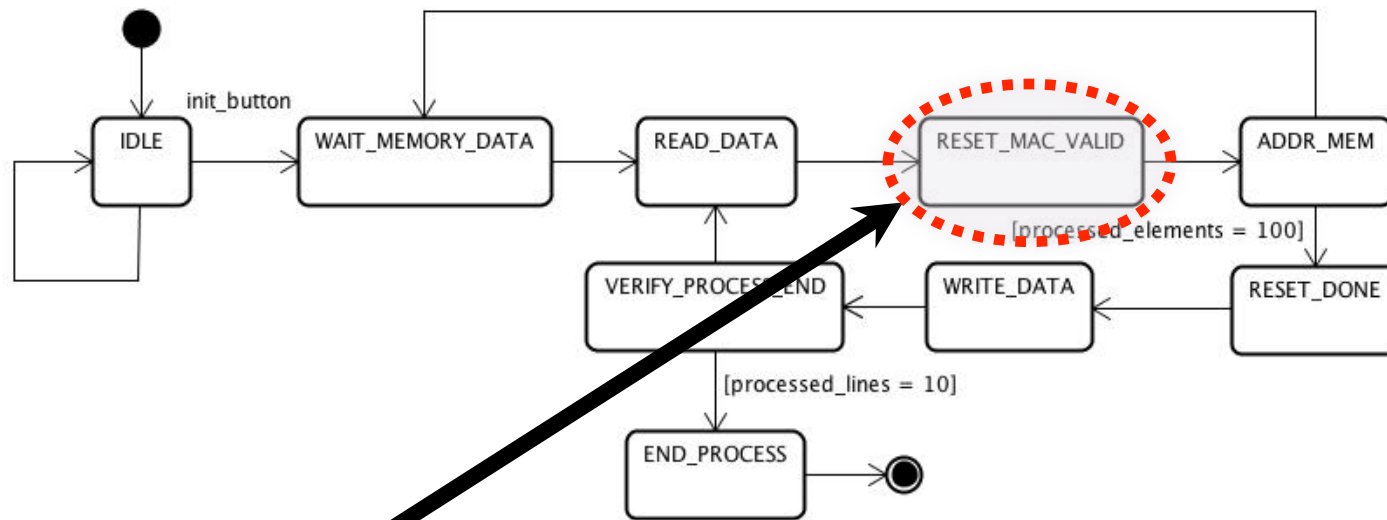
```



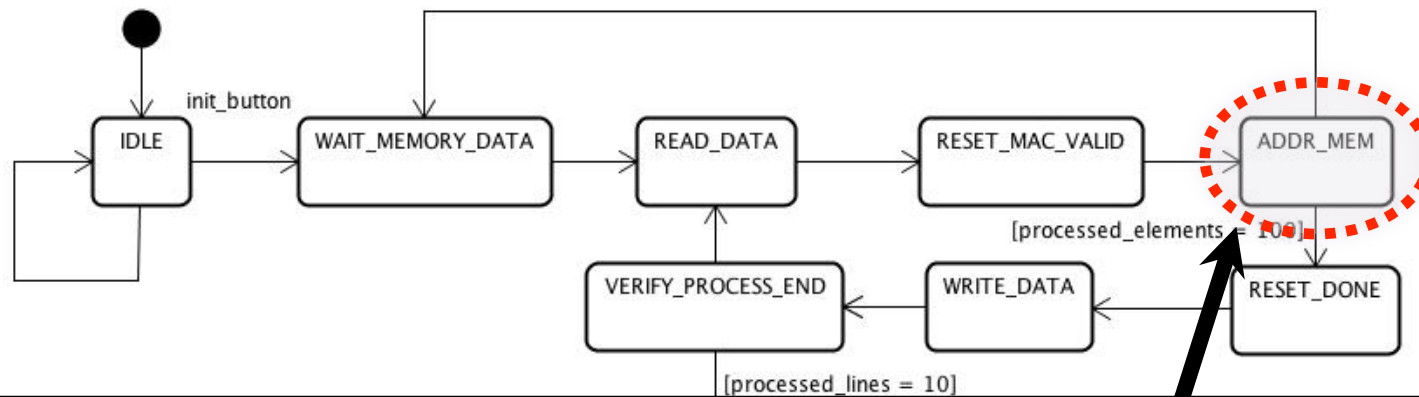
```
when WAIT_MEMORY_DATA =>
    -- espera um ciclo, tempo de resposta da memoria
    Sctrl_mem_state <= READ_DATA;
```



```
when READ_DATA =>
    -- indica para o mac que os dados sao validos
    Pmac_vld      <= '1';
    -- incrementa contador de operacoes do MAC
    Vop_counter   := Vop_counter + 1;
    -- vai enderecar novas palavras
    Sctrl_mem_state <= RESET_MAC_VALID;
```

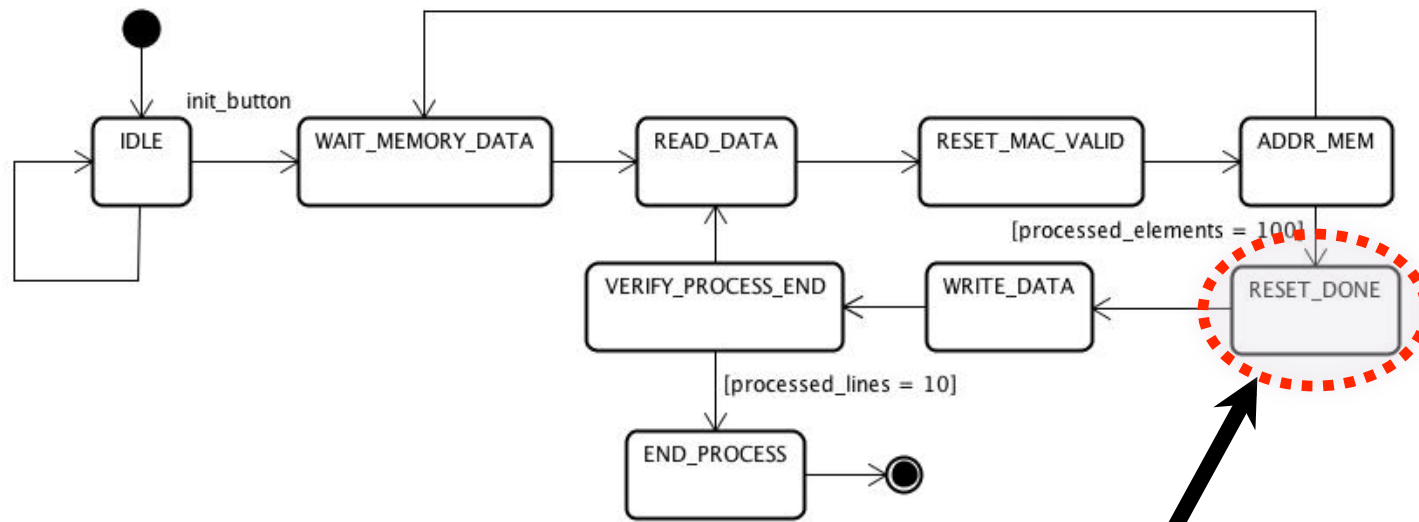


```
when RESET_MAC_VALID =>
    Pmac_vld      <= '0';
    Sctrl_mem_state <= ADDR_MEM;
```



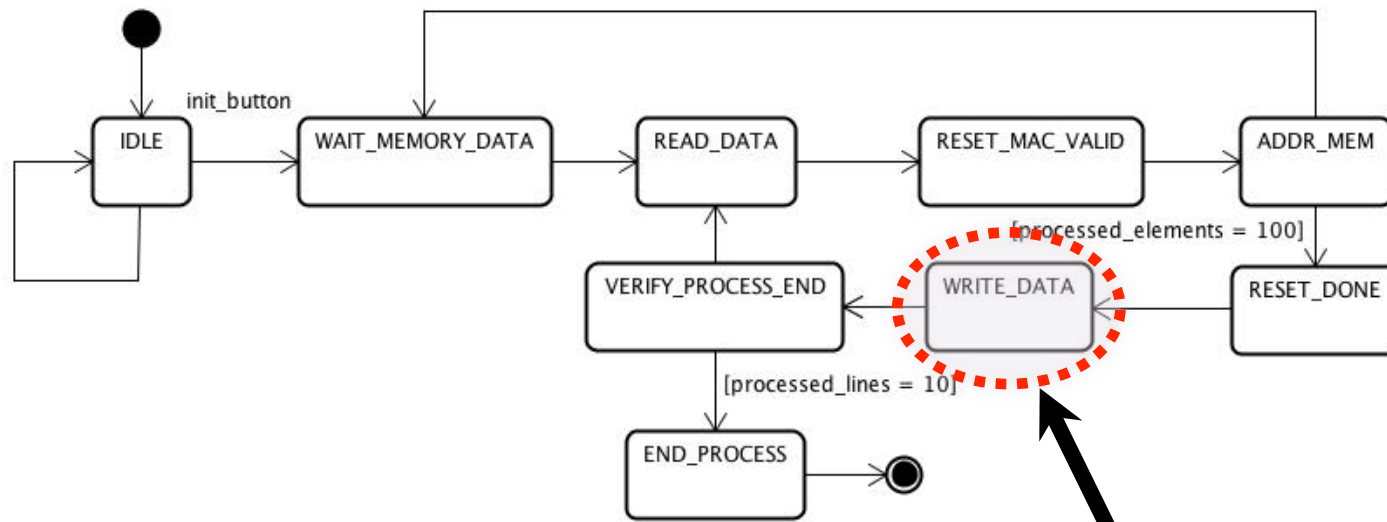
```

when ADDR_MEM =>
    Pmac_vld          <= '0';
    -- endereca a matriz
    Saddr_mem_matrix  <= Saddr_mem_matrix + "0000000001";
    -- endereca o vetor
    Saddr_mem_vector   <= Saddr_mem_vector + "0000001";
    -- se ja realizou 100 operacoes
    if (Vop_counter = 100) then
        -- zera contador e escreve na memoria de saida
        Vop_counter    := 0;
        Pdone          <= '1';
        Sctrl_mem_state <= RESET_DONE;
    else
        -- retorna a uma nova leitura da memoria
