**Projetos Eletrônicos Digitais**

****

|  |  |  |  |
| --- | --- | --- | --- |
| **Digitado por:** | **Professor:** | **Versão** | **Ano** |
| Ricardo Kim | Renato Giacomini | 1.0 | 2017 |

**ÍNDICE**

[1. Apresentação e Ementa 3](#_Toc484708807)

[2. Very High Speed Integrated Circuit 4](#_Toc484708808)

[3. Lab aula 2 6](#_Toc484708809)

[3.1. Programa 1 (push buttom como clock) 6](#_Toc484708810)

[3.2. Programa 2 (“Mudar” Frequência 50mhz para 475khz ) 6](#_Toc484708811)

[3.3. Programa 3 (Projeto transmissão AM, código em desenvolvimento) 7](#_Toc484708812)

[4. Lab aula 3 8](#_Toc484708813)

[4.1. Tema: 8](#_Toc484708814)

[4.2. Algoritmo – (PWM) 9](#_Toc484708815)

[5. Lab Aula 4 [Tx] 10](#_Toc484708816)

[6. Lab 5 [RX] 12](#_Toc484708817)

[7. Lab 7 – Trabalho (incompleto) 16](#_Toc484708818)

[8. Video 20](#_Toc484708819)

# Apresentação e Ementa

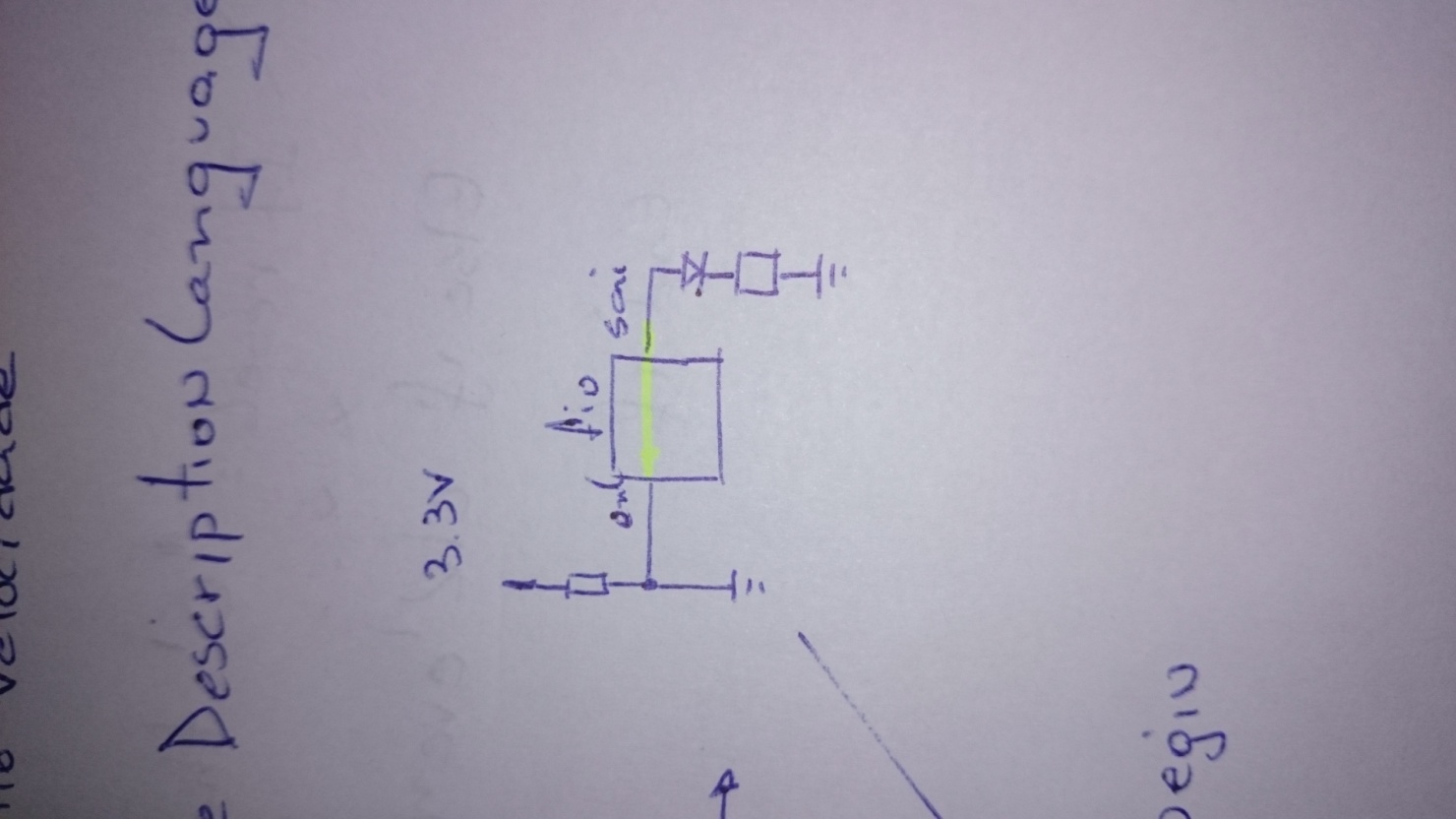
* Contato: [renato@fei.edu.br](mailto:renato@fei.edu.br) ; 99987-0280
* Livros:
  + Sistemas Digitais – Princípios e Aplicações, Tocci.
* Ementa:
  + Tecnologia
    - Analógicos
    - Digitais
    - Up
    - Lógica Programação
  + Softwares
    - Quartus II 13.0
  + Linguagens
    - Verilog
    - VHDL
  + Projetos
    - Pwm
    - Serial
    - Video
    - Servo Motor
* Referências
  + <HTTP://altera.com>
  + FPGA
  + Tecnologia Arm

# Very High Speed Integrated Circuit

Grupo criado pelo DOD para cuidar dos projetos de sistemas digitais de dH0 Velocidade.

Very High Speed Hardware Description Laguage

Ex:



Entity fio is

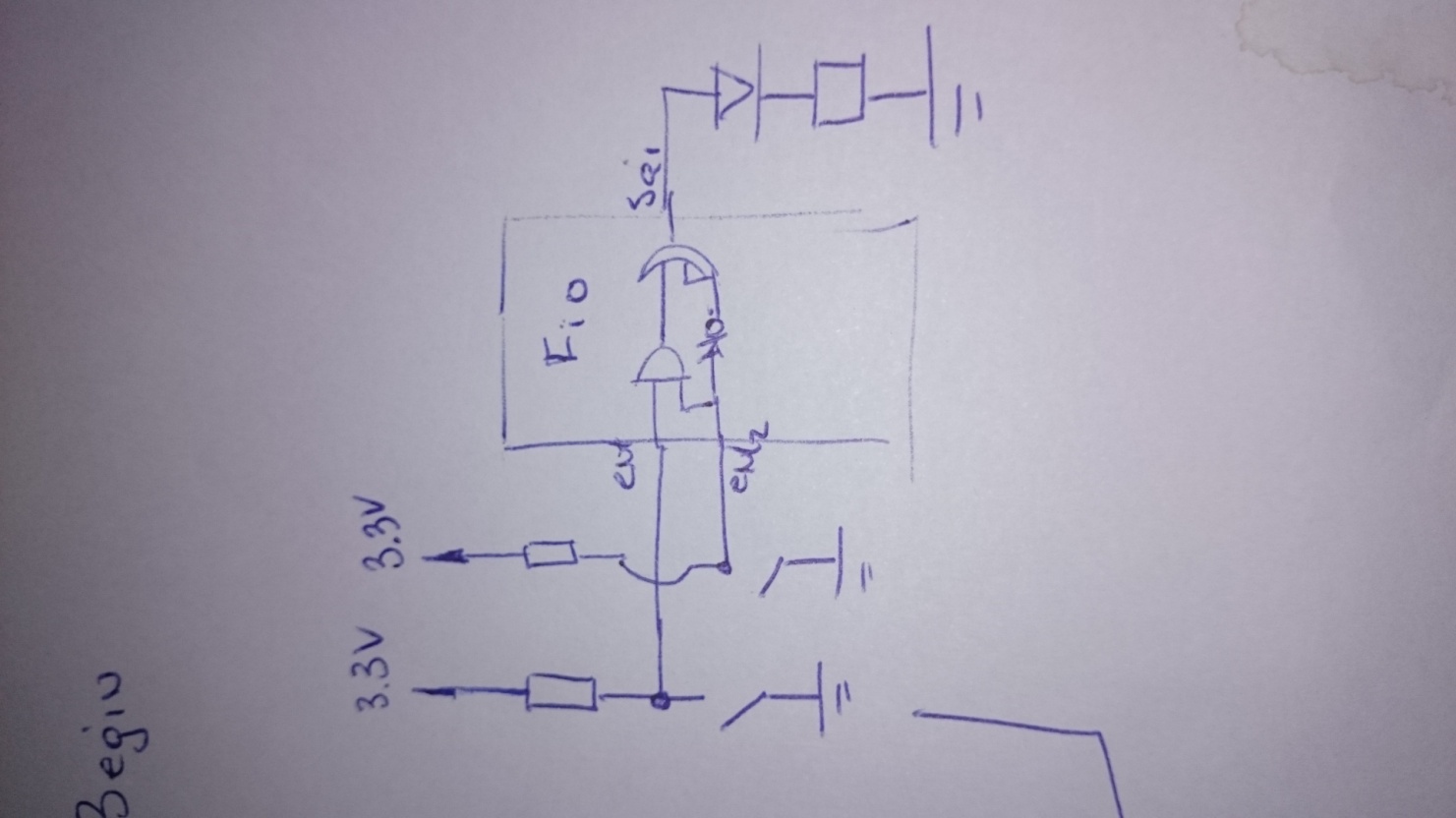
Port (ent: in bit; sai: out bit);

