```
library IEEE;
  use IEEE.STD_LOGIC_1164.ALL;
  use IEEE.STD_LOGIC_ARITH.ALL;
  use IEEE.STD_LOGIC_UNSIGNED.ALL;

  -- programme pour tester le bus parallèle et les cartes DDS AD9852--
  entity XILINX_DDS_AD9852 is

  Port (
  MASTER_CLOCK : in STD_LOGIC;
  reset : in STD_LOGIC;
  dout : out std_logic_vector(7 downto 0);
  d_address : out std_logic_vector(7 downto 0);
  BP0 : in STD_LOGIC;
  DECODE_A0_A1_IN : in STD_LOGIC_VECTOR (1 downto 0);
  ADDRESS_BUS_CARD_in : in STD_LOGIC_VECTOR (5 downto 0);
  ADDRESS_BUS_CARD : out STD_LOGIC_VECTOR (5 downto 0);
  DECODE_A0_A1_OUT : out STD_LOGIC_VECTOR (1 downto 0);
  OUTPUT_GND : out STD_LOGIC_VECTOR (15 downto 0)
  );

  end XILINX_DDS_AD9852;

  architecture Behavioral of XILINX_DDS_AD9852 is
  TYPE mem_data IS ARRAY (0 TO 7) OF std_logic_vector(7 DOWNTO 0);
  --déclaration de zone mémoire pour les datas--
  TYPE mem_address IS ARRAY (0 TO 7) OF std_logic_vector(7 DOWNTO 0);
  --déclaration de zone mémoire pour les adresses--
  SIGNAL count_1MHz : integer := 255; --déclaration d'un compteur 8
bits--
  SIGNAL clock_1MHz_int : STD_LOGIC :='0'; -- signal issus du  diviseur
d'horloge--
  SIGNAL compte : STD_LOGIC_VECTOR (3 DOWNTO 0):="0000";

  constant data : mem_data := (
  ("01001111"), --data PLL x 15 clk = 20MHz x15= 300MHz
  ("00001111"), --data amplitude;
  ("00001000"), --Fout = 10MHz AD9852
  ("10001000"), --Fout = 10MHz AD9852
  ("10001000"), --Fout = 10MHz AD9852
  ("10001000"), --Fout = 10MHz AD9852
  ("10001000"), --Fout = 10MHz AD9852
  ("10001000")); --Fout = 10MHz AD9852

  constant address : mem_address := (
  ("00011110"), -- adresse HEX=0x1E; AD9852 parallel address
  ("00100001"), -- adresse HEX=0x21; AD9852 parallel address
  ("00001001"), -- adresse HEX=0x09; AD9852 parallel address
  ("00001000"), -- adresse HEX=0x08; AD9852 parallel address
  ("00001101"), -- adresse HEX=0x07; AD9852 parallel address
  ("00000110"), -- adresse HEX=0x06; AD9852 parallel address
  ("00000101"), -- adresse HEX=0x05; AD9852 parallel address
  ("00000100")); -- adresse HEX=0x04; AD9852 parallel address

  begin
  -- GND ----------
  OUTPUT_GND <= (others => '0');
```

```
  ADDRESS_BUS_CARD <= ADDRESS_BUS_CARD_in; --adresse de décodage des
DDS--
  DECODE_A0_A1_OUT <= DECODE_A0_A1_IN; -- initialisation des  données à
transmettre A0 A1--
  --STROBE_out <= clock_1MHz_int;

  --Diviseur par 50 50MHz/50 = 1MHz
  PROCESS
  BEGIN
  wait until MASTER_CLOCK'EVENT and MASTER_CLOCK = '1';
  IF count_1MHz < 50 THEN
  count_1MHz <= count_1MHz + 1;
  ELSE
  count_1MHz <= 0;
  END IF;
  IF count_1MHz < 25 THEN
  clock_1MHz_int <='0';
  ELSE
  clock_1MHz_int <='1' ;
  end if;
  end process;


  --compteur et envoi des datas sure le bus parallèle--
  process(clock_1MHz_int,reset)
  begin
  if reset ='1' then
  compte <= "0000";
  dout <="00000000";
  d_address <="00000000";

  elsif clock_1MHz_int'event and clock_1MHz_int='1' then

  IF compte < 8 and BP0 ='1' THEN
  dout <= data(CONV_INTEGER(compte));
  d_address <= address(CONV_INTEGER(compte));
  end if;
  IF compte > 7 THEN
  compte <= "0000";
  ELSE
  compte <=compte + 1;
  END IF;
  end if;
  end process;

  end Behavioral;
```