

Disciplina: PCS 3335 – Laboratório Digital A	
Prof.: <i>Glauber De Bona</i>	Data: 11/07
Turma: <i>Glauber - T04</i>	Bancada: 08
Membros:	
<i>11261531 - Enzo Bustos Da Silva</i>	
<i>10379694 - Davi Augusto Bandeira</i>	



P2 **Relógio Digital**

1. Introdução

Este experimento servirá como avaliação P2 para os alunos do Laboratório Digital, visando consolidar os conhecimentos sobre a dinâmica da Unidade de Controle e Fluxo de Dados com suas implementações em sala de aula no painel de montagens, com circuitos integrados, e na placa FPGA.

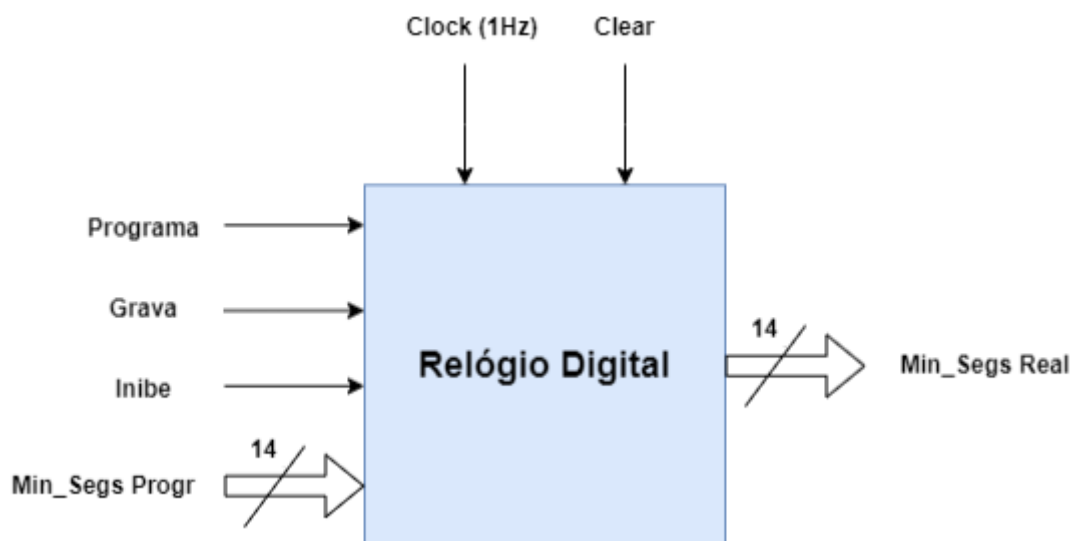
Neste experimento será montado um circuito de Relógio Digital, que mostrará os minutos e segundos em um display de 7 segmentos no painel.

Apesar deste relatório ter sido feito em dupla, a prova e o relatório final será feito individualmente em duas datas. O integrante 11261531 - *Enzo Bustos Da Silva* irá comparecer na primeira semana no dia 12/07 e o aluno 10379694 - *Davi Augusto Bandeira* na segunda semana no dia 19/07.

2. Objetivo

O objetivo do projeto consiste em montar um relógio digital que registra minutos e segundos, explicitando-os em um display de 7 segmentos. Para tal, será feita uma divisão entre Unidade de Controle (UC) e Fluxo de Dados (FD) para tratar, respectivamente, dos sinais de controle e a entrada/saída de dados.

A figura abaixo, retirada da apostila, mostra o diagrama de blocos em alto nível do projeto.



3. Planejamento

a) Descrição funcional do sistema

O funcionamento do relógio digital depende dos controles que são aplicados. Tendo, por princípio, dois modos de operação que dependem do sinal "Programa": Modo Programação (em nível lógico alto) e Modo Operação (em nível lógico baixo).