        Sctrl_mem_state <= WAIT_MEMORY_DATA;
    end if;
  
```



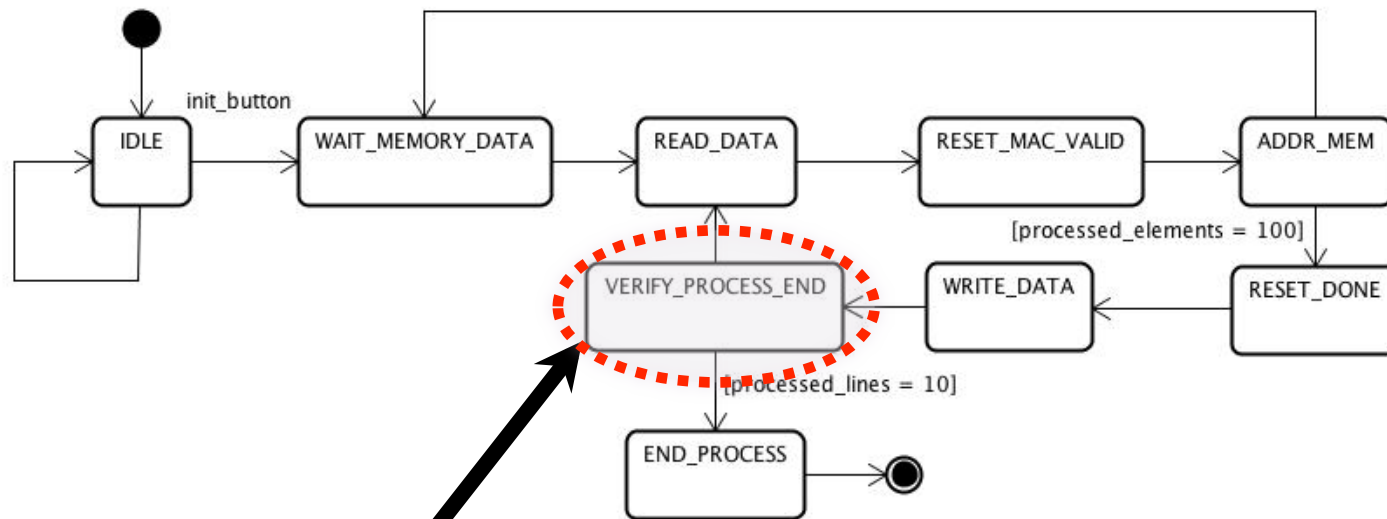
```

when RESET_DONE =>
    Sctrl_mem_state    <= WRITE_DATA;
    Saddr_mem_vector   <= "00000000";
    Pdone <= '0';
  
```



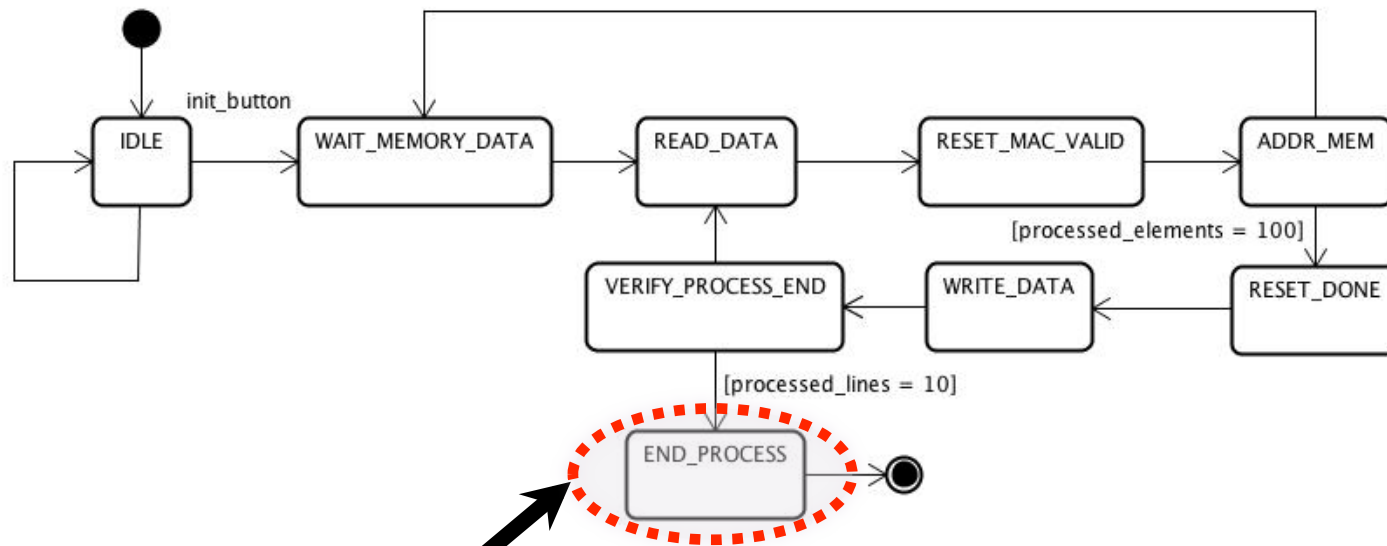
```

when WRITE_DATA =>
    Pdone          <= '0';
    -- sinaliza pedido de escrita na memoria
    Pwr_mem_out    <= '1';
    -- vai verificar se finalizou operacoes
    Sctrl_mem_state <= VERIFY_PROCESS_END;
  
```



```

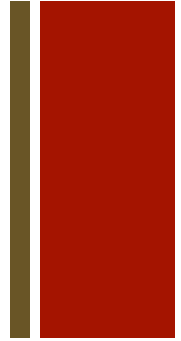
when VERIFY_PROCESS_END =>
    Pwr_mem_out      <= '0';
    Saddr_mem_out    <= Saddr_mem_out + "0001";
    -- verifica se ja finalizou operacao
    if ( Saddr_mem_out = "1001" ) then
        -- finalizou operacao (10 escritas)
        Sctrl_mem_state <= END_PROCESS;
    else
        -- ainda nao finalizou
        Sctrl_mem_state <= READ_DATA;
    end if;
  
```



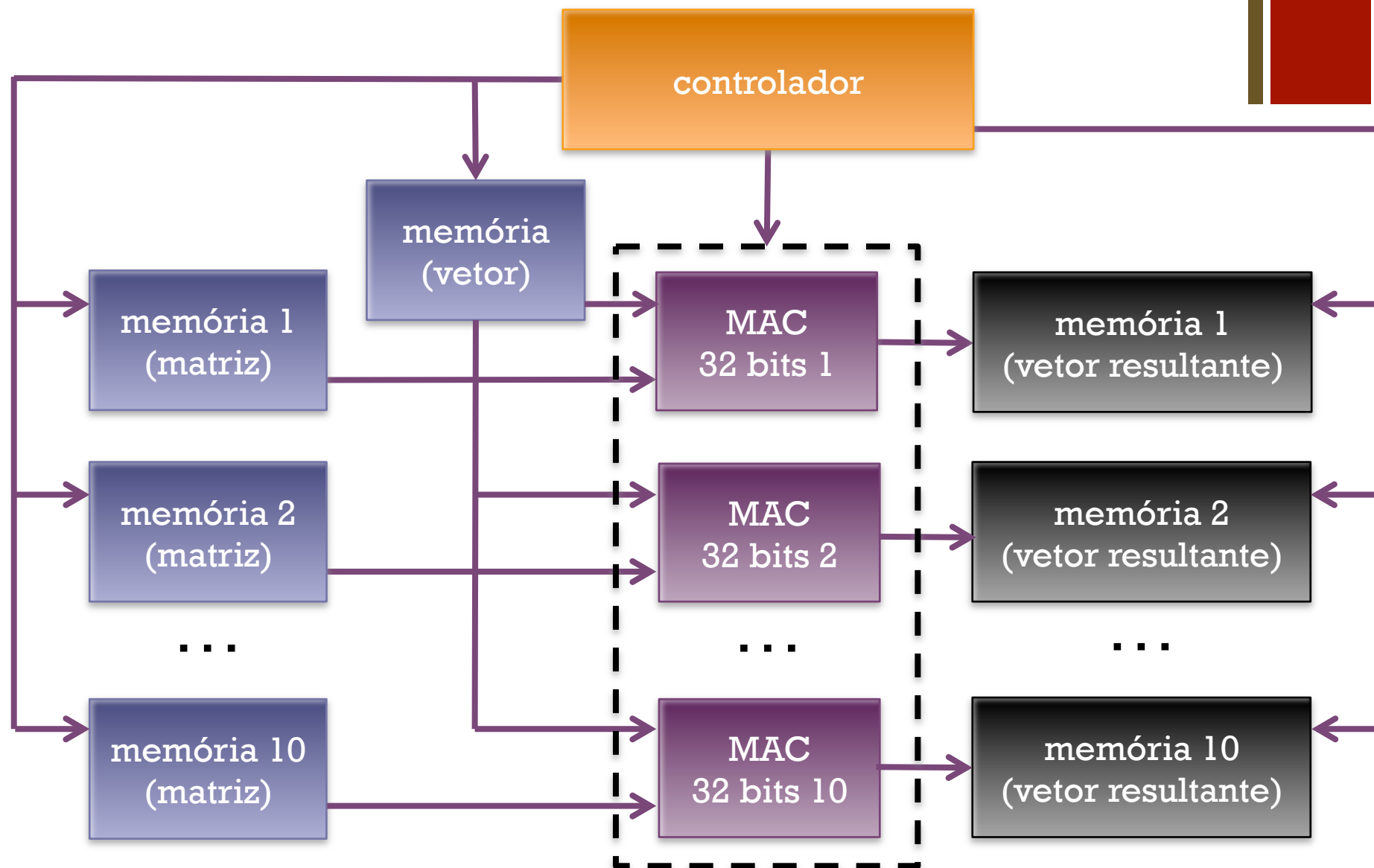
```
when END_PROCESS =>
    -- operacao finalizada, se mantem no estado
    Sctrl_mem_state <= END_PROCESS;
```

+ Discussão!

- E agora? Como podemos aumentar consideravelmente o desempenho dessa solução?

