UNIVERSIDADE DE SÃO PAULO – USP
INSTITUTO DE CIÊNCIAS MATEMÁTICAS E DE COMPUTAÇÃO
DEPARTAMENTO DE SISTEMAS DE COMPUTAÇÃO

# RELATÓRIO
# SSC 118 – SISTEMAS DIGITAIS

Docente:
Vanderlei Bonato

Discentes :
André Miguel: 8626249
Giovane Cunha Mocelin: 8778382
Wesley Tiozzo: 8077925

São Carlos
2013

## 1. Identificação

**Título: The Legend of Link**

## 2. Introdução

O presente relatório tem como objetivo explicar o funcionamento do jogo implementado em linguagem VHDL em conjunto com o software Quartus.

## 3. Modelagem do problema

O jogo baseia-se no estilo "Shoot'Em Up", na qual o jogador controla um personagem que faz uso de um arco e flechas para combater seus inimigos.
Comandos: W para se mover para cima
S para se mover para baixo
Space Bar para atirar flechas

## 4. Experimentos e resultados

### 4.1 Teste dos registradores em FPGA

## 4.2 Código VHDL

**-- Processo responsável por movimentação do personagem (Link)**
**-- Se a tecla S for pressionada e o Link estiver em posição menor do que 1159 que seria a posição final do vetor de 1200 posições, a posição do Link incrementa em 40, tornando possível a movimentação na tela para baixo.**
**-- Se a tecla W for pressionada e o Link estiver em posição maior do que 39, que seria a primeira linha do vetor, a posição decrementa 40, tornando possível a movimentação para cima.**

```
PROCESS (clk, reset)

BEGIN


ELSIF (clk'event) and (clk = '1') THEN

        CASE LINKESTADO IS
                WHEN x"00" =>
                        IF (GAMESTART = '0') THEN
                                LINKESTADO <= x"01";
                        ELSE
                                CASE key IS
                                        WHEN x"53" => -- (S) BAIXO
                                                IF (LINKPOS < 1159) THEN
                                                        LINKPOS <= LINKPOS + x"28";  -- LINKPOS + 40
                                                END IF;
                                        WHEN x"57" => -- (W) CIMA
                                                IF (LINKPOS > 39) THEN
                                                        LINKPOS <= LINKPOS - x"28";  -- LINKPOS - 40
                                                END IF;

                                        WHEN OTHERS =>
                                END CASE;
                                LINKESTADO <= x"01";
                        END IF;

                WHEN x"01" =>  -- DELAY

                        IF DELAY1 >= x"000011FF" THEN
                                DELAY1 <= x"00000000";
                                LINKESTADO <= x"00";
                        ELSE
                                DELAY1 <= DELAY1 + x"01";
                        END IF;

                WHEN OTHERS =>
        END CASE;
END IF;

END PROCESS;
```

