

Multiplicação de Matrizes

VHDL, ASM, PO-PC, Pipeline e Basys2

Cindy Evelyn Peterson (219155) e Maurício Izé (273168)

Problema

- **Entrada**

- Duas matrizes 4x4
- 32 números unsigned de 4 bits

- **Saída**

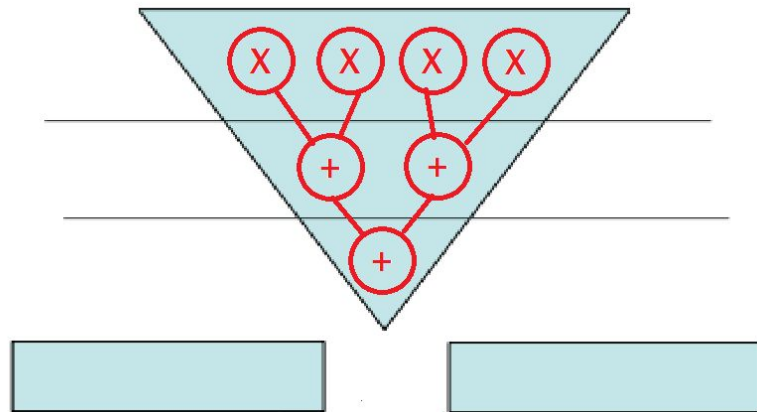
- Uma matriz 4x4 com o resultado da multiplicação das matrizes de entrada
- 16 números unsigned de 10 bits

Implementação

Um pipeline de 3 estágios

8 entradas por ciclo de clock

- **Estágio 1:**
 - 4 multiplicadores de 4 bits
- **Estágio 2:**
 - 2 somadores de 8 bits
- **Estágio 3:**
 - 1 somador de 9 bits



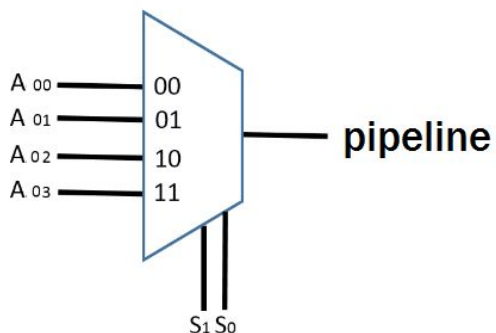
A cada ciclo de clock é calculado um número da matriz resultante (latência = 3 ciclos)

Implementação

Os 32 números das matrizes de entrada foram armazenados em 32 signals de 4 bits.

```
SIGNAL A00 : STD_LOGIC_VECTOR(3 DOWNTO 0)
SIGNAL A01 : STD_LOGIC_VECTOR(3 DOWNTO 0)
SIGNAL A02 : STD_LOGIC_VECTOR(3 DOWNTO 0)
SIGNAL A03 : STD_LOGIC_VECTOR(3 DOWNTO 0)
...
```

E selecionados por 4 multiplexadores 4-1



Métricas

- **21 ciclos de clock**
 - 3 ciclos de latência
 - 16 ciclos para calcular os 16 números da matriz resultante
 - 2 ciclos de gerenciamento (reset, estado final)
- **Report**
 - Minimum period: 7.346ns
 - Maximum Frequency: 136.123MHz
 - Tempo de execução: 154 ns

