



PROJETO DE RELÓGIO DIGITAL

Experimento #P2

PCS3335 - Laboratório Digital A

14/07/2022

Natanael Magalhães Cardoso, 8914122

Renato Naves Fleury, 11805269

Professor: Glauber de Bona

Turma: 10

Bancada: B3

Universidade de São Paulo

Escola Politécnica

Departamento de Eng. de
Computação e Sistemas Digitais



UNIVERSIDADE DE SÃO PAULO

ESCOLA POLITÉCNICA

Departamento de Eng. de Computação e Sistemas Digitais



Projeto de Relógio Digital

Natanael Magalhães Cardoso, Renato Naves Fleury

1. INTRODUÇÃO

Uma FPGA é um circuito integrado criado para ser configurado por um projetista após sua fabricação com uma linguagem de descrição de hardware. Neste projeto, será usada uma FPGA na implementação da unidade de controle de um circuito digital de um relógio digital. Já o fluxo de dados será implementado utilizando circuitos integrados não programáveis (CIs TTL da família 74).

2. OBJETIVOS

O objetivo desse projeto consiste em projetar o circuito digital de um relógio com dois dígitos para os segundos e dois dígitos para os minutos. Além disso, o relógio deve ser programável e controlável.

3. PLANEJAMENTO

3.1. DESCRIÇÃO FUNCIONAL

O sistema digital em questão tem como principal função a contagem do tempo. Ele pode ser controlado por 5 sinais de controle (clock, Clear, Programa, Grava, Inibe) e possui dois modos de operação, o modo "Programação" (Programa = nível lógico "1") e o modo "Operação" (Programa = nível lógico "0").

Quando no modo Programação, é possível configurar o valor de minutos e segundos desejados, basta ajustar os valores desejados na entrada de dados e colocar o sinal Grava em nível lógico "1", no clock seguinte o sinal de entrada será configurado na saída. O sinal Grava só possui efeito nesse modo de operação.

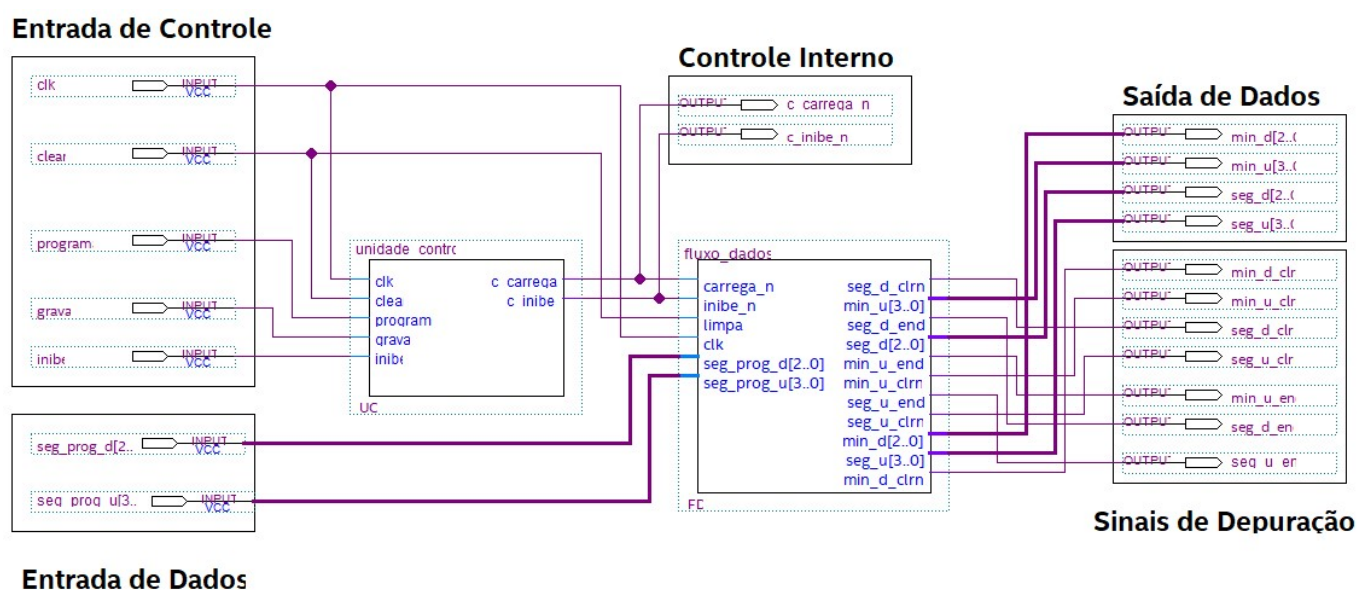
Já o modo Operação funciona da seguinte maneira: se o sinal Inibe estiver em nível lógico baixo a saída contará o tempo a partir do valor de tempo configurado, caso contrário, se o sinal Inibe

estiver em nível lógico alto, a contagem não ocorrerá.

