

A decorative graphic consisting of thin, grey, stylized circuit lines with small circles at the ends, extending horizontally from the left and right sides of the central black box.

SISTEMAS EMBARCADOS

PROF. JOSENALDE OLIVEIRA

ADS-UFRN

NIOS-II: AMPLIANDO AS CAPACIDADES

- A configuração básica no soft-processor inclui:
 - Clock, JTAG_UART, memória interna (onchip mem) na **DE2**, limitada a +- 52,5 KB
 - São blocos chamados M4K, que possuem palavra de 4.096 bits, 512 bytes ou 0,5 KB. Como são 105 blocos, daria um total de $0,5 * 105 = 52,5$ KB
 - Além destes, recomenda-se a adição do componente SYSID e INTERVAL TIMER, para identificar unicamente cada arquitetura, evitando conflitos, e um temporizador, para ser possível utilizar arquivos .h e funções de temporização, clock etc.
 - Isto restringe o código fonte a versão reduzida do C (sem float por exemplo), com pouca adição de device drivers entre outras limitações.
- **Uma generalização do SoC que inclua SDRAM e periféricos (chaves, botões, displays) amplia a possibilidade de projetos**

QSYS

Use	Connections	Name	Description	Export	Clock	Base	End	IRQ
<input checked="" type="checkbox"/>		clock	Clock Source					
		clk_in	Clock Input	clk				
		clk_in_reset	Reset Input	reset				
		clk	Clock Output	<i>Double-click to export</i>	clock			
		clk_reset	Reset Output	<i>Double-click to export</i>				
<input checked="" type="checkbox"/>		cpu	Nios II Processor					
		clk	Clock Input	<i>Double-click to export</i>	clock			
		reset_n	Reset Input	<i>Double-click to export</i>	[clk]			
		data_master	Avalon Memory Mapped Master	<i>Double-click to export</i>	[clk]			
		instruction_master	Avalon Memory Mapped Master	<i>Double-click to export</i>	[clk]			
		jtag_debug_module_reset	Reset Output	<i>Double-click to export</i>	[clk]			
		jtag_debug_module	Avalon Memory Mapped Slave	<i>Double-click to export</i>	[clk]	0x0000_8800	0x0000_8fff	
		custom_instruction_master	Custom Instruction Master	<i>Double-click to export</i>				
<input checked="" type="checkbox"/>		onchipmemory	On-Chip Memory (RAM or ROM)					
		clk1	Clock Input	<i>Double-click to export</i>	clock			
		s1	Avalon Memory Mapped Slave	<i>Double-click to export</i>	[clk1]	0x0000_0000	0x0000_4faf	
		reset1	Reset Input	<i>Double-click to export</i>	[clk1]			
<input checked="" type="checkbox"/>		jtag_uart	JTAG UART					
		clk	Clock Input	<i>Double-click to export</i>	clock			
		reset	Reset Input	<i>Double-click to export</i>	[clk]			
		avalon_jtag_slave	Avalon Memory Mapped Slave	<i>Double-click to export</i>	[clk]	0x0000_9118	0x0000_911f	
<input checked="" type="checkbox"/>		green_leds	PIO (Parallel IO)					
		clk	Clock Input	<i>Double-click to export</i>	clock			
		reset	Reset Input	<i>Double-click to export</i>	[clk]			
		s1	Avalon Memory Mapped Slave	<i>Double-click to export</i>	[clk]	0x0000_9100	0x0000_910f	
		external_connection	Conduit	green_leds				
<input checked="" type="checkbox"/>		sysid_qsys_0	System ID Peripheral					
		clk	Clock Input	<i>Double-click to export</i>	clock			
		reset	Reset Input	<i>Double-click to export</i>	[clk]			
		control_slave	Avalon Memory Mapped Slave	<i>Double-click to export</i>	[clk]	0x0000_9110	0x0000_9117	
<input checked="" type="checkbox"/>		sys_clk_timer	Interval Timer					
		clk	Clock Input	<i>Double-click to export</i>	clock			
		reset	Reset Input	<i>Double-click to export</i>	[clk]			
		s1	Avalon Memory Mapped Slave	<i>Double-click to export</i>	[clk]	0x0000_9000	0x0000_901f	
<input checked="" type="checkbox"/>		red_leds	PIO (Parallel IO)					
		clk	Clock Input	<i>Double-click to export</i>	clock			
		reset	Reset Input	<i>Double-click to export</i>	[clk]			
		s1	Avalon Memory Mapped Slave	<i>Double-click to export</i>	[clk]	0x0000_90f0	0x0000_90ff	
		external_connection	Conduit	red_leds				

