VLSI Lab Report Assignment 3 Annexure 2

BCSE 4th Year 2nd Semester

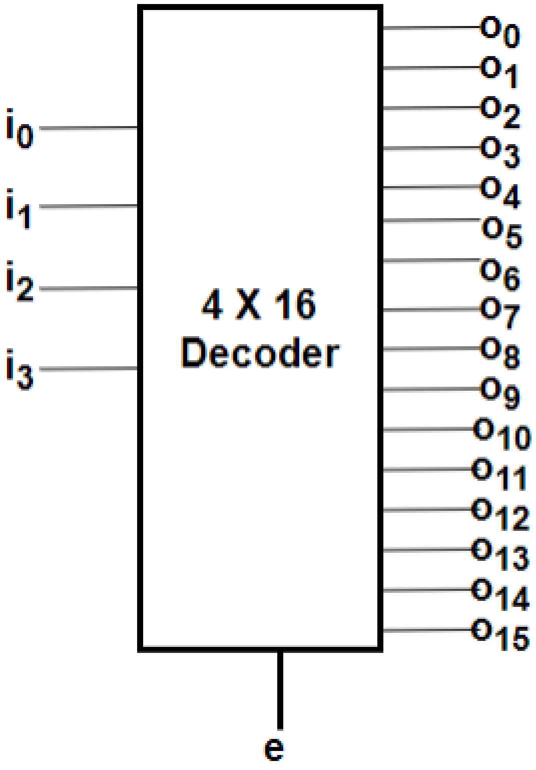
Name: ***Priti Shaw***
Roll No. : ***001710501076***
Batch: ***A3***

1. Design 4X16 Decoder using function and procedure.

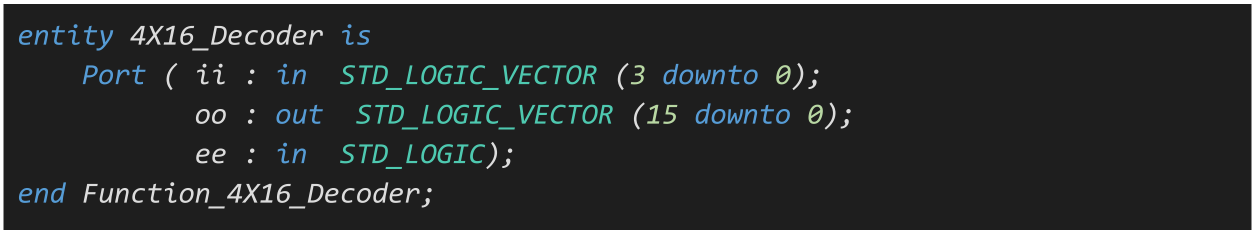
Description

A decoder is a combinational circuit that has N input lines and a maximum of 2 N output lines. One of these outputs will be active high based on the combination of inputs present when the decoder is enabled. That means the decoder detects a particular code. A 1 to 2 decoder has 1 input line and 2 output lines. An enable input is provided to switch the decoder on and off.