Métricas

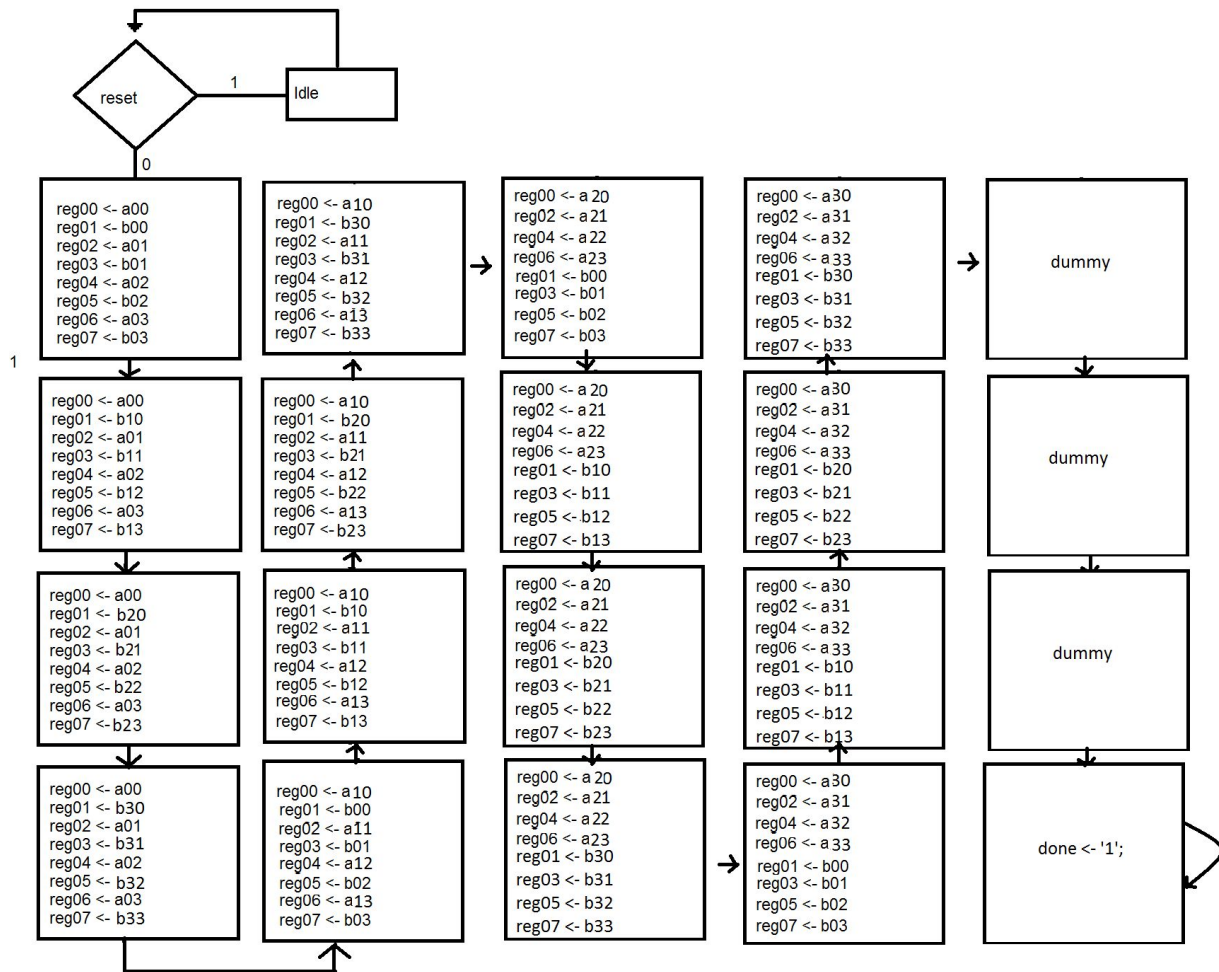
- **Área**

- 4 multiplicadores (resultado em 8 bits)
- 2 somadores (resultado em 9 bits)
- 1 somador (resultado em 10 bits)
- 8 multiplexadores
- 8 registradores 4 bits (estágio de entrada)
- 4 registradores 8 bits (estágio 1)
- 2 registradores 9 bits (estágio 2)
- 1 registrador 10 bits (estágio 3)

