

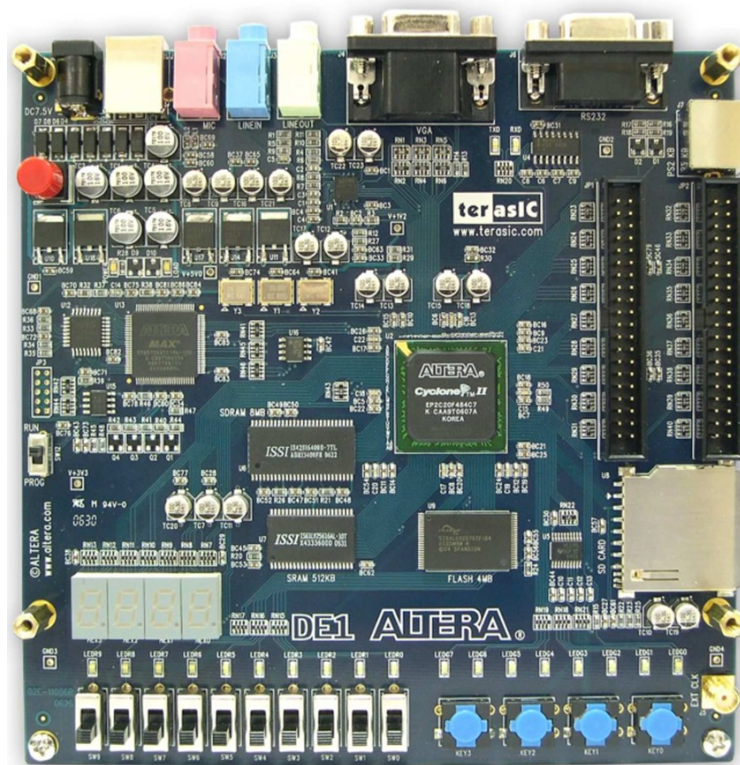
arquitetura de computadores

UTFPR – DAELN – Engenharia de Computação/Eletrônica
prof. Ofuchi; prof. Rafael; prof. Juliano;

rev 4

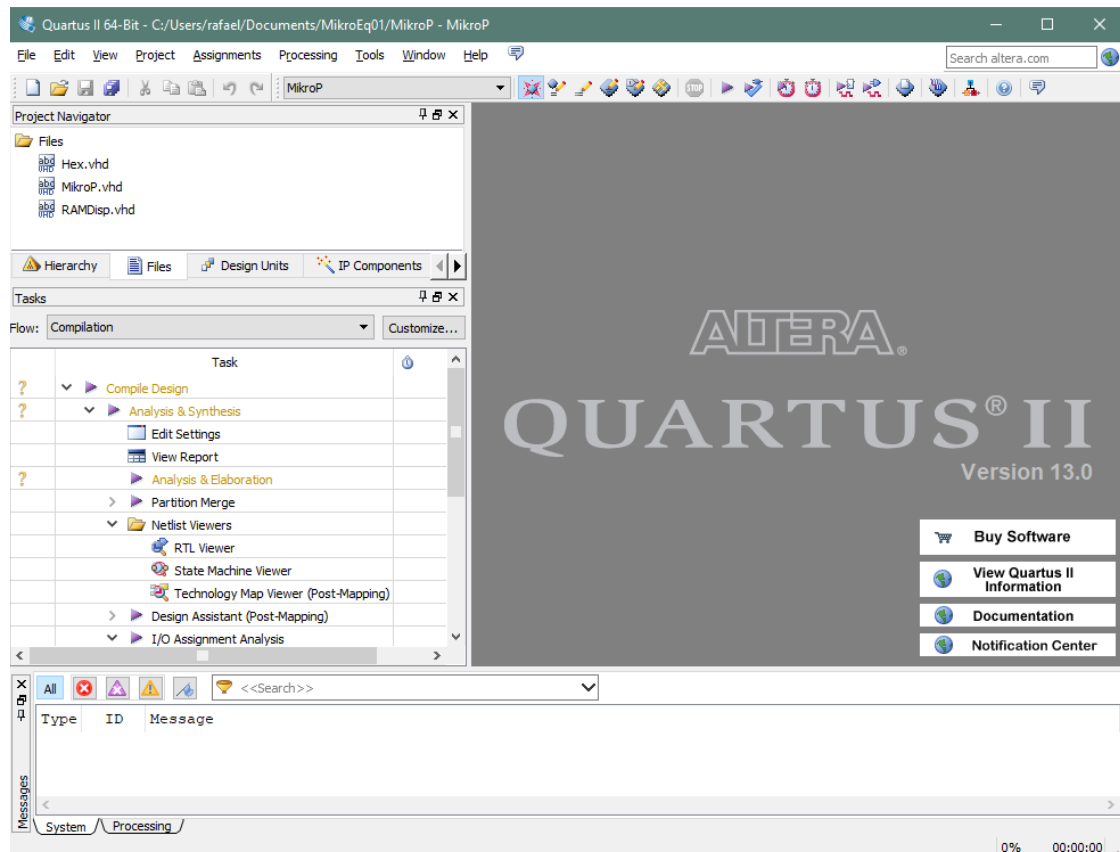
μProcessador 3.5 Circuitos Síncronos e Programação da FPGA