Dentro do Modo Programação, existem duas funções diferentes em função do comando "Grava". Caso este sinal esteja em nível lógico alto, temos

a função de gravação, em que a entrada “Min_Seg_Prog” é copiada para a saída “Min_Seg_Real”, sendo este comando utilizado para definir um horário desejado pelo usuário. No caso em que “Grava” fica em nível lógico baixo, o relógio digital apenas retém o valor da saída, ou seja, o relógio fica com a sua contagem travada. Isto foi definido como escolha de projeto, já que não foi especificado no enunciado se a contagem deveria continuar no modo Programação.

Assim, apenas quando o sistema está no Modo Operação é que o relógio é, de fato, incrementado conforme o sinal de clock. A exceção é caso o comando “Inibe” esteja ativado, o que interrompe a contagem. Ademais, é evidente que o comando “Grava” não tem efeito no sistema em Modo Operação, assim como “Inibe” também não tem efeito no Modo Programação.

O comando de “Clear” funciona como um reset global do sistema, zerando a saída de todos os contadores TTL 74161. Vale destacar que este Clear é síncrono e só ocorre nas bordas de subidas do clock de 1 Hz.

Finalmente, vale destacar que os 14 bits da entrada/saída encapsulam o horário da seguinte maneira:

- 4 bits para unidade (0 a 9 -> “0000” a “1001”)
- 3 bits para dezena (0 a 5 -> “0000” a “101”)

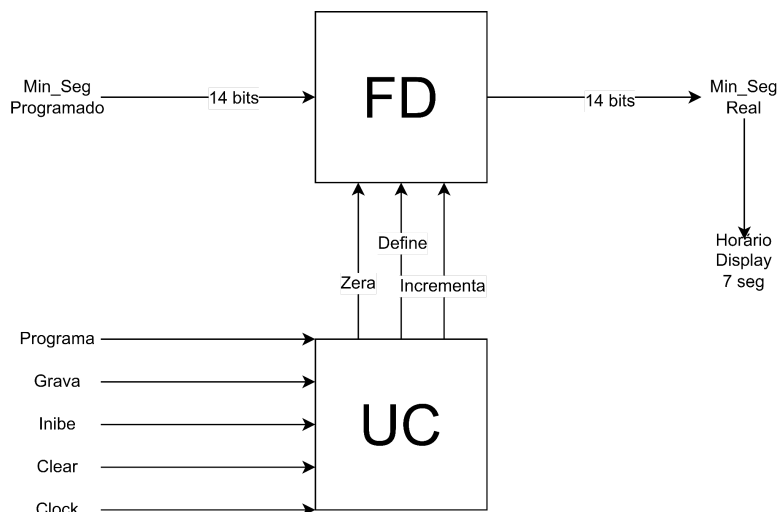
De forma que são 7 bits para os segundos (menos significativos do vetor) e os outros 7 para os minutos (mais significativos).

b) Diagrama de Blocos do sistema completo (FD + UC)