O sinal clear zera as saídas no clock seguinte.

3.2. DIAGRAMA DE BLOCOS

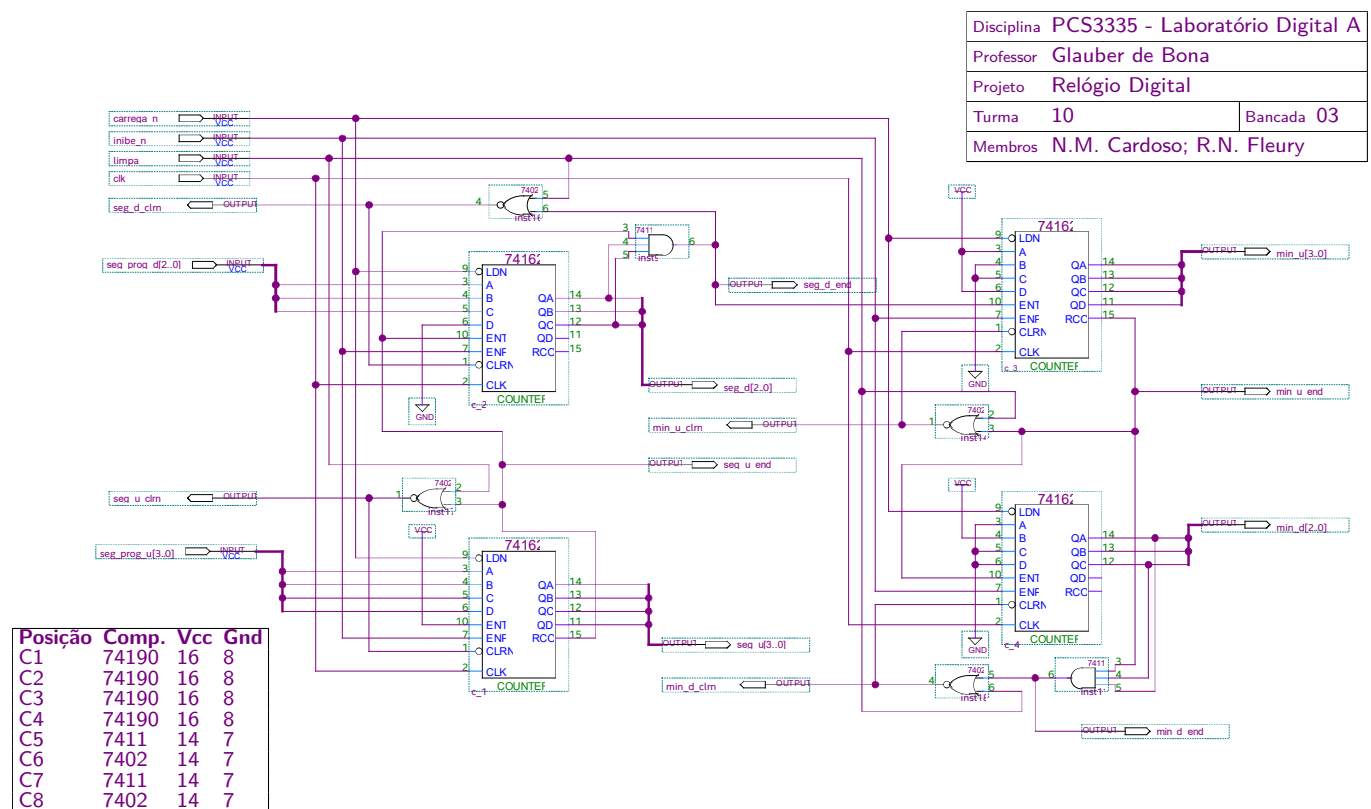
A Fig. 1 mostra o diagrama de blocos do circuito do relógio digital. O circuito recebe os sinais de controle clock, clear, programa, grava e inibe. Além disso, também recebe como entrada de dados 14 bits que correspondem aos valores de minutos e segundos a serem programados. Como saída, o circuito fornece 14 bits de dados que correspondem ao tempo atual que muda a cada clock, e sinais de depuração. A Unidade de Controle controla o Fluxo de Dados a partir dos sinais $\overline{c_carrega}$ e $\overline{c_inibe}$.