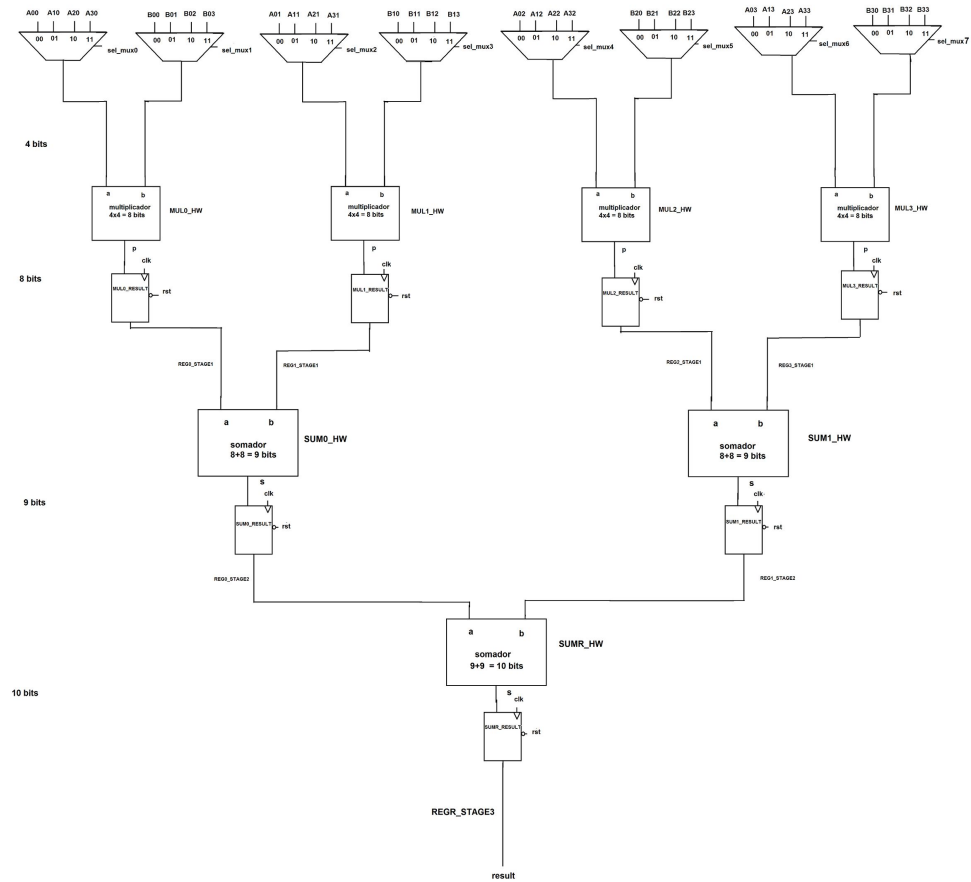
- **Report**

- # Comparators: 12
- 4-bit comparator greater: 12
- Flip-Flops: 127
- 4-bit adder: 12
- # IOs: 33
- # of 4 input LUTs: 204