End fio;

Architecture Arc of fio is Begin

Sai <= ent

Ent Arc



Architecture Arc of fio is

Begin

Sai <= Not ent2 or

(ent AND ent2)

End Arc

Ou

Architecture Arc of fio is

Begin

A <= ent AND ent2

Sai <= A or not ent2

End Arc

**Sequencia:**

Process (clk, reset) ::<- dentro da função define os eventos a serem observados

Begin

If reset = “0” then :: isso monitora o evento do botão reset.

::instruções aqui

else if clk event and clk = ‘1’ then :: isso aqui monitora o evento o clock

:: instruções aqui

end if

# Lab aula 2

## Programa 1 (push buttom como clock)

entity Proj1 is

port

(

saida:buffer integer range 0 to 7;

clk:in bit

);

end Proj1;

architecture b of Proj1 is

Begin

Process (clk)

Begin

if (clk'event and clk = '0') then

saida <= saida+1;

end if;

end process;

end b;

## Programa 2 (“Mudar” Frequência 50mhz para 475khz )

Entity Proj1 is

port

(

clk:in bit;

saida: buffer bit

);

end Proj1;

architecture b of Proj1 is

signal count : integer range 0 to 53;

Begin Process (clk)

Begin

if (clk'event and clk = '0') then

if (count < 53) then

count <= count + 1;

else

saida <= not saida ;-- comentario

count <= 0;

end if;

end if;

end process;

end b;

## Programa 3 (Projeto transmissão AM, código em desenvolvimento)

Entity Proj1 is

port

(

clk:in bit;

--x: buffer integer range 0 to 1

s1: buffer bit;

s2: buffer bit;

s3: buffer bit

);

end Proj1;

architecture b of Proj1 is

signal count : integer range 0 to 254;

Begin

Process (clk)

Begin

if (clk'event and clk = '0') then

--saida <= saida + 1;

if (count < 36) then

count <= count + 1;

s1 <= '1';

if (count rem 12 = 0) then

--s1 <= not s1;

end if;

if (count rem 3 = 0) then

--s2 <= not s2;

end if;

if (count rem 6 = 0) then

--s3 <= not s3;

end if;

else

if (s1 = '1') then

s1<='0';

end if;

if (s2 = '1') then

s2 <= '0';

end if;

if (s3 = '1') then

s3 <= '0';

end if;

if (count < 72) then

count <= count + 1;

else

count <= 0;

end if;

end if;

end if;

end process;

end b;

# Lab aula 3

## Tema:

Como controlar continuamente um sinalfísico utilizando apenas uma saída de 1 bit?

Problemas:

Realizar o controle de brilho contínuo para o display numérico. Utilizar os push-bottons para aumentar ou diminuir o brilho

Soluções:

Frequencia que acente o led

PWM (Pulse Width Modulation)

Média,

PS: Não é analógico de verdade é persistência visual.

## Algoritmo – (PWM)

entity aula3 is

port (

clk: in bit;

--nivel de brilho

--caixa de inputs

input1 : in bit;

input2 : in bit;

--outputs

led: buffer bit

);

end aula3;

architecture Kim of aula3 is

--clock progressivo

signal newClockSignal: bit;

signal ctlBright: integer range 0 to 1023 ;

signal clockCountReducer : integer range 0 to 125000;

signal ledTimerOn : integer range 0 to 1023;

--inicio da architecture

begin

-- 200 mhz era muito lento professor pediu para

-- a velocidade

-- Processo de controle [baixa clock de 50 mhz para 400 hz]

process (clk)

begin

if(clk'event and clk='0') then

clockCountReducer <= clockCountReducer + 1;\

if(count200 < 62500) then

newClockSignal <= '1';

else

newClockSignal <= '0';

end if;

end if;

end process;

-- Processo de controle [caixa de inputs]

process (newClockSignal)

begin if (newClockSignal'event and newClockSignal='0')then

if (input1 = '0') and (ctlBright < 1023) then

ctlBright <= ctlBright + 1;

elsif (input2 = '0') and (ctlBright > 0) then

ctlBright <= ctlBright - 1;

end if;

end if;

end process;

-- Processo de controle [pwm - Pulse Width Modulation]

process (clk)

begin if (clk' event and clk='0') then

ledTimerOn <= ledTimerOn + 1;

if (ledTimerOn < ctlBright ) then

led <= '1';

else

led <= '0';

end if;

end if;

end process;

end Kim;

# Lab Aula 4 [Tx]

ENTITY Projeto4 is

PORT

(

sw0, sw1, sw2, sw3, sw4, sw5, sw6, sw7 : in BIT;

clk: in BIT;

send : IN BIT;

Tx,ledTransmission : out BIT

);

END Projeto4;

ARCHITECTURE Kim OF Projeto4 is

--Reservado somente para o novo clock

--PROCESSO 1

CONSTANT Cons\_Count : INTEGER := 50000;

SIGNAL nClk: INTEGER RANGE 0 to Cons\_Count;

SIGNAL sClk: BIT;

--Fim da reserva

--reservado para transmissao

--PROCESSO 2

SIGNAL byte: INTEGER RANGE 0 to 7;

--fim da reserva

BEGIN

--Esse processo faz a conversao de 50Mhz para 1 Khz

PROCESS(clk)

BEGIN

IF(clk'EVENT and clk='0') THEN

IF (nClk > Cons\_Count/2) THEN

nClk <= 0;

sClk <= not sClk;

ELSE

nClk <= nClk + 1;

END IF;

END IF;

END PROCESS;

-- Esse processo faz a leitura dos dados do switch

PROCESS (sClk)

BEGIN

IF (sClk'EVENT and sClk ='0') THEN

IF(send = '0') THEN --quando botao pressionado..