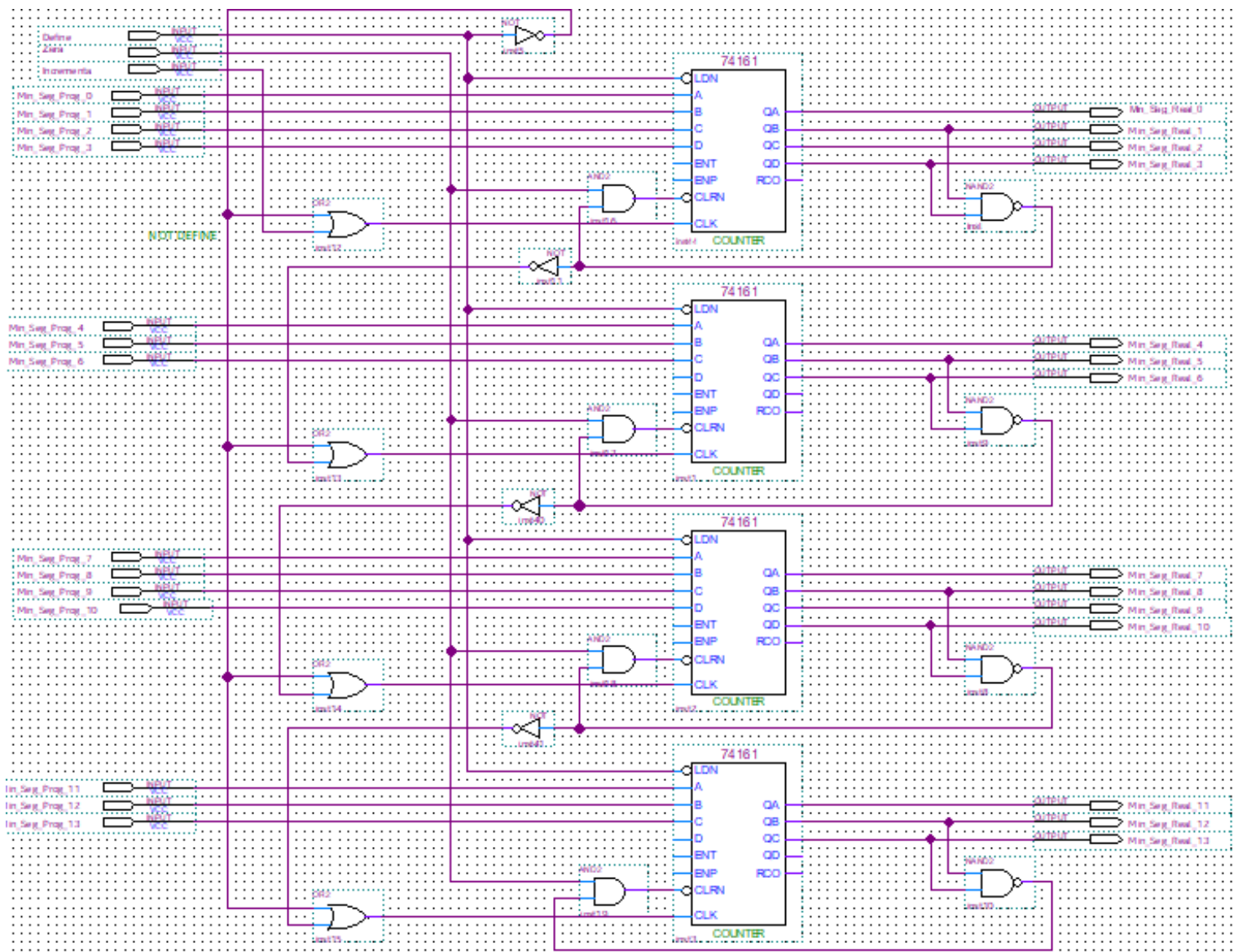
A partir do diagrama de alto nível, foi montado um desenho esquemático que mostra como será feita a divisão do projeto em UC e FD.

Desta forma, a Unidade de Controle receberá os sinais de comando “Programa”, “Grava”, “Inibe”, “Clear” e “Clock” como entrada e terá como saída para o Fluxo de Dados os sinais intermediários “Zera”, “Define” e “Incrementa”. Estas saídas são determinadas em função do estado em qual o sistema se encontra (que por sua vez depende das entradas de controle).

Nesse contexto, o Fluxo de Dados recebe como entrada, além dos sinais intermediários vindos da UC, os 14 bits de dados “Min_Seg_Prog” e tem como saída os 14 bits “Min_Seg_Real” que irão para o display de 7 segmentos, que se encontrará no Painel de Montagens.



c) Diagrama Lógico do Fluxo de Dados



Tem-se acima o Diagrama Lógico do Fluxo de Dados com os seguintes sinais intermediários de controle vindo da UC:

- Incrementa: permite a contagem, indicando que 1 segundo se passou;
- Zera: zera a contagem, dando clear em todos contadores;
- Define: permite a gravação, isto é, define o valor da saída conforme a entrada.

