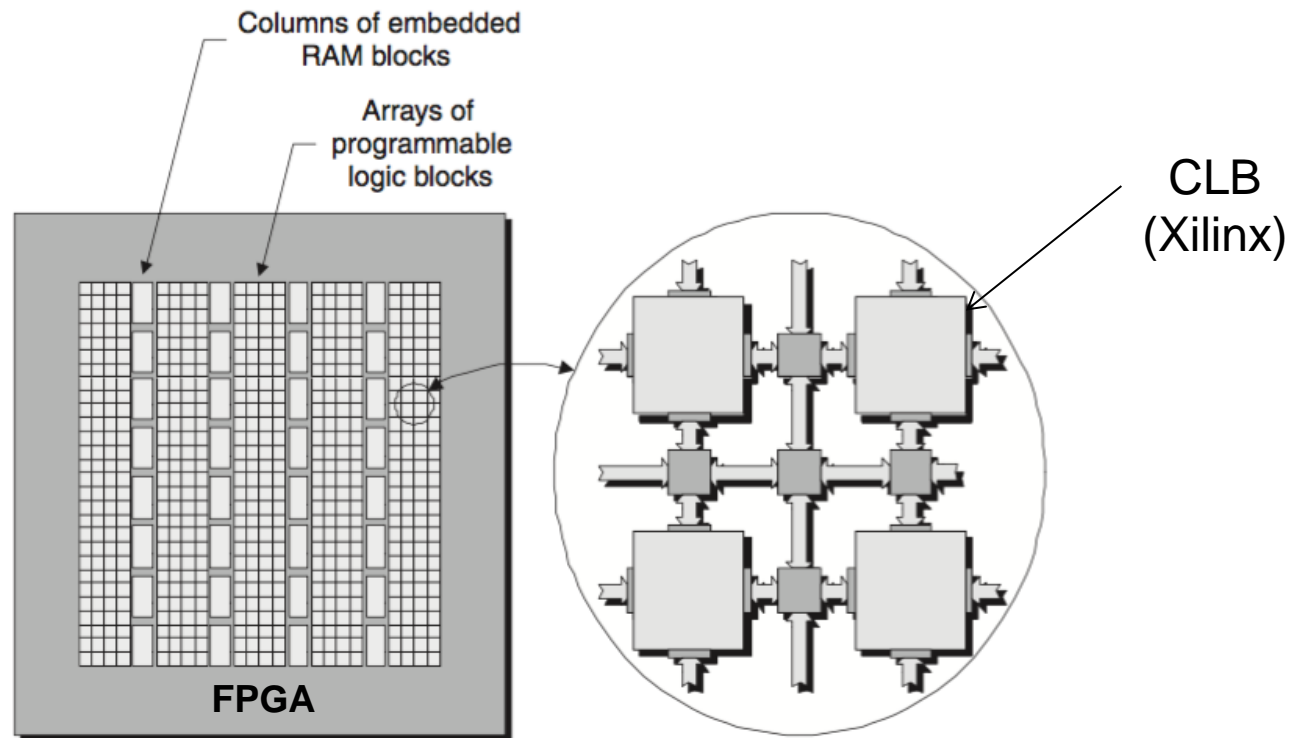


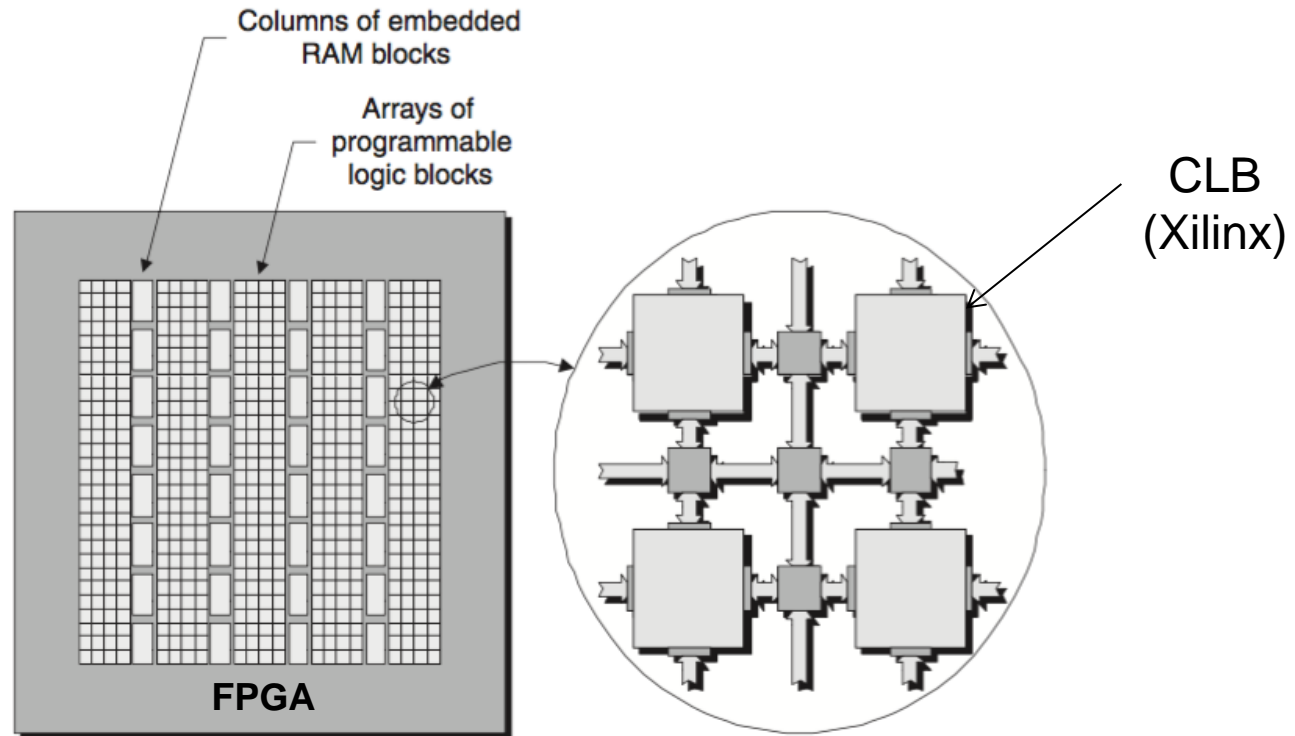
Trabalho 1 – parte 3

- A tarefa deste trabalho será fazer a prototipação em FPGA do processador MIPS
- Utilizaremos as Block RAMs do FPGA para implementar as memórias de instruções e dados



Trabalho 1 – parte 3

- ❑ A memória fornecida (Memory.vhd) está descrita de maneira que o sintetizador (XST) faça a inferência de *block* RAMs
- ❑ Desta maneira as memórias serão implementadas utilizando os blocos de RAM internos ao FPGA ao invés de utilizar LUTs



Trabalho 1 – parte 3

- ❑ O número de palavras das memórias é especificados através do parâmetro generic SIZE

```