**-- Processo responsável pela flecha que será atirada pelo Link**
**-- Se a barra de espaço for pressionada, a flecha aparecerá na frente do Link.**
**-- Se a flecha for atirada mas sua posição não for igual a posição final da linha (39), incrementa a posição da flecha, fazendo com que ela siga em frente.**
**-- Se a posição da flecha coincidir com a posição de um dos três inimigos,  é ativada uma flag de acerto e o sinal da flecha recebe 0.**

```
PROCESS (clk, reset)

BEGIN

IF RESET = '1' THEN
	FLECHACHAR <= "01100101";
	FLECHACOR <= "1111"; -- 1111 Branco
	FLECHAPOS <= x"04B0";  --fora da tela
	DELAY2 <= x"00000000";
	FLECHAESTADO <= x"00";
	FLECHASINAL <= '0';
	ACERTOU1	<= '0';
	ACERTOU2	<= '0';
	ACERTOU3	<= '0';

ELSIF (clk'event) and (clk = '1') THEN

				CASE FLECHAESTADO iS
					WHEN x"00" =>
						IF (GAMESTART = '0') THEN
							FLECHAESTADO <= x"03";
						ELSE
							--Atirou flecha?
							IF (key = x"00" AND FLECHASINAL = '0') THEN    --espaco
								FLECHASINAL <= '1';
								FLECHAPOS <= LINKPOS; --Flecha comeca na
frente do link
							END IF;

							FLECHAESTADO <= x"01";
						END IF;

					WHEN x"01" =>
						-- INCREMENTA A POS DA FLECHA
							IF ((NOT((conv_integer(FLECHAPOS) MOD 40) = 39)) AND
FLECHASINAL = '1') THEN
								FLECHAPOS <= FLECHAPOS + x"01";
								FLECHAESTADO <= x"02";

							ELSE
								FLECHASINAL <= '0';
								FLECHAPOS <= x"04B0";    --fora da tela
								FLECHAESTADO <= x"03";
							END IF;

					WHEN x"02" =>
						-- destroi flecha?
						IF ((ENEMY1POSA = FLECHAPOS) OR (ENEMY1POS =
FLECHAPOS)) THEN    --FLECHA E ENEMY1
							FLECHASINAL <= '0';
							ACERTOU1	<= '1';
							FLECHAPOS <= x"04B0";
						ELSIF ((ENEMY2POSA = FLECHAPOS) OR (ENEMY2POS =
FLECHAPOS)) THEN    --FLECHA E ENEMY2
							FLECHASINAL <= '0';
							ACERTOU2	<= '1';
							FLECHAPOS <= x"04B0";
						ELSIF ((ENEMY3POSA = FLECHAPOS) OR (ENEMY3POS =
FLECHAPOS)) THEN    --FLECHA E ENEMY3
							FLECHASINAL <= '0';
							ACERTOU3	<= '1';
							FLECHAPOS <= x"04B0";
						END IF;

						FLECHAESTADO <= x"03";

					WHEN x"03" =>

							IF((ENEMY1SINAL = '0') AND (ACERTOU1	= '1'))
THEN
```

```
                                                ACERTOU1        <= '0';
                                        END IF;

                                        IF((ENEMY2SINAL = '0') AND (ACERTOU2        = '1'))
THEN
                                                ACERTOU2        <= '0';
                                        END IF;

                                        IF((ENEMY3SINAL_VAR = '1') AND (ACERTOU3
= '1')) THEN     --enemy3sinal_var se perder vida
                                                ACERTOU3        <= '0';
                                        END IF;

                                        FLECHAESTADO <= x"04";

                                WHEN x"04" =>  -- Delay da Flecha

                                        IF DELAY2 >= x"000000FFF" THEN
                                                DELAY2 <= x"00000000";
                                                FLECHAESTADO <= x"00";
                                        ELSE
                                                DELAY2 <= DELAY2 + x"01";
                                        END IF;
                                WHEN OTHERS =>
                                        FLECHAESTADO <= x"00";

                        END CASE;
END IF;

END PROCESS;
```

**-- Processo responsável pelo inimigo número 1**
**-- Os inimigos aparecem randomicamente na tela, para isso é feito um cálculo que se encontra no código abaixo.**
**-- Caso o inimigo for acertado, seu sinal recebe 0 e sua posição é mandada para fora da tela.**
**-- Se o inimigo chegar no início da tela e seu sinal estiver em 1, ou seja, o inimigo ainda está vivo, o jogo acaba.**
**-- Se o inimigo estiver vivo, com sinal = 1, ele decrementa uma posição, fazendo com que se movimente para a esquerda.**

```
PROCESS (clk, reset)

BEGIN

IF RESET = '1' THEN
        ENEMY1CHAR <= "01100001";
        ENEMY1COR <= "0001"; -- 0 VERMELHO
        ENEMY1POS <= x"04BB";  --fora da tela
        DELAY3 <= x"00000000";
        ENEMY1ESTADO <= x"00";
        ENEMY1SINAL <= '0';
        ENDGAME1 <= '0';

ELSIF (clk'event) and (clk = '1') THEN

                        CASE ENEMY1ESTADO IS
                                WHEN x"00" =>
                                        IF (GAMESTART = '0') THEN
                                                ENEMY1ESTADO <= x"02";
                                        ELSE
                                                --Nao tem enemy1 na tela?
```

```
                                                IF (ENEMY1SINAL = '0') THEN    --esta fora tela
                                                        ENEMY1SINAL <= '1';
                                                        ENEMY1POS <=
std_logic_vector(to_unsigned((40*(((conv_integer(rd)+111) MOD 30))+39),16));
                                                END IF;
                                                ENEMY1ESTADO <= x"01";
                                        END IF;
                                WHEN x"01" =>
                                        -- INCREMENTA A POS DO ENEMY1
                                                IF (ACERTOU1 = '1') THEN    --FLECHA E ENEMY1
                                                        ENEMY1SINAL <= '0';
                                                        ENEMY1POS <= x"04BB";
                                                        --Soma PONTO;
                                                ELSIF (((conv_integer(ENEMY1POS) MOD 40) = 0) AND
(ENEMY3SINAL = '1')) THEN
                                                        ENDGAME1 <= '1';
                                                ELSIF ENEMY1SINAL = '1' THEN
                                                        ENEMY1POS <= ENEMY1POS - x"01";
                                                END IF;

                                                ENEMY1ESTADO <= x"02";

                                WHEN x"02" =>  -- Delay da Flecha

                                        IF DELAY3 >= x"0000AFFF" THEN
                                                DELAY3 <= x"00000000";
                                                ENEMY1ESTADO <= x"00";
                                        ELSE
                                                DELAY3 <= DELAY3 + x"01";
                                        END IF;

                                WHEN OTHERS =>
                                        ENEMY1ESTADO <= x"00";

                        END CASE;
END IF;

END PROCESS;
```