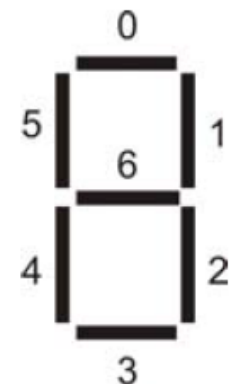
Setar vetores
reset/exception
para SDRAM

Apenas data master

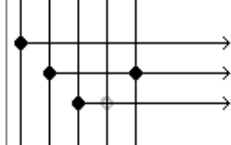
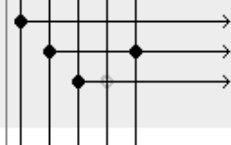
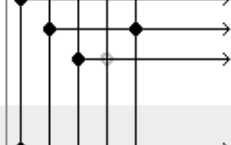
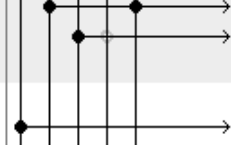
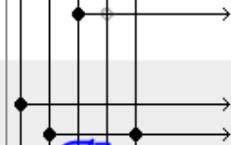
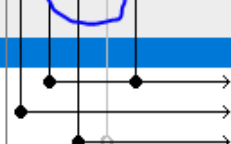

- O timer é referenciado na especificação do BSP
- Aqui, os leds verdes e vermelhos da placa DE2 são exportados e podem ser referenciados no código fonte, tal como identificados no system.h

QSYS

- Toggle switch[17..0]
- Push-buttons key[3..1]
 - NF
- Displays 7-SEG
 - Anodo comum (liga com 0)
 - HEX0[6..0] a HEX7[6..0]



- 8 displays de 7 segmentos.
 - HEX0 até HEX7.
- Sinais do *display* ativos em ZERO.
- Sinais de cada *led* no display:
 - Ex. HEX0[0] até HEX0[6]

<input checked="" type="checkbox"/>		<div>seven_seg_3</div> <div>clk</div> <div>reset</div> <div>s1</div> <div>external_connection</div>	PIO (Parallel IO) Clock Input Reset Input Avalon Memory Mapped Slave Conduit	<div>Double-click to export</div> <div>Double-click to export</div> <div>Double-click to export</div> <div>seven_seg_3</div>	clock [clk] [clk]	<div>0x0000_9050</div>	0x0000_905f		
<input checked="" type="checkbox"/>		<div>seven_seg_4</div> <div>clk</div> <div>reset</div> <div>s1</div> <div>external_connection</div>	PIO (Parallel IO) Clock Input Reset Input Avalon Memory Mapped Slave Conduit	<div>Double-click to export</div> <div>Double-click to export</div> <div>Double-click to export</div> <div>seven_seg_4</div>	clock [clk] [clk]	<div>0x0000_9060</div>	0x0000_906f		
<input checked="" type="checkbox"/>		<div>seven_seg_5</div> <div>clk</div> <div>reset</div> <div>s1</div> <div>external_connection</div>	PIO (Parallel IO) Clock Input Reset Input Avalon Memory Mapped Slave Conduit	<div>Double-click to export</div> <div>Double-click to export</div> <div>Double-click to export</div> <div>seven_seg_5</div>	clock [clk] [clk]	<div>0x0000_9070</div>	0x0000_907f		
<input checked="" type="checkbox"/>		<div>seven_seg_6</div> <div>clk</div> <div>reset</div> <div>s1</div> <div>external_connection</div>	PIO (Parallel IO) Clock Input Reset Input Avalon Memory Mapped Slave Conduit	<div>Double-click to export</div> <div>Double-click to export</div> <div>Double-click to export</div> <div>seven_seg_6</div>	clock [clk] [clk]	<div>0x0000_9090</div>	0x0000_909f		
<input checked="" type="checkbox"/>		<div>seven_seg_7</div> <div>clk</div> <div>reset</div> <div>s1</div> <div>external_connection</div>	PIO (Parallel IO) Clock Input Reset Input Avalon Memory Mapped Slave Conduit	<div>Double-click to export</div> <div>Double-click to export</div> <div>Double-click to export</div> <div>seven_seg_7</div>	clock [clk] [clk]	<div>0x0000_9080</div>	0x0000_908f		
<input checked="" type="checkbox"/>		<div>sdram</div> <div>clk</div> <div>reset</div> <div>s1</div> <div>wire</div>	SDRAM Controller Clock Input Reset Input Avalon Memory Mapped Slave Conduit	<div>Double-click to export</div> <div>Double-click to export</div> <div>Double-click to export</div> <div>sdram</div>	clock [clk] [clk] [clk]	<div>0x0800_0000</div>	0x0fff_ffff		
<input checked="" type="checkbox"/>		<div>lcd</div> <div>reset</div> <div>clk</div> <div>control_slave</div> <div>external</div>					0x0000_9020	0x0000_902f	