DATA_MEMORY: entity work.Memory
    generic map (
        SIZE => 128,      -- Memory depth
        ...
```

Disponibilidade de Block RAM no FPGA Spartan-6 LX16: 72000 Bytes

Trabalho 1 – parte 3

- O arquivo de imagem pode ser gerado pelo MARS (mesmo arquivo utilizado na simulação VHDL)
- Cada linha do arquivo de imagem da memória corresponde a uma palavra (hexadecimal)
- O arquivo de imagem das memórias deve conter um número de linhas igual ou maior que a profundidade da memória (parâmetro SIZE)
- Exemplo

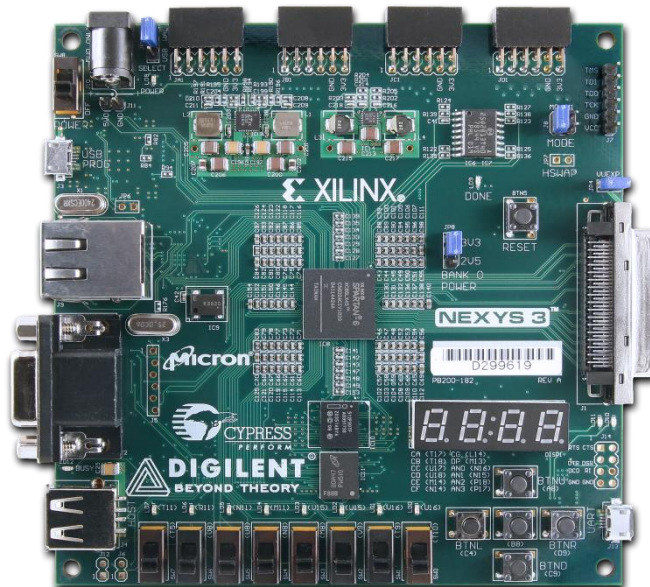
```
DATA_MEMORY: entity work.Memory  
    generic map (  
        SIZE => 128,  
        ...  
    )
```