A partir dos registradores e de circuitos que usam o sinal de clock, teremos uma pausa para verificar como estes conceitos formam a base para a geração de circuitos em HW propriamente dito. Para permitir que um circuito descrito em VHDL seja gravado numa FPGA (Cyclone II da Altera - EP2C20F484C7) será usado o ambiente integrado Quartus. Vamos usar a placa CycloneII FPGA Starter Board. Para este componente deverá ser usada a versão 13, service pack1 (o Quartus Prime só funciona para a Cyclone IV em diante).



De modo similar ao que consideramos até aqui para a geração de estímulos utilizando test bench como *toplevel*, vamos usar os sinais da placa (chaves) e excitar alguns circuitos combinacionais e sequenciais e liga-los a sinais elétricos propriamente ditos (disponíveis em LEDs e displays). O esquemático da placa está disponível no Moodle.

Inicialmente, após baixar e descompactar o projeto exemplo na pasta de trabalho, e abrí-lo no Quartus.

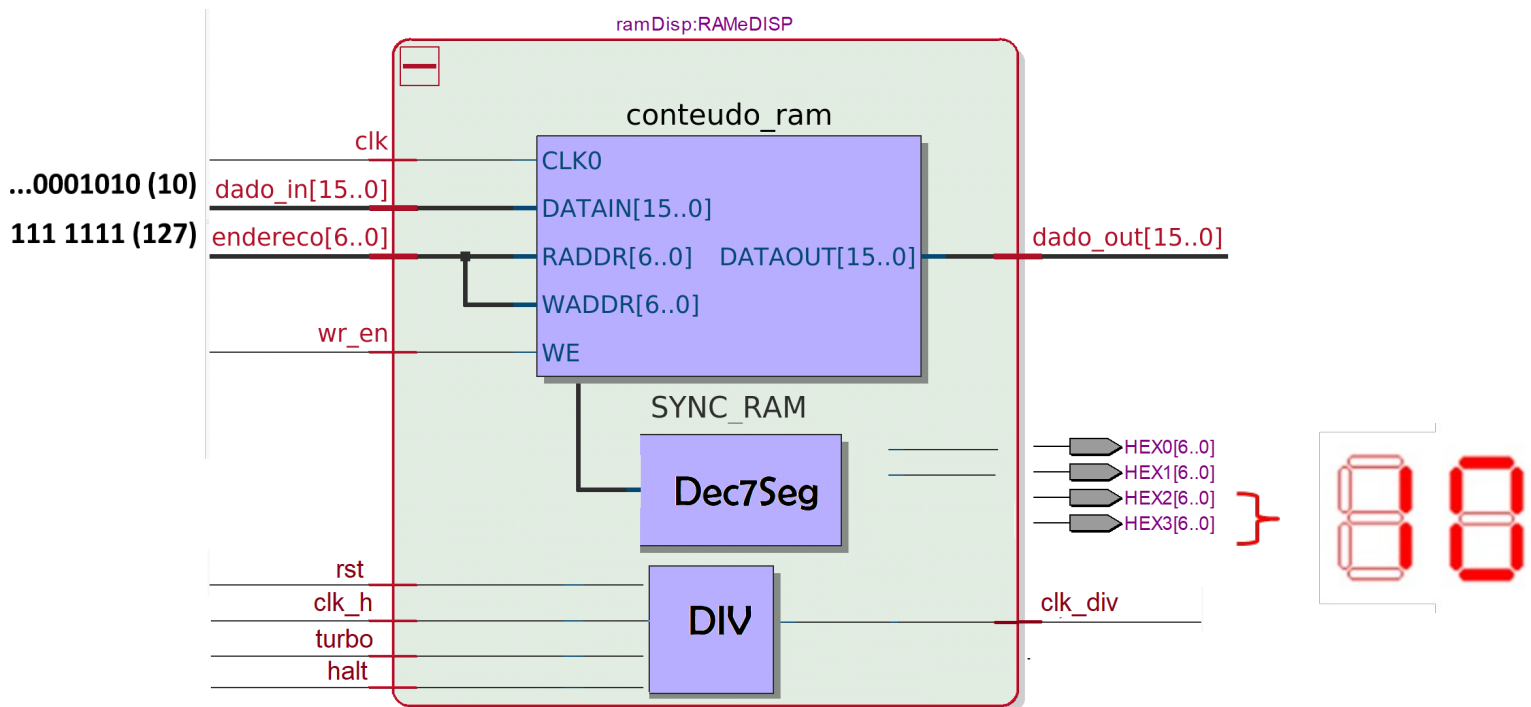


Além de usar chaves, push buttons LEDs e display de 7 segmentos este componente encapsula um divisor de clock principal da placa.

Alguns LEDs indicam apenas um contador global de ciclos e chaves de parada (halt) e “turbo”. A história da tecla turbo é bastante interessante no contexto da disciplina. Vale a leitura do verbete:

https://en.wikipedia.org/wiki/Turbo_button

A entidade RAMEDisp tem também um periférico de saída que é um registrador. Neste caso, ao escrever no endereço 127d (não será necessário fazer modificação do barramento de endereços por enquanto) o dado é apresentado em decimal num par de displays de 7 segmentos da placa Cyclone II Starter Board. A conversão é feita no modo flash (em um ciclo) para simplificar a temporização e evitar potenciais problemas com a temporização do processador implementado. O display corresponde, portanto, a um periférico mapeado em memória. As escritas são feitas em um endereço específico e a decodificação fica encapsulados nesta entidade (Dec7Seg).