Block Diagram

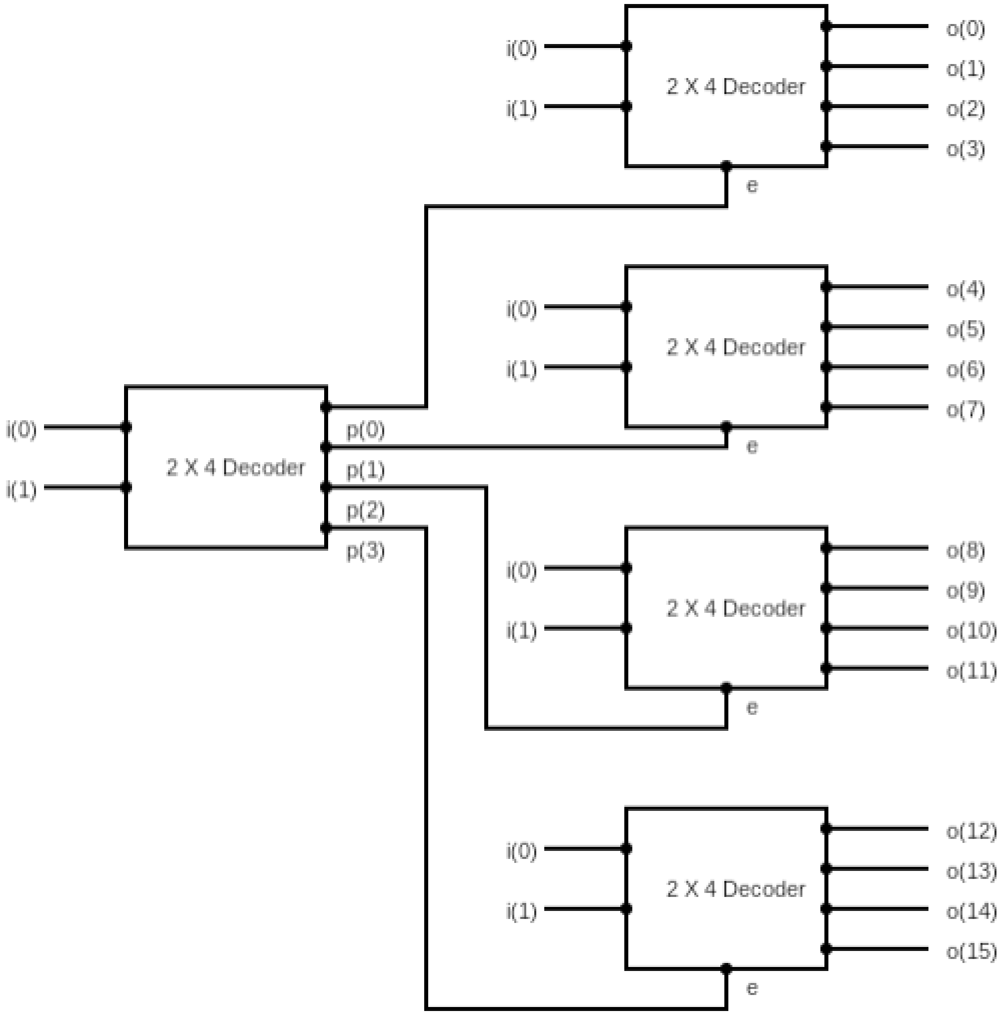


Entity



```
entity 4X16_Decoder is
    Port ( ii : in  STD_LOGIC_VECTOR (3 downto 0);
           oo : out  STD_LOGIC_VECTOR (15 downto 0);
           ee : in  STD_LOGIC);
end Function_4X16_Decoder;
```