CASE byte is

WHEN 0 =>

IF (sw7 = '1') THEN

Tx <= '1';

ELSE

Tx <= '0';

END IF;

WHEN 1 =>

IF (sw6 = '1') THEN

Tx <= '1';

ELSE

Tx <= '0';

END IF;

WHEN 2 =>

IF (sw5 = '1') THEN

Tx <= '1';

ELSE

Tx <= '0';

END IF;

WHEN 3 =>

IF (sw4 = '1') THEN

Tx <= '1';

ELSE

Tx <= '0';

END IF;

WHEN 4 =>

IF (sw3 = '1') THEN

Tx <= '1';

ELSE

Tx <= '0';

END IF;

WHEN 5 =>

IF (sw2 = '1') THEN

Tx <= '1';

ELSE

Tx <= '0';

END IF;

WHEN 6 =>

IF (sw1 = '1') THEN

Tx <= '1';

ELSE

Tx <= '0';

END IF;

WHEN 7 =>

IF (sw0 = '1') THEN

Tx <= '1';

ELSE

Tx <= '0';

END IF;

WHEN others =>

end case;

ledTransmission <= '1';

ELSE

Tx <='0';

ledTransmission <= '0';

END IF;

byte <= byte + 1;

END IF;

END PROCESS;

END Kim;

# Lab 5 [RX]

--Tx trabalhar o transmissor no Tx em 1 bus

--Rx trabalhar o Rx com clock superior do Tx, normalmente com 4x~10x mais

--dessa forma o receptor pode detectar a borda de 0

--indicando o start bit

ENTITY Projeto4 is

PORT

(

--ELEMENTOS COMUNS A TODOS

clk: in BIT;

--ELEMENTOS DO Tx

sw0, sw1, sw2, sw3, sw4, sw5, sw6, sw7 : in BIT;

send : IN BIT;

Tx,ledTransmission : out BIT;

--ELEMENTOS DO Rx

led0, led1, led2,led3, led4, led5, led6, led7: OUT BIT;

Rx: IN BIT;

-- elementos de debug

txview,rxview : OUT BIT

);

END Projeto4;

ARCHITECTURE Kim OF Projeto4 is

--PROCESSOS DO TX

--Reservado somente para o novo clock

--PROCESSO 1

CONSTANT Cons\_Count : INTEGER := 50000;

SIGNAL nClk: INTEGER RANGE 0 to Cons\_Count;

SIGNAL sClk: BIT;

--Fim da reserva

--reservado para transmissao

--PROCESSO 2

SIGNAL byte: INTEGER RANGE 0 to 10;

--PROCESSOS DO RX

--Clock do RX

--PROCESSO 3

--CONSTANT rx\_count : INTEGER := (Cons\_Count); --clock de leitura 10x mais rapida do tx

CONSTANT rx\_count : INTEGER := 5000; --clock de leitura 10x mais rapida do tx

SIGNAL rxNClock : INTEGER RANGE 0 to rx\_count;

SIGNAL rxSClock : BIT;

--PROCESSO 4

SIGNAL init\_flag: BIT := '0';

SIGNAL rxByte: INTEGER RANGE 0 to 150 := 0;

--fim da reserva

BEGIN

rxview <= rxSClock;

txview <= sCLk;

--Esse processo faz a conversao de 50MHz para 500Hz

PROCESS(clk) --PROCESSO 1

BEGIN

IF(clk'EVENT and clk='0') THEN

IF (nClk > Cons\_Count/2) THEN

nClk <= 0;

sClk <= not sClk;

ELSE

nClk <= nClk + 1;

END IF;

END IF;

END PROCESS;

PROCESS (sClk) --PROCESSO 2 esse processo faz a leitura dos dados do switch

BEGIN

IF (sClk'EVENT and sClk ='0') THEN

IF(send = '0') THEN --quando botao pressionado..

CASE byte is

WHEN 0 =>

Tx <= '1';

WHEN 1 =>

IF (sw7 = '1') THEN

Tx <= '1';

ELSE

Tx <= '0';

END IF;

WHEN 2 =>

IF (sw6 = '1') THEN

Tx <= '1';

ELSE

Tx <= '0';

END IF;

WHEN 3 =>

IF (sw5 = '1') THEN

Tx <= '1';

ELSE

Tx <= '0';

END IF;

WHEN 4 =>

IF (sw4 = '1') THEN

Tx <= '1';

ELSE Tx <= '0';

END IF;

WHEN 5 =>

IF (sw3 = '1') THEN

Tx <= '1';

ELSE

Tx <= '0';

END IF;

WHEN 6 =>

IF (sw2 = '1') THEN

Tx <= '1';

ELSE

Tx <= '0';

END IF;

WHEN 7 =>

IF (sw1 = '1') THEN

Tx <= '1';

ELSE

Tx <= '0';

END IF;

WHEN 8 =>

IF (sw0 = '1') THEN

Tx <= '1';

ELSE

Tx <= '0';

END IF;

WHEN 9 =>

Tx <= '0';

WHEN 10 =>

Tx <= '0';

byte <= 0;

WHEN others =>

end case;

ledTransmission <= '1';

ELSE

Tx <='0';

ledTransmission <= '0';

END IF;

byte <= byte + 1;

END IF;

END PROCESS;

PROCESS(clk) --PROCESSO 3

BEGIN

IF(clk'EVENT and clk='0') THEN

IF (rxNClock > rx\_count/2) THEN

rxNClock <= 0;

rxSClock <= not rxSClock;

ELSE

rxNClock <= rxNClock + 1;

END IF;

END IF;

END PROCESS;

PROCESS(rxSClock) --PROCESSO 4

BEGIN

IF (rxSClock'EVENT and rxSClock ='0' ) THEN

IF (Rx = '1' AND init\_flag ='0') THEN

init\_flag <= '1';

rxByte <= 0;

ELSE

rxByte <= rxByte + 1;

CASE rxByte is

WHEN 4 =>

WHEN 14 =>

IF (Rx = '1') THEN

led7 <= '1';

ELSE

led7 <= '0';

END IF;

WHEN 24 =>

IF (Rx = '1') THEN

led6 <= '1';

ELSE

led6 <= '0';

END IF;

WHEN 34 =>

IF (Rx = '1') THEN

led5 <= '1';

ELSE

led5 <= '0';

END IF;

WHEN 44 =>

IF (Rx = '1') THEN

led4 <= '1';

ELSE

led4 <= '0';

END IF;

WHEN 54 =>

IF (Rx = '1') THEN

led3 <= '1';

ELSE

led3 <= '0';

END IF;

WHEN 64 =>

IF (Rx = '1') THEN

led2 <= '1';

ELSE

led2 <= '0';

END IF;

WHEN 74 =>

IF (Rx = '1') THEN

led1 <= '1';

ELSE

led1 <= '0';

END IF;

WHEN 84 =>

IF (Rx = '1') THEN

led0 <= '1';

ELSE

led0 <= '0';

END IF;

WHEN 94 =>

WHEN 104 =>

init\_flag <= '0';

rxByte <= 0;

WHEN others =>

end case;

END IF;

END IF;

END PROCESS;

END Kim;

# Lab 7 – Trabalho (incompleto)

Controlar os Leds de uma placa, usando as chaves da outra (10 leds). Conforme a chave for mudada em uma placa, o led deve mudar imediatamente na outra.

Transmite apenas 1 byte por vez.

Melhor projeto = +0,5 no semestre

Requisitos:

Documentação (clareza de código)

Funcionalidade

Confiabilidade

--Tx trabalhar o transmissor no Tx em 1 bus

--Rx trabalhar o Rx com clock superior do Tx, normalmente com 4x~10x mais

--dessa forma o receptor pode detectar a borda de 0

--indicando o start bit

ENTITY Projeto4 is

PORT

(

--ELEMENTOS COMUNS A TODOS

clk: in BIT;

--ELEMENTOS DO Tx

sw0, sw1, sw2, sw3, sw4, sw5, sw6, sw7, sw8, sw9 : in BIT;

send : IN BIT;

Tx,ledTransmission : out BIT;

--ELEMENTOS DO Rx

led0, led1, led2,led3, led4, led5, led6, led7, led8, led9: OUT BIT;

Rx: IN BIT;

-- elementos de debug

txview,rxview : OUT BIT

);

END Projeto4;

ARCHITECTURE Kim OF Projeto4 is

--PROCESSOS DO TX

--Reservado somente para o novo clock

--PROCESSO 1

CONSTANT Cons\_Count : INTEGER := 50000;

SIGNAL nClk: INTEGER RANGE 0 to Cons\_Count;

SIGNAL sClk: BIT;

--Fim da reserva

--reservado para transmissao

--PROCESSO 2

SIGNAL byte: INTEGER RANGE 0 to 60;

--PROCESSOS DO RX

--Clock do RX

--PROCESSO 3

--CONSTANT rx\_count : INTEGER := (Cons\_Count); --clock de leitura 10x mais rapida do tx

CONSTANT rx\_speed: INTEGER := (100);

CONSTANT rx\_count : INTEGER := (Cons\_Count/rx\_speed); --clock de leitura 10x mais rapida do tx

SIGNAL rxNClock : INTEGER RANGE 0 to rx\_count;

SIGNAL rxSClock : BIT;

--PROCESSO 4

SIGNAL init\_flag: BIT := '0';

SIGNAL rxByte: INTEGER RANGE 0 to 2200 := 0;

-- registrador de entrada;

SIGNAL byte\_receiver: BIT\_VECTOR (0 TO 22);

--fim da reserva

BEGIN

rxview <= rxSClock;

txview <= sCLk;