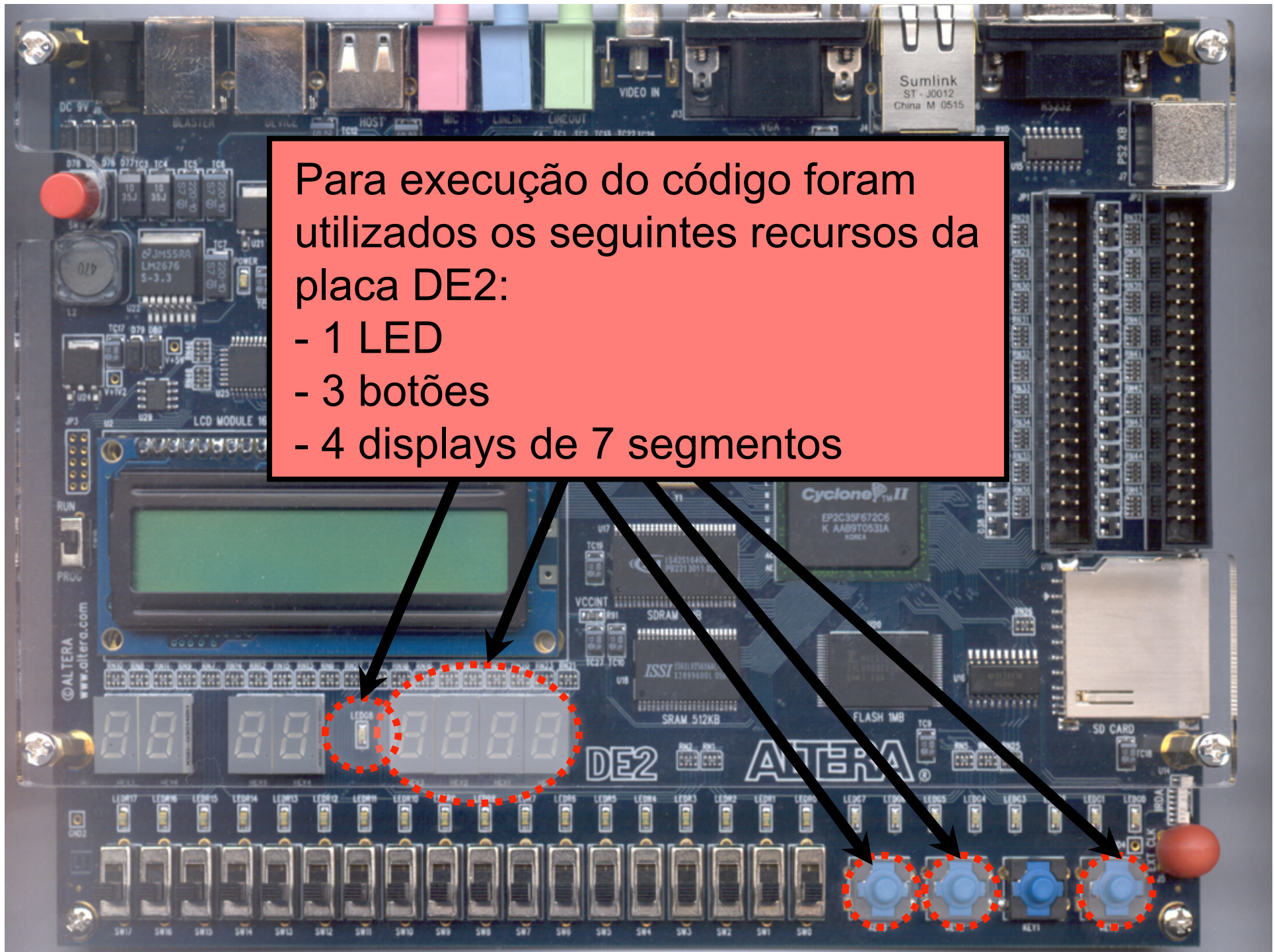


+ Arquitetura da solução proposta



Para execução do código foram utilizados os seguintes recursos da placa DE2:

- 1 LED
- 3 botões
- 4 displays de 7 segmentos



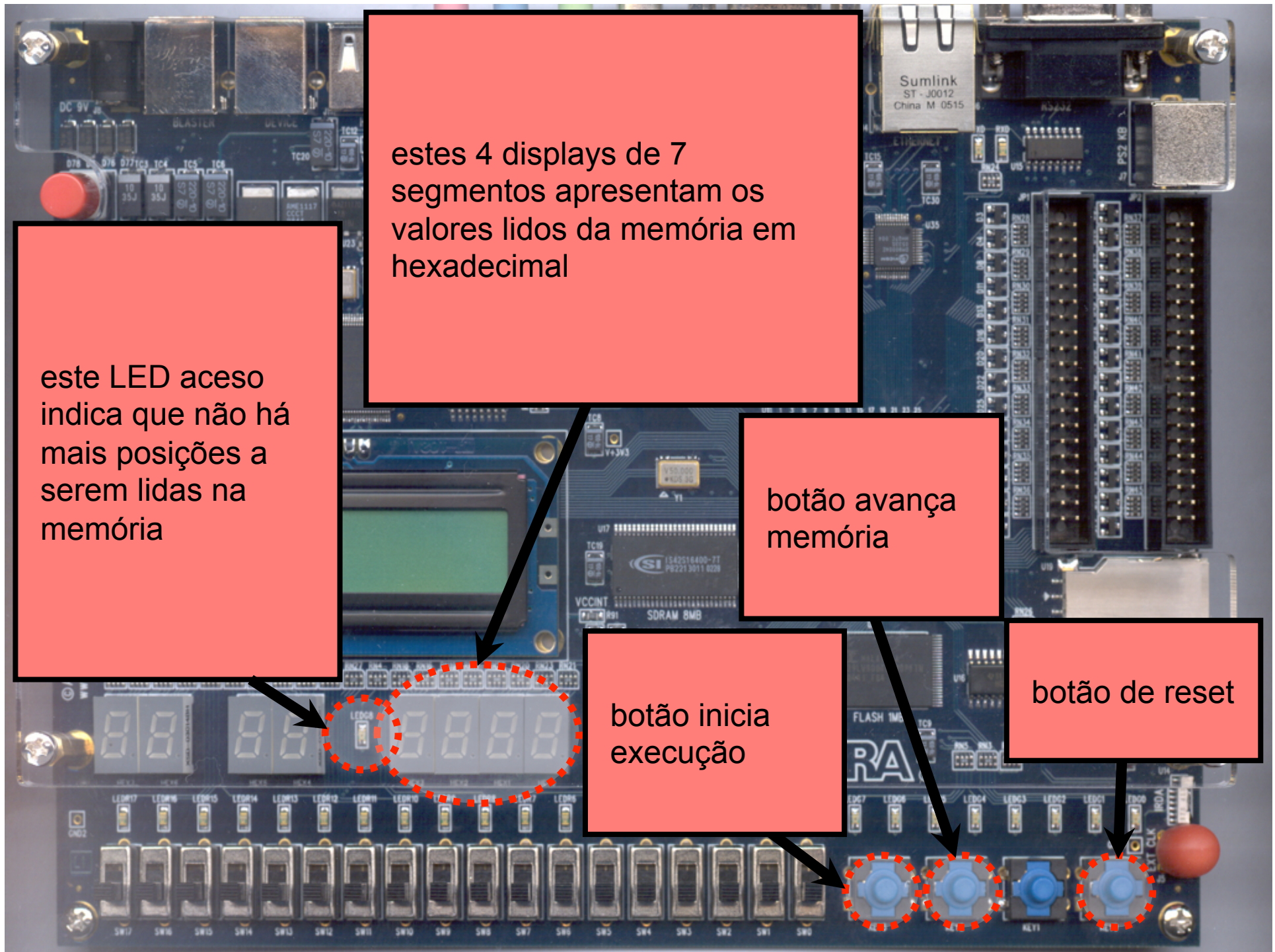
este LED aceso indica que não há mais posições a serem lidas na memória

estes 4 displays de 7 segmentos apresentam os valores lidos da memória em hexadecimal

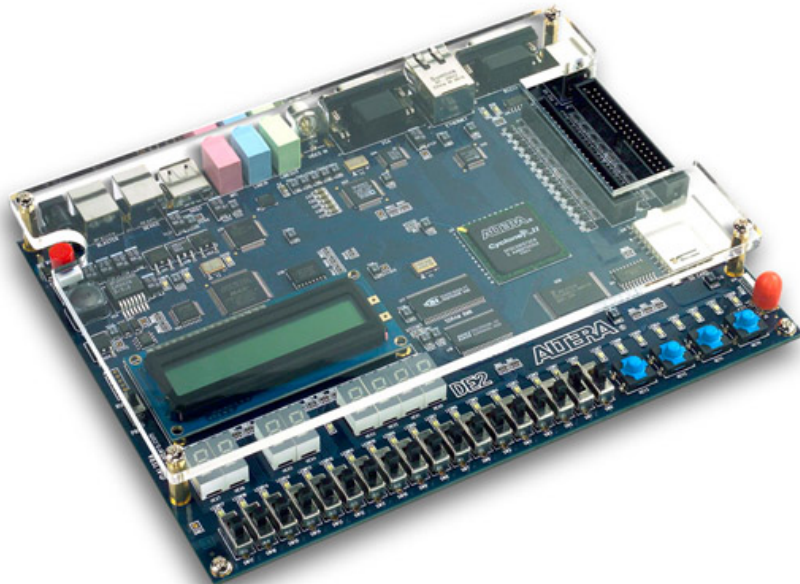
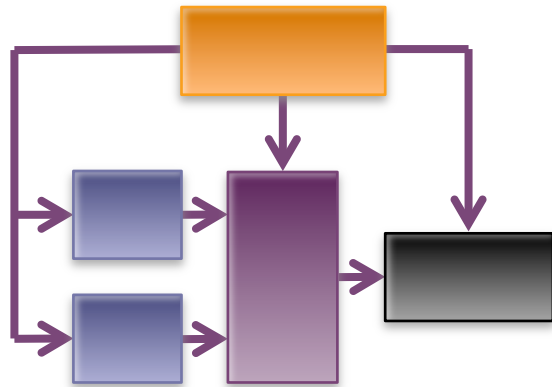
botão avança memória

botão inicia execução

botão de reset



+ Módulos de apoio para execução na placa DE2

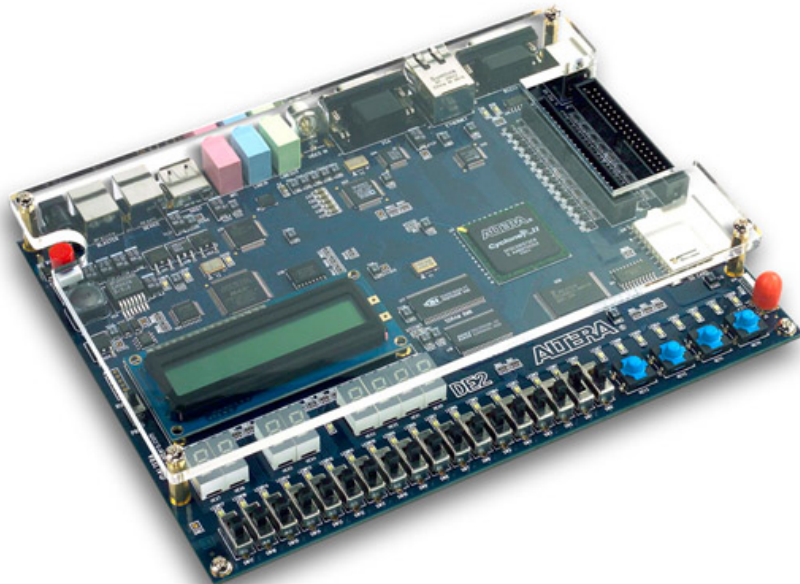
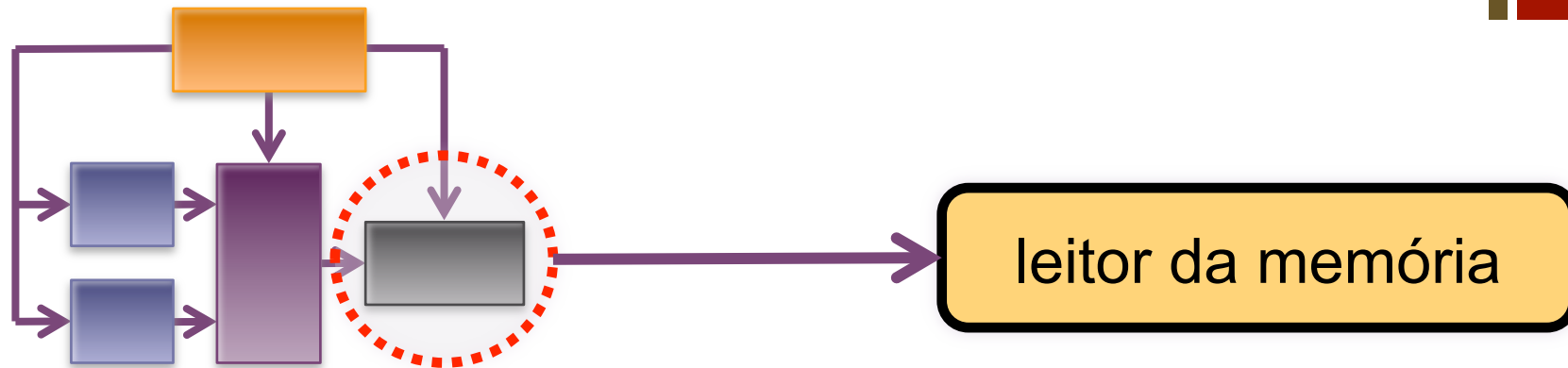


leitor da memória

decodificador BCD