Altera Avalon LCD 16207 - lcd

Altera Avalon LCD 16207

altera_avalon_lcd_16207

Documentation

Block Diagram


Show signals

Description

The Optrex 16207 LCD Controller core provides the hardware interface required for a Nios II processor to display characters on an Optrex 16207 (or equivalent)

- Atenção especial à configuração da SDRAM

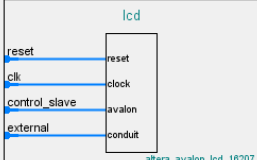
Altera Avalon LCD 16207 - lcd

 **Altera Avalon LCD 16207**
altera_avalon_lcd_16207

[Documentation](#)

Block Diagram

☐ Show signals



Description

The Optrex 16207 LCD Controller core provides the hardware interface required for a Nios II processor to display characters on an Optrex 16207 (or equivalent) 16x2-character LCD panel.

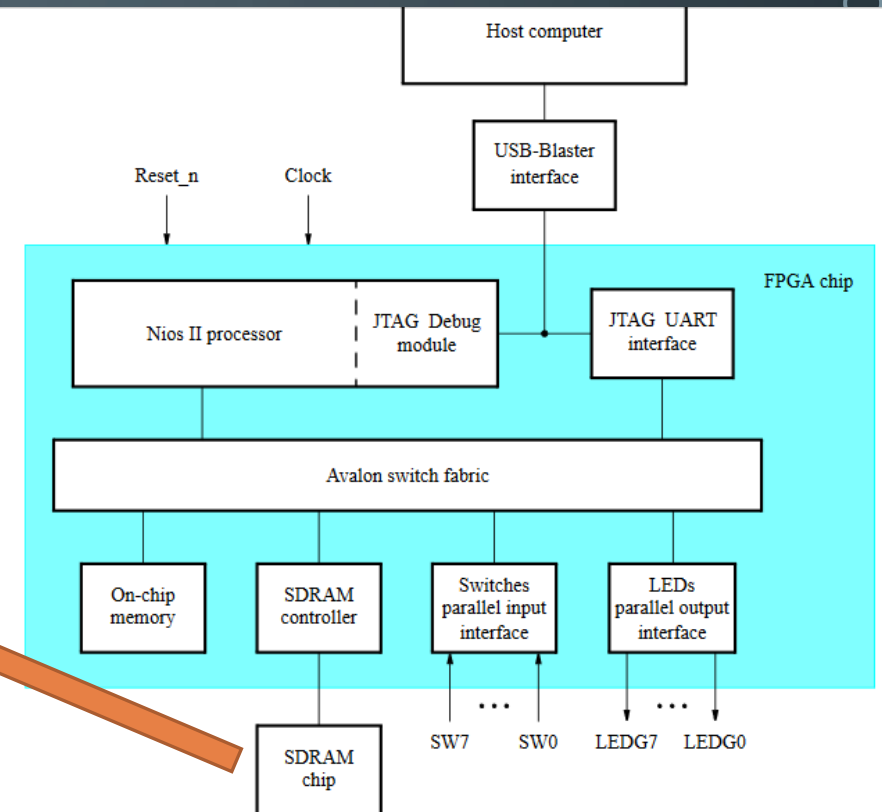
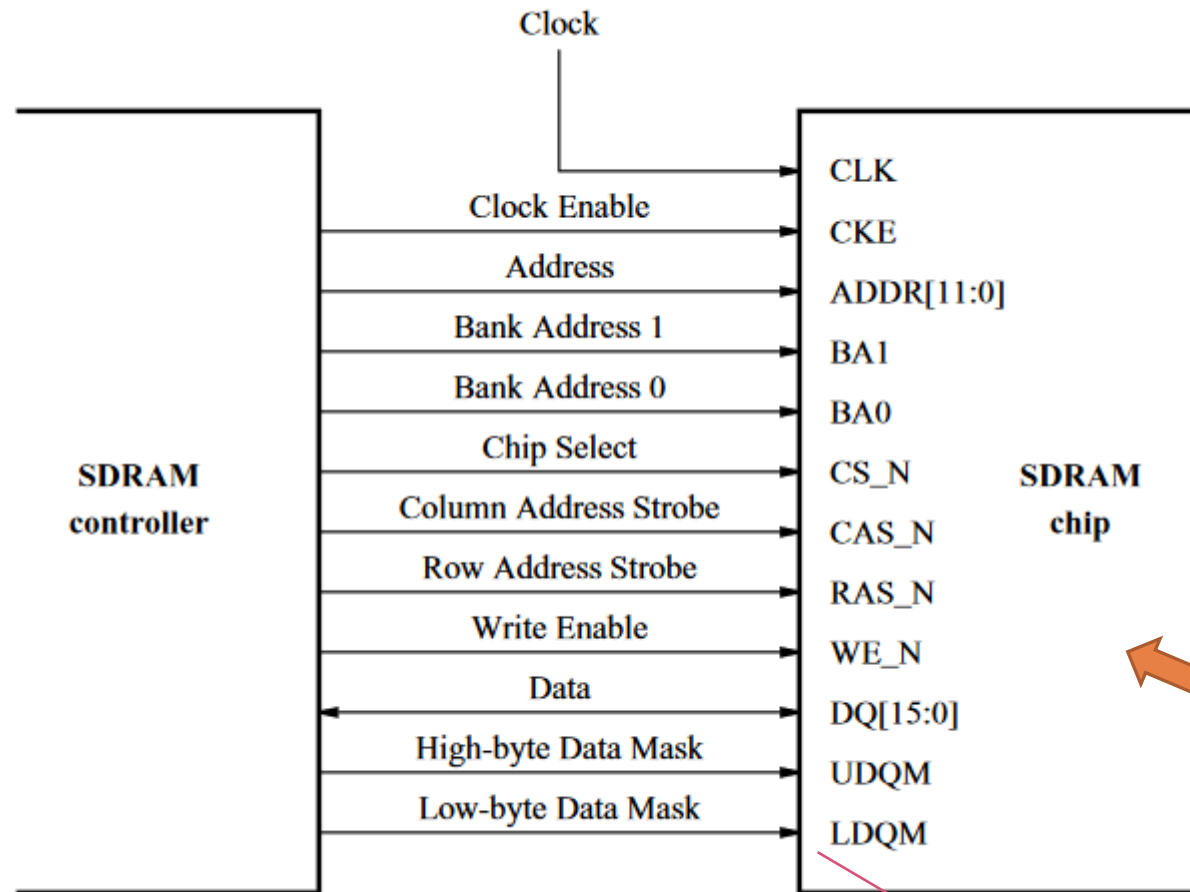
Device drivers are provided in the HAL system library for the Nios II processor.

