```
--Descrição do circuito
--Feito por Mario Raffo 2011
--Modificado WJC 2012

LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;
USE IEEE.STD_LOGIC_ARITH.ALL;
USE IEEE.NUMERIC_STD.ALL;

ENTITY fibonacci IS
      PORT (SIGNAL write_enable_i: IN STD_LOGIC;
            SIGNAL data_in        : IN NATURAL;
            SIGNAL read_enable_i  : IN STD_LOGIC;
            SIGNAL data_out       : OUT NATURAL;
            SIGNAL status_o       : OUT STD_LOGIC_VECTOR(1 DOWNTO
0);             SIGNAL irq_o          : OUT STD_LOGIC);
END fibonacci;

ARCHITECTURE behavior_w OF fibonacci IS

  CONSTANT PERIOD : TIME := 10 ns;

BEGIN

      PROCESS

            VARIABLE   n_anterior1: NATURAL :=0;
            VARIABLE   n_anterior2: NATURAL :=0;
            VARIABLE   n_fibonacci: NATURAL :=0;
            VARIABLE   n_max      : NATURAL :=0;

      BEGIN

                        status_o <= "00";
                        irq_o <= '0';
                        data_out <= 0;


                        WAIT FOR 1*PERIOD;

                        WHILE(write_enable_i/='1') LOOP
                          WAIT FOR 1*PERIOD;
                        END LOOP;

                        n_anterior1 :=1;
                        n_anterior2 :=0;
                        n_max :=data_in;
                        status_o <= "01";
                        WAIT FOR 1*PERIOD;

                        IF (n_max =0) or (n_max=1) THEN
                              n_fibonacci :=n_max;
                        ELSE
                              WHILE (n_max /= 1) LOOP
                              n_fibonacci :=n_anterior1
+n_anterior2;
```

```
                        n_anterior2 :=n_anterior1;
                        n_anterior1 :=n_fibonacci;
                        n_max :=n_max-1;
                        END LOOP;
                    END IF;

                    data_out <= n_fibonacci;
                    irq_o <= '1';
                    status_o <= "10";
                    WAIT FOR 1*PERIOD;

                    WHILE (read_enable_i/='1') LOOP
                       WAIT FOR 1*PERIOD;
                    END LOOP;

                    WAIT FOR 1*PERIOD;

        END PROCESS;


END behavior_w;
```