Vale ressaltar que, apesar do contador 74161 possuir um clear assíncrono, o sinal intermediário “Zera” vem da UC de forma sincronizada ao clock, então na prática o clear ocorre de forma sincronizada, mesmo que o FD não receba diretamente o sinal de clock, já que “Zera” é atualizado conforme.

Ademais, os sinais intermediários “Define” e “Zera” são ativos em baixo.

d) Carta de tempos do Fluxo de Dados

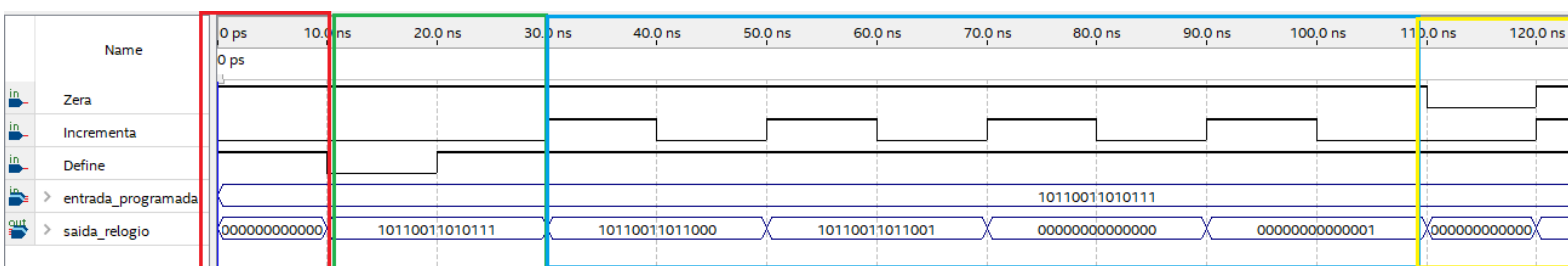
Foi dividido 4 situações na carta de tempos do FD para exemplificar seu funcionamento. Vale destacar novamente que, por mais que o FD não receba diretamente o clock e este não esteja presente na simulação a seguir, todas as transições de valor dos sinais intermediários vindos da UC são feitas de forma sincronizada, portanto no esquema abaixo ele se encontra implícito.

Inicialmente, no retângulo vermelho, todos sinais intermediários estão desativados, logo a saída não se altera.

No momento em que, no retângulo verde, o sinal “Define” (ativo em baixo) é ativo, a entrada programada de 14 bits passa para a saída do relógio. Definimos a entrada para “10110011010111”, correspondente à “59:57”. É evidente que, nesta situação, o sistema se encontra no modo de programação, mais especificamente no estado de gravação.

Em seguida, no retângulo azul, o sinal “Incrementa” dá algumas bordas de subida, fazendo contagens. Percebe-se que o relógio retorna para “00:00” ao atingir o limite em $t = 70\text{ns}$ e continua incrementando. O sistema se encontra, portanto, no modo de operação.

Finalmente, no retângulo amarelo, é ativado o sinal “Zera” (ativo em baixo), zerando a contagem.



e) Tabela de testes Fluxo de Dados

Na tabela de testes, é inicialmente feito uma limpa no sistema, seguido de uma gravação do horário “59:57” (10110011010111 em binário), de forma que após 3 incrementações a saída volta para “00:00”, testando o caso mais extremo possível, onde todos os contadores chegam ao limite de contagem.

Ressalta-se que “Zera” e “Define” são ativos em baixo.