■ **Figura 1:** Diagrama de blocos do circuito do Relógio Digital

3.3. DIAGRAMA LÓGICO

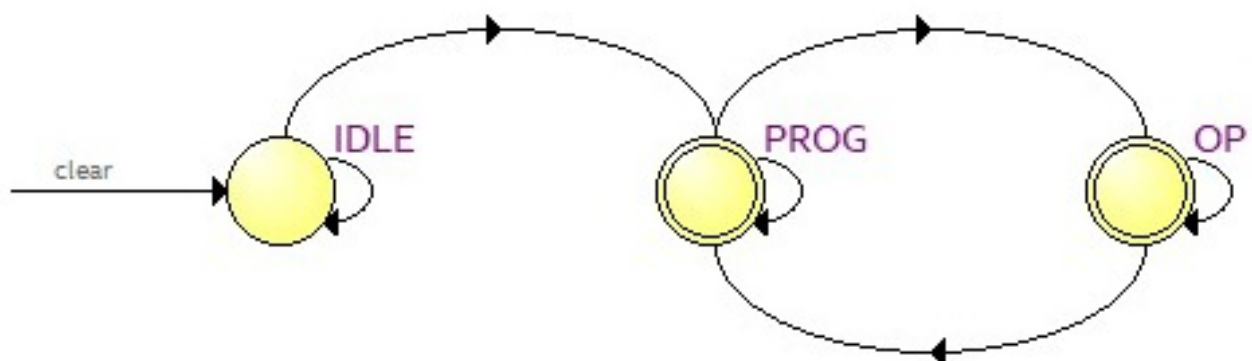
A Fig.2 mostra a implementação do fluxo de dados utilizando-se 4 contadores 74190, 2 CIs de portas AND3 7411 e 2 CIs de portas NOR2 7411.



■ **Figura 2:** Diagrama lógico do Fluxo de Dados

3.4. DIAGRAMA DE ESTADOS

A máquina de estados implementada na unidade de controle é composta por 3 estados, o primeiro consiste em um estado inicial do qual só se pode ir para o próximo estado que é o estado de programação onde o valor de minutos e segundos é carregado nos contadores do fluxo de dados, o outro estado é o de operação no qual o relógio começa a contar. Ao resetar o circuito, ele volta para o estado IDLE e só volta para o estado OP depois de passar pelo estado PROG. Isto é, o relógio só volta a contar depois de ser reprogramado.



■ **Figura 3:** Diagrama de estados da unidade de controle

3.5. SIMULAÇÃO

As Figs. 4, 5 e 6 mostram as cartas de tempos da simulação do fluxo de dados, unidade de controle e circuito completo, respectivamente.