debounce

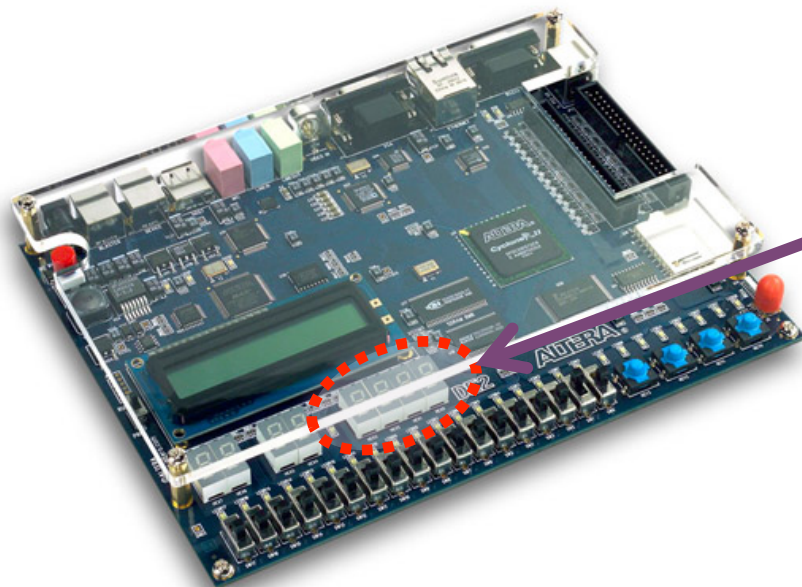
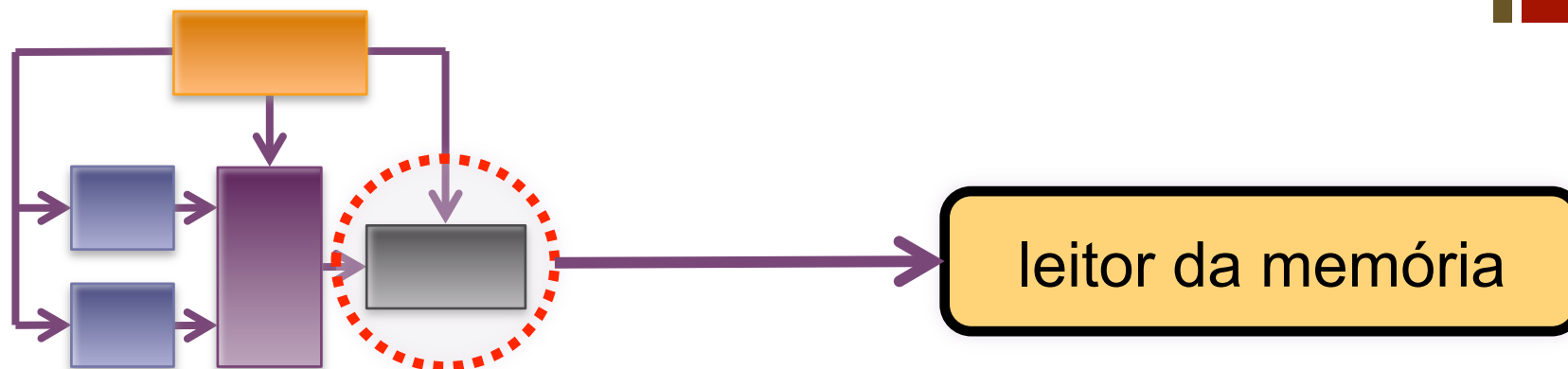
+ Módulos de apoio para execução na placa DE2



decodificador BCD

debounce

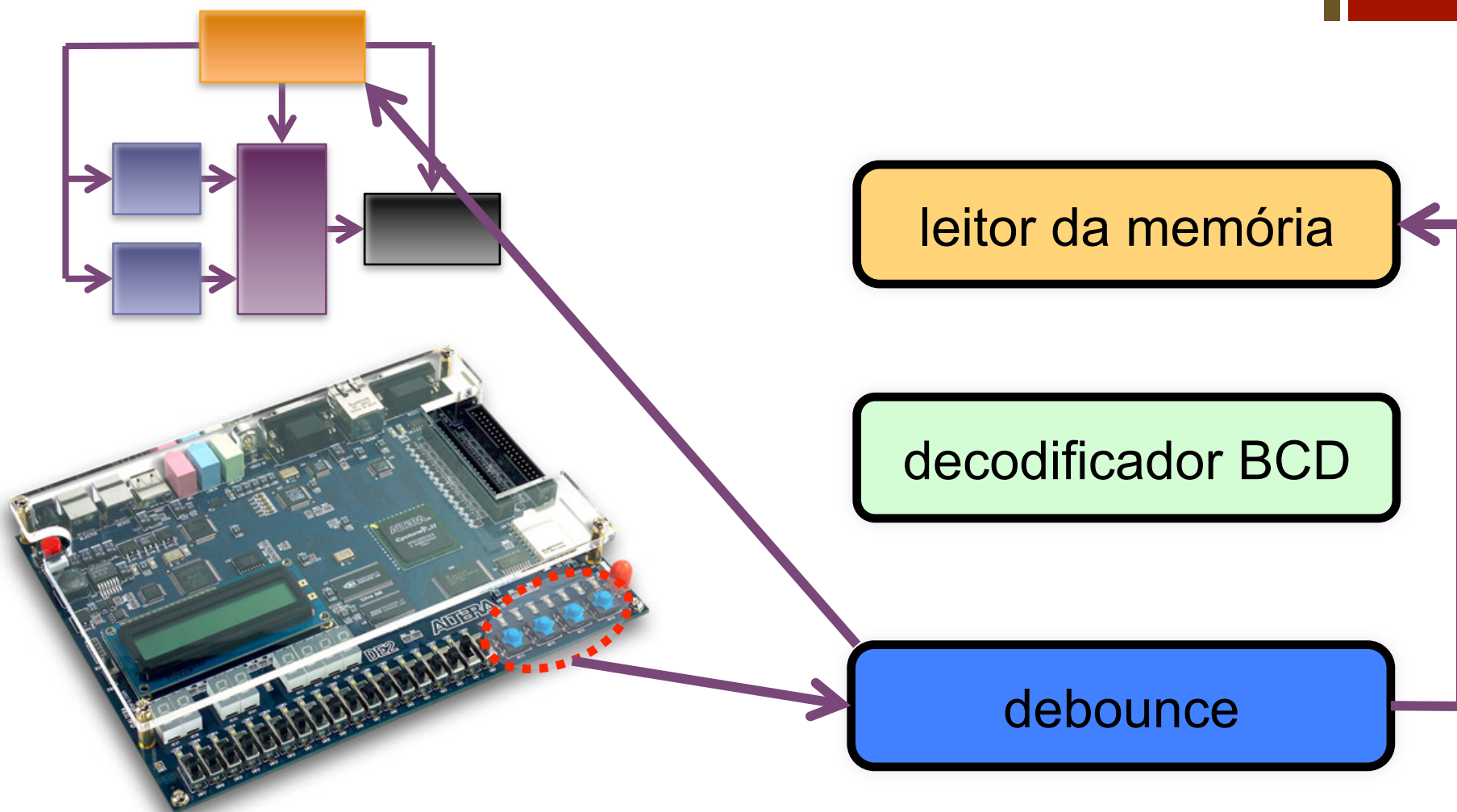
+ Módulos de apoio para execução na placa DE2



decodificador BCD

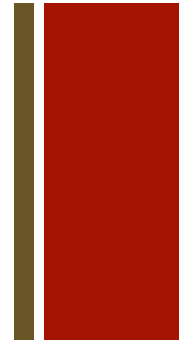
debounce

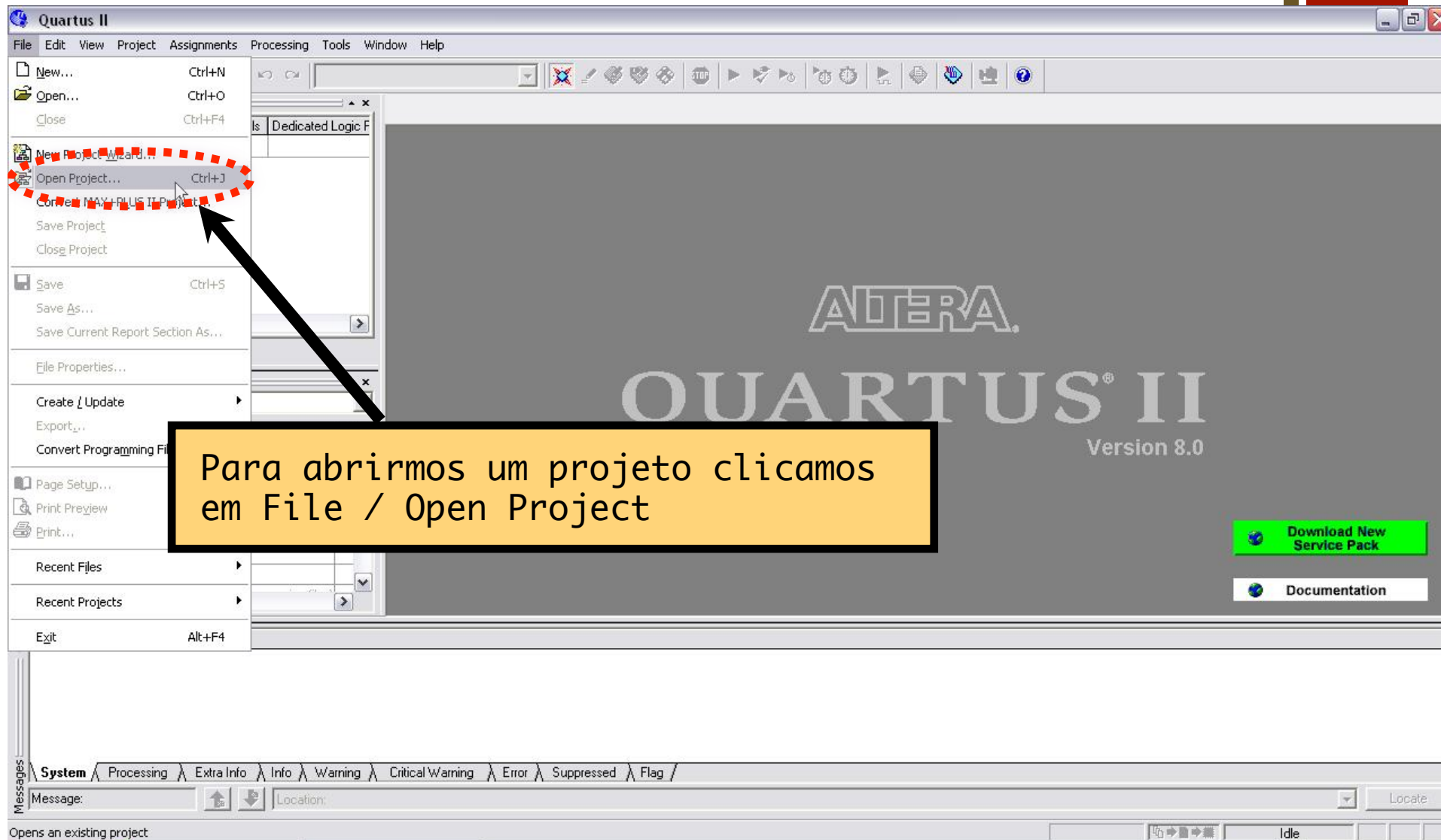
+ Módulos de apoio para execução na placa DE2



+ Simulação

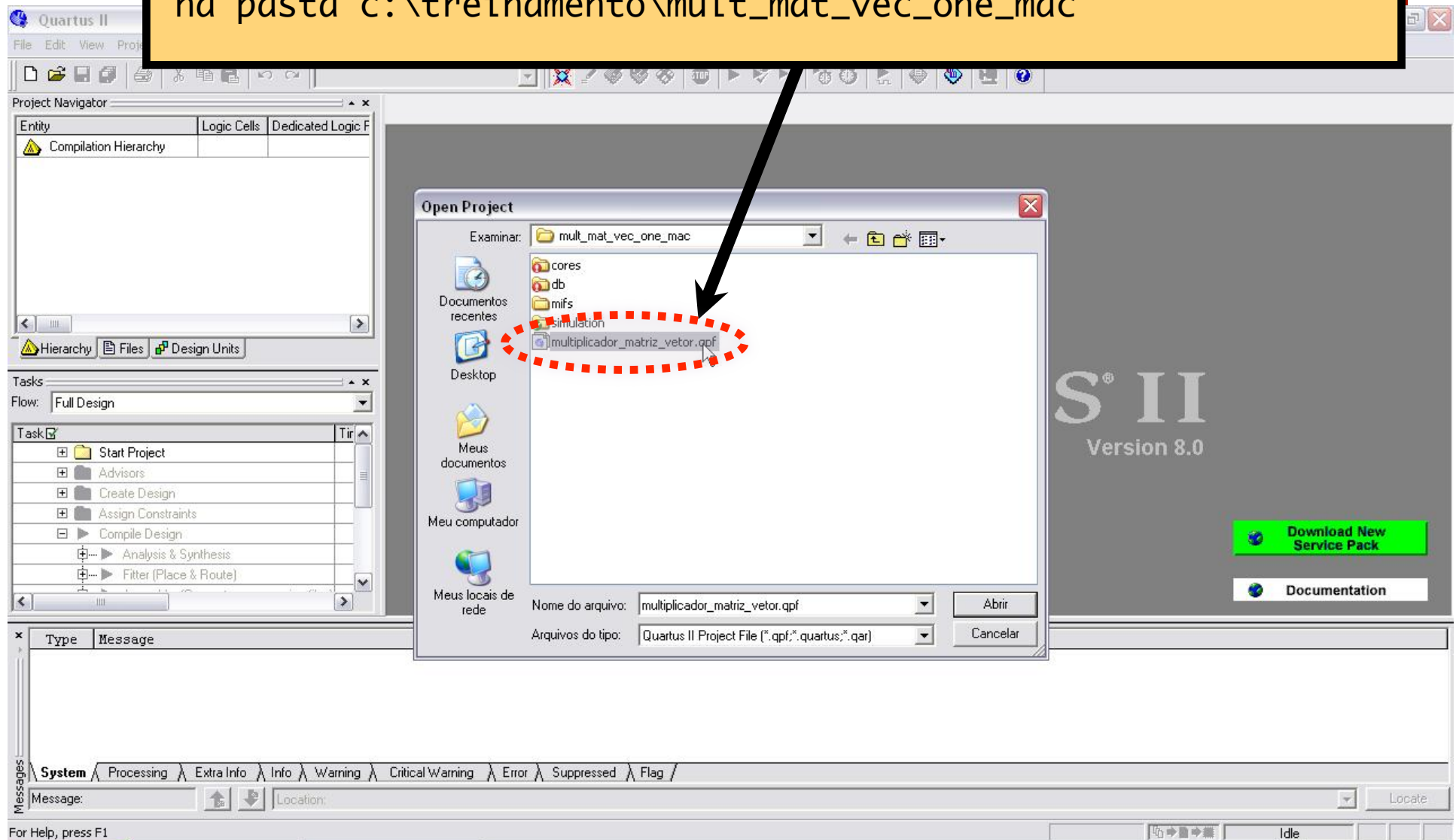
- Antes de baixarmos nosso código na plataforma alvo, podemos realizar a simulação dos cores desenvolvidos em uma ferramenta de simulação. Esta simulação nos permite identificar de antemão se o módulo realiza corretamente a tarefa para qual foi desenvolvido.