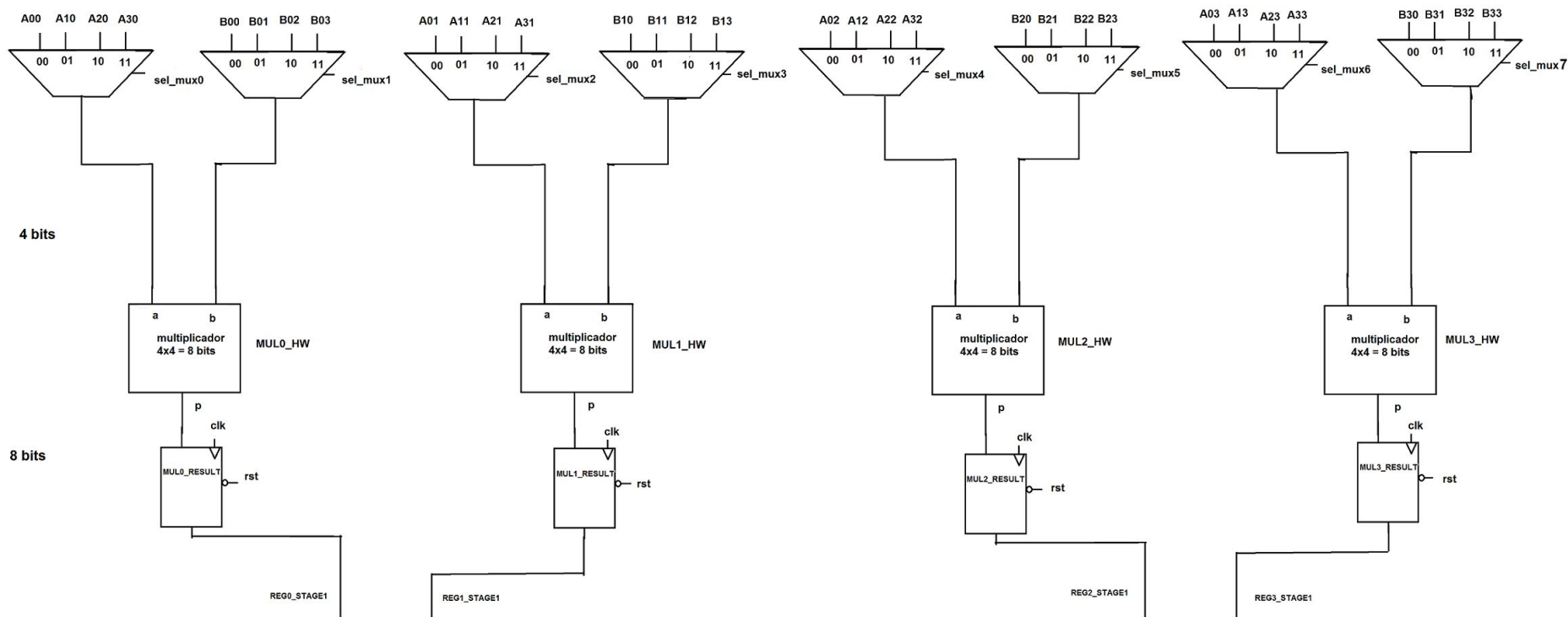
ASM



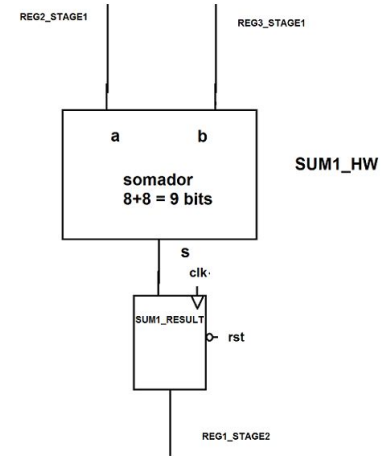
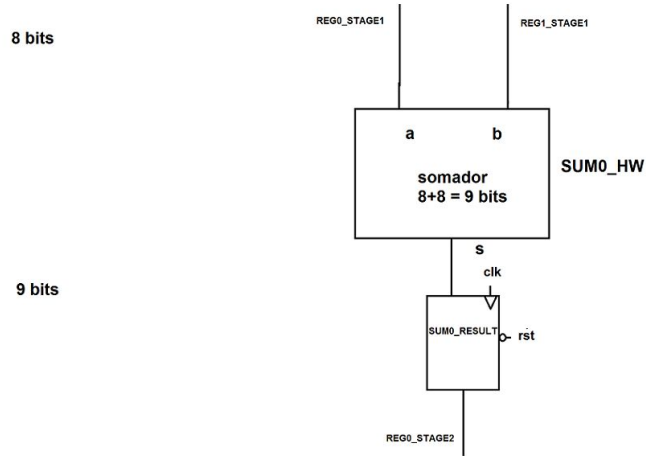
PO