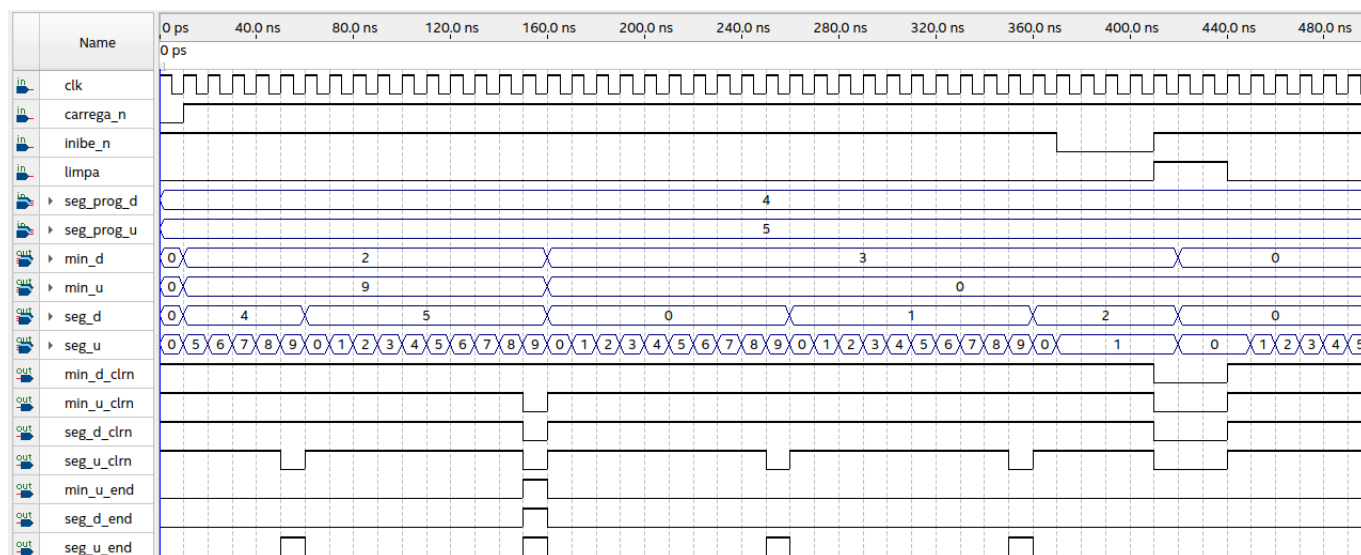


Figura 4: Carta de tempos do fluxo de dados

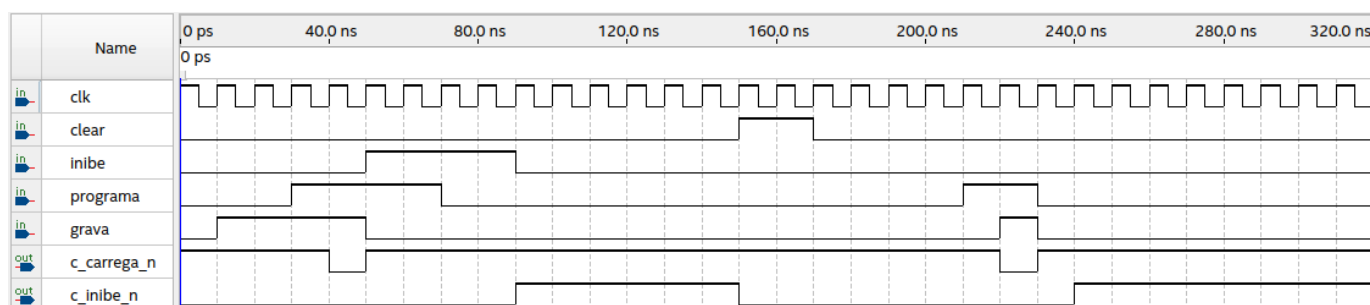
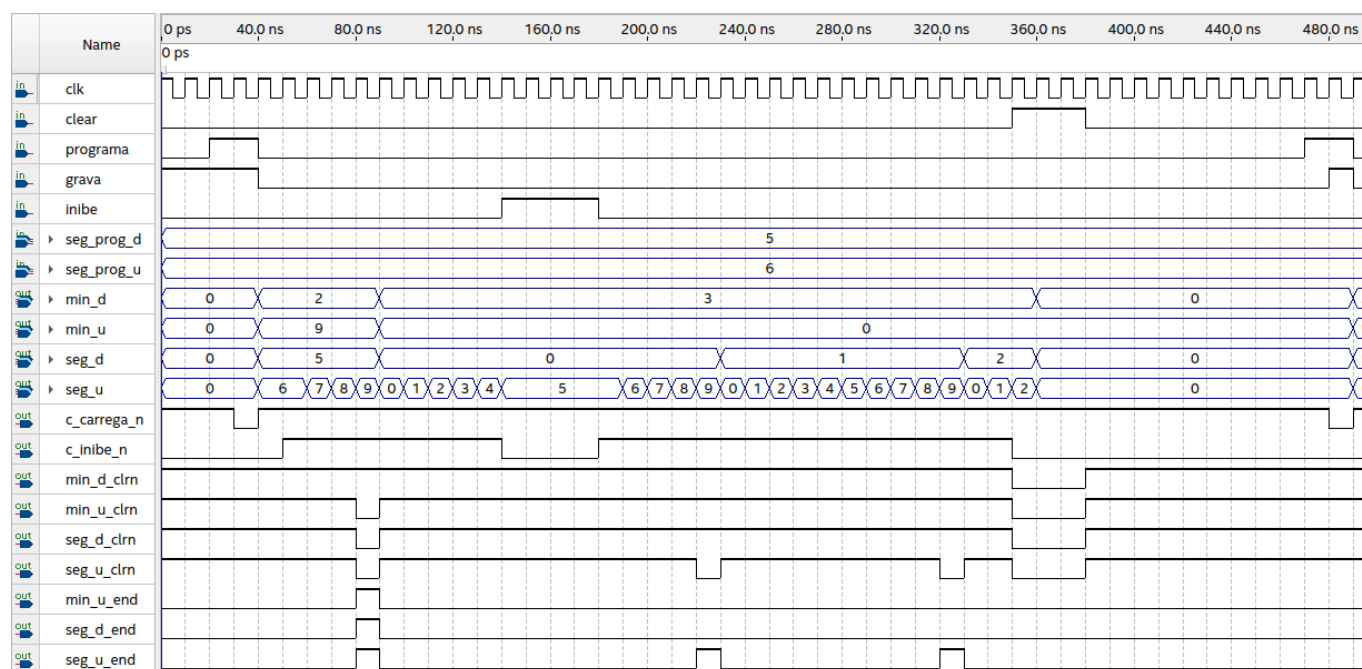