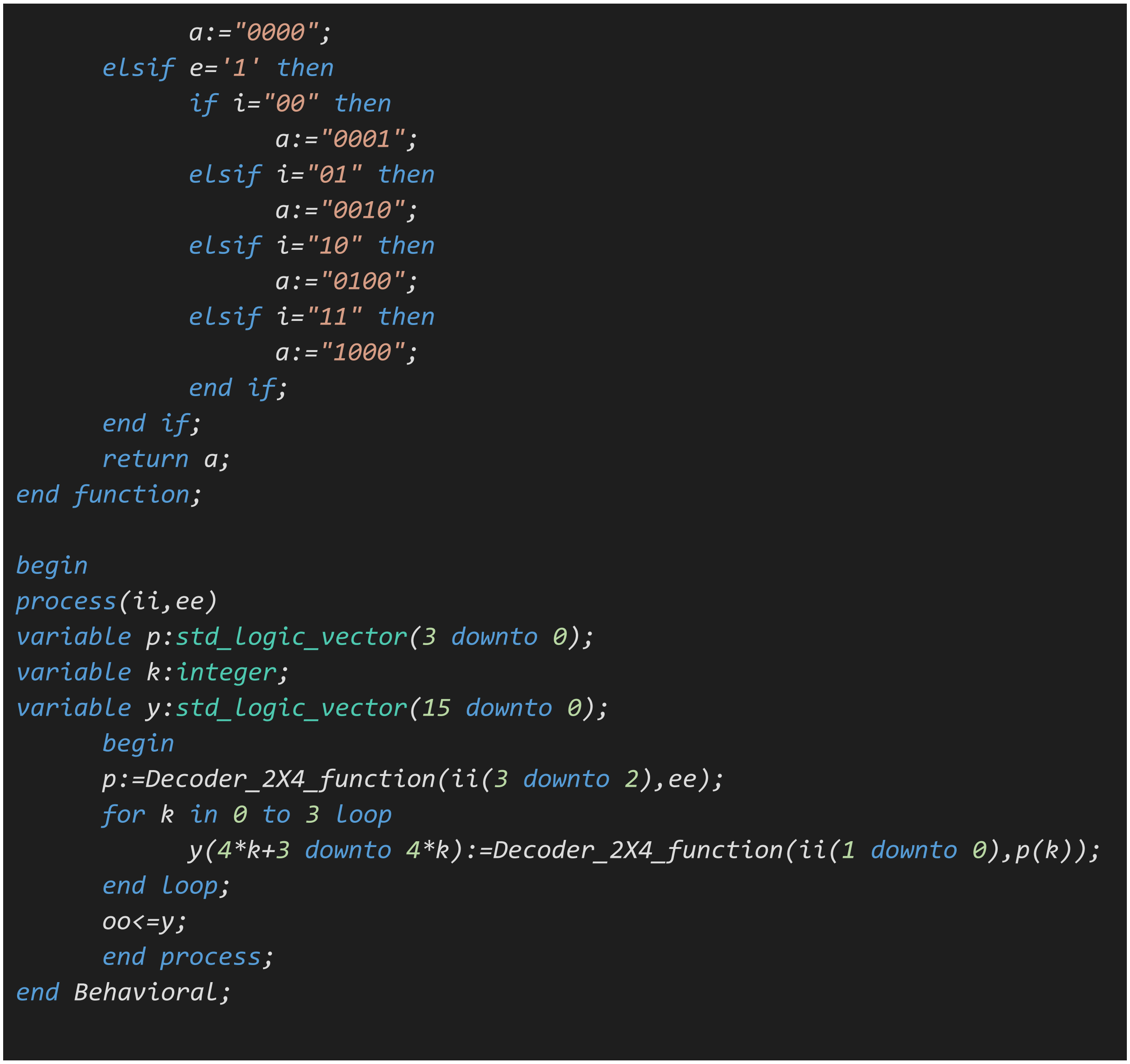
Circuit Diagram



Architecture

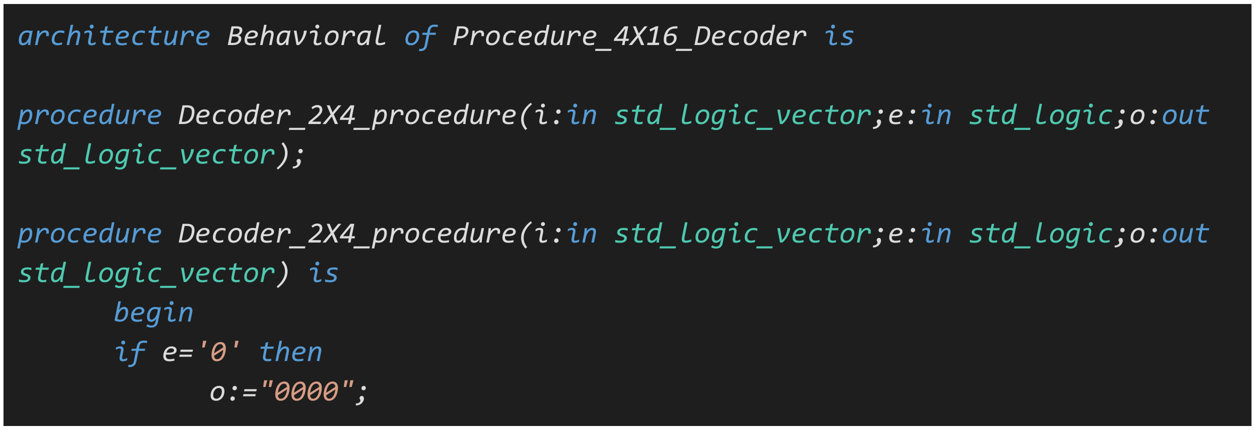
a) Using function of 2X4 decoders only

```
architecture Behavioral of Function_4X16_Decoder is

function Decoder_2X4_function(i:in std_logic_vector;e:in std_logic)return
std_logic_vector;

function Decoder_2X4_function(i:in std_logic_vector;e:in std_logic)return
std_logic_vector is
      variable a:std_logic_vector(3 downto 0);
      begin
      if e='0' then
```



```
            a:="0000";
      elsif e='1' then
            if i="00" then
                  a:="0001";
            elsif i="01" then
                  a:="0010";
            elsif i="10" then
                  a:="0100";
            elsif i="11" then
                  a:="1000";
            end if;
      end if;
      return a;
end function;

begin
process(ii,ee)
variable p:std_logic_vector(3 downto 0);
variable k:integer;
variable y:std_logic_vector(15 downto 0);
      begin
      p:=Decoder_2X4_function(ii(3 downto 2),ee);
      for k in 0 to 3 loop
            y(4*k+3 downto 4*k):=Decoder_2X4_function(ii(1 downto 0),p(k));
      end loop;
      oo<=y;
      end process;
end Behavioral;
```