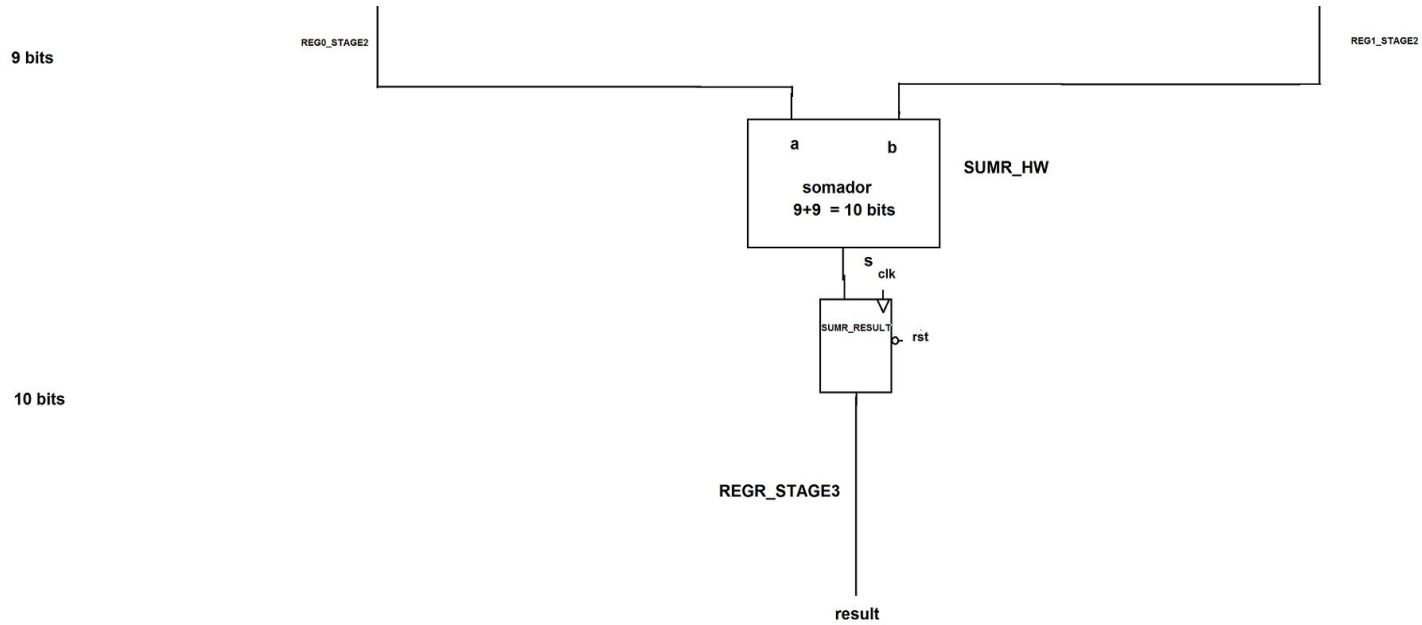
PO - Estágio 1



PO - Estágio 2



PO - Estágio 3

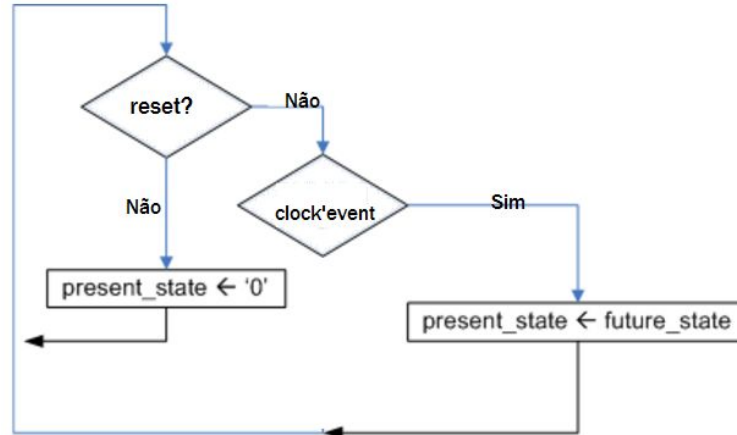


PC

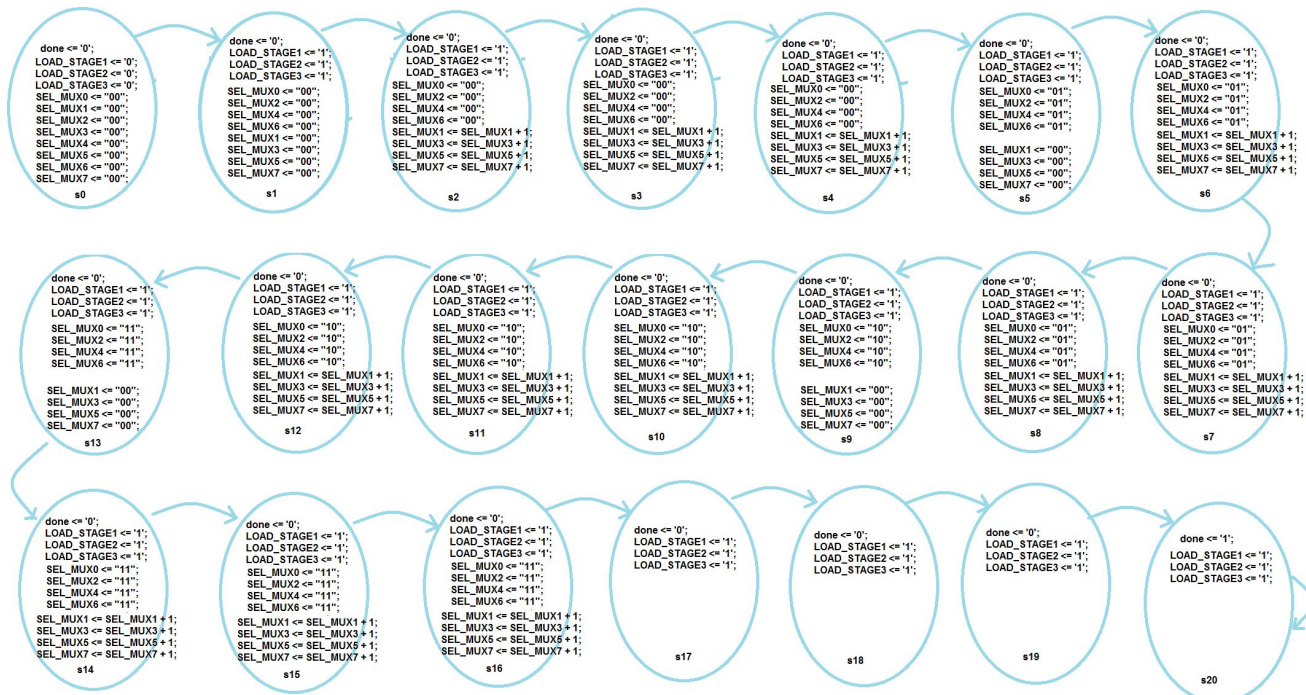
- **FSM com 21 estados**

- 1 estado inicial (reset)
- 16 estados para gerenciar fluxo dos 16 números da matriz resultante (1 número por ciclo)
- 3 estados finais 'dummy' por causa da latência inicial de 3 ciclos do pipeline
- 1 estado para armazenamento do resultado final

- **Controle da FSM**



PC



reset é desativado

1º clock rise após o reset ser desativado, os multiplicadores já possuem o resultado, mas os registradores ainda não deram o load deles