Figura 5: Carta de tempos da unidade de controle



■ **Figura 6:** Carta de tempos do circuito completo

3.6. TESTES

As Tabelas 1, 2 e ?? mostram as tabelas de testes da unidade de controle, do fluxo de dados e do circuito completo, respectivamente.

■ Tabela 1: Tabela de testes da unidade de controle

Entradas					Saídas	
clk	clear	inibe	programa	grava	c_carrega_n	c_inibe_n
↑	0	0	0	0	1	0
↑	0	0	0	1	1	0
↑	0	0	1	1	1	0
↑	0	0	1	1	0	0
↑	0	1	1	0	1	0
↑	0	1	0	0	1	0
↑	0	0	0	0	1	1
↑	1	0	0	0	1	0
↑	0	0	1	0	1	0
↑	0	0	1	1	0	0
↑	0	0	0	0	1	0
↑	0	0	0	0	1	1

■ **Tabela 2:** Tabela de testes do fluxo de dados com sinais os seguintes sinais: (1) clk, (2) clear, (3) carrega_n, (4) inibe_n, (5) seg_prog_d, (6) seg_prog_u, (7) min_d, (8) min_u, (9) seg_d, (10) seg_u, (11) min_d_clrn, (12) min_u_clrn, (13) seg_d_clrn, (14) seg_u_clrn, (15) min_u_end, (16) min_d_end, (17) seg_u_end

Entradas						Saídas				Depuração						
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
↑	1	×	×	×	×	000	0000	000	0	1	1	1	1	0	0	0
↑	0	0	1	101	0110	000	0000	000	0	1	1	1	1	0	0	0
↑	0	0	1	101	0110	010	1001	101	0	1	1	1	1	0	0	0
↑	0	0	1	101	0110	010	1001	101	1	1	1	1	1	0	0	0
↑	0	0	1	101	0110	010	1001	101	0	1	1	1	1	0	0	0
↑	0	0	1	101	0110	010	1001	101	0	1	1	1	1	0	0	0
↑	0	0	1	101	0110	010	1001	101	0	0	0	0	0	1	1	1
↑	0	0	1	101	0110	011	0000	000	0	1	1	1	1	0	0	0
↑	0	0	1	101	0110	011	0000	000	0	1	1	1	1	0	0	0
↑	0	0	1	101	0110	011	0000	000	0	1	1	1	1	0	0	0
↑	0	0	1	101	0110	011	0000	000	0	1	1	1	1	0	0	0
↑	0	0	1	101	0110	011	0000	000	0	1	1	1	1	0	0	0
↑	0	0	0	101	0110	011	0000	000	0	1	1	1	1	0	0	0
↑	0	0	1	101	0110	011	0000	000	0	1	1	1	1	0	0	0
↑	0	0	1	101	0110	011	0000	000	0	1	1	1	1	0	0	0
↑	0	0	1	101	0110	011	0000	000	0	1	1	1	1	0	0	0
↑	0	0	1	101	0110	011	0000	000	0	1	1	1	1	0	0	0
↑	0	0	1	101	0110	011	0000	000	0	1	1	1	0	0	0	1

■ **Tabela 3:** Tabela de testes no circuito completo com os seguintes sinais: (1) clk, (2) clear, (3) programa, (4) grava, (5) inibe, (6) seg_prog_d, (7) seg_prog_u, (8) min_d, (9) min_u, (10) seg_d, (11) seg_u, (12) c_carrega_n, (13) c_inibe_n, (14) min_d_clrn, (15) min_u_clrn, (16) seg_d_clrn, (17) seg_u_clrn, (18) min_u_end, (19) min_d_end, (20) seg_u_end