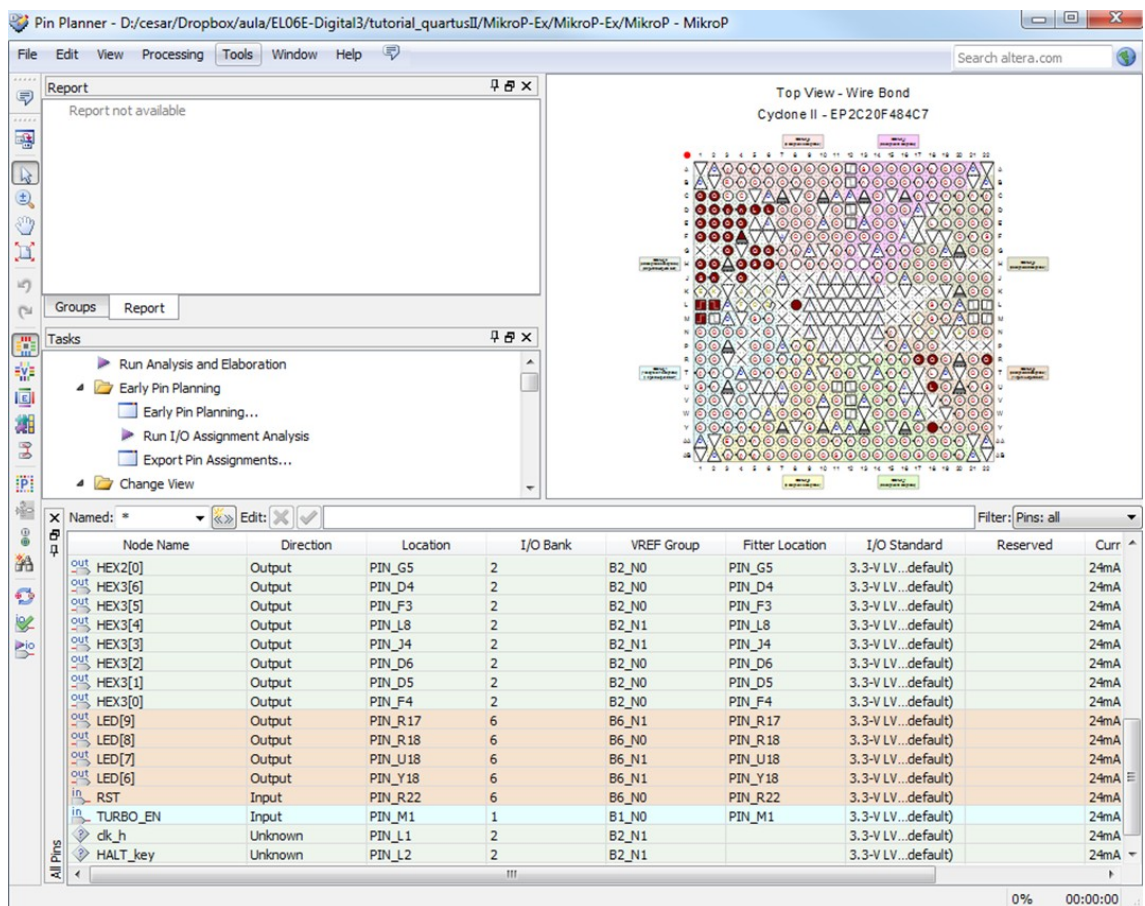
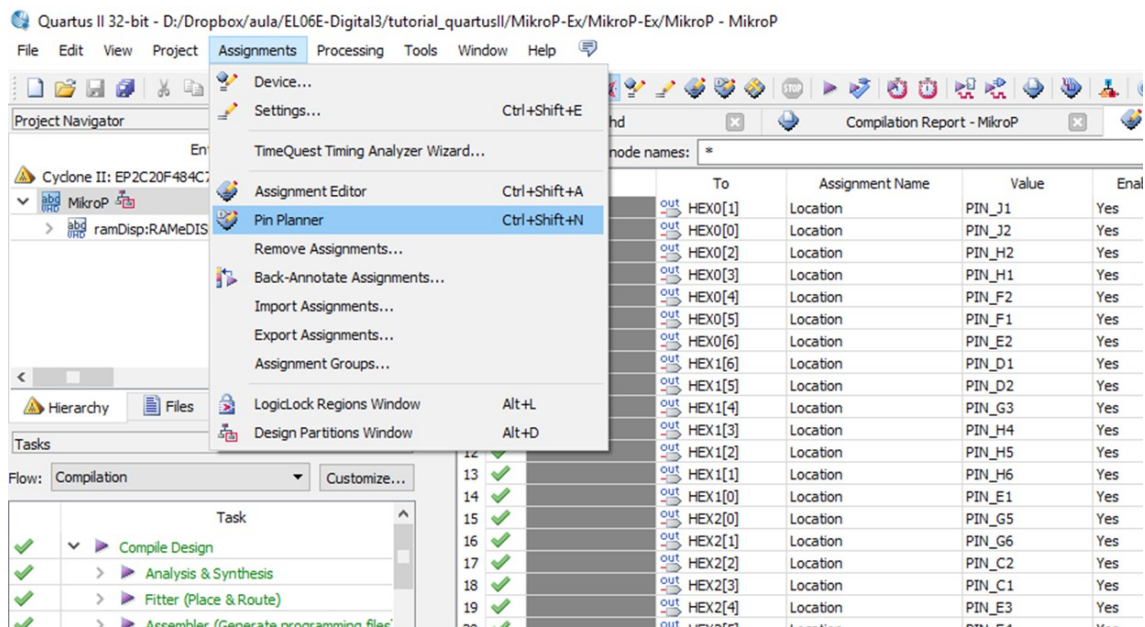
Em VHDL, a nova interface da RAM fica como a seguir.

```
entity ram is
  port(
    clk      : in std_logic;
    endereco : in unsigned(6 downto 0);
    wr_en    : in std_logic;
    dado_in  : in unsigned(15 downto 0);
    dado_out : out unsigned(15 downto 0);

    HEX0      : out std_logic_vector (6 downto 0);
    HEX1      : out std_logic_vector (6 downto 0);
    HEX2      : out std_logic_vector (6 downto 0);
    HEX3      : out std_logic_vector (6 downto 0);
    halt      : in std_logic;
    turbo     : in std_logic;
    clk_h     : in std_logic;
    clk_div   : out std_logic;
    rst       : in std_logic;
  );
end entity;
```