--Esse processo faz a conversao de 50MHz para 500Hz

PROCESS(clk) --PROCESSO 1

BEGIN

IF(clk'EVENT and clk='0') THEN

IF (nClk > Cons\_Count/2) THEN

nClk <= 0;

sClk <= not sClk;

ELSE

nClk <= nClk + 1;

END IF;

END IF;

END PROCESS;

PROCESS (sClk) --PROCESSO 2 esse processo faz a leitura dos dados do switch

--Tx

BEGIN

IF (sClk'EVENT and sClk ='0') THEN

IF(send = '0') THEN --quando botao pressionado..

byte <= byte + 1;

CASE byte is

-- byte 1

WHEN 0 =>

Tx <= '1';

WHEN 1 =>

Tx <= sw7;

WHEN 2 =>

Tx <= sw6;

WHEN 3 =>

Tx <= sw5;

WHEN 4 =>

Tx <= sw4;

WHEN 5 =>

Tx <= sw3;

WHEN 6 =>

Tx <= sw2;

WHEN 7 =>

Tx <= sw1;

WHEN 8 =>

Tx <= sw0;

WHEN 9 =>

Tx <= '0';

WHEN 10 =>

Tx <= '0';

--byte <= 0;

-- byte 2

WHEN 11 =>

Tx <= '1';

WHEN 12 =>

Tx <= sw9;

WHEN 13 =>

Tx <= sw8;

WHEN 14 =>

Tx <= '0';

WHEN 15 =>

Tx <= '0';

WHEN 16 =>

Tx <= '0';

WHEN 17 =>

Tx <= '0';

WHEN 18 =>

Tx <= '0';

WHEN 19 =>

Tx <= '0';

WHEN 20 =>

Tx <= '0';

WHEN 21 =>

Tx <= '0';

byte <= 0;

WHEN others =>

end case;

ledTransmission <= '1';

ELSE

Tx <='0';

ledTransmission <= '0';

byte <= 0;

END IF;

END IF;

END PROCESS;

PROCESS(clk) --PROCESSO 3

BEGIN

IF(clk'EVENT and clk='0') THEN

IF (rxNClock > rx\_count/2) THEN

rxNClock <= 0;

rxSClock <= not rxSClock;

ELSE

rxNClock <= rxNClock + 1;

END IF;

END IF;

END PROCESS;

PROCESS(rxSClock) --PROCESSO 4

--Rx

BEGIN

IF (rxSClock'EVENT and rxSClock ='0' ) THEN

IF (Rx = '1' AND init\_flag ='0') THEN

init\_flag <= '1';

rxByte <= 0;

led0 <= '0';

led1 <= '0';

led2 <= '0';

led3 <= '0';

led4 <= '0';

led5 <= '0';

led6 <= '0';

led7 <= '0';

led8 <= '0';

led9 <= '0';

ELSE

rxByte <= rxByte + 1;

CASE rxByte is

WHEN 4 \*rx\_speed / 10 =>

--start bit deslocada

byte\_receiver(0) <= Rx;

WHEN 14 \* rx\_speed / 10 =>

byte\_receiver(8) <= Rx;

WHEN 24 \* rx\_speed / 10 =>

byte\_receiver(7) <= Rx;

WHEN 34 \* rx\_speed / 10 =>

byte\_receiver(6) <= Rx;

WHEN 44 \* rx\_speed / 10 =>

byte\_receiver(5) <= Rx;

WHEN 54 \* rx\_speed / 10 =>

byte\_receiver(4) <=Rx;

WHEN 64 \* rx\_speed / 10 =>

byte\_receiver(3) <= Rx;

WHEN 74 \* rx\_speed / 10 =>

byte\_receiver(2) <= Rx;

WHEN 84 \* rx\_speed / 10 =>

byte\_receiver(1) <= Rx;

WHEN 94 \* rx\_speed / 10 =>

byte\_receiver(9) <= Rx;

--end bit

WHEN 104 \* rx\_speed / 10 =>

byte\_receiver(10) <= Rx;

--end bit

--segundo byte recebimento

WHEN 114 \* rx\_speed / 10 =>

byte\_receiver(11)

-- start bit

WHEN 124 \* rx\_speed / 10 =>

--bit xyyyyyyy

byte\_receiver(19) <= Rx;

WHEN 134 \* rx\_speed / 10 =>

--bit yxyyyyyy

byte\_receiver(18) <= Rx;

WHEN 144 \* rx\_speed / 10 =>

byte\_receiver(17) <= Rx;

--bit yyxyyyyy

WHEN 154 \* rx\_speed / 10 =>

byte\_receiver(16) <= Rx;

--bit yyyxyyyy

WHEN 164 \* rx\_speed / 10 =>

byte\_receiver(15) <= Rx;

--bit yyyyxyyy

WHEN 174 \* rx\_speed / 10 =>

byte\_receiver(14) <= Rx;

--bit yyyyyxyy

WHEN 184 \* rx\_speed / 10 =>

byte\_receiver(13) <= Rx;

--bit yyyyyyxy

WHEN 194 \* rx\_speed / 10 =>

byte\_receiver(12) <= Rx;

--bit yyyyyyyx

WHEN 204 \* rx\_speed / 10 =>

byte\_receiver(20) <= Rx;

--end bit

WHEN 214 \* rx\_speed / 10 =>

byte\_receiver(21) <= Rx;

--end bit

init\_flag <= '0';

rxByte <= 0;

WHEN others =>

end case;

END IF;

END IF;

END PROCESS;

END Kim;

# Video

**ENTITY** Projeto\_Base **IS**

**PORT**

**(**

-- INDICANDO QUE SERA USADO O CLOCK DE 27 MHZ

CLOCK\_27 **:** **IN** bit\_vector **(**0 **to** 1**);**

CLOCK\_50 **:** **IN** bit**;**

-- PINOS DE SAIDA DO SINCRONISMO

VGA\_VS**,** VGA\_HS **:** **BUFFER** BIT**;**

-- Controles de cor

VGA\_R**:** **out** integer **range** 15 **downto** 0 **:=** 0**;**

VGA\_G**:** **out** integer **range** 15 **downto** 0 **:=** 0**;**

VGA\_B**:** **out** integer **range** 15 **downto** 0 **:=** 0**;**

-- VGA\_R: BUFFER bit\_vector (0 to 3);

-- VGA\_G: BUFFER bit\_vector (0 to 3);

-- VGA\_B: BUFFER bit\_vector (0 to 3);

-- leds confirmacao visual

LEDG**:** **BUFFER** bit\_vector **(**0 **to** 7**);**

-- botoes de movimento

KEY**:** **IN** bit\_vector **(**0 **to** 3**)**

**);**

**END** Projeto\_Base**;**

**ARCHITECTURE** Kim **OF** Projeto\_Base **IS**

-- controle de clock do video (compensar 27 MHz)

**SIGNAL** countHS**:** integer **RANGE** 0 **TO** 809**;**

**SIGNAL** countVS**:** integer **RANGE** 0 **TO** 525**;**

-- controle de sinal

**SIGNAL** enableHS**:** bit**;**

**SIGNAL** enableVS**:** bit**;**

-- Controle do display

**SIGNAL** drawHS**:** integer **RANGE** 0 **TO** 640 **:=** 0**;**

**SIGNAL** drawVS**:** integer **RANGE** 0 **TO** 480 **:=** 0**;**

**SIGNAL** showColor**:** bit**;**

**SIGNAL** hideColor**:** bit**;**

**SIGNAL** showDisplay**:** boolean**;**

-- Controle do desenho

**SIGNAL** startController**:** bit**;**

**SIGNAL** xInit**,** xEnd**:** integer **RANGE** 0 **TO** 640**;**

**SIGNAL** yInit**,** yEnd**:** integer **RANGE** 0 **TO** 480**;**

**SIGNAL** quadrado**:** boolean**;**

-- Controle de Direcao

**CONSTANT** cima**:** integer **:=** 0**;**

**CONSTANT** baixo**:** integer **:=** 1**;**

**CONSTANT** esquerda**:** integer **:=** 2**;**

**CONSTANT** direita**:** integer **:=** 3**;**

**CONSTANT** tempo**:** integer **:=** 700000**;**

**SIGNAL** moveDelay**:** integer **RANGE** 0 **TO** tempo**;**

-- ================================

-- Desenhos

-- ================================

--Pacman

**SIGNAL** pacmanFrame**:** integer **RANGE** 0 **TO** 3**;**

**TYPE** cor **IS**

**RECORD**

r**:** integer **RANGE** 15 **DOWNTO** 0**;**

g**:** integer **RANGE** 15 **DOWNTO** 0**;**

b**:** integer **RANGE** 15 **DOWNTO** 0**;**

**END** **RECORD;**

**TYPE** matriz **IS** **ARRAY(**0 **TO** 9**,** 0 **TO** 9**)** **OF** cor**;**

**TYPE** frames **IS** **ARRAY(**0 **TO** 3**)** **OF** matriz**;**

**CONSTANT** pacman**:** frames **:=** **(**

0 **=>** **(**

0 **=>** **(**0 **=>** **(**0**,**0**,**15**),** 1 **=>** **(**0**,**0**,**15**),** 2 **=>** **(**0**,**0**,**15**),** 3 **=>** **(**15**,**15**,**0**),** 4 **=>** **(**15**,**15**,**0**),** 5 **=>** **(**15**,**15**,**0**),** 6 **=>** **(**15**,**15**,**0**),** 7 **=>** **(**0**,**0**,**15**),** 8 **=>** **(**0**,**0**,**15**),** 9 **=>** **(**0**,**0**,**15**)),**