Entradas							Saídas				Depuração								
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	
↑	1	×	×	×	×	×	000	0000	000	0000	1	0	0	0	0	0	0	0	
↑	0	1	0	0	101	0110	000	0000	000	0000	1	0	1	1	1	1	0	0	
↑	0	1	1	0	101	0110	000	0000	000	0000	0	0	1	1	1	1	0	0	
↑	0	1	1	0	101	0110	010	1001	101	0110	1	0	1	1	1	1	0	0	
↑	0	1	1	0	101	0110	010	1001	101	0110	1	1	1	1	1	1	0	0	
↑	0	1	1	0	101	0110	010	1001	101	0111	1	0	1	1	1	1	0	0	
↑	0	1	1	0	101	0110	010	1001	101	1000	1	0	1	1	1	1	0	0	
↑	0	1	1	0	101	0110	010	1001	101	1001	1	0	0	0	0	0	1	1	
↑	0	1	1	0	101	0110	011	0000	000	0000	1	0	1	1	1	1	0	0	
↑	0	1	1	0	101	0110	011	0000	000	0001	1	0	1	1	1	1	0	0	
↑	0	1	1	0	101	0110	011	0000	000	0010	1	0	1	1	1	1	0	0	
↑	0	1	1	0	101	0110	011	0000	000	0011	1	0	1	1	1	1	0	0	
↑	0	1	1	0	101	0110	011	0000	000	0100	1	0	1	1	1	1	0	0	
↑	0	1	1	1	101	0110	011	0000	000	0101	1	0	1	1	1	1	0	0	
↑	0	1	1	0	101	0110	011	0000	000	0101	1	0	1	1	1	1	0	0	
↑	0	1	1	0	101	0110	011	0000	000	0110	1	0	1	1	1	1	0	0	
↑	0	1	1	0	101	0110	011	0000	000	0111	1	0	1	1	1	1	0	0	
↑	0	1	1	0	101	0110	011	0000	000	1000	1	0	1	1	1	1	0	0	
↑	0	1	1	0	101	0110	011	0000	000	1001	1	0	1	1	1	0	0	0	

3.7. LEVANTAMENTO DOS MATERIAIS NECESSÁRIOS

■ **Tabela 4:** Unidades requeridas para cada CI

Slot	Operação	CI	Un. Requeridas	Un. Disponíveis
1	Contador	74190	1	1
2	Contador	74190	1	1
3	Contador	74190	1	1
4	Contador	74190	1	1
5	AND3	7411	1	2
6	OR2	7402	2	4
7	AND3	7411	1	2
8	OR2	7402	2	4

Para garantir que o circuito projetado respeite as restrições de montagem, fizemos um levantamento dos recursos necessários para este circuito mostrado na Tabela 4. Ela mostra a quantidade de unidades lógicas requeridas para cada CI utilizado. As especificações de cada CI foi obtido pelos respectivos *datasheets*.

3.8. MONTAGEM E DEPURAÇÃO

O circuito será montado e depurado por partes. A Tabela 5 mostra a correlação dos sinais de depuração e dos LEDs da placa de montagem do fluxo de dados e a Tabela 6 mostra a designação dos pinos da unidade de controle. Os sinais de saída (segundos e minutos) serão mostrados nos displays da placa de montagem.

■ **Tabela 5:** Tabela de correlação dos sinais de depuração, entrada e saída com os LEDs, chaves e displays da placa de montagem do Fluxo de Dados

Sinal	Endereço da Porta	LED/Display/Chave	Tipo E/S/D
<i>Seg_prog_u</i> [0]	74162:1:3	Chave 0	Entrada
<i>Seg_prog_u</i> [1]	74162:1:4	Chave 1	Entrada
<i>Seg_prog_u</i> [2]	74162:1:5	Chave 2	Entrada
<i>Seg_prog_u</i> [3]	74162:1:6	Chave 3	Entrada
<i>Seg_prog_d</i> [0]	74162:2:3	Chave 4	Entrada
<i>Seg_prog_d</i> [1]	74162:2:4	Chave 5	Entrada
<i>Seg_prog_d</i> [2]	74162:2:5	Chave 6	Entrada
<i>Seg_u</i> [0]	74162:1:14	Display D0:1	Saída
<i>Seg_u</i> [1]	74162:1:13	Display D0:2	Saída
<i>Seg_u</i> [2]	74162:1:12	Display D0:3	Saída
<i>Seg_u</i> [3]	74162:1:11	Display D0:4	Saída
<i>Seg_d</i> [0]	74162:2:14	Display D1:1	Saída
<i>Seg_d</i> [1]	74162:2:13	Display D1:2	Saída
<i>Seg_d</i> [2]	74162:2:12	Display D1:3	Saída
<i>Min_u</i> [0]	74162:3:14	Display D2:1	Saída
<i>Min_u</i> [1]	74162:3:13	Display D2:2	Saída
<i>Min_u</i> [2]	74162:3:12	Display D2:3	Saída
<i>Min_u</i> [3]	74162:3:11	Display D2:4	Saída
<i>Min_d</i> [0]	74162:4:14	Display D3:1	Saída
<i>Min_d</i> [1]	74162:4:13	Display D3:2	Saída
<i>Min_d</i> [2]	74162:4:12	Display D3:3	Saída
<i>Seg_u_clrn</i>	7402:1:1	LED0	Depuração
<i>Seg_d_clrn</i>	7402:1:4	LED1	Depuração
<i>Min_u_clrn</i>	7402:2:1	LED2	Depuração
<i>Min_d_clrn</i>	7402:2:4	LED3	Depuração
<i>Seg_u_end</i>	74162:1:15	LED4	Depuração
<i>Seg_d_end</i>	7411:1:6	LED5	Depuração
<i>Min_u_end</i>	74162:3:15	LED6	Depuração
<i>Min_d_end</i>	7411:2:6	LED7	Depuração