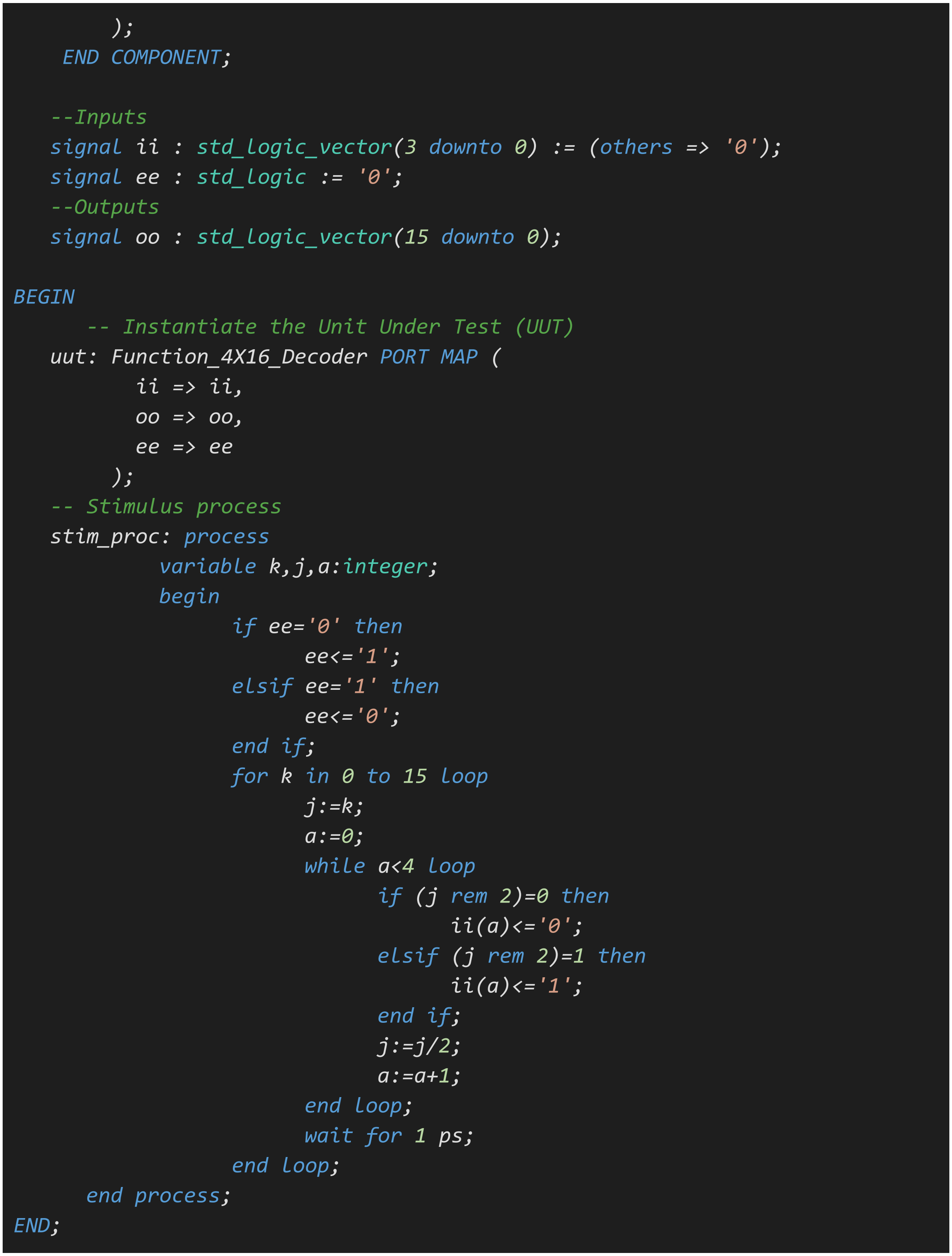
b) Using procedure of 2X4 decoders only



```
architecture Behavioral of Procedure_4X16_Decoder is

procedure Decoder_2X4_procedure(i:in std_logic_vector;e:in std_logic;o:out
std_logic_vector);

procedure Decoder_2X4_procedure(i:in std_logic_vector;e:in std_logic;o:out
std_logic_vector) is
      begin
      if e='0' then
            o:="0000";
      elsif e='1' then
            if i="00" then
                  o:="0001";
            elsif i="01" then
                  o:="0010";
            elsif i="10" then
                  o:="0100";
            elsif i="11" then
                  o:="1000";
            end if;
      end if;
end procedure;

begin
process(ii,ee)
variable p:std_logic_vector(3 downto 0);
variable k:integer;
variable y:std_logic_vector(15 downto 0);
      begin
            Decoder_2X4_procedure(ii(3 downto 2),ee,p);
            for k in 0 to 3 loop
                  Decoder_2X4_procedure(ii(1 downto 0),p(k),y(4*k+3 downto
4*k));
            end loop;
            oo<=y;
      end process;
end Behavioral;
```