There are no user-configurable settings.

(No parameters)


CONTROLADOR DE MEMÓRIA RAM DINÂMICA

- SDRAM 8-Mbyte Single Data Rate Synchronous Dynamic RAM memory chip (1M x 16 bits x 4 bancos)



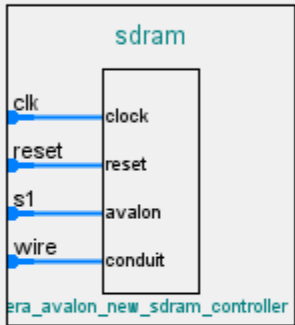
Na documentação da placa, divide em 02 pinos, mas o componente SDRAM no Quartus possui um array DQM[1..0], o qual deve ter a pinagem ajustada

CONTROLADOR DE MEMÓRIA RAM DINÂMICA: DE2

**SDRAM Controller**
altera_avalon_new_sdram_controller

Documentation

Block Diagram
☐ Show signals



era_avalon_new_sdram_controller

Memory Profile **Timing**

Data Width
Bits: 16 **DQ[15..0]**

Architecture
Chip select: 1 **CS_N**
Banks: 4

Address Width
Row: 12 **ADDR[11..0]**
Column: 8


Generic Memory model (simulation only)
☐ Include a functional memory model in the system testbench