Incrementa	Zera	Define	Entrada (14 bits)	Saída (14 bits)
X	0	X	X	00000000000000
0	1	0	10110011010111	10110011010111
1 (↑)	1	1	X	10110011011000
1 (↑)	1	1	X	10110011011001
1 (↑)	1	1	X	00000000000000

f) Código VHDL da Unidade de Controle

```
library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.numeric_std.all;

entity uc_vhdl_rel is
    port (
        clock, clear, programa, grava, inibe: in std_logic; -- entradas UC
        incrementa, zera, define: out std_logic -- saídas UC pro FD
    );
end uc_vhdl_rel;

architecture arch of uc_vhdl_rel is
    -- estados
    type state_type is (zerado, programacao, gravacao, operacao, inibicao);
    signal PS: state_type;
begin
    transition_of_states: process (clock, clear, programa, grava, inibe)
    begin
        if (rising_edge(clock)) then
            if (clear = '1') then
                PS <= zerado;
            else -- clear = '0'
                if (programa = '1') then
                    if (grava = '0') then
                        PS <= programacao;
                    else -- grava = '0'
                        PS <= gravacao;
                    end if;
                else -- programa = '0'
                    if (inibe = '0') then
                        PS <= operacao;
                    else
                        PS <= inibicao;
                    end if;
                end if;
            end if;
        end if;
    end process transition_of_states;

    action_of_states: process (PS)
    begin
        case PS is
            when zerado =>
                zera <= '0'; -- clear ativo em baixo
                incrementa <= '0';
                define <= '1';

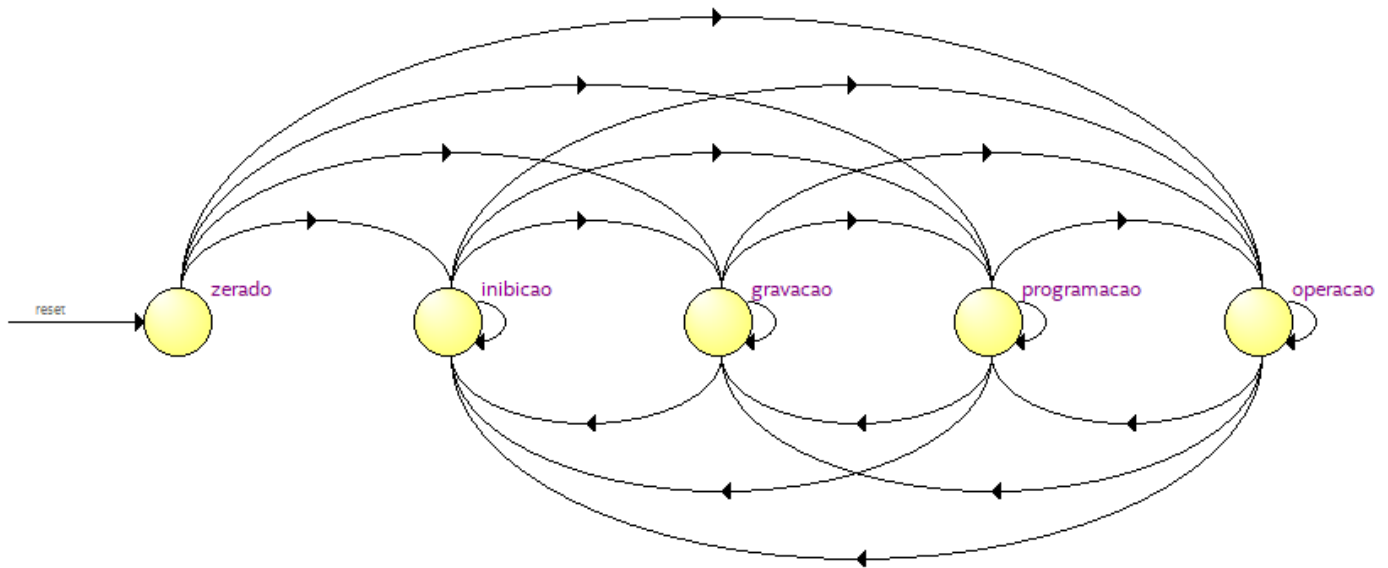
            when programacao =>
                zera <= '1';
                incrementa <= '0';
                define <= '1';

            when gravacao =>
                zera <= '1';
                incrementa <= '0';
                define <= '0'; -- load ativo em baixo

            when operacao =>
                zera <= '1';
                incrementa <= clock; -- permite contagem em funcao do clock
                define <= '1';

            when inibicao =>
                zera <= '1';
                incrementa <= '0';
                define <= '1';
        end case;
    end process action_of_states;
end arch;
```