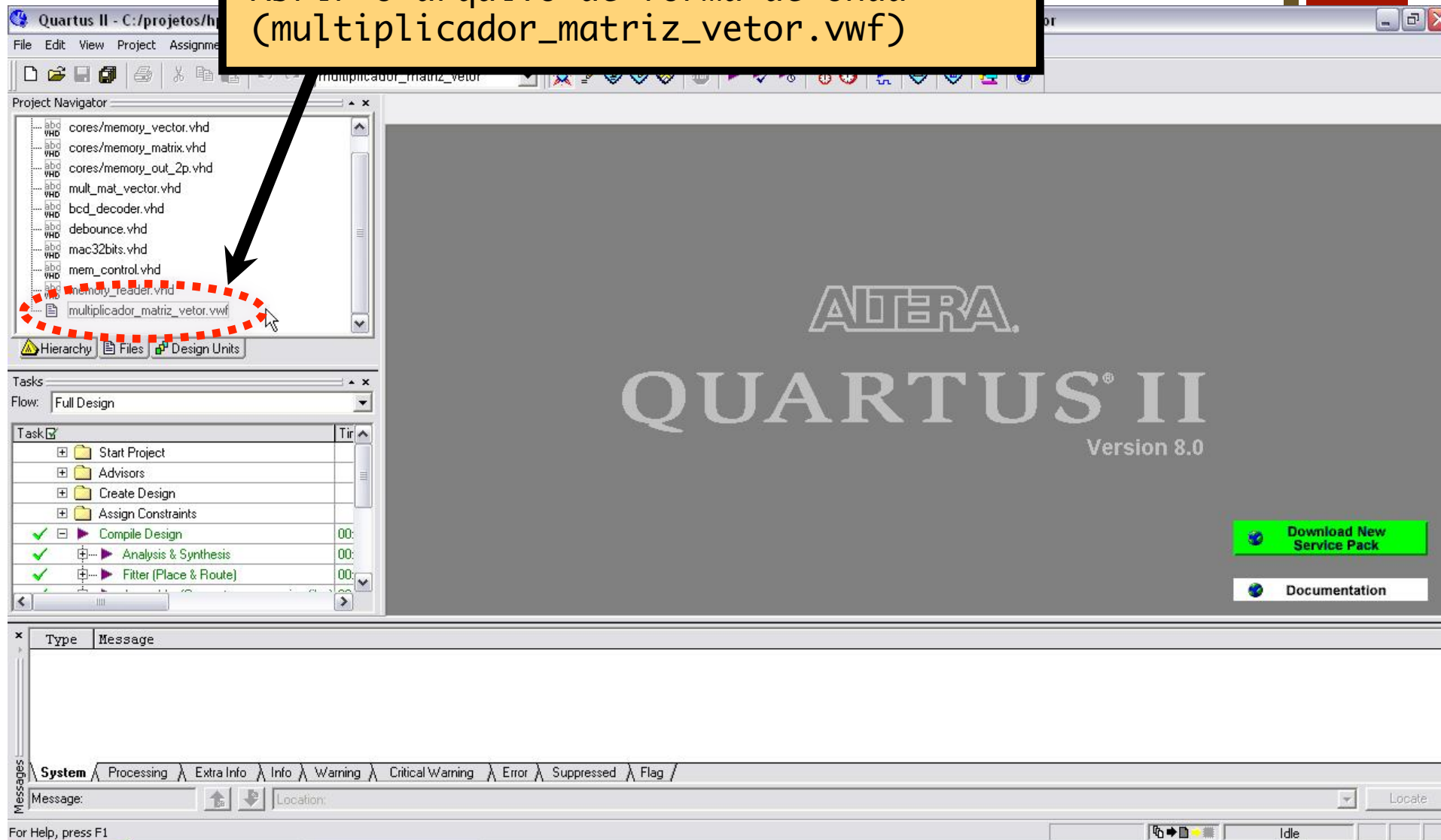


Abrir o projeto multiplicador_matriz_vetor.qpf localizado na pasta c:\treinamento\mult_mat_vec_one_mac





Abrir o arquivo de forma de onda
(multiplicador_matriz_vetor.vwf)





Para melhorar a visualização
clique neste ícone (full-screen)

Quartus II - C:/projetos/hpcin/treinamento(quartus8)/projetos/m

File Edit View Project Assignments Processing Tools Window Help

multiplicador_matriz_vetor

Project Navigator

- cores/memory_vector.vhd
- cores/memory_matrix.vhd
- cores/memory_out_2p.vhd
- mult_mat_vector.vhd
- bcd_decoder.vhd
- debounce.vhd
- mac32bits.vhd
- mem_control.vhd
- memory_reader.vhd
- multiplicador_matriz_vetor.vwf

Hierarchy Files Design Units

Tasks

Flow: Full Design

Task

- Start Project
- Advisors
- Create Design
- Assign Constraints
- Compile Design 00:
- Analysis & Synthesis 00:
- Filter (Place & Route) 00:

multiplicador_matriz_vetor.vwf

Master Time Bar: 0 ps Pointer: 7.58 ns Interval: 7.58 ns Start: End:

Name	0 ps	80,0 ns	160,0 ns	240,0 ns	320,0 ns	400,0 ns	480,0 ns	560,0 ns	640,0 ns	720,0 ns
0 Pclk										
1 Prst										
2 ...it_button										
3 ...ry_button										
4 ...mac_out B										
37 Pdisplay0										
45 Pdisplay1										
53 Pdisplay2										
61 Pdisplay3										
69 Pdone										
70 Pend_led										
71 Pmac_vld										
72 ...trllPdone										