Memory Size = 8 MBytes
4194304 x 16
64 MBits

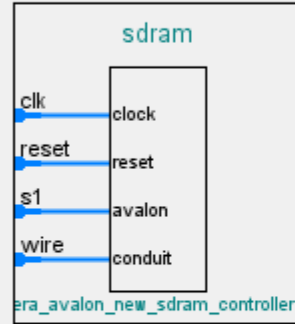
Presets

☐ Project
Library

- Four SDR 100 8MByte x16 chips
- Micron MT8LSDT 1664HG module
- single Alliance AS4LC1M16S1 10 chip
- single
- single
- single
- single

**SDRAM Controller**
altera_avalon_new_sdram_controller

Block Diagram
☐ Show signals



era_avalon_new_sdram_controller

Memory Profile **Timing**

CAS latency cycles::
☐ 1
☐ 2
☒ 3

Initialization refresh cycles: 2

Issue one refresh command every: 15.625 us

Delay after powerup, before initialization: 200.0 us

Duration of refresh command (t_rfc): 60.0 ns

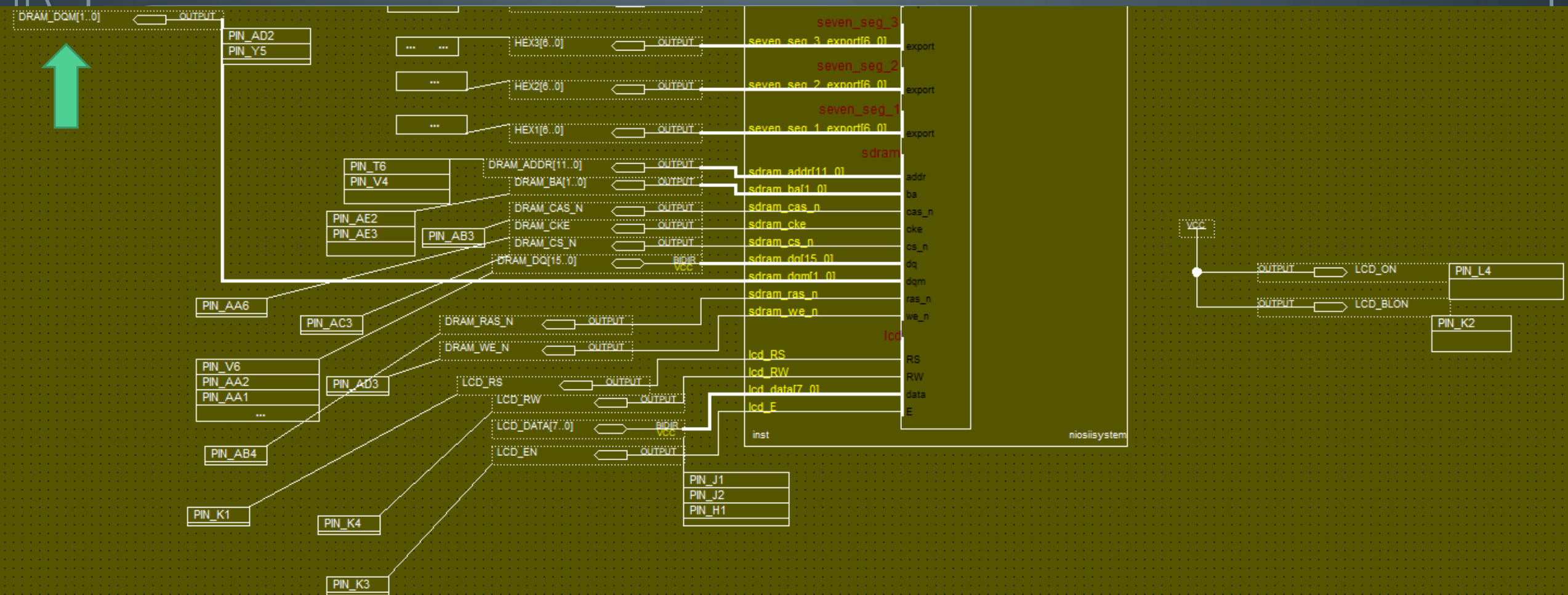
Duration of precharge command (t_rp): 18.0 ns