1 **=>** **(**0 **=>** **(**0**,**0**,**15**),** 1 **=>** **(**15**,**15**,**0**),** 2 **=>** **(**15**,**15**,**0**),** 3 **=>** **(**15**,**15**,**0**),** 4 **=>** **(**15**,**15**,**0**),** 5 **=>** **(**15**,**15**,**0**),** 6 **=>** **(**15**,**15**,**0**),** 7 **=>** **(**15**,**15**,**0**),** 8 **=>** **(**15**,**15**,**0**),** 9 **=>** **(**0**,**0**,**15**)),**

2 **=>** **(**0 **=>** **(**0**,**0**,**15**),** 1 **=>** **(**15**,**15**,**0**),** 2 **=>** **(**15**,**15**,**0**),** 3 **=>** **(**15**,**15**,**0**),** 4 **=>** **(**15**,**15**,**0**),** 5 **=>** **(**15**,**15**,**0**),** 6 **=>** **(**15**,**15**,**0**),** 7 **=>** **(**15**,**15**,**0**),** 8 **=>** **(**15**,**15**,**0**),** 9 **=>** **(**0**,**0**,**15**)),**

3 **=>** **(**0 **=>** **(**15**,**15**,**0**),** 1 **=>** **(**15**,**15**,**0**),** 2 **=>** **(**15**,**15**,**0**),** 3 **=>** **(**15**,**15**,**0**),** 4 **=>** **(**15**,**15**,**0**),** 5 **=>** **(**15**,**15**,**0**),** 6 **=>** **(**15**,**15**,**0**),** 7 **=>** **(**15**,**15**,**0**),** 8 **=>** **(**15**,**15**,**0**),** 9 **=>** **(**15**,**15**,**0**)),**

4 **=>** **(**0 **=>** **(**15**,**15**,**0**),** 1 **=>** **(**15**,**15**,**0**),** 2 **=>** **(**15**,**15**,**0**),** 3 **=>** **(**15**,**15**,**0**),** 4 **=>** **(**15**,**15**,**0**),** 5 **=>** **(**15**,**15**,**0**),** 6 **=>** **(**15**,**15**,**0**),** 7 **=>** **(**15**,**15**,**0**),** 8 **=>** **(**15**,**15**,**0**),** 9 **=>** **(**15**,**15**,**0**)),**

5 **=>** **(**0 **=>** **(**15**,**15**,**0**),** 1 **=>** **(**15**,**15**,**0**),** 2 **=>** **(**15**,**15**,**0**),** 3 **=>** **(**15**,**15**,**0**),** 4 **=>** **(**15**,**15**,**0**),** 5 **=>** **(**15**,**15**,**0**),** 6 **=>** **(**15**,**15**,**0**),** 7 **=>** **(**15**,**15**,**0**),** 8 **=>** **(**15**,**15**,**0**),** 9 **=>** **(**15**,**15**,**0**)),**

6 **=>** **(**0 **=>** **(**15**,**15**,**0**),** 1 **=>** **(**15**,**15**,**0**),** 2 **=>** **(**15**,**15**,**0**),** 3 **=>** **(**15**,**15**,**0**),** 4 **=>** **(**0**,**0**,**15**),** 5 **=>** **(**0**,**0**,**15**),** 6 **=>** **(**15**,**15**,**0**),** 7 **=>** **(**15**,**15**,**0**),** 8 **=>** **(**15**,**15**,**0**),** 9 **=>** **(**15**,**15**,**0**)),**

7 **=>** **(**0 **=>** **(**0**,**0**,**15**),** 1 **=>** **(**15**,**15**,**0**),** 2 **=>** **(**15**,**15**,**0**),** 3 **=>** **(**0**,**0**,**15**),** 4 **=>** **(**0**,**0**,**15**),** 5 **=>** **(**0**,**0**,**15**),** 6 **=>** **(**0**,**0**,**15**),** 7 **=>** **(**15**,**15**,**0**),** 8 **=>** **(**15**,**15**,**0**),** 9 **=>** **(**0**,**0**,**15**)),**

8 **=>** **(**0 **=>** **(**0**,**0**,**15**),** 1 **=>** **(**15**,**15**,**0**),** 2 **=>** **(**0**,**0**,**15**),** 3 **=>** **(**0**,**0**,**15**),** 4 **=>** **(**0**,**0**,**15**),** 5 **=>** **(**0**,**0**,**15**),** 6 **=>** **(**0**,**0**,**15**),** 7 **=>** **(**0**,**0**,**15**),** 8 **=>** **(**15**,**15**,**0**),** 9 **=>** **(**0**,**0**,**15**)),**

9 **=>** **(**0 **=>** **(**0**,**0**,**15**),** 1 **=>** **(**0**,**0**,**15**),** 2 **=>** **(**0**,**0**,**15**),** 3 **=>** **(**0**,**0**,**15**),** 4 **=>** **(**0**,**0**,**15**),** 5 **=>** **(**0**,**0**,**15**),** 6 **=>** **(**0**,**0**,**15**),** 7 **=>** **(**0**,**0**,**15**),** 8 **=>** **(**0**,**0**,**15**),** 9 **=>** **(**0**,**0**,**15**))**

**),**

1 **=>** **(**

0 **=>** **(**0 **=>** **(**0**,**0**,**15**),** 1 **=>** **(**0**,**0**,**15**),** 2 **=>** **(**0**,**0**,**15**),** 3 **=>** **(**15**,**15**,**0**),** 4 **=>** **(**15**,**15**,**0**),** 5 **=>** **(**15**,**15**,**0**),** 6 **=>** **(**15**,**15**,**0**),** 7 **=>** **(**0**,**0**,**15**),** 8 **=>** **(**0**,**0**,**15**),** 9 **=>** **(**0**,**0**,**15**)),**

1 **=>** **(**0 **=>** **(**0**,**0**,**15**),** 1 **=>** **(**15**,**15**,**0**),** 2 **=>** **(**15**,**15**,**0**),** 3 **=>** **(**15**,**15**,**0**),** 4 **=>** **(**15**,**15**,**0**),** 5 **=>** **(**15**,**15**,**0**),** 6 **=>** **(**15**,**15**,**0**),** 7 **=>** **(**15**,**15**,**0**),** 8 **=>** **(**15**,**15**,**0**),** 9 **=>** **(**0**,**0**,**15**)),**

2 **=>** **(**0 **=>** **(**0**,**0**,**15**),** 1 **=>** **(**15**,**15**,**0**),** 2 **=>** **(**15**,**15**,**0**),** 3 **=>** **(**15**,**15**,**0**),** 4 **=>** **(**15**,**15**,**0**),** 5 **=>** **(**15**,**15**,**0**),** 6 **=>** **(**15**,**15**,**0**),** 7 **=>** **(**15**,**15**,**0**),** 8 **=>** **(**15**,**15**,**0**),** 9 **=>** **(**0**,**0**,**15**)),**

3 **=>** **(**0 **=>** **(**15**,**15**,**0**),** 1 **=>** **(**15**,**15**,**0**),** 2 **=>** **(**15**,**15**,**0**),** 3 **=>** **(**15**,**15**,**0**),** 4 **=>** **(**15**,**15**,**0**),** 5 **=>** **(**15**,**15**,**0**),** 6 **=>** **(**15**,**15**,**0**),** 7 **=>** **(**15**,**15**,**0**),** 8 **=>** **(**15**,**15**,**0**),** 9 **=>** **(**15**,**15**,**0**)),**