os registradores do primeiro estágio carregam o resultado da multiplicação

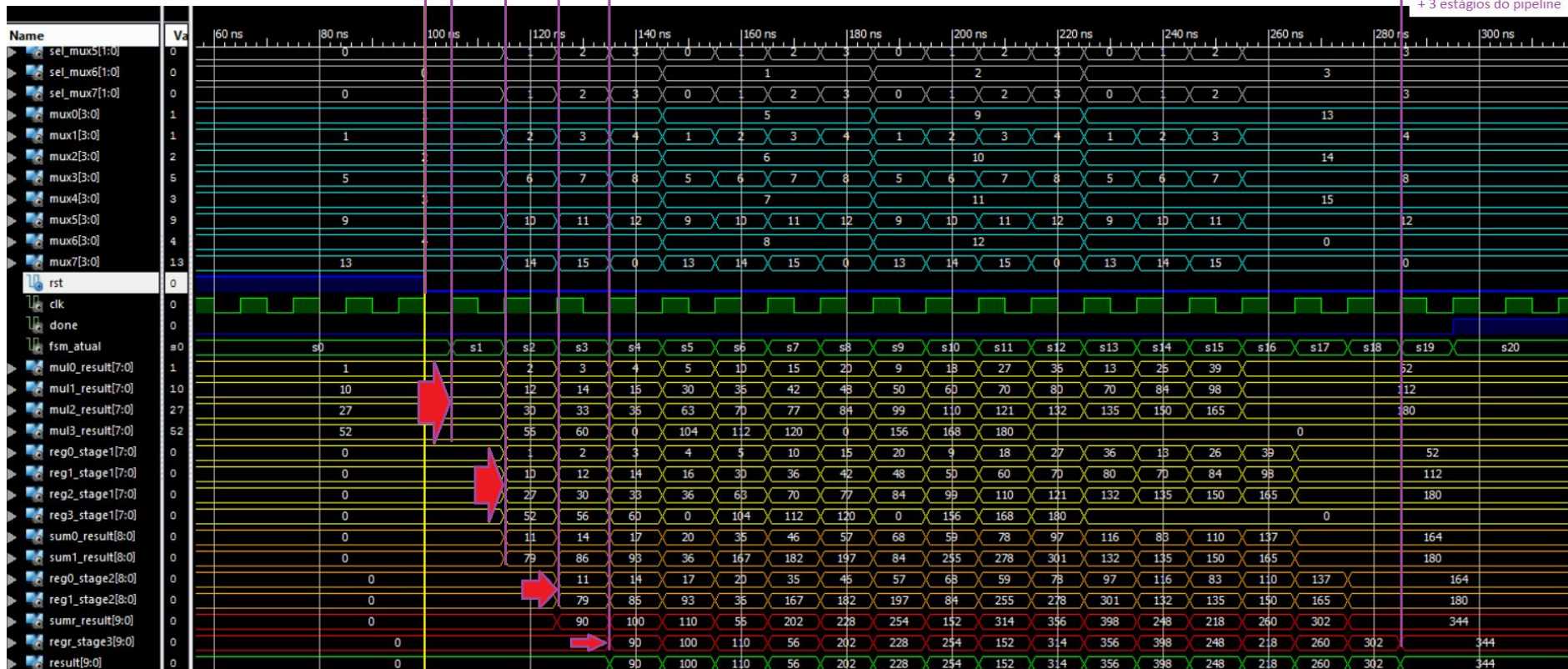
os registradores do segundo estágio carregam o resultado da soma

o registrador do terceiro e último estágio carrega o resultado final do primeiro elemento da matriz resultante