Uma sugestão é, antes de modificar qualquer coisa, verificar o funcionamento da placa executando direto os procedimentos das seções “**Compilação do arquivo a ser gravado**” e “**Gravação do Arquivo Gerado na FPGA**” ao final deste documento para testar o processo.

Como base neste ponto de partida, é possível modificar detalhes dos circuitos combinacionais e sequenciais bem como os pinos que receberão os sinais do projeto> Para a atribuição dos pinos, há o *Pin Planner* e o *Assignment Editor* conforme as telas a seguir.



Pin Planner - D:/cesar/Dropbox/aula/EL06E-Digital3/tutorial_quartusII/MikroP-Ex/MikroP - MikroP

File Edit View Processing Tools Window Help

Search altera.com

Report

Report not available

Groups Report

Tasks

- Run Analysis and Elaboration
 - Early Pin Planning
 - Early Pin Planning...
 - Run I/O Assignment Analysis
 - Export Pin Assignments...
- Change View

Top View - Wire Bond
Cyclone II - EP2C20F484C7

Named: * Edit: [X] [Y] [Z]

Filter: Pins: all

Node Name	Direction	Location	I/O Bank	VREF Group	Fitter Location	I/O Standard	Reserved	Curr
HEX2[0]	Output	PIN_G5	2	B2_N0	PIN_G5	3.3-V LV...default		24mA
HEX3[6]	Output	PIN_D4	2	B2_N0	PIN_D4	3.3-V LV...default		24mA
HEX3[5]	Output	PIN_F3	2	B2_N0	PIN_F3	3.3-V LV...default		24mA
HEX3[4]	Output	PIN_L8	2	B2_N1	PIN_L8	3.3-V LV...default		24mA
HEX3[3]	Output	PIN_J4	2	B2_N1	PIN_J4	3.3-V LV...default		24mA
HEX3[2]	Output	PIN_D6	2	B2_N0	PIN_D6	3.3-V LV...default		24mA
HEX3[1]	Output	PIN_D5	2	B2_N0	PIN_D5	3.3-V LV...default		24mA
HEX3[0]	Output	PIN_F4	2	B2_N0	PIN_F4	3.3-V LV...default		24mA
LED[9]	Output	PIN_R17	6	B6_N1	PIN_R17	3.3-V LV...default		24mA
LED[8]	Output	PIN_R18	6	B6_N0	PIN_R18	3.3-V LV...default		24mA
LED[7]	Output	PIN_U18	6	B6_N1	PIN_U18	3.3-V LV...default		24mA
LED[6]	Output	PIN_Y18	6	B6_N1	PIN_Y18	3.3-V LV...default		24mA
RST	Input	PIN_R22	6	B6_N0	PIN_R22	3.3-V LV...default		24mA
TURBO_EN	Input	PIN_M1	1	B1_N0	PIN_M1	3.3-V LV...default		24mA
clk_h	Unknown	PIN_L1	2	B2_N1		3.3-V LV...default		24mA
HALT_key	Unknown	PIN_L2	2	B2_N1		3.3-V LV...default		24mA