4 **=>** **(**0 **=>** **(**15**,**15**,**0**),** 1 **=>** **(**15**,**15**,**0**),** 2 **=>** **(**15**,**15**,**0**),** 3 **=>** **(**15**,**15**,**0**),** 4 **=>** **(**15**,**15**,**0**),** 5 **=>** **(**15**,**15**,**0**),** 6 **=>** **(**15**,**15**,**0**),** 7 **=>** **(**15**,**15**,**0**),** 8 **=>** **(**15**,**15**,**0**),** 9 **=>** **(**15**,**15**,**0**)),**

5 **=>** **(**0 **=>** **(**15**,**15**,**0**),** 1 **=>** **(**15**,**15**,**0**),** 2 **=>** **(**15**,**15**,**0**),** 3 **=>** **(**15**,**15**,**0**),** 4 **=>** **(**15**,**15**,**0**),** 5 **=>** **(**0**,**0**,**15**),** 6 **=>** **(**15**,**15**,**0**),** 7 **=>** **(**15**,**15**,**0**),** 8 **=>** **(**15**,**15**,**0**),** 9 **=>** **(**15**,**15**,**0**)),**

6 **=>** **(**0 **=>** **(**15**,**15**,**0**),** 1 **=>** **(**15**,**15**,**0**),** 2 **=>** **(**15**,**15**,**0**),** 3 **=>** **(**15**,**15**,**0**),** 4 **=>** **(**0**,**0**,**15**),** 5 **=>** **(**0**,**0**,**15**),** 6 **=>** **(**0**,**0**,**15**),** 7 **=>** **(**15**,**15**,**0**),** 8 **=>** **(**15**,**15**,**0**),** 9 **=>** **(**15**,**15**,**0**)),**

7 **=>** **(**0 **=>** **(**0**,**0**,**15**),** 1 **=>** **(**15**,**15**,**0**),** 2 **=>** **(**15**,**15**,**0**),** 3 **=>** **(**15**,**15**,**0**),** 4 **=>** **(**0**,**0**,**15**),** 5 **=>** **(**0**,**0**,**15**),** 6 **=>** **(**0**,**0**,**15**),** 7 **=>** **(**15**,**15**,**0**),** 8 **=>** **(**15**,**15**,**0**),** 9 **=>** **(**0**,**0**,**15**)),**

8 **=>** **(**0 **=>** **(**0**,**0**,**15**),** 1 **=>** **(**15**,**15**,**0**),** 2 **=>** **(**15**,**15**,**0**),** 3 **=>** **(**0**,**0**,**15**),** 4 **=>** **(**0**,**0**,**15**),** 5 **=>** **(**0**,**0**,**15**),** 6 **=>** **(**0**,**0**,**15**),** 7 **=>** **(**0**,**0**,**15**),** 8 **=>** **(**15**,**15**,**0**),** 9 **=>** **(**0**,**0**,**15**)),**

9 **=>** **(**0 **=>** **(**0**,**0**,**15**),** 1 **=>** **(**0**,**0**,**15**),** 2 **=>** **(**0**,**0**,**15**),** 3 **=>** **(**0**,**0**,**15**),** 4 **=>** **(**0**,**0**,**15**),** 5 **=>** **(**0**,**0**,**15**),** 6 **=>** **(**0**,**0**,**15**),** 7 **=>** **(**0**,**0**,**15**),** 8 **=>** **(**0**,**0**,**15**),** 9 **=>** **(**0**,**0**,**15**))**

**),**

2 **=>** **(**

0 **=>** **(**0 **=>** **(**0**,**0**,**15**),** 1 **=>** **(**0**,**0**,**15**),** 2 **=>** **(**0**,**0**,**15**),** 3 **=>** **(**15**,**15**,**0**),** 4 **=>** **(**15**,**15**,**0**),** 5 **=>** **(**15**,**15**,**0**),** 6 **=>** **(**15**,**15**,**0**),** 7 **=>** **(**0**,**0**,**15**),** 8 **=>** **(**0**,**0**,**15**),** 9 **=>** **(**0**,**0**,**15**)),**

1 **=>** **(**0 **=>** **(**0**,**0**,**15**),** 1 **=>** **(**15**,**15**,**0**),** 2 **=>** **(**15**,**15**,**0**),** 3 **=>** **(**15**,**15**,**0**),** 4 **=>** **(**15**,**15**,**0**),** 5 **=>** **(**15**,**15**,**0**),** 6 **=>** **(**15**,**15**,**0**),** 7 **=>** **(**15**,**15**,**0**),** 8 **=>** **(**15**,**15**,**0**),** 9 **=>** **(**0**,**0**,**15**)),**

2 **=>** **(**0 **=>** **(**0**,**0**,**15**),** 1 **=>** **(**15**,**15**,**0**),** 2 **=>** **(**15**,**15**,**0**),** 3 **=>** **(**15**,**15**,**0**),** 4 **=>** **(**15**,**15**,**0**),** 5 **=>** **(**15**,**15**,**0**),** 6 **=>** **(**15**,**15**,**0**),** 7 **=>** **(**15**,**15**,**0**),** 8 **=>** **(**15**,**15**,**0**),** 9 **=>** **(**0**,**0**,**15**)),**

3 **=>** **(**0 **=>** **(**15**,**15**,**0**),** 1 **=>** **(**15**,**15**,**0**),** 2 **=>** **(**15**,**15**,**0**),** 3 **=>** **(**15**,**15**,**0**),** 4 **=>** **(**15**,**15**,**0**),** 5 **=>** **(**15**,**15**,**0**),** 6 **=>** **(**15**,**15**,**0**),** 7 **=>** **(**15**,**15**,**0**),** 8 **=>** **(**15**,**15**,**0**),** 9 **=>** **(**15**,**15**,**0**)),**

4 **=>** **(**0 **=>** **(**15**,**15**,**0**),** 1 **=>** **(**15**,**15**,**0**),** 2 **=>** **(**15**,**15**,**0**),** 3 **=>** **(**15**,**15**,**0**),** 4 **=>** **(**15**,**15**,**0**),** 5 **=>** **(**15**,**15**,**0**),** 6 **=>** **(**15**,**15**,**0**),** 7 **=>** **(**15**,**15**,**0**),** 8 **=>** **(**15**,**15**,**0**),** 9 **=>** **(**15**,**15**,**0**)),**

5 **=>** **(**0 **=>** **(**15**,**15**,**0**),** 1 **=>** **(**15**,**15**,**0**),** 2 **=>** **(**15**,**15**,**0**),** 3 **=>** **(**15**,**15**,**0**),** 4 **=>** **(**15**,**15**,**0**),** 5 **=>** **(**15**,**15**,**0**),** 6 **=>** **(**15**,**15**,**0**),** 7 **=>** **(**15**,**15**,**0**),** 8 **=>** **(**15**,**15**,**0**),** 9 **=>** **(**15**,**15**,**0**)),**

6 **=>** **(**0 **=>** **(**15**,**15**,**0**),** 1 **=>** **(**15**,**15**,**0**),** 2 **=>** **(**15**,**15**,**0**),** 3 **=>** **(**15**,**15**,**0**),** 4 **=>** **(**15**,**15**,**0**),** 5 **=>** **(**0**,**0**,**15**),** 6 **=>** **(**15**,**15**,**0**),** 7 **=>** **(**15**,**15**,**0**),** 8 **=>** **(**15**,**15**,**0**),** 9 **=>** **(**15**,**15**,**0**)),**

7 **=>** **(**0 **=>** **(**0**,**0**,**15**),** 1 **=>** **(**15**,**15**,**0**),** 2 **=>** **(**15**,**15**,**0**),** 3 **=>** **(**15**,**15**,**0**),** 4 **=>** **(**0**,**0**,**15**),** 5 **=>** **(**0**,**0**,**15**),** 6 **=>** **(**0**,**0**,**15**),** 7 **=>** **(**15**,**15**,**0**),** 8 **=>** **(**15**,**15**,**0**),** 9 **=>** **(**0**,**0**,**15**)),**

