Experiment No. 4
Hamming Code Generator
ECE 446

Peter Chinetti

September 22, 2014

Date Performed: September 16, 2014
Instructor: Professor Shanechi

1 Introduction

Whenever data is transmitted from one location to another, either between components in a single system or between separate computers, ensuring that the data arrives at its destination error free is of critical importance. As a set of data is being transmitted over some form of communication line, ambient electronic noise may cause one or more bits in the transmission to become corrupted. Often, this corruption takes the form of a bit flip, either from a one to a zero, or vice versa. The probability of one such error occurring for an individual bit in a given transmission is very small, but as the size of the transmission increases, so does the probability of a bit flip occurring somewhere in the transmission. Detecting and/or correcting these bit flips is the goal of every error detection and correction code.

2 Procedure

a. Write VHDL to implement encoder/decoder logic.

b. Assign pins to ports

c. Simulate

d. Program and Test

3 Equipment

- PC
- Spartan-3