0% 00:00:00

Quartus II 64-Bit - D:/Dropbox/aula/EL06E-Digital3/tutorial_quartusII/MikroP-Ex/MikroP

File Edit View Project Assignments Processing Tools Window Help

Project Navigator

Cyclone II: EP2C20F484C7

MikroP

Assignment Editor

Pin Planner

From	To	Assignment Name	Value	Enabled	Entity
HEX0[1]	Location	PIN_J1	Yes		
HEX0[0]	Location	PIN_J2	Yes		
HEX0[2]	Location	PIN_H2	Yes		
HEX0[3]	Location	PIN_H1	Yes		
HEX0[4]	Location	PIN_F2	Yes		
HEX0[5]	Location	PIN_F1	Yes		
HEX0[6]	Location	PIN_E2	Yes		
HEX1[6]	Location	PIN_D1	Yes		
HEX1[5]	Location	PIN_D2	Yes		
HEX1[4]	Location	PIN_G3	Yes		
HEX1[3]	Location	PIN_H4	Yes		
HEX1[2]	Location	PIN_H5	Yes		
HEX1[1]	Location	PIN_H6	Yes		
HEX1[0]	Location	PIN_E1	Yes		
HEX2[0]	Location	PIN_G5	Yes		
HEX2[1]	Location	PIN_G6	Yes		
clk_h	Location	PIN_L1	Yes		
TURBO_EN	Location	PIN_M1	Yes		
LED[9]	Location	PIN_R17	Yes		
LED[8]	Location	PIN_R18	Yes		
LED[7]	Location	PIN_U18	Yes		
LED[6]	Location	PIN_Y18	Yes		
HALT_KEY	Location	PIN_L22	Yes		
<<new>>	<<new>>	<<new>>			

Os pinos da placa (chaves, botões, clock, LEDs e Displays) já estão mapeados neste projeto, mas podem ser modificados. Os assinalamentos dos sinais aos pinos deve seguir o roteamento da placa disponível no manual e resumido a seguir (DE1_UserManual_v1018.pdf):

Signal Name	FPGA Pin No.	Description
SW[0]	PIN_L22	Toggle Switch[0]
SW[1]	PIN_L21	Toggle Switch[1]
SW[2]	PIN_M22	Toggle Switch[2]
SW[3]	PIN_V12	Toggle Switch[3]
SW[4]	PIN_W12	Toggle Switch[4]
SW[5]	PIN_U12	Toggle Switch[5]
SW[6]	PIN_U11	Toggle Switch[6]
SW[7]	PIN_M2	Toggle Switch[7]
SW[8]	PIN_M1	Toggle Switch[8]
SW[9]	PIN_L2	Toggle Switch[9]

Signal Name	FPGA Pin No.	Description
KEY[0]	PIN_R22	Pushbutton[0]
KEY[1]	PIN_R21	Pushbutton[1]
KEY[2]	PIN_T22	Pushbutton[2]
KEY[3]	PIN_T21	Pushbutton[3]

Signal Name	FPGA Pin No.	Description
LEDR[0]	PIN_R20	LED Red[0]
LEDR[1]	PIN_R19	LED Red[1]
LEDR[2]	PIN_U19	LED Red[2]
LEDR[3]	PIN_Y19	LED Red[3]
LEDR[4]	PIN_T18	LED Red[4]
LEDR[5]	PIN_V19	LED Red[5]
LEDR[6]	PIN_Y18	LED Red[6]
LEDR[7]	PIN_U18	LED Red[7]
LEDR[8]	PIN_R18	LED Red[8]
LEDR[9]	PIN_R17	LED Red[9]
LEDG[0]	PIN_U22	LED Green[0]
LEDG[1]	PIN_U21	LED Green[1]
LEDG[2]	PIN_V22	LED Green[2]
LEDG[3]	PIN_V21	LED Green[3]
LEDG[4]	PIN_W22	LED Green[4]
LEDG[5]	PIN_W21	LED Green[5]
LEDG[6]	PIN_Y22	LED Green[6]
LEDG[7]	PIN_Y21	LED Green[7]