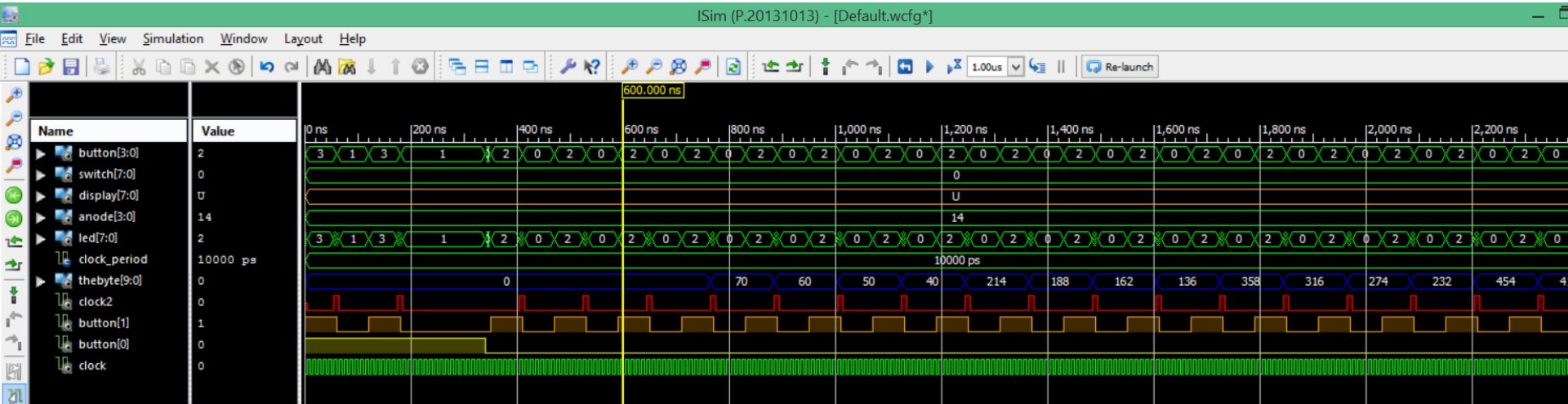
matriz calculada

19 ciclos de clock

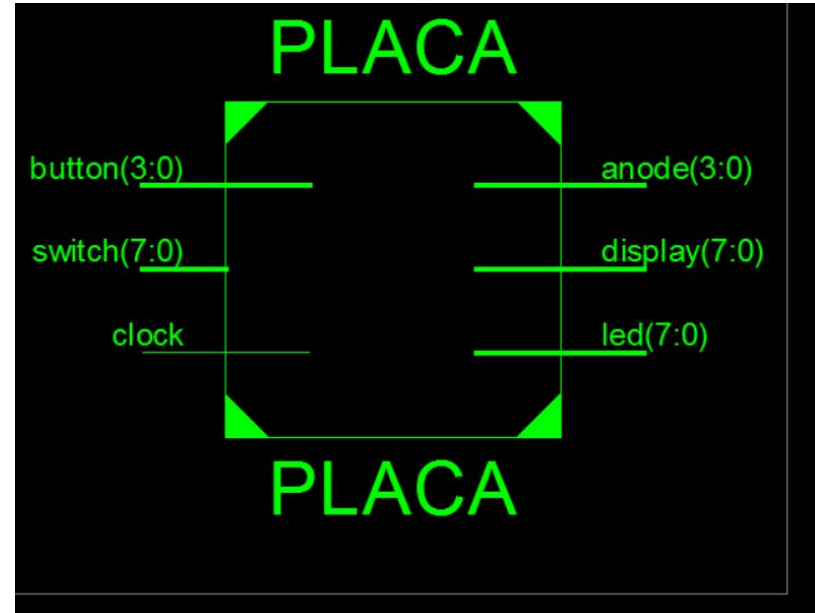
16 elementos da matriz
+ 3 estágios do pipeline