-- Memory depth

O arquivo de imagem deve ter pelo menos 128 linhas

Trabalho 1 – parte 3

- ❑ A prototipação será feita utilizando a placa Nexys 3 (NUPEDEE)
 - ❑ Nesta placa o FPGA opera a 100MHz, no entanto nosso projeto deve operar a 25 MHz
 - ❑ Para dividir a frequência da placa utilizaremos um dos DCMs (*Digital Clock Manager*) do FPGA



Trabalho 1 – parte 3

- ❑ Para utilizar o DCM do FPGA, deve-se adicionar ao projeto componente *ClockManager*
 - ❑ ClockManager.vhd (*moodle*)

```
entity ClockManager is
    port (
        -- Board clock (100MHz)
        clk_100MHz      : in      std_logic;

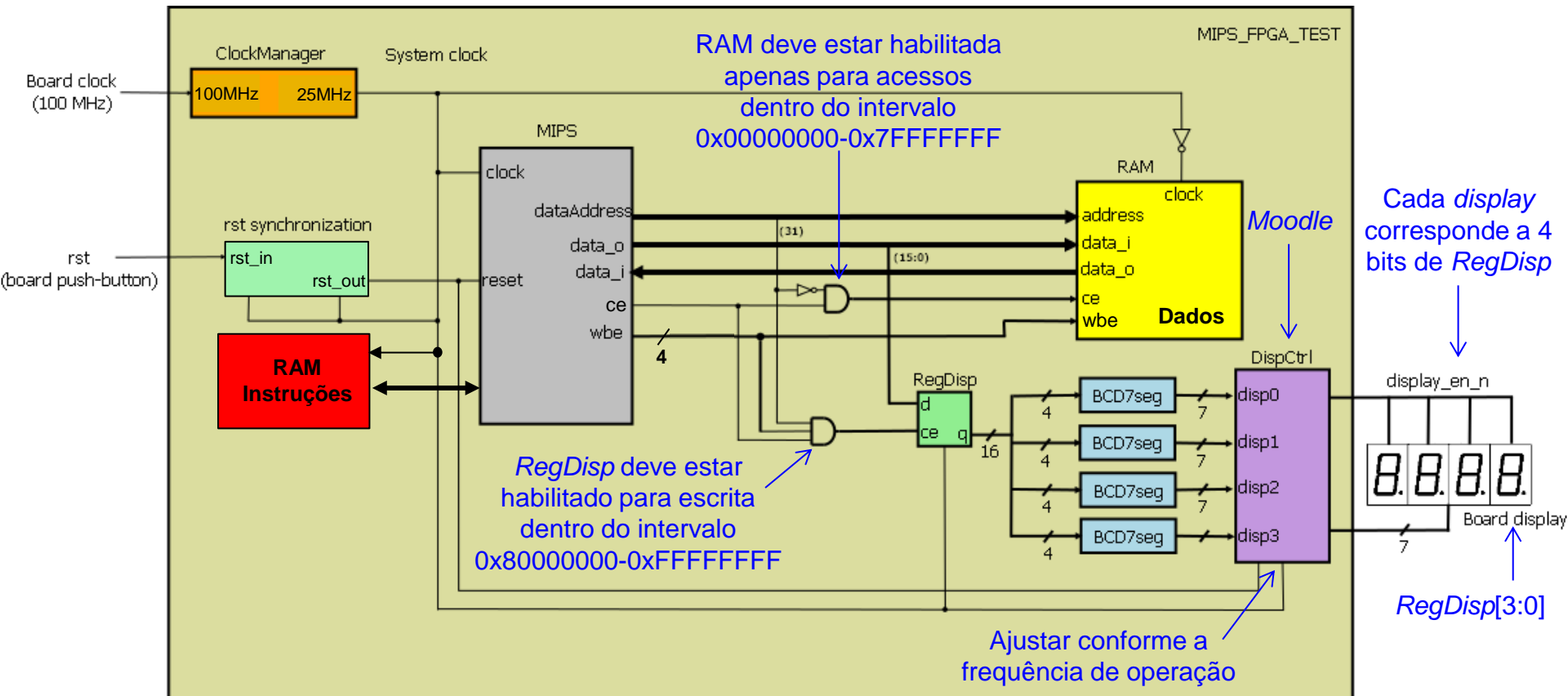
        -- Project clock
        clk_50MHz       : out     std_logic;
        clk_25MHz       : out     std_logic;
        clk_10MHz       : out     std_logic;
        clk_5MHz        : out     std_logic
    );
end ClockManager;
```