73 ...cIPoutput										

Messages

Type Message

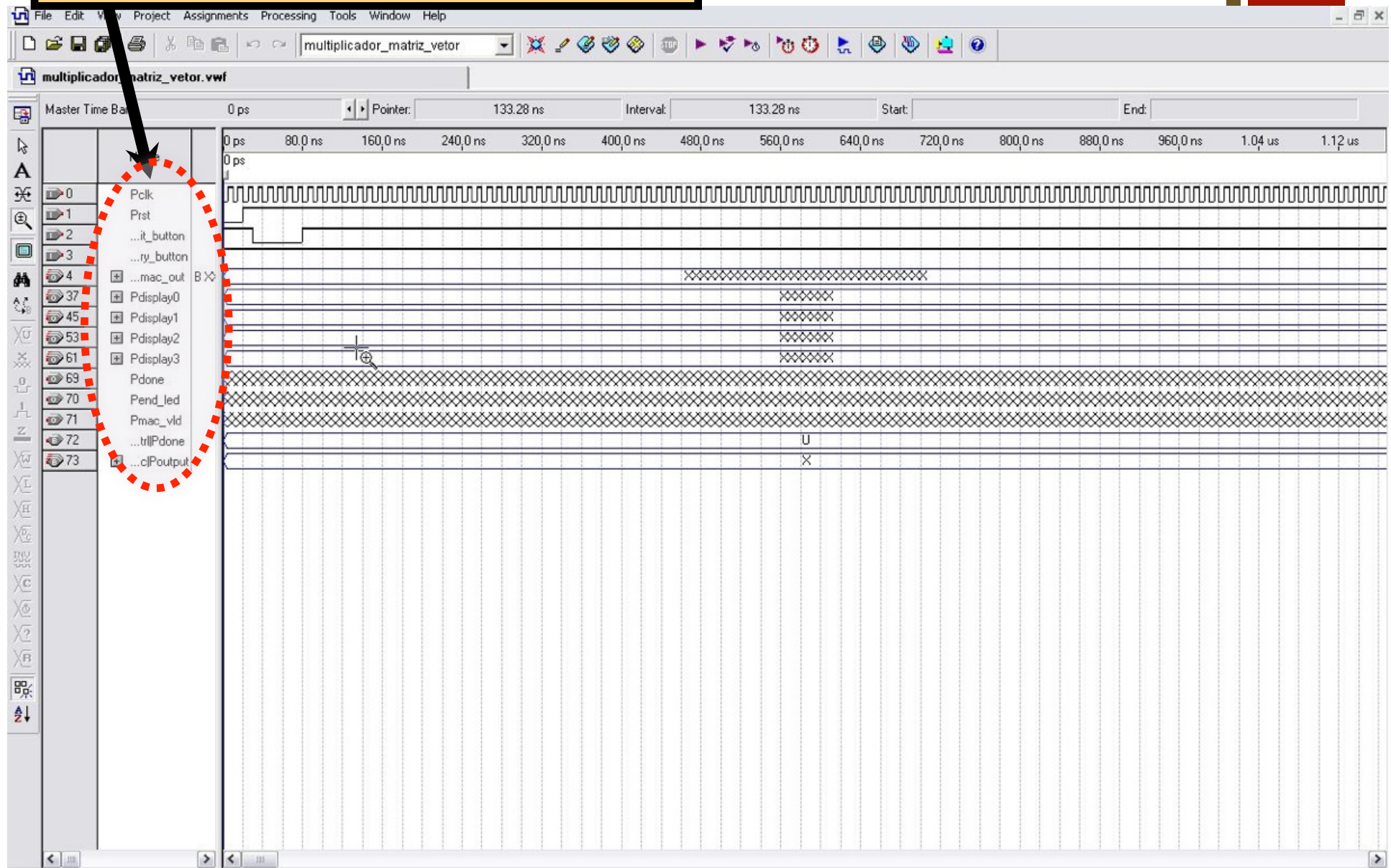
System Processing Extra Info Info Warning Critical Warning Error Suppressed Flag

Message: Location: Locate

Turns Full Screen mode on and off

Idle

sinais utilizados na simulação

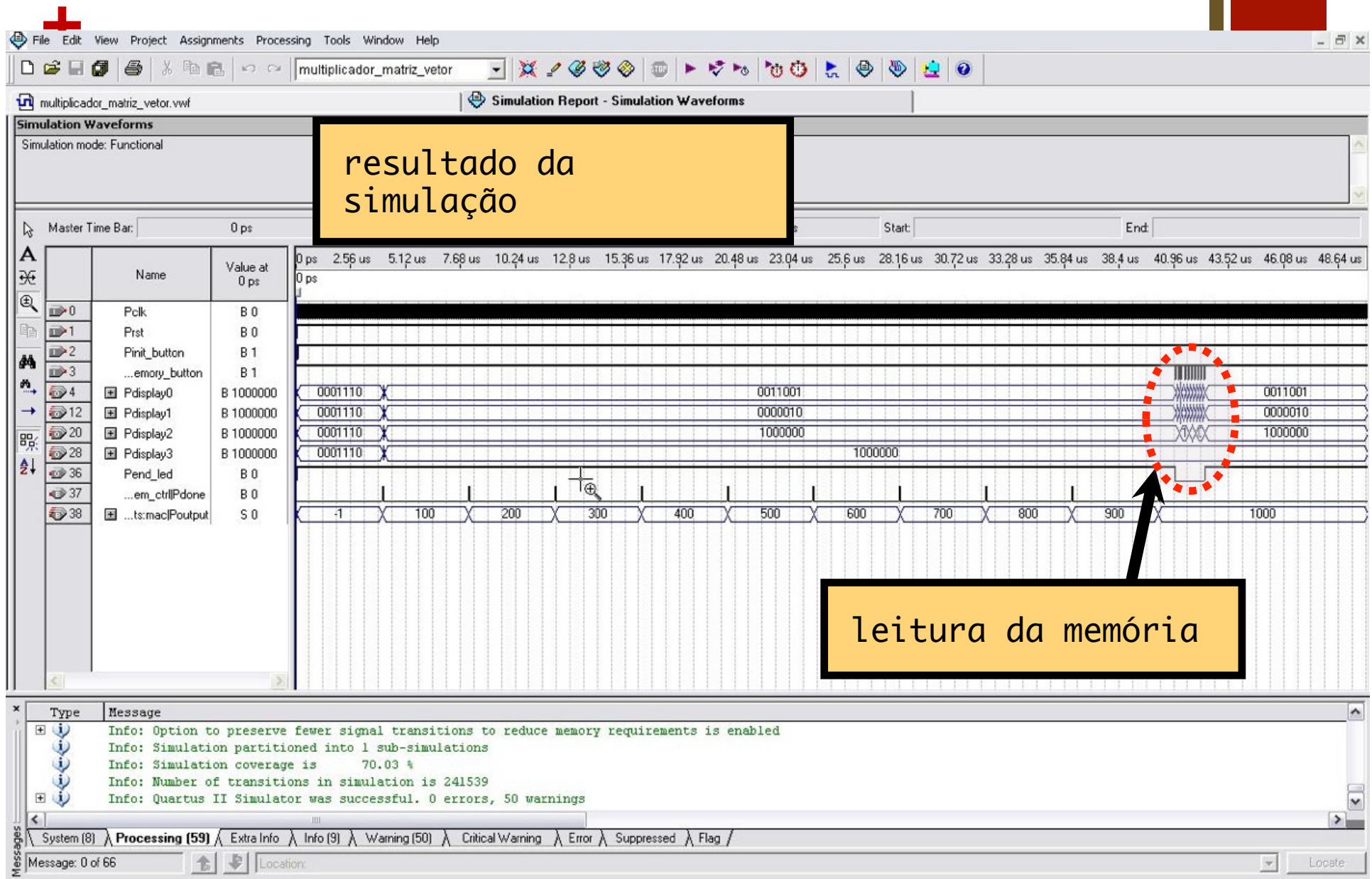


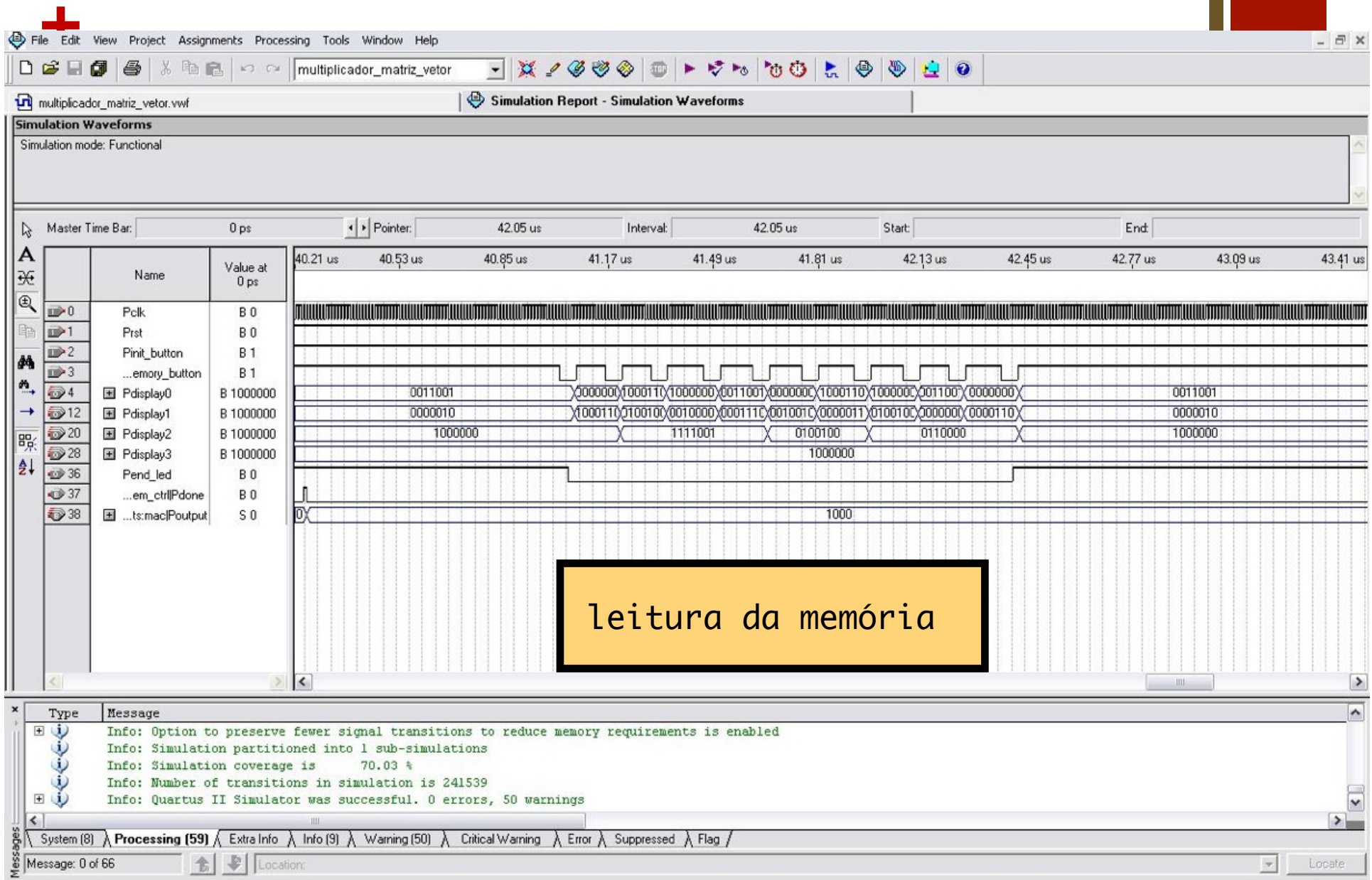
The screenshot displays the Intel Quartus II IDE interface. The 'Processing' menu is open, and the 'Generate Functional Simulation Netlist' option is highlighted with a red dashed box. An arrow points from a yellow text box to this option. The text box contains the following text:

é necessário gerar o modelo de simulação das memórias utilizadas no módulo antes de simulá-lo.

The Messages window at the bottom shows the following error message:

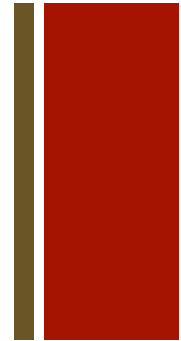
```
Info: *****
Info: Running Quartus II Simulator
Info: Command: quartus_sim --read_settings_files=on --write_settings_files=off multiplicador_matriz_vetor -c multiplicador_matriz_vetor
Error: Run Generate Functional Simulation Netlist (quartus_map multiplicador_matriz_vetor --generate_functional_sim_netlist) to generate functional simulation netlist for
Error: Quartus II Simulator was unsuccessful. 1 error, 0 warnings
```





+ Execução no FPGA

- Como fazemos para baixar nosso projeto para a placa?