## -- Processo responsável pelo inimigo 2
## -- Este inimigo tem o código similar ao primeiro inimigo, porém ele consegue atirar flechas.

```
PROCESS (clk, reset)

BEGIN

IF RESET = '1' THEN
        ENEMY2CHAR <= "01100010";
        ENEMY2COR <= "0101"; -- 0 ROXO
        ENEMY2POS <= x"04BB";  --fora da tela
        DELAY4 <= x"00000000";
        ENEMY2ESTADO <= x"00";
        ENEMY2SINAL <= '0';
        ENDGAME2 <= '0';

ELSIF (clk'event) and (clk = '1') THEN

                        CASE ENEMY2ESTADO IS
                                WHEN x"00" =>
                                        IF (GAMESTART = '0') THEN
                                                ENEMY2ESTADO <= x"02";
                                        ELSE
                                                --Nao tem enemy2 na tela?
                                                IF (ENEMY2SINAL = '0') THEN    --nao esta fora tela
```

```
                                                ENEMY2SINAL <= '1';
                                                ENEMY2POS <=
std_logic_vector(to_unsigned((40*((conv_integer(rd) MOD 30))+39),16));
                                        END IF;

                                                ENEMY2ESTADO <= x"01";
                                END IF;

                        WHEN x"01" =>
                                -- INCREMENTA A POS DO ENEMY1
                                IF (ACERTOU2 = '1') THEN    --FLECHA E ENEMY1
                                        ENEMY2SINAL <= '0';
                                        ENEMY2POS <= x"04BB";
                                        -- SOME PONTO
                                ELSIF (((conv_integer(ENEMY2POS) MOD 40) = 0) AND
(ENEMY2SINAL = '1')) THEN
                                        ENDGAME2 <= '1';
                                ELSIF (ENEMY2SINAL = '1') THEN          --FAZER com
que ele e a flecha nao andem juntos
                                        ENEMY2POS <= ENEMY2POS - x"01";
                                END IF;

                                ENEMY2ESTADO <= x"02";

                        WHEN x"02" =>  -- Delay    Fazer um delay dentro desse para o enemy, usar esse
para a flecha

                                IF DELAY4 >= x"0000FFFF" THEN
                                        DELAY4 <= x"00000000";
                                        ENEMY2ESTADO <= x"00";
                                ELSE
                                        DELAY4 <= DELAY4 + x"01";
                                END IF;



                        WHEN OTHERS =>
                                ENEMY2ESTADO <= x"00";

                END CASE;
END IF;

END PROCESS;
```

**-- Processo responsável pela flecha do Inimigo 2**
**-- Sua posição é iniciada na frente do Inimigo 2**
**-- Quando sua posição se encontra com o Link, o jogo encerra.**
**-- Se a flecha não atingir posições iniciais do vetor (... MOD 40) = 0, ela continuará decrementando uma posição, assim a flecha continua seguindo em frente.**

```
PROCESS (clk, reset)

BEGIN

IF RESET = '1' THEN
        ENEMY2FLECHACHAR <= "01100100";
        ENEMY2FLECHACOR <= "0110"; -- 0 X
        ENEMY2FLECHAPOS <= x"0000";  --fora da tela
        DELAY5 <= x"00000000";
        ENEMY2FLECHAESTADO <= x"00";
        ENEMY2FLECHASINAL <= '0';
        ENDGAME4 <= '0';

ELSIF (clk'event) and (clk = '1') THEN
```

```
                CASE ENEMY2FLECHAESTADO IS
                        WHEN x"00" =>
                                IF (GAMESTART = '0') THEN
                                        ENEMY2FLECHAESTADO <= x"02";
                                ELSE
                                        --Atirou flecha?
                                        IF (ENEMY2FLECHASINAL = '0') THEN
                                                ENEMY2FLECHASINAL <= '1';
                                                ENEMY2FLECHAPOS <= ENEMY2POS -
x"0001"; --Flecha comeca na frente do enemy2
                                        END IF;

                                        ENEMY2FLECHAESTADO <= x"01";
                                END IF;

                        WHEN x"01" =>
                                -- INCREMENTA A POS DA FLECHA
                                        IF (ENEMY2FLECHAPOS = LINKPOS) THEN
                                                ENDGAME4 <= '1';
                                        ELSIF ((NOT((conv_integer(ENEMY2FLECHAPOS) MOD
40) = 0)) AND ENEMY2FLECHASINAL = '1') THEN
                                                ENEMY2FLECHAPOS <= ENEMY2FLECHAPOS
- x"0001";
                                        ELSE
                                                ENEMY2FLECHAPOS <= x"04B0";     --fora da
tela
                                                ENEMY2FLECHASINAL <= '0';
                                        END IF;

                                        ENEMY2FLECHAESTADO <= x"02";

                        WHEN x"02" =>  -- Delay da Flecha

                                IF DELAY5 >= x"00001FFF" THEN
                                        DELAY5 <= x"00000000";
                                        ENEMY2FLECHAESTADO <= x"00";
                                ELSE
                                        DELAY5 <= DELAY5 + x"01";
                                END IF;
                        WHEN OTHERS =>
                                ENEMY2FLECHAESTADO <= x"00";

                END CASE;
END IF;

END PROCESS;
```