8 **=>** **(**0 **=>** **(**0**,**0**,**15**),** 1 **=>** **(**15**,**15**,**0**),** 2 **=>** **(**15**,**15**,**0**),** 3 **=>** **(**15**,**15**,**0**),** 4 **=>** **(**0**,**0**,**15**),** 5 **=>** **(**0**,**0**,**15**),** 6 **=>** **(**0**,**0**,**15**),** 7 **=>** **(**15**,**15**,**0**),** 8 **=>** **(**15**,**15**,**0**),** 9 **=>** **(**0**,**0**,**15**)),**

9 **=>** **(**0 **=>** **(**0**,**0**,**15**),** 1 **=>** **(**0**,**0**,**15**),** 2 **=>** **(**0**,**0**,**15**),** 3 **=>** **(**0**,**0**,**15**),** 4 **=>** **(**0**,**0**,**15**),** 5 **=>** **(**0**,**0**,**15**),** 6 **=>** **(**0**,**0**,**15**),** 7 **=>** **(**0**,**0**,**15**),** 8 **=>** **(**0**,**0**,**15**),** 9 **=>** **(**0**,**0**,**15**))**

**),**

3 **=>** **(**

0 **=>** **(**0 **=>** **(**0**,**0**,**15**),** 1 **=>** **(**0**,**0**,**15**),** 2 **=>** **(**0**,**0**,**15**),** 3 **=>** **(**15**,**15**,**0**),** 4 **=>** **(**15**,**15**,**0**),** 5 **=>** **(**15**,**15**,**0**),** 6 **=>** **(**15**,**15**,**0**),** 7 **=>** **(**0**,**0**,**15**),** 8 **=>** **(**0**,**0**,**15**),** 9 **=>** **(**0**,**0**,**15**)),**

1 **=>** **(**0 **=>** **(**0**,**0**,**15**),** 1 **=>** **(**15**,**15**,**0**),** 2 **=>** **(**15**,**15**,**0**),** 3 **=>** **(**15**,**15**,**0**),** 4 **=>** **(**15**,**15**,**0**),** 5 **=>** **(**15**,**15**,**0**),** 6 **=>** **(**15**,**15**,**0**),** 7 **=>** **(**15**,**15**,**0**),** 8 **=>** **(**15**,**15**,**0**),** 9 **=>** **(**0**,**0**,**15**)),**

2 **=>** **(**0 **=>** **(**0**,**0**,**15**),** 1 **=>** **(**15**,**15**,**0**),** 2 **=>** **(**15**,**15**,**0**),** 3 **=>** **(**15**,**15**,**0**),** 4 **=>** **(**15**,**15**,**0**),** 5 **=>** **(**15**,**15**,**0**),** 6 **=>** **(**15**,**15**,**0**),** 7 **=>** **(**15**,**15**,**0**),** 8 **=>** **(**15**,**15**,**0**),** 9 **=>** **(**0**,**0**,**15**)),**

3 **=>** **(**0 **=>** **(**15**,**15**,**0**),** 1 **=>** **(**15**,**15**,**0**),** 2 **=>** **(**15**,**15**,**0**),** 3 **=>** **(**15**,**15**,**0**),** 4 **=>** **(**15**,**15**,**0**),** 5 **=>** **(**15**,**15**,**0**),** 6 **=>** **(**15**,**15**,**0**),** 7 **=>** **(**15**,**15**,**0**),** 8 **=>** **(**15**,**15**,**0**),** 9 **=>** **(**15**,**15**,**0**)),**

4 **=>** **(**0 **=>** **(**15**,**15**,**0**),** 1 **=>** **(**15**,**15**,**0**),** 2 **=>** **(**15**,**15**,**0**),** 3 **=>** **(**15**,**15**,**0**),** 4 **=>** **(**15**,**15**,**0**),** 5 **=>** **(**15**,**15**,**0**),** 6 **=>** **(**15**,**15**,**0**),** 7 **=>** **(**15**,**15**,**0**),** 8 **=>** **(**15**,**15**,**0**),** 9 **=>** **(**15**,**15**,**0**)),**

5 **=>** **(**0 **=>** **(**15**,**15**,**0**),** 1 **=>** **(**15**,**15**,**0**),** 2 **=>** **(**15**,**15**,**0**),** 3 **=>** **(**15**,**15**,**0**),** 4 **=>** **(**15**,**15**,**0**),** 5 **=>** **(**15**,**15**,**0**),** 6 **=>** **(**15**,**15**,**0**),** 7 **=>** **(**15**,**15**,**0**),** 8 **=>** **(**15**,**15**,**0**),** 9 **=>** **(**15**,**15**,**0**)),**

6 **=>** **(**0 **=>** **(**15**,**15**,**0**),** 1 **=>** **(**15**,**15**,**0**),** 2 **=>** **(**15**,**15**,**0**),** 3 **=>** **(**15**,**15**,**0**),** 4 **=>** **(**15**,**15**,**0**),** 5 **=>** **(**15**,**15**,**0**),** 6 **=>** **(**15**,**15**,**0**),** 7 **=>** **(**15**,**15**,**0**),** 8 **=>** **(**15**,**15**,**0**),** 9 **=>** **(**15**,**15**,**0**)),**

7 **=>** **(**0 **=>** **(**0**,**0**,**15**),** 1 **=>** **(**15**,**15**,**0**),** 2 **=>** **(**15**,**15**,**0**),** 3 **=>** **(**15**,**15**,**0**),** 4 **=>** **(**15**,**15**,**0**),** 5 **=>** **(**15**,**15**,**0**),** 6 **=>** **(**15**,**15**,**0**),** 7 **=>** **(**15**,**15**,**0**),** 8 **=>** **(**15**,**15**,**0**),** 9 **=>** **(**0**,**0**,**15**)),**

8 **=>** **(**0 **=>** **(**0**,**0**,**15**),** 1 **=>** **(**15**,**15**,**0**),** 2 **=>** **(**15**,**15**,**0**),** 3 **=>** **(**15**,**15**,**0**),** 4 **=>** **(**15**,**15**,**0**),** 5 **=>** **(**15**,**15**,**0**),** 6 **=>** **(**15**,**15**,**0**),** 7 **=>** **(**15**,**15**,**0**),** 8 **=>** **(**15**,**15**,**0**),** 9 **=>** **(**0**,**0**,**15**)),**

9 **=>** **(**0 **=>** **(**0**,**0**,**15**),** 1 **=>** **(**0**,**0**,**15**),** 2 **=>** **(**0**,**0**,**15**),** 3 **=>** **(**15**,**15**,**0**),** 4 **=>** **(**15**,**15**,**0**),** 5 **=>** **(**15**,**15**,**0**),** 6 **=>** **(**15**,**15**,**0**),** 7 **=>** **(**0**,**0**,**15**),** 8 **=>** **(**0**,**0**,**15**),** 9 **=>** **(**0**,**0**,**15**))**

**)**

**);**

**BEGIN**

-- Processo de controle do HSync

**PROCESS(**CLOCK\_27**(**0**))**

**BEGIN**

**IF(**CLOCK\_27**(**0**)**'**EVENT** and CLOCK\_27**(**0**)** **=** '1'**)** **THEN**

-- ========================================

-- parte do controle externo das imagens

-- ========================================

showDisplay **<=** **(**enableHS and enableVS**)** **=**'1'**;**

showColor **<=** '1'**;** --(enableHS and enableVS);

hideColor **<=** '0'**;**

-- ========================================

-- parte do controle do desenho

-- ========================================

quadrado **<=** **(**xInit **<=** drawHS and drawHS **<** xEnd and yInit **<=** drawVS and drawVS **<** yEnd**);**

-- constrole de contagem

**IF** **(**countHS **<** 809**)** **THEN**

countHS **<=** countHS **+** 1**;**

**ELSE**

countHS **<=** 0**;**

**END** **IF;**

-- controle do Sincronismo

**IF** **(**0 **<=** countHS and countHS **<=** 102**)** **THEN**

VGA\_HS **<=** '0'**;**

**ELSE**

VGA\_HS **<=** '1'**;**

**END** **IF;**

-- controle do enable do HSync : 640 px (793 - 153 = 640)