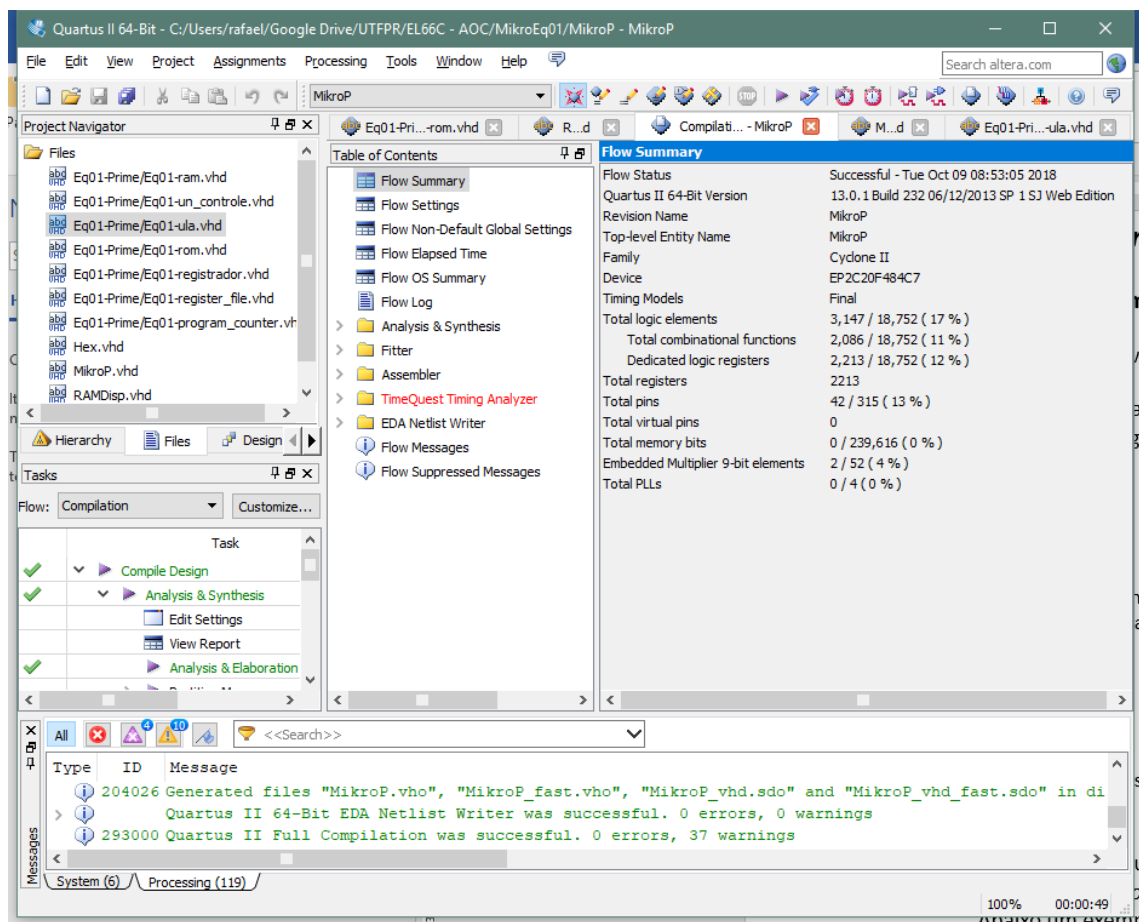
Signal Name	FPGA Pin No.	Description
CLOCK_27	PIN_D12, PIN_E12	27 MHz clock input
CLOCK_50	PIN_L1	50 MHz clock input
CLOCK_24	PIN_A12, PIN_B12	24 MHz clock input from USB Blaster
EXT_CLOCK	PIN_M21	External (SMA) clock input

Algumas sugestões de objetivos a serem implementados:

- 1 - Alterar as chaves para outra posição
Botão de reset
Chave de parada
- 2 - Contar até 64 ao invés de até 16
Mostrar no display
Mostrar nos LEDs
- 3 - Mudar a velocidade da contagem
Alterar o clock inserindo um outro patamar intermediário de frequência, selecionável a partir de outra chave.
- 4 - Alterar a base de contagem para base 7.