Quartus II - C:/projetos/hpcin/treinamento(quartus8)/projetos/mult_mat_vec_one_mac/multiplicador_matriz_vetor - multiplicador_matriz_vetor - [Compilation Repo]

File Edit View Project Assignments Processing Tools Window Help

Project Navigator

Entity	Logic Cells	Dedicated Lo
Cyclone II: EP2C35F672C6		
mult_mat_vector	134 (1)	82 (0)

Tasks

Flow: Full Design

Task	Time
Start Project	
Advisors	
Create Design	
Assign Constraints	
Compile Design	00:
Analysis & Synthesis	00:
Filter (Place & Route)	00:

Compilation Report - Flow Summary

Flow Summary

Flow Status	Successful - Mon Aug 04 16:38:41 2008
Quartus II Version	8.0 Build 215 05/29/2008 SJ Full Version
Revision Name	multiplicador_matriz_vetor
Top-level Entity Name	mult_mat_vector
Device	Cyclone II EP2C35F672C6
Logic Cells	134 (< 1 %)
Block RAMs	216 (< 1 %)
Block RAMs	216 (< 1 %)
Block RAMs	216 (< 1 %)
Total pins	33 / 475 (7 %)
Total virtual pins	0
Total memory bits	17,760 / 483,840 (4 %)
Embedded Multiplier 9-bit elements	2 / 70 (3 %)
Total PLLs	0 / 4 (0 %)

clique neste botão para iniciar o processo de programação do FPGA

Messages

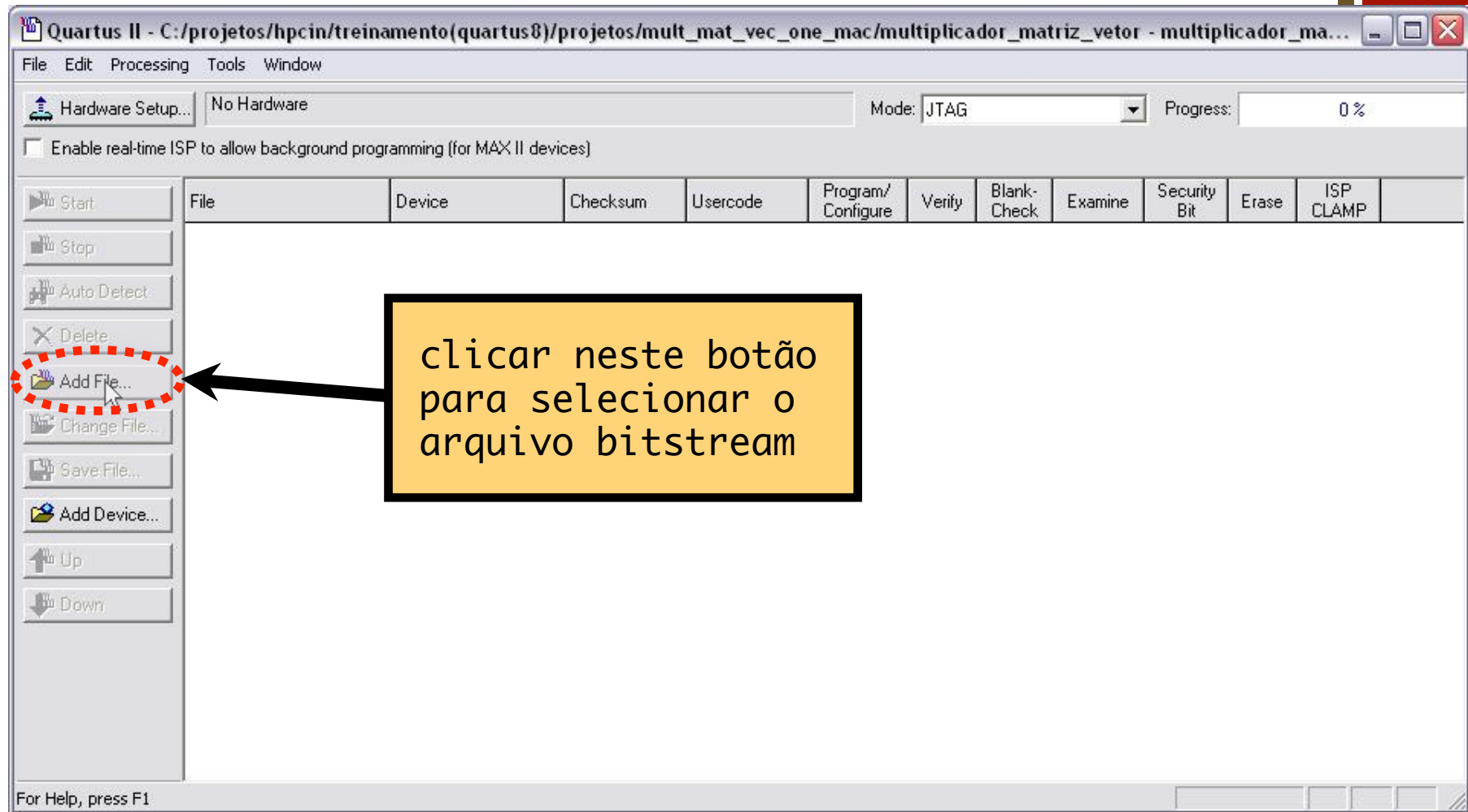