**-- Processo responsável pelo inimigo 3**
**-- O inimigo 3 segue o mesmo modelo de código que o inimigo 1, porém esse tem mais vida, tornando a batalha mais difícil.**
**-- Se a variável de acerto for igual a 1, sua vida decrementa 1, sendo que o total é 3 vidas.**
**-- Se ele morrer, o inimigo reaparece com sua vida cheia.**

```
PROCESS (clk, reset)

BEGIN

IF RESET = '1' THEN
        ENEMY3CHAR <= "01100011";
        ENEMY3COR <= "1111"; -- 0 Branco
        ENEMY3POS <= x"04BB";  --fora da tela
        DELAY6 <= x"00000000";
        ENEMY3ESTADO <= x"00";
```

```
            ENEMY3VIDA <= "11";
            ENEMY3SINAL <= '0';
            ENEMY3SINAL_VAR <= '0';
            ENDGAME3 <= '0';

ELSIF (clk'event) and (clk = '1') THEN

                        CASE ENEMY3ESTADO IS
                                    WHEN x"00" =>
                                                IF (GAMESTART = '0') THEN
                                                            ENEMY3ESTADO <= x"02";
                                                ELSE
                                                            --Nao tem enemy3 na tela?
                                                            IF (ENEMY3SINAL = '0') THEN    --esta fora tela
                                                                        ENEMY3SINAL <= '1';
                                                                        ENEMY3POS <=
std_logic_vector(to_unsigned((40*(((conv_integer(rd)+280) MOD 30))+39),16));
                                                            END IF;

                                                            ENEMY3ESTADO <= x"01";
                                                END IF;
                                    WHEN x"01" =>
                                                -- INCREMENTA A POS DO ENEMY3
                                                            IF (ACERTOU3 = '1') THEN    --FLECHA E ENEMY3
                                                                        ENEMY3VIDA <= ENEMY3VIDA - "01";
                                                                        ENEMY3SINAL_VAR <= '1';
                                                                        IF (ENEMY3VIDA = "00") THEN
                                                                                    ENEMY3SINAL <= '0';
                                                                                    ENEMY3POS <= x"04BB";
                                                                                    ENEMY3VIDA <= "11";
                                                                        END IF;
                                                            ELSIF (((conv_integer(ENEMY3POS) MOD 40) = 0) AND
(ENEMY3SINAL = '1')) THEN
                                                                        ENDGAME3 <= '1';
                                                            ELSIF ENEMY3SINAL = '1' THEN
                                                                        ENEMY3POS <= ENEMY3POS - x"01";
                                                            END IF;

                                                            ENEMY3ESTADO <= x"02";

                                    WHEN x"02" =>  -- Delay da Flecha

                                                IF DELAY6 >= x"0000FFFF" THEN
                                                            DELAY6 <= x"00000000";
                                                            ENEMY3SINAL_VAR <= '0';
                                                            ENEMY3ESTADO <= x"00";
                                                ELSE
                                                            DELAY6 <= DELAY6 + x"01";
                                                END IF;

                                    WHEN OTHERS =>
                                                ENEMY3ESTADO <= x"00";

                        END CASE;
END IF;

END PROCESS;
```

## 5. Conclusões

Em suma, como pode-se notar, a implementação do jogo foi de grande importância e de grande interesse para um aluno de ciência da computação, pois foi aprendido os conceitos básicos da linguagem VHDL e também sobre programação de jogos digitais.