Compilação do arquivo a ser gravado

Após as modificações sugeridas “execute a síntese”. Para isso, usar o menu **Processing > Start Compilation**. Observe no relatório final da síntese a quantidade de elementos lógicos utilizados.



Alguns Warnings (novos) podem ocorrer. A maioria diz respeito

apenas ao fato de ser usada uma versão de avaliação do pacote de software de síntese Quartus e detalhes de simulação que não foram definidos.

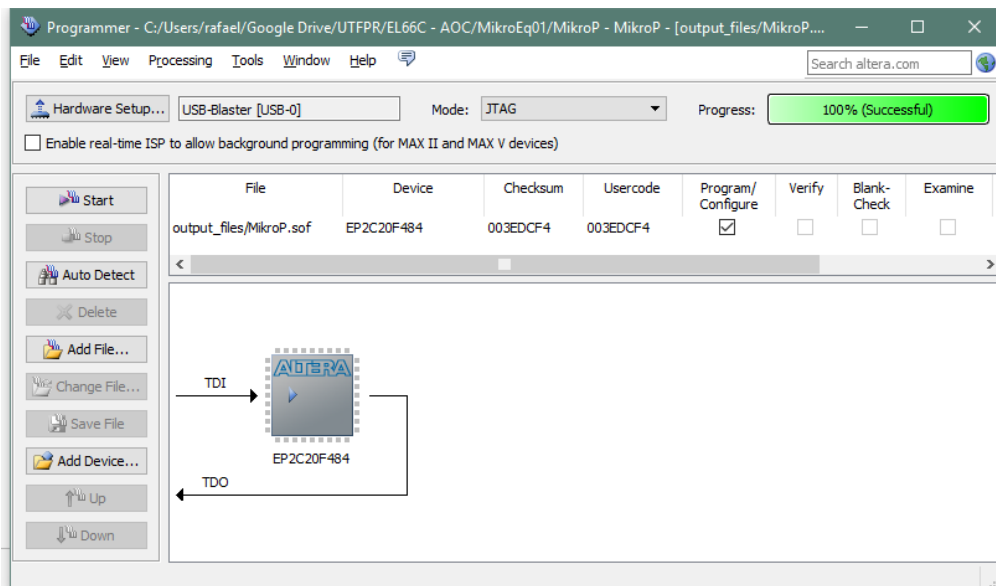
Se todos os sinais foram ligados corretamente, a síntese do circuito gerará um arquivo que pode ser gravado no componente. Ter certeza disso pode levar algum tempo e a necessidade de ir colocando sinais de diagnóstico em LEDs caso o projeto não esteja bem organizado.

A gravação do arquivo gerado na FPGA

Para gravar o arquivo gerado: MikroP.sof basta usar o programador (menu **Tools > Programmer**). Abaixo é mostrada a tela do Programador.

Clique duas vezes no arquivo .sof para ter certeza de que a versão no diretório corrente de trabalho está sendo usada. Com a placa conectada a qualquer porta USB, use o botão “**Auto Detect**” e o componente deve aparecer conforme a figura, com um triângulo azul sobre o chip.

A chave SW12 da placa deve estar na posição “RUN”. Clique no botão **Start** para descarregar o arquivo para o componente.



Após a gravação, o circuito funciona diretamente na placa. O reset está mapeado no botão KEY0 e as chaves de halt e turbo estão mapeadas nas chaves SW9 e SW8, respectivamente.

Arquivos a Entregar

Além da demonstração em sala, postar um único ZIP contendo toda a pasta do projeto em Quartus da equipe com um cabeçalho descrevendo a funcionalidade implementada (Obs: para que o arquivo Zip fique menor, pode apagar a pasta “simulation”).