Type	Message
Info	Info: tsu for register "mac32bits:mac Saccumulator[0]" (data pin = "Prst", clock pin = "Pclk") is 6.441 ns
Info	Info: tco from clock "Pclk" to destination pin "Pdisplay0[0]" through memory "memory_out_2p:mem_out altsyncram:altsyncram_component altsyncram_dall:auto_generate
Info	Info: th for register "mac32bits:mac Poutput[9]" (data pin = "Prst", clock pin = "Pclk") is -3.522 ns
Info	Info: Quartus II Classic Timing Analyzer was successful. 0 errors, 1 warning
Info	Info: Quartus II Full Compilation was successful. 0 errors, 55 warnings

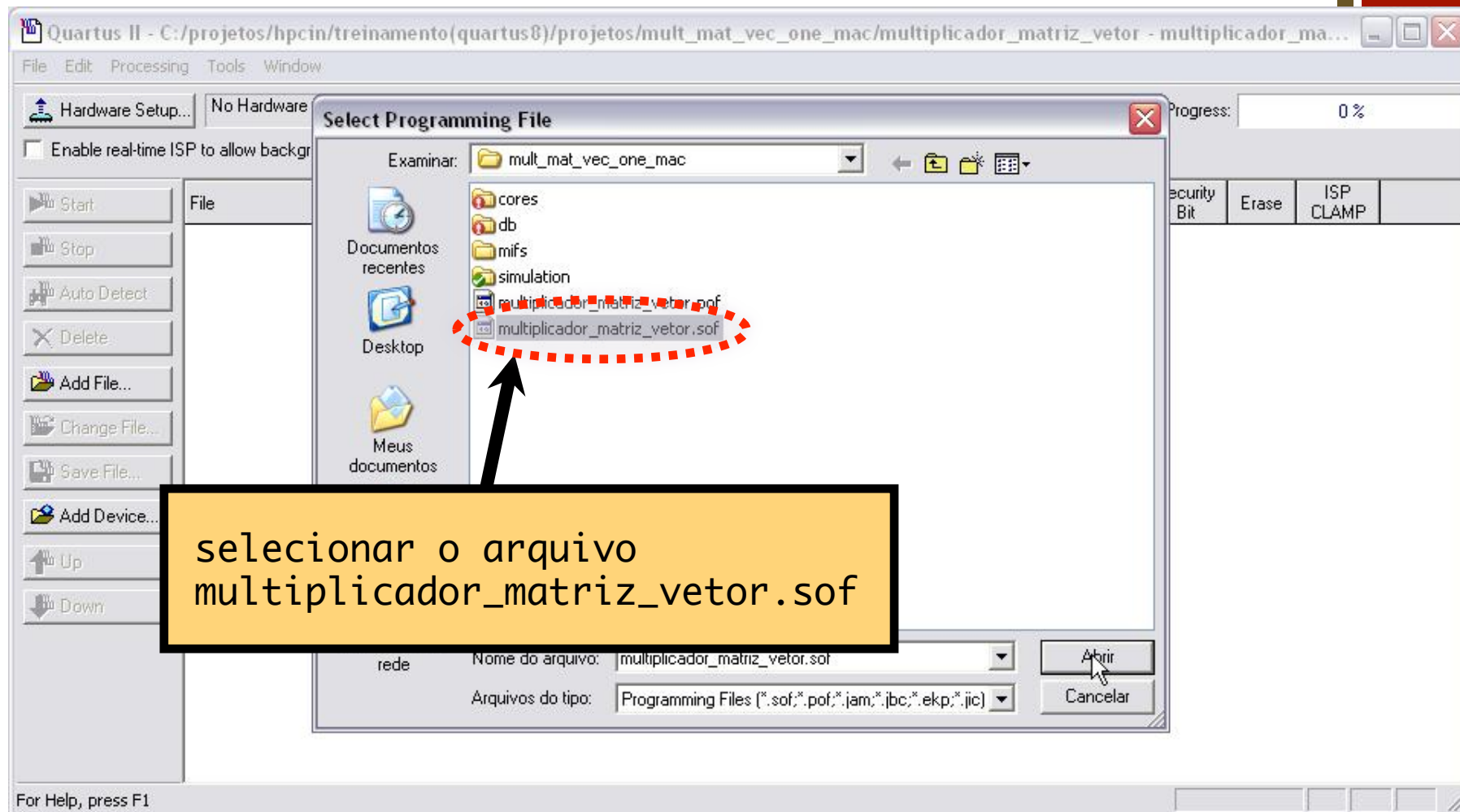
System (2) Processing (96) Extra Info Info (91) Warning (5) Critical Warning Error Suppressed (6) Flag

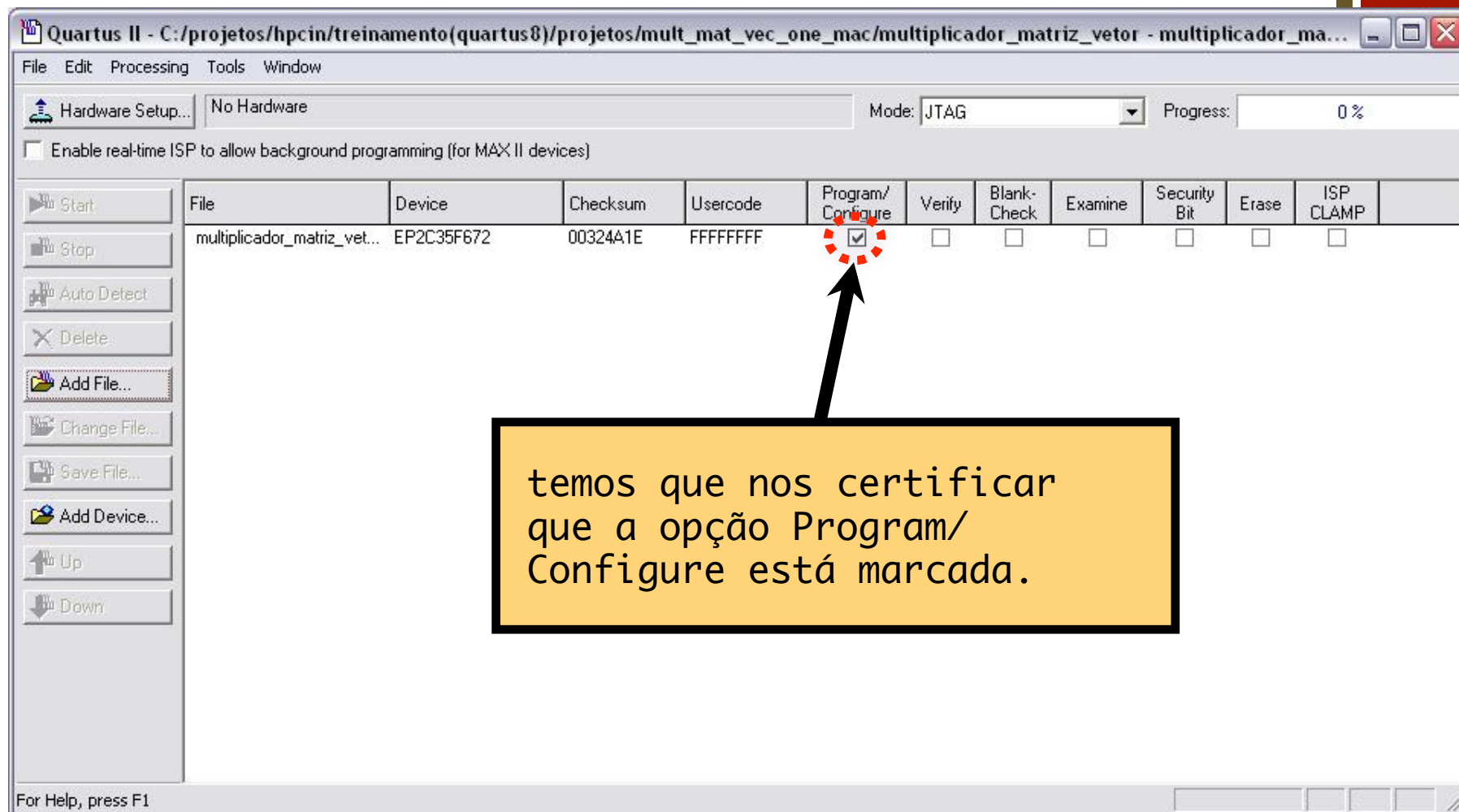
Message: 0 of 475

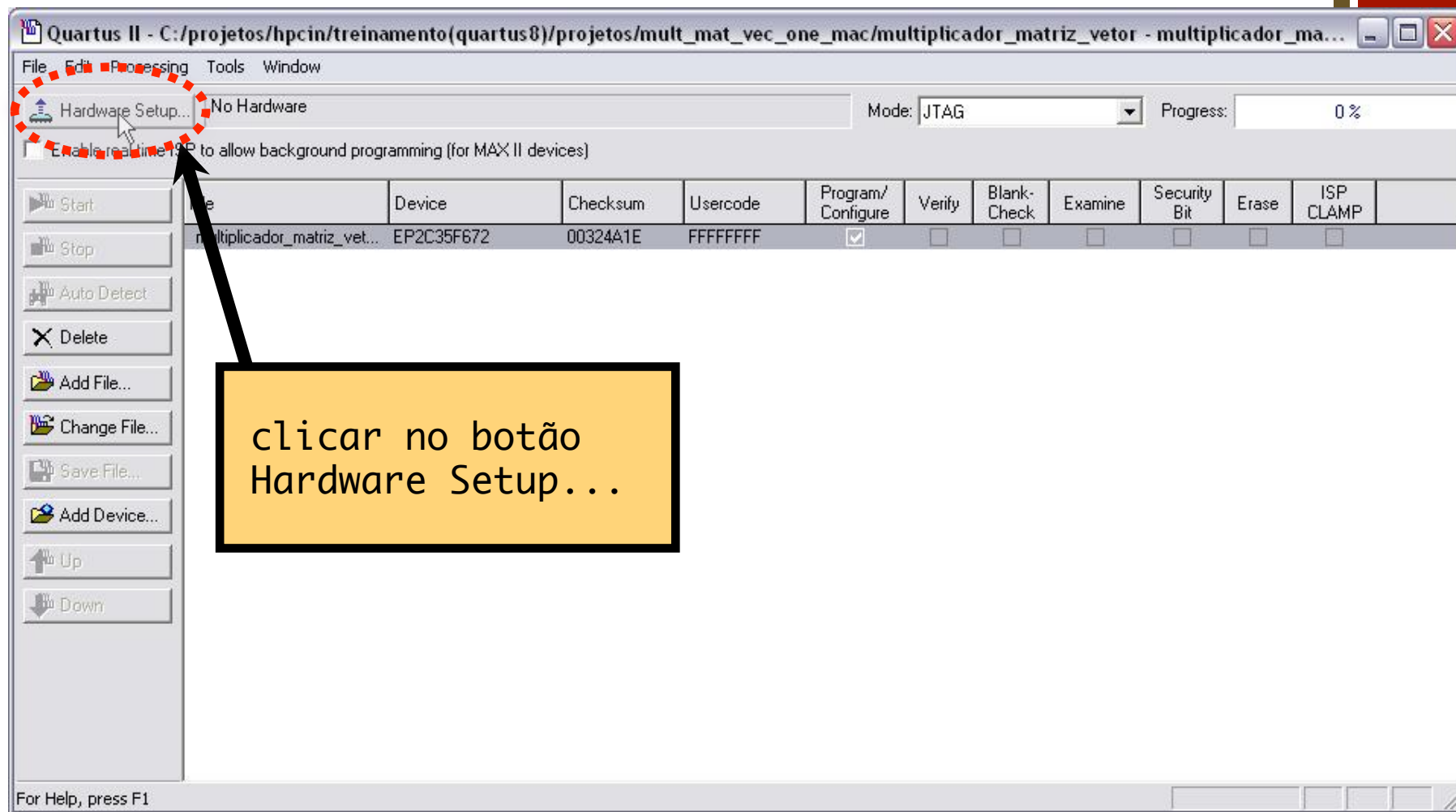
Open a Programmer window

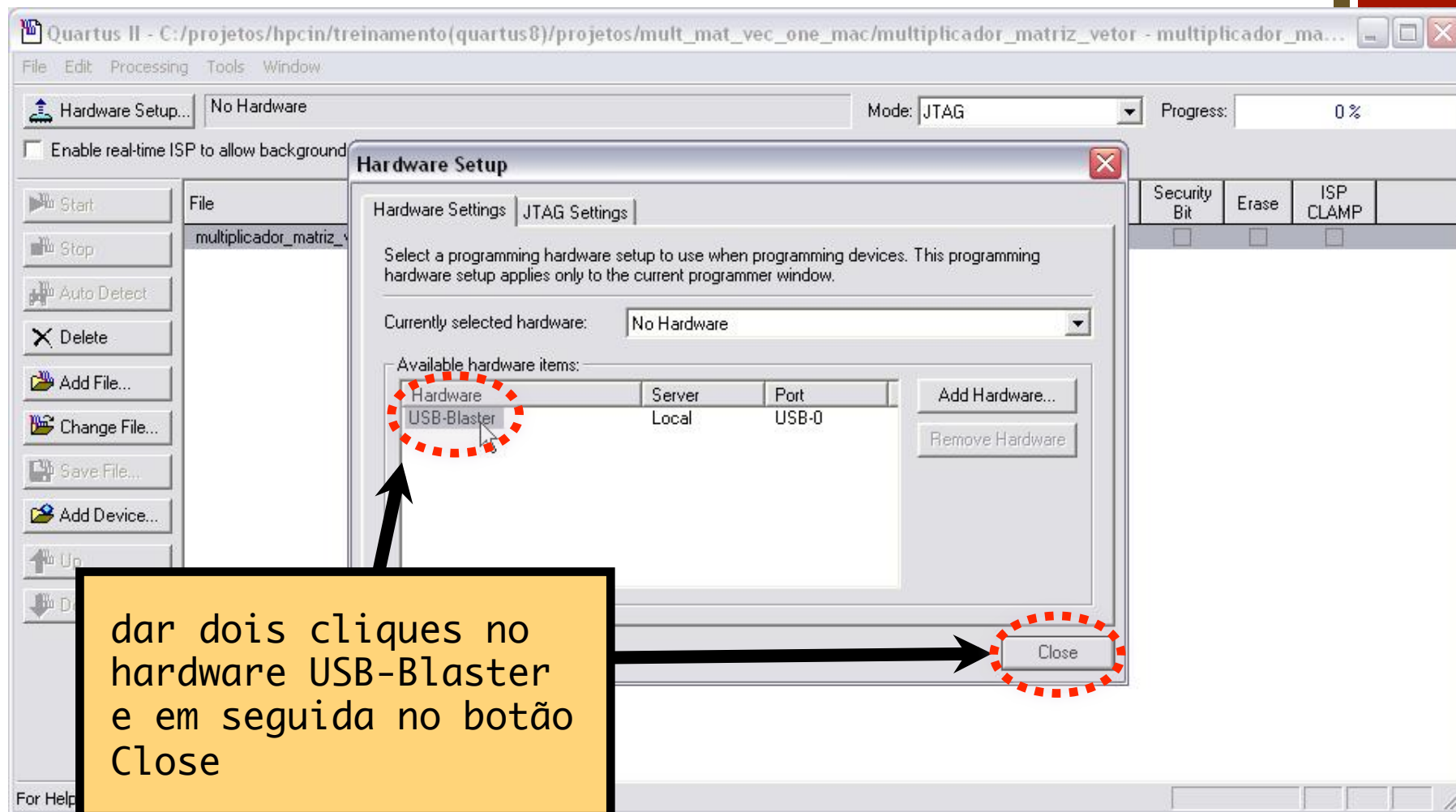
Idle

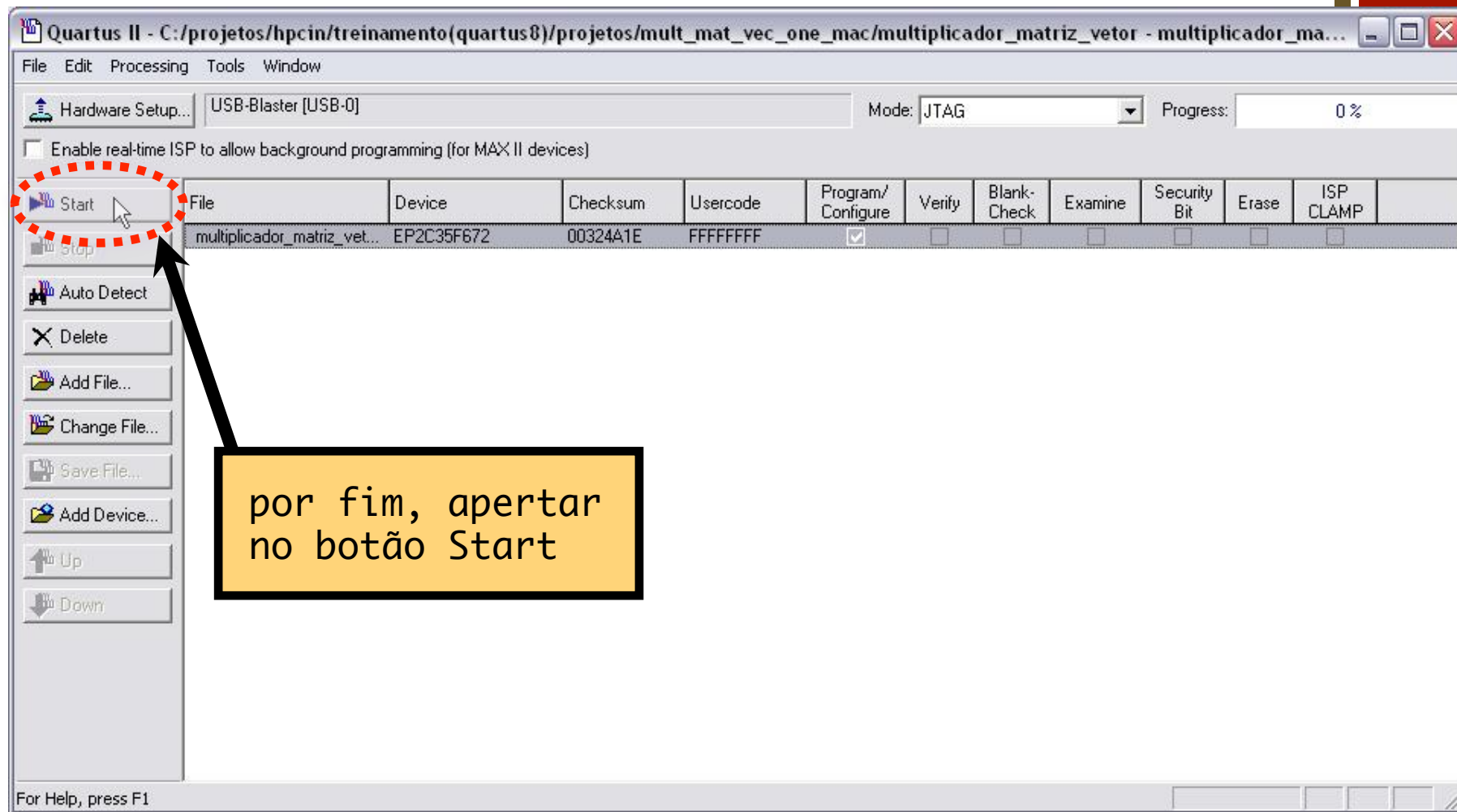












este LED aceso indica que não há mais posições a serem lidas na memória

estes 4 displays de 7 segmentos apresentam os valores lidos da memória em hexadecimal

botão avança memória

botão inicia execução

botão de reset