ACTIVE to READ or WRITE delay (t_rcd): 18.0 ns

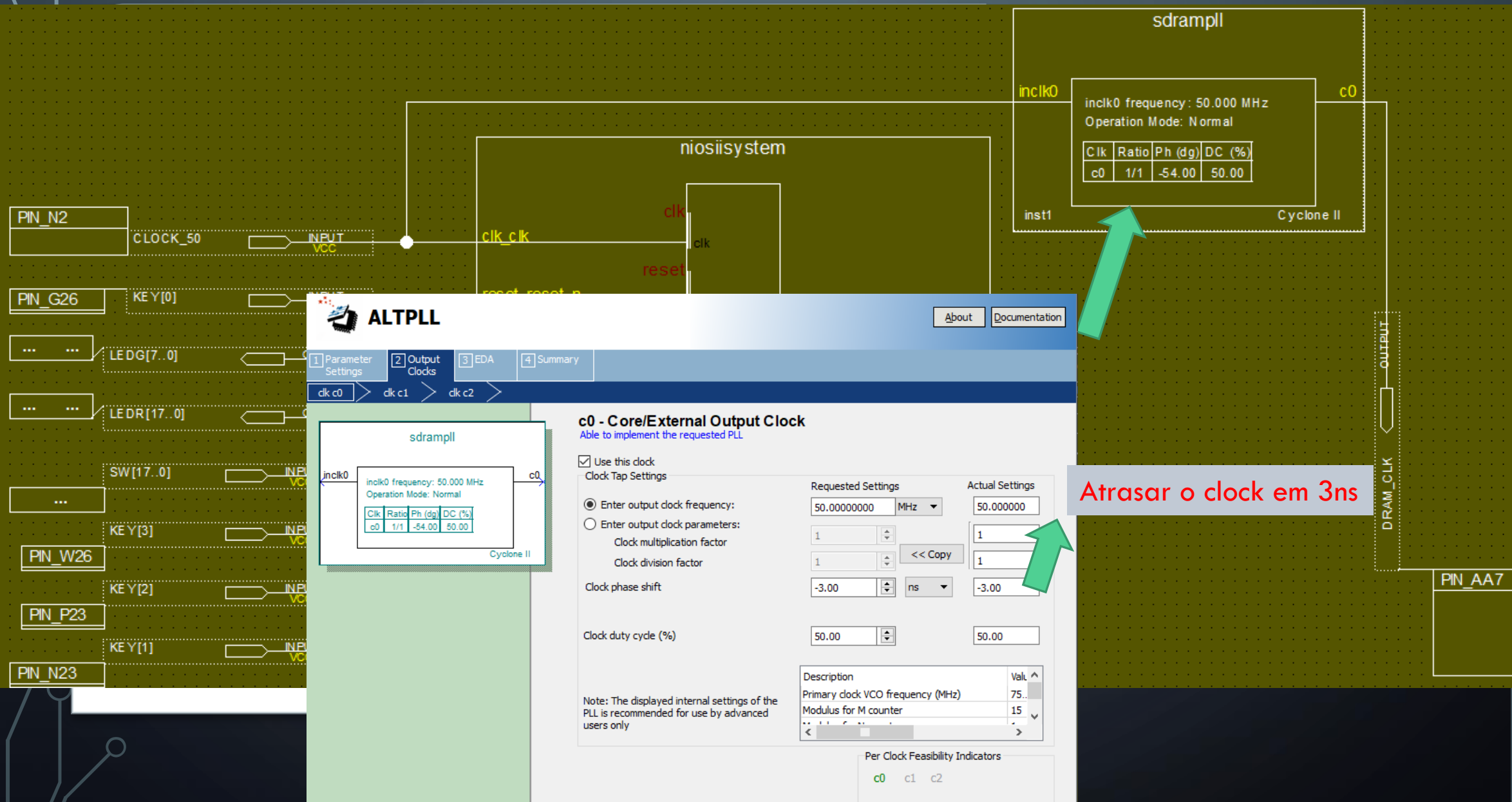
Access time (t_ac): 5.5 ns

Write recovery time (t_wr, no auto precharge): 14.0 ns

COMPONENTE .BSF GERADO COM O SOC



COMPONENTE .BSF GERADO COM O SOC – CLOCK DA SDRAM



RETORNANDO AO QUARTUS

- Adicionar ao projeto o arquivo .qip do SoC criado no QSYS
- Colocar na pasta de projeto o arquivo de2.qsf
- Compilar para atribuição automática de pinos, com os nomes tal como no QSF
- Fazer o deploy (Tools-Programmer) para a placa
- Desenvolver o software o qual será executado como uma Nios II Application