■ Tabela 6: Tabela de designação de pinos da Unidade de Controle para a placa FPGA DE0-CV com Cyclone V 5CEBA4F23C7

Sinal	Código Interface	Código FPGA	Tipo E/C
clk	GPIO_0_D0	N16	Entrada
clear	GPIO_0_D4	D17	Entrada
programa	GPIO_0_D5	K20	Entrada
grava	GPIO_0_D6	K21	Entrada
inibe	GPIO_0_D7	K22	Entrada
c_carrega	GPIO_1_D4	A13	Controle
c_inibe	GPIO_1_D5	B13	Controle

4. MONTAGEM E RESULTADOS

O fluxo de dados foi montado e testado por partes de acordo com a seguinte sequência seguinte sequência: (1) dezena de segundo; (2) dezena e unidade de segundo; (3) dezena de minuto e (4) dezena e unidade de minuto.

O objetivo dos teste das dezenas foi certificar a contagem de 0 a 5 e, durante a monagem e testes, as entradas dos CIs 74162 (contador) e 7411 (and) que dependiam do sinal RCO do dígito de unidade foram substituídos por VCC. Já o objetivo do teste dos dígitos de dezena e unidade foi testar a contagem de 0 a 60 com a devida sincronização dos dois dígitos.

Foi considerado o funcionamento correto da Unidade de Controle, já que ela havia passado por simulações e o circuito completo já havia sido construído e testado com sucesso em um open lab. Desta forma, os testes mencionados com o fluxo de dados foram feitos já com os sinais de controle provenientes da UC e os sinais externos (clock, reset, programa, grava e inibe) foram inseridos pelo Analog Discovery.

Não houve nenhum problema durante os testes individuais, apenas na execução do circuito completo foi detectado um teste com o sinal clear, na qual o dígito de unidade de minuto não estava sendo limpo. O problema foi resolvindo com a substituição do contador (CI 74162) por outro.

O circuito completo (Fig. B) foi testado de acordo com a Tabela 3 com uma modificação do valor inicial do minuto para 59. Desta forma, foram testadas a troca de segundo, a troca de minuto e a troca de hora.

Além disso, não houve uso dos CI's conversores de tensão durante a montagem, pois o circuito já foi projetado para evitar o seu uso.

5. CONCLUSÃO

Neste experimento, um circuito de relógio digital foi projetado e implementado usando a separação em dois componentes: a Unidade de Controle e o Fluxo de Dados. O primeiro foi projetado em VHDL e implementado em uma FPGA Cyclone V e o segundo foi implementado no painel de montagens usando circuitos integrados TTL. O resultado da execução do circuito completo está condizente e satisfatória com relação às simulações. Além disso, a decisão de não usar conversores de tensão funcionou corretamente, já que poucos CI's foram utilizados e as ligações em série foram evitadas, sendo substituídas por ligações em paralelo para evitar queda de tensão.