g) Diagrama de Estados da Unidade de Controle



Acima, tem-se o diagrama de estados gerado no Quartus. A máquina de estados é do tipo Moore, dado que as saídas só dependem do estado atual.

As 2 tabelas foram adicionadas com o fito de complementar a compreensão da máquina de estados. A primeira mostra como as entradas de comando (na vertical) indicam o estado atual (na horizontal). A segunda em como as saídas (na horizontal) são determinadas pelo estado atual (na vertical).

↓ Entradas / Estados →	Zerado	Programação	Gravação	Operação	Inibição
Clear	1	0	0	0	0
Programa	X	1	1	0	0
Grava	X	0	1	X	X
Inibe	X	X	X	0	1

↓ Estados / Saídas →	Zera	Incrementa	Define
Zerado	0	0	1
Programação	1	0	1
Gravação	1	0	0
Operação	1	Clock	1
Inibição	1	0	1

h) Carta de tempos da Unidade de Controle

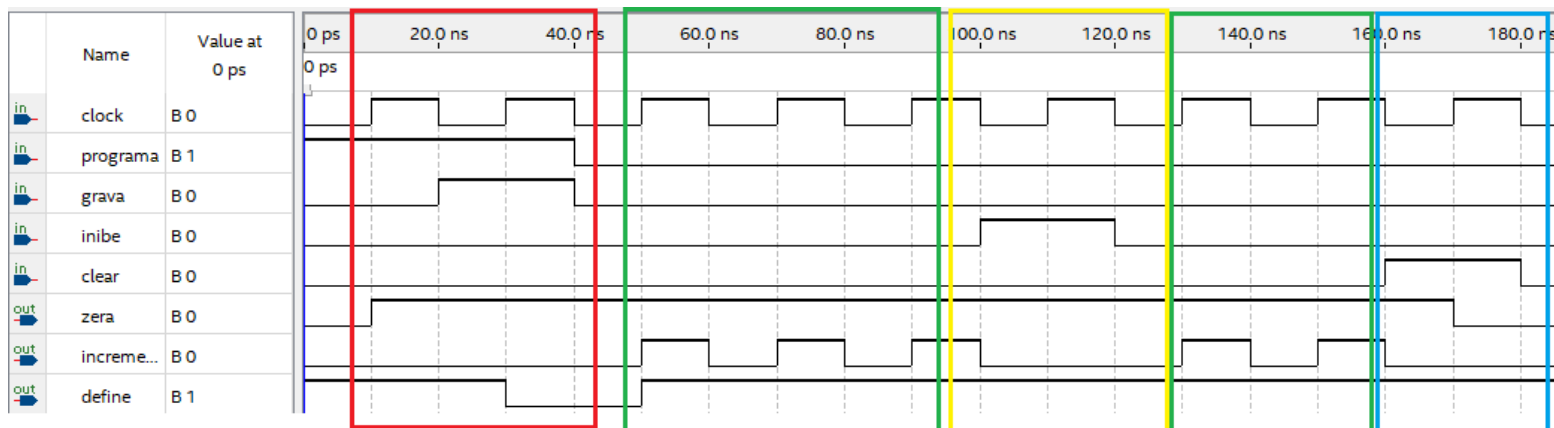
Foram divididas algumas situações nesta carta de tempos da UC para exemplificar seu funcionamento.