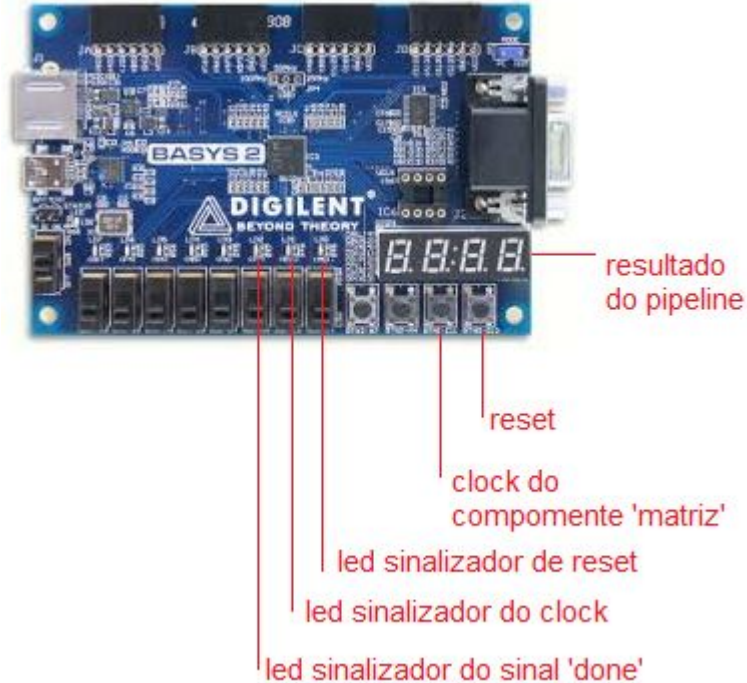
Simulação



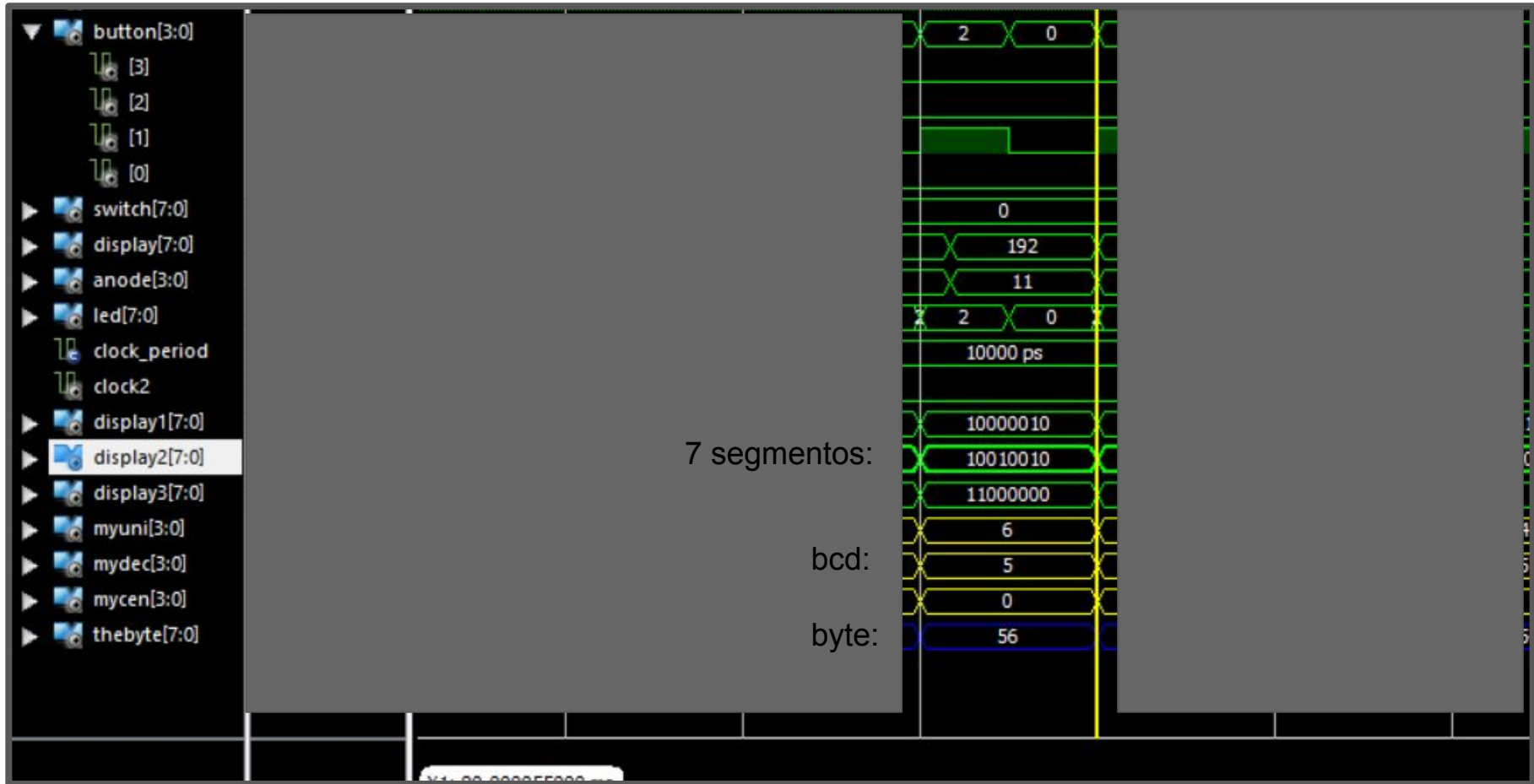
Implementação na Basys2



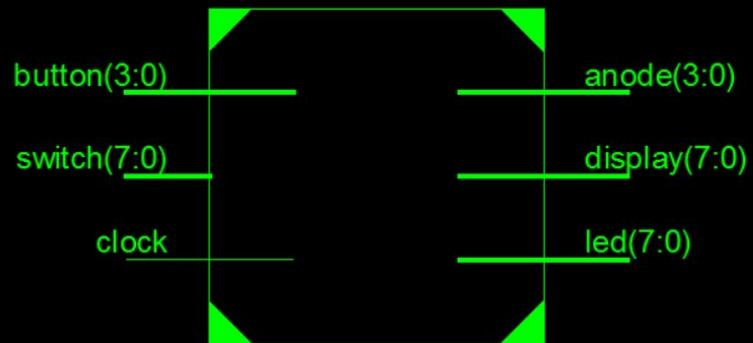
Implementação na Basys2



Representação de byte para BCD e 7 seg:

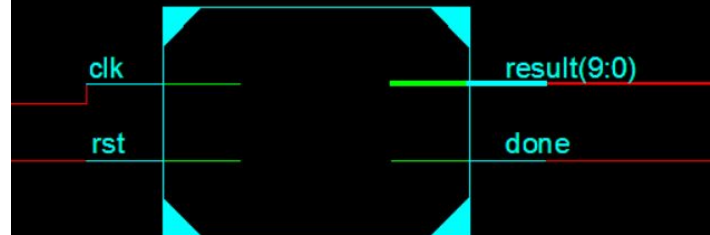


PLACA



PLACA

MATRIZ



matriz_hw

Exemplo 1

Matrix A =

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	0

Matrix B =

15	14	13	12
11	10	9	8
7	6	5	4
3	2	1	0

A x B =

70	60	50	40
214	188	162	136
358	316	274	232
454	412	370	328

Fim