**IF** **(** 153 **<=** countHS and countHS **<** 793 **)** **THEN**

enableHS **<=** '1'**;**

**ELSE**

-- testar consistencia do startdraw

-- drawHS <= 0;

enableHS **<=** '0'**;**

**END** **IF;**

**END** **IF;**

**END** **PROCESS;**

-- Processo de controle do hSinc

**PROCESS** **(**VGA\_HS**)**

**BEGIN**

**IF** **(**VGA\_HS'**EVENT** and VGA\_HS **=** '1'**)** **THEN**

-- constrole de contagem

**IF** **(**countVS **<** 525**)** **THEN**

countVS **<=** countVS **+** 1**;**

**ELSE**

countVS **<=** 0**;**

**END** **IF;**

-- controle do Sincronismo

**IF** **(**0 **<=** countVS and countVS **<=** 2**)** **THEN**

VGA\_VS **<=** '0'**;**

**ELSE**

VGA\_VS **<=** '1'**;**

**END** **IF;**

-- controle do enable do HSync : 480 px (515-35 = 480)

**IF** **(** 35 **<=** countVS and countVS **<** 515 **)** **THEN**

enableVS **<=** '1'**;**

**ELSE**

-- testar consistencia do start draw

-- drawVS <= 0;

enableVS **<=** '0'**;**

**END** **IF;**

**END** **IF;**

**END** **PROCESS;**

-- processo de desenhar o quadrado

**PROCESS** **(**CLOCK\_27**(**0**))**

-- PROCESS (CLOCK\_50)

**BEGIN**

-- obs

-- showColor corresponte a operacao AND enableVS e enableHS avaliando bits quando estiverem ativos

-- showDisplay corresponde operacao AND enableVS e enableHS em boleano

**IF** **(**CLOCK\_27**(**0**)**'**EVENT** and CLOCK\_27**(**0**)** **=** '0'**)** **THEN**

**IF(**enableHS **=** '0'**)** **THEN**

drawHS **<=** 0**;**

**END** **IF;**

**IF(**enableVS **=** '0'**)** **THEN**

drawVS **<=** 0**;**

**END** **IF;**

**IF** **(**showDisplay**)** **THEN**

**IF** **(**drawHS **<** 639**)** **THEN**

drawHS **<=** drawHS **+** 1**;**

**ELSE**

**IF** **(**drawVS **<** 480**)** **THEN**

drawVS **<=** drawVS **+** 1**;**

**ELSE**

-- drawVS <= 0;

**END** **IF;**

-- drawHS <= 0;

**END** **IF;**

-- IF (xInit <= drawHS and drawHS < xEnd ) and (yInit <= drawVS and drawVS < yEnd) then

-- VGA\_R <= 15;

-- VGA\_G <= 0;

-- VGA\_B <= 15;

-- else

-- VGA\_R <= 0;

-- VGA\_G <= 0;

-- VGA\_B <= 0;

-- end if;

**IF** not **(**

**((** 0 **<=** drawHS and drawHS **<** 159 **)** and **(**0 **<=** drawVS and drawVS **<** 111**))** or

**((**xInit **<=** drawHS and drawHS **<** xEnd **)** and **(**yInit **<=** drawVS and drawVS **<** yEnd**))**

**)** **THEN**

VGA\_R **<=** 0**;**

VGA\_G **<=** 0**;**

VGA\_B **<=** 15**;**

**END** **IF;**

**IF** **(** 0 **<=** drawHS and drawHS **<** 159 **)** and **(**0 **<=** drawVS and drawVS **<** 111**)** **THEN**

VGA\_R **<=** sonic**(**0**)(**drawVS**,** drawHS**).**r**;**

VGA\_G **<=** sonic**(**0**)(**drawVS**,** drawHS**).**g**;**

VGA\_B **<=** sonic**(**0**)(**drawVS**,** drawHS**).**b**;**

**END** **IF;**

**IF** **(**xInit **<=** drawHS and drawHS **<** xEnd **)** and **(**yInit **<=** drawVS and drawVS **<** yEnd**)** **then**

-- IF (315 <= drawHS and drawHS < 325 ) and (315 <= drawVS and drawVS < 325) then

VGA\_R **<=** pacman**(**pacmanFrame**)(**drawVS **-** yInit**,** drawHS **-** xInit**).**r**;**

VGA\_G **<=** pacman**(**pacmanFrame**)(**drawVS **-** yInit**,** drawHS **-** xInit**).**g**;**

VGA\_B **<=** pacman**(**pacmanFrame**)(**drawVS **-** yInit**,** drawHS **-** xInit**).**b**;**

**END** **if;**

**ELSE**

-- eh obrigatorio manter o nivel logico em 0 quando nao for autorizado o desenho

-- caso constrario nao eh projetado no display

VGA\_R **<=** 0**;**

VGA\_G **<=** 0**;**

VGA\_B **<=** 0**;**

**END** **IF;**

**END** **IF;**

**END** **PROCESS;**

-- controle do processo de movimento

**PROCESS** **(**CLOCK\_27**(**0**))**

**BEGIN**

**IF(**CLOCK\_27**(**0**)**'**EVENT** and CLOCK\_27**(**0**)** **=** '0'**)** **THEN**

**IF** **(**not startController **=** '1'**)** **THEN**

startController **<=** '1'**;**

xInit **<=** 315**;**

xEnd **<=** 325**;**

yInit **<=** 235**;**

yEnd **<=** 245**;**

moveDelay **<=** 0**;**

pacmanFrame **<=**0 **;**

**ELSE**

**IF** **((**KEY**(**3**)** OR KEY**(**2**)** OR KEY**(**1**)** OR KEY**(**0**))** **=** '1' **)THEN**

**IF** **(**moveDelay **<** tempo**)** **THEN**

moveDelay **<=** moveDelay **+** 1**;**

**ELSE**

moveDelay **<=** 0**;**

**END** **IF;**

**END** **IF;**

-- cima

**IF** **(**KEY**(**3**)** **=** '0'**)** **THEN**

**IF** **(**yInit **>** 0 and moveDelay **=** tempo**)** **THEN**

yInit **<=** yInit **-** 1**;**

yEnd **<=** yEnd **-** 1**;**

**IF** **(**pacmanFrame **<** 4**)** **THEN**

pacmanFrame **<=** pacmanFrame **+** 1**;**

**ELSE**

pacmanFrame **<=** pacmanFrame **+** 0**;**

**END** **IF;**

**END** **IF;**

LEDG**(**5**)** **<=** '1'**;**

**ELSE**

LEDG**(**5**)** **<=** '0'**;**

**END** **IF;**

-- baixo

**IF(**KEY**(**2**)** **=** '0'**)** **THEN**

**IF** **(**yEnd **<** 480 and moveDelay **=** tempo**)** **THEN**

yInit **<=** yInit **+** 1**;**

yEnd **<=** yEnd **+** 1**;**

**IF** **(**pacmanFrame **<** 4**)** **THEN**

pacmanFrame **<=** pacmanFrame **+** 1**;**

**ELSE**

pacmanFrame **<=** pacmanFrame **+** 0**;**

**END** **IF;**

**END** **IF;**

LEDG**(**7**)** **<=** '1'**;**

**ELSE**

LEDG**(**7**)** **<=** '0'**;**

**END** **IF;**

-- esquerda

**IF** **(**KEY**(**1**)** **=** '0'**)** **THEN**

**IF** **(**xInit **>** 0 and moveDelay **=** tempo**)** **THEN**

xInit **<=** xInit **-** 1**;**

xEnd **<=** xEnd **-** 1**;**

**IF** **(**pacmanFrame **<** 4**)** **THEN**

pacmanFrame **<=** pacmanFrame **+** 1**;**

**ELSE**

pacmanFrame **<=** pacmanFrame **+** 0**;**

**END** **IF;**

**END** **IF;**

LEDG**(**3**)** **<=** '1'**;**

**ELSE**

LEDG**(**3**)** **<=** '0'**;**

**END** **IF;**

-- direita

**IF** **(**KEY**(**0**)** **=** '0'**)** **THEN**

**IF** **(**xEnd **<** 640 and moveDelay **=** tempo**)** **then**

xInit **<=** xInit **+** 1**;**

xEnd **<=** xEnd **+** 1**;**

**IF** **(**pacmanFrame **<** 4**)** **THEN**

pacmanFrame **<=** pacmanFrame **+** 1**;**

**ELSE**

pacmanFrame **<=** pacmanFrame **+** 0**;**

**END** **IF;**

**END** **IF;**

LEDG**(**1**)** **<=** '1'**;**

**ELSE**

LEDG**(**1**)** **<=** '0'**;**

**END** **IF;**

**END** **IF;**

**END** **IF;**

**END** **PROCESS;**

**END** Kim**;**