TestBench

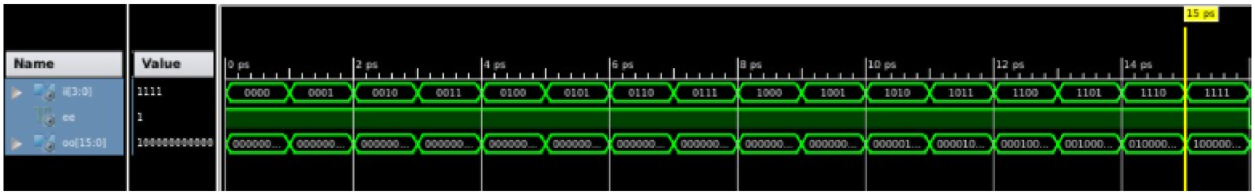
```
ENTITY Function_4X16_Decoder_TestBench IS
END Function_4X16_Decoder_TestBench;

ARCHITECTURE behavior OF Function_4X16_Decoder_TestBench IS

    -- Component Declaration for the Unit Under Test (UUT)
    COMPONENT Function_4X16_Decoder
    PORT(
         ii : IN  std_logic_vector(3 downto 0);
         oo : OUT  std_logic_vector(15 downto 0);
         ee : IN  std_logic
```



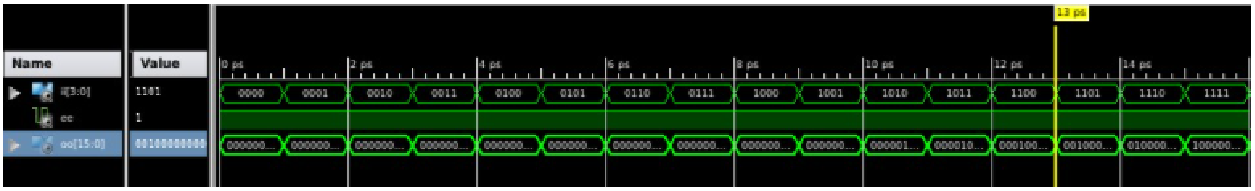
```
        );
    END COMPONENT;

   --Inputs
   signal ii : std_logic_vector(3 downto 0) := (others => '0');
   signal ee : std_logic := '0';
   --Outputs
   signal oo : std_logic_vector(15 downto 0);

BEGIN
      -- Instantiate the Unit Under Test (UUT)
   uut: Function_4X16_Decoder PORT MAP (
          ii => ii,
          oo => oo,
          ee => ee
        );
   -- Stimulus process
   stim_proc: process
            variable k,j,a:integer;
            begin
                  if ee='0' then
                        ee<='1';
                  elsif ee='1' then
                        ee<='0';
                  end if;
                  for k in 0 to 15 loop
                        j:=k;
                        a:=0;
                        while a<4 loop
                              if (j rem 2)=0 then
                                    ii(a)<='0';
                              elsif (j rem 2)=1 then
                                    ii(a)<='1';
                              end if;
                              j:=j/2;
                              a:=a+1;
                        end loop;
                        wait for 1 ps;
                  end loop;
      end process;
END;
```

Timing Diagram

a) Using function of 2X4 decoders only



b) Using procedure of 2X4 decoders only


13 ps
Name
Value
ii[3:0]
1101
ee
1
oo[15:0]
0010000000000
0 ps
2 ps
4 ps
6 ps
8 ps
10 ps
12 ps
14 ps
0000
0001
0010
0011
0100
0101
0110
0111
1000
1001
1010
1011
1100
1101
1110
1111
000000...
000000...
000000...
000000...
000000...
000000...
000000...
000000...
000000...
000000...
000001...
000010...
000100...
001000...
010000...
100000...