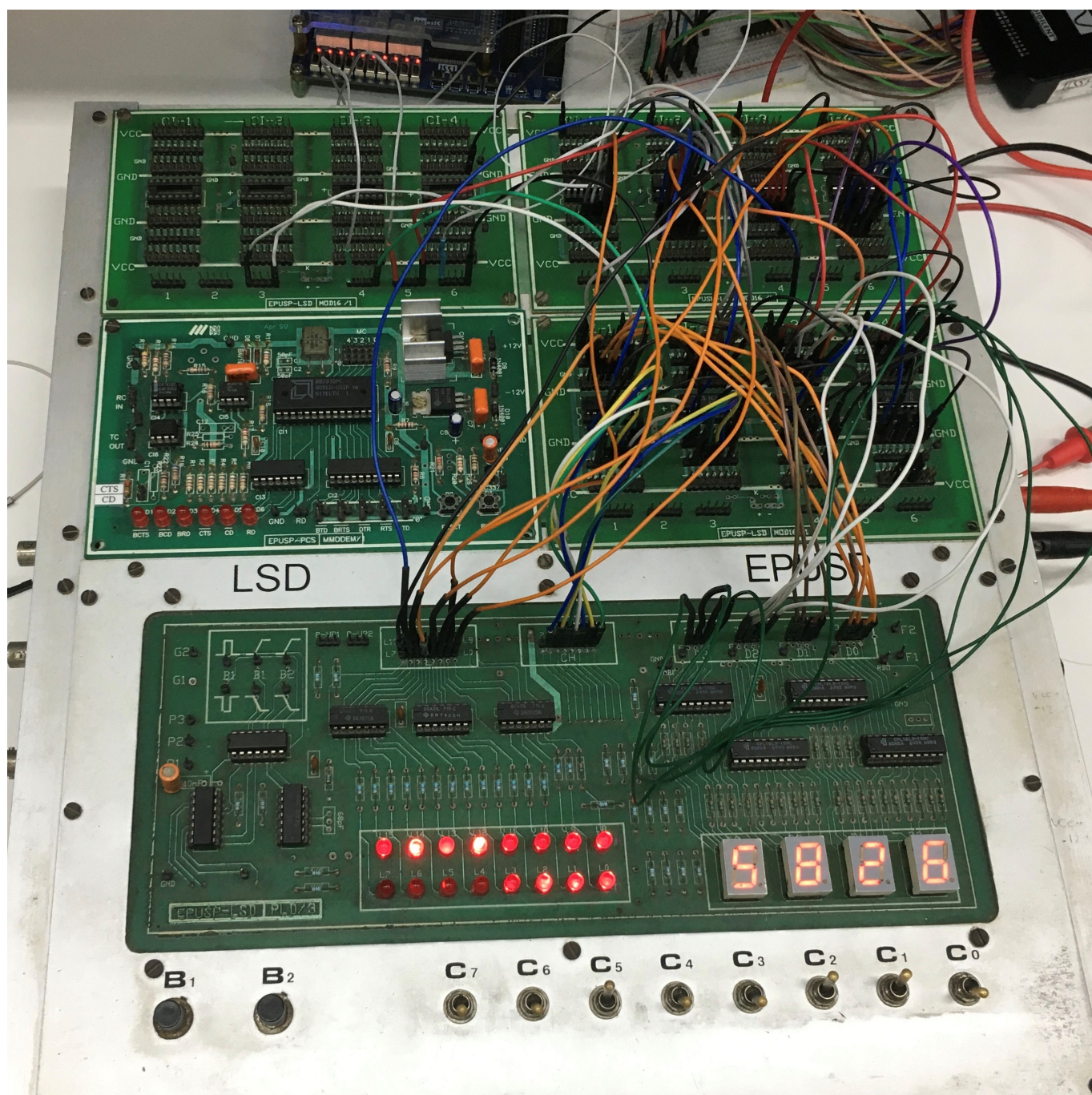
APÊNDICE

A. DESCRIÇÕES VHDL

Listing 1: Descrição da Unidade de Controle

```
1  library IEEE;
2  use IEEE.std_logic_1164.all;
3  use IEEE.numeric_std.all;
4
5  entity unidade_controle is
6      port (
7          clk, clear, programa, grava, inibe: in std_logic;
8          c_carrega_n, c_inibe_n: out std_logic
9      );
10 end unidade_controle;
11
12 architecture unidade_controle_arch of unidade_controle is
13     type estado_tipo is (PROG, OP, IDLE);
14     signal estado_atual, prox_estado: estado_tipo;
15 begin
16     sincrono: process(clk, clear, prox_estado)
17     begin
18         if clear='1' then
19             estado_atual <= IDLE;
20         elsif clk'event and clk='1' then
21             estado_atual <= prox_estado;
22         end if;
23     end process;
24
25     prox_estado <=
26         IDLE when estado_atual = IDLE and programa = '0' else
27         PROG when programa = '1' else
28         OP;
29     c_inibe_n <= not inibe when estado_atual = OP else '0';
30     c_carrega_n <= not grava when estado_atual = PROG else '1';
31 end architecture;
```

B. MONTAGEM DO CIRCUITO



■ Figura 7: Montagem do circuito completo



Universidade de São Paulo

Escola Politécnica

Departamento de Eng. de Computação e Sistemas Digitais