EXEMPLO DE CÓDIGO

```
#include <stdio.h>
#include <string.h>
#include "altera_avalon_pio_regs.h"
//#include <altera_avalon_lcd_16207.h>
//#include <altera_avalon_lcd_16207_regs.h>
//#include <altera_avalon_lcd_16207_fd.h>
#include "system.h"
#include <unistd.h> // usleep.h

int main()
{
    /*Nios II acessa o controlador de LCD como um dispositivo STDOUT, e pode ser acessado com métodos de arquivo, such as printf().
    O controlador LCD pode ser facilmente integrado à arquitetura no QSYS. Se escolher no BSP Reduced Drives, não usa LCD*/

    double x = 10.2;
    FILE *lcd_d;
    lcd_d = fopen(LCD_NAME, "w");

    if (lcd_d == NULL) {
        fprintf(stderr, "Erro abertura\n");
        return 0;
    }
    fprintf(lcd_d, "TADS UFRN \n"); // se o tamanho da string exibida não ultrapassar 16, não gera scroll
    fprintf(lcd_d, "x: %.2f", x); // segunda linha
    fclose(lcd_d);

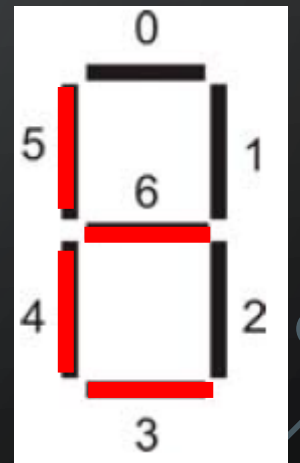
    while(1) { // Este laço exibe permanentemente a frase TADS UFRN nos displays de 7 segmentos
        IOWR_ALTERA_AVALON_PIO_DATA(SEVEN_SEG_7_BASE, 0x07); // t
        IOWR_ALTERA_AVALON_PIO_DATA(SEVEN_SEG_6_BASE, 0x08); // A
        IOWR_ALTERA_AVALON_PIO_DATA(SEVEN_SEG_5_BASE, 0x21); // d
        IOWR_ALTERA_AVALON_PIO_DATA(SEVEN_SEG_4_BASE, 0x12); // S
        IOWR_ALTERA_AVALON_PIO_DATA(SEVEN_SEG_3_BASE, 0x41); // U
        IOWR_ALTERA_AVALON_PIO_DATA(SEVEN_SEG_2_BASE, 0x0E); // F
        IOWR_ALTERA_AVALON_PIO_DATA(SEVEN_SEG_1_BASE, 0x2F); // r
        IOWR_ALTERA_AVALON_PIO_DATA(SEVEN_SEG_0_BASE, 0x2B); // n
    }
    return 0;
}
```

Entendendo o display
HEX -> BIN

0x07 = 0 0 0 0 0 1 1 1
 = x 6 5 4 3 2 1 0

Liga: 0, desliga 1

Logo:



EXERCÍCIO DE FIXAÇÃO

1) Com base na arquitetura de SoC desenvolvida, desenvolver um letreiro digital com o texto desejado, limitado a 8 caracteres e que seja factível de exibição num display de 7 segmentos

1.1) deve haver uma chave TOGGLE (SWITCH) que permita escolher o sentido do letreiro – E->D ou D->E

1.2) as letras devem ir trocando de displays, com uma frequência (usleep) desejada

1.3) no LCD pode ser exibido o sentido da direção escolhida com as palavras LEFT e RIGHT

ESTE PROJETO DEVE SER DESENVOLVIDO EM C, FAZENDO O LETREIRO FUNCIONAR. DEIXAM-SE AS CONSTANTES ASSOCIADAS A CADA LETRA PRONTAS NO CÓDIGO, E NO LABORATÓRIO IREMOS PASSAR PARA A PLACA E FAZER AJUSTES