Inicialmente, no retângulo vermelho, observa-se que a saída “Define” só é ativada na segunda borda de subida do clock, quando “Programa” e “Grava” são ativadas, indicando o modo “Gravação”. Vale apontar que, na primeira borda de subida do clock, apenas “Programa” estava ativo, indicando o modo “Programação”, que apenas retém a saída do relógio, conforme especificado anteriormente no item a) como escolha de projeto.

No retângulo verde a seguir, entramos no estado “Operação” dado que o comando “Programa” está em nível lógico ‘0’. De fato, é observado que o clock é passado diretamente à saída “Incrementa”, indicando que o FD está implementando a contagem de segundos na mesma frequência do clock.

Prosseguindo, no retângulo amarelo, a entrada “Inibe” é ativada, o que para a contagem, dado que a saída “Incrementa” se anula. No segundo retângulo verde (logo em seguida), “Inibe” é desativado e o sistema volta a incrementar.

Finalmente, no retângulo azul, é dado o comando clear, que ativa o sinal intermediário “Zera”, o colocando para ‘0’, já que este é ativo em baixo.



i) Tabela de testes para o sistema completo

A tabela de testes do sistema completo é dada abaixo, onde apenas os sinais de entrada e saídas de alto nível são explicitados.

Inicialmente, é feita uma limpa no sistema na primeira linha com o comando Clear. Vale ressaltar que este é ativo em nível lógico ‘1’, por mais que internamente no circuito este sinal é negado para entrar no *clear* dos contadores. A UC é quem realiza esta dinâmica.

Em seguida, na segunda linha, entra-se no estado de programação, que retém a saída, mas com a gravação desativada. É apenas na terceira linha que, de fato, a saída recebe a entrada, no estado de gravação. Neste exemplo gravamos o caso extremo que corresponde ao horário “59:57” para verificar a transição para “00:00”, conforme o esperado de um relógio.

A partir disso, incrementações são feitas até que isto ocorra. Vale observar que na linha 5, o comando “Inibe” é ativado, retendo a saída no valor da saída, até que este comando seja desativado.

#	Clock	Clear	Programa	Grava	Inibe	Entrada (14 bits)	Saída (14 bits)
1	↑	1	X	X	X	X	00000000000000
2	↑	0	1	0	X	X	00000000000000
3	↑	0	1	1	X	10110011010111	10110011010111
4	↑	0	0	X	0	X	10110011011000
5	↑	0	0	X	1	X	10110011011000
6	↑	0	0	X	0	X	10110011011001
7	↑	0	0	X	0	X	00000000000000
8	↑	0	0	X	0	X	00000000000001

4. Relatório:

a. Montagem

- UC: foi feita a montagem da Unidade de Controle através da FPGA, para isso foi necessário criar um Pin Planner para cada uma das entradas e saídas.
- Entradas:
 - Programa - SW9 (AB12)
 - Grava - SW8 (AB13)
 - Inibe - SW7 (AA13)
 - Clear - SW6 (AA14)
- Saídas
 - Incrementa - GPIO_1_D1 (A12)
 - Define - GPIO_1_D3 (B12)
 - Zera - GPIO_1_D5 (B13)
 - Clock - Foi utilizado o Analog Discovery para criação do sinal de 1 Hz que foi colocado na entrada D13 da FPGA

×	Named: *	▼	⚡	Edit: ✕	✓			
⚡	Node Name	Direction	Location	I/O Bank	VREF Group	Pin Location	I/O Standard	
in	clear	Input	PIN_AA14	4A	B4A_N0	PIN_AA14	2.5 V	
in	clock	Input	PIN_D13	7A	B7A_N0	PIN_D13	2.5 V	
out	define	Output	PIN_B12	7A	B7A_N0	PIN_B12	2.5 V	
in	grava	Input	PIN_AB13	4A	B4A_N0	PIN_AB13	2.5 V	
out	incrementa	Output	PIN_A12	7A	B7A_N0	PIN_A12	2.5 V	
in	inibe	Input	PIN_AA13	4A	B4A_N0	PIN_AA13	2.5 V	
in	programa	Input	PIN_AB12	4A	B4A_N0	PIN_AB12	2.5 V	
out	zera	Output	PIN_B13	7A	B7A_N0	PIN_B13	2.5 V	
	<<new node>>							