Trabalho 1 – parte 3

- ❑ Para simular componentes internos do FPGA (e.g. DCM) será necessário utilizar uma biblioteca específica da Xilinx presente no simulador ISIM (ISE)
 - ❑ No caso de outros simuladores, não há necessidade de utilizar o DCM durante a simulação
-

Trabalho 1 – parte 3

- ❑ Para que seja possível verificar o funcionamento da prototipação utilizaremos os *displays* da placa

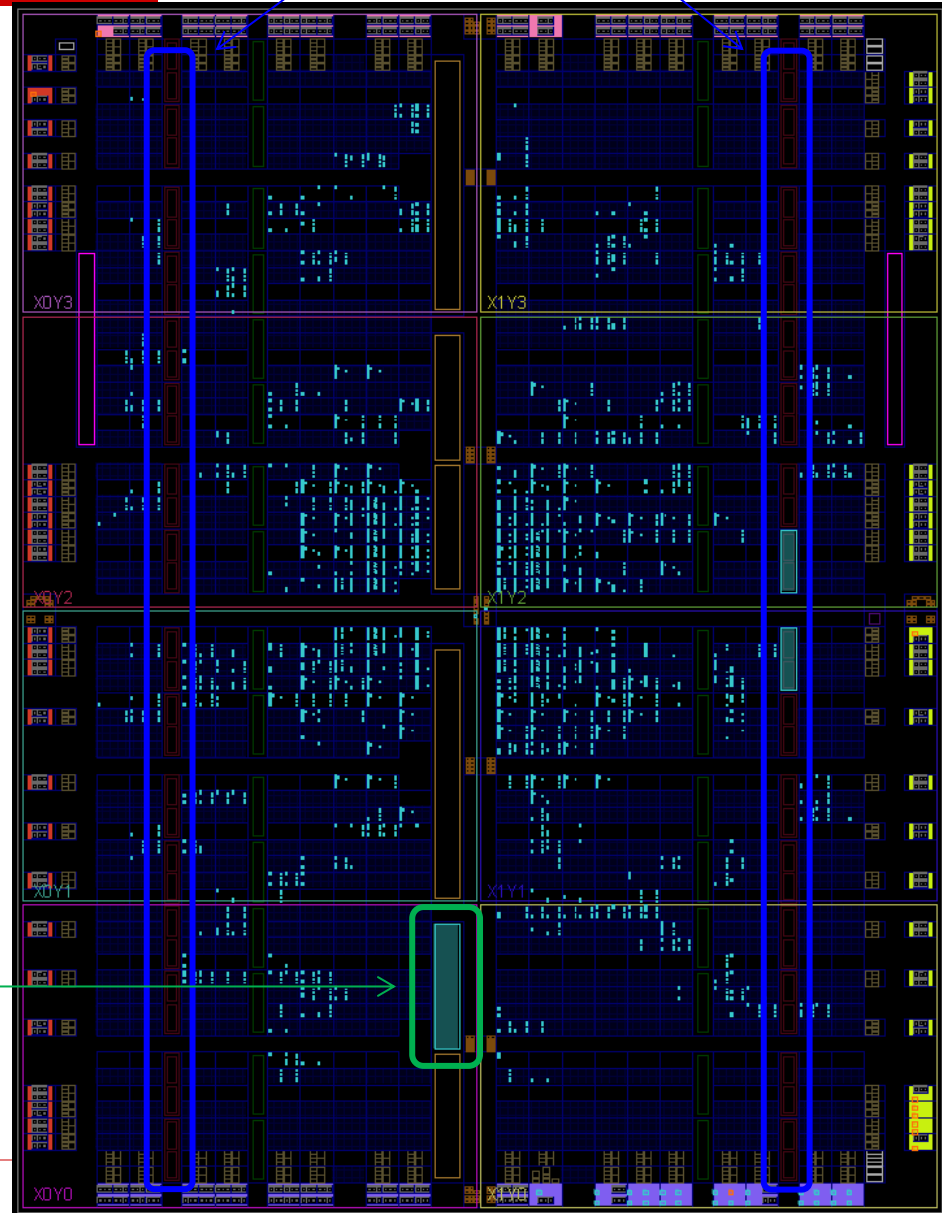


Trabalho 1 – parte 3

Block RAMs

- ❑ *Implemented design*
- ❑ Spartan 6 (LX16)

DCM



Trabalho 1 – parte 3

□ Aplicação

- Implementar um contador hexadecimal com incremento de 1 segundo (C ou *assembly*)
- Considerar o tempo de execução das instruções e a frequência a fim de atingir um incremento o mais próximo possível de 1 segundo

□ Sequência sugerida para o desenvolvimento

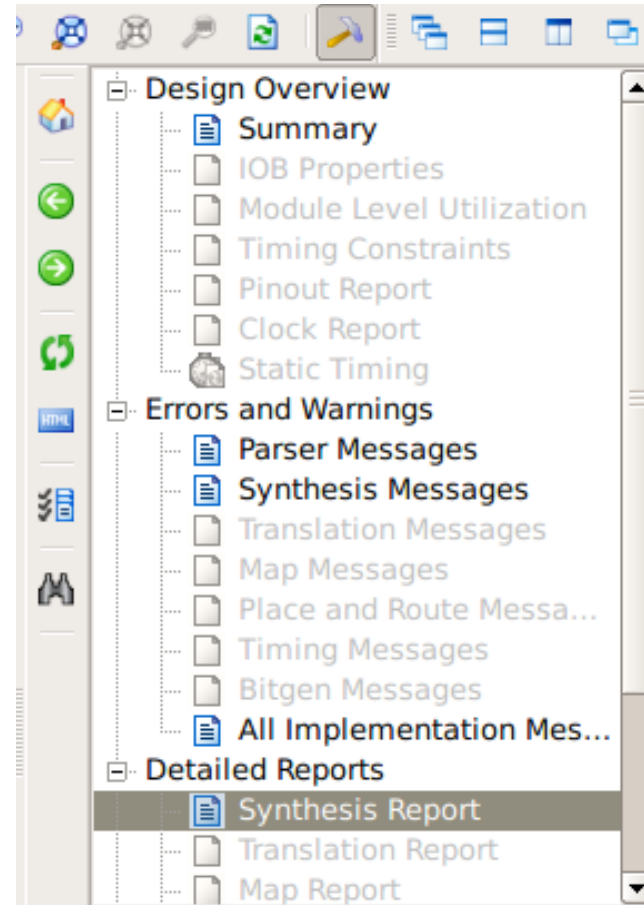
1. Adicionar DCM e sincronização do *reset* ao projeto atual e *simular*
 2. Criar a *entity* MIPS_FPGA_TEST contendo MIPS, memórias, DCM e sincronização do *reset*. Criar *test bench* para gerar *clock* e *reset* para a *entity* MIPS_FPGA_TEST e *simular*
 3. Adicionar a parte relativa à interface com os *displays* e *simular*
 4. Prototipar
 - Os arquivos de imagem das memórias devem estar no mesmo diretório do arquivo Memory.vhd
 - Atenção aos *warnings*!
-

Trabalho 1 – parte 3

☐ *Synthesis Report*

☐ Apresentar

Fiquem atento aos
warnings e ao *report*
do *clock*



Trabalho 1 – parte 3

□ Grupos de 2 alunos

- Mesmos grupos da parte 1
- Apresentação da descrição do funcionamento será **IMPRETERIVELMENTE** dia 11/4
- A nota do trabalho dará **ENORME ÊNFASE** à **prototipação**
 - Se a prototipação não funciona, não há o que apresentar
- Para a apresentação, trazer o bitstream (.bit) para que seja feito o download na placa Nexys
- Em relação às dúvidas, sejam pontuais