- FD: foi feita a montagem do Fluxo de Dados através do Painel de Montagens, para isso foi reservado alguns componentes eletrônicos (CI's) que vão ser usados na experiências, são eles:
 - 4 Contadores (74161)
 - 1 Porta NAND (7400)
 - 1 Porta NOT (7404)
 - 1 Porta AND (7408)
 - 1 Porta OR (7432)

b. Montagem no laboratório

Primeiramente foi configurada a FPGA conforme descrito anteriormente, passando o programa em VHDL para a placa da FPGA, foram então realizados testes para garantir o funcionamento da Unidade de Controle do sistema digital, que após validada passou-se para a próxima etapa.

A seguir fez-se todas as conexões necessárias para funcionamento dos sinais intermediários, utilizando-se para isso a placa de conversão que contém o TTL 74365 para converter os sinais vindos da placa da FPGA para a placa de montagens.

Feita a conversão então foram colocados todos os contadores e testados um a um cada um desses componentes para evitar erros posteriores e, estando todos funcionais, passou-se para a próxima etapa de colocar as demais portas lógicas e iniciar o cabeamento do restante do Fluxo de Dados.

c. Problemas

Durante a montagem do Fluxo de Dados houveram problemas que consumiram muito tempo do laboratório, dentre eles vale citar:

- Ambas as portas NAND e NOT estavam com defeitos e precisaram ser testadas individualmente para perceber-se esse problema, o que consumiu bastante tempo até ser percebido e essas portas serem trocadas por outras portas funcionais.
- Mal contato: diversas vezes houveram problemas no clear tanto do terceiro quanto do segundo contador por conta de mal contato de fio que não pode ser resolvido devido ao problema abaixo.
- Durante o laboratório passamos por carência de jumpers e de fios, e os monitores estavam sendo frequentemente abordados para saciar os alunos destes componentes.

d. Resultados e Conclusão

O resultado da montagem foi parcialmente satisfatório.

A contagem de segundos, tanto nas dezenas, quanto nas unidades, ficou totalmente funcional tendo as unidades contando corretamente de 0-9 e as dezenas sendo contadas corretamente de 0-5, concluindo que o circuito estava funcionando corretamente para esses casos.

A contagem dos minutos, por sua vez, foi um tanto mais turbulenta. O CLEAR do terceiro contador, que funcionava como contagem das unidades dos minutos, não tinha seu CLEAR corretamente implementado e, por conta disso, estava contando de 0-F, ou seja, ele não parava a contagem em 9 como o esperado. Já para as dezenas dos minutos, não houve tempo hábil para debugar o funcionamento do CLOCK do quarto contador que estava incorreto e, dessa forma, as dezenas de minutos ficaram congeladas, não estando no estado ótimo que seria uma contagem de 0-5.

Considerando apenas a parte funcional do relógio, ou seja, parcialmente as unidades de minutos e tanto as dezenas, quanto as unidades de segundos havia também um correto funcionamento das chaves de CLEAR e INIBE, que corretamente foram responsáveis por zerar o relógio e congelar seu funcionamento, respectivamente.

A outra funcionalidade que usava a chave de GRAVA não pode ser testada, pois não houve tempo hábil para sua implementação de forma satisfatória.

Apesar dessas falhas, pode-se concluir que os erros foram puramente de montagem, uma vez que conceitualmente ambos os circuitos, tanto da UC, quanto da FD funcionam corretamente se compilados utilizando-se a plataforma fornecida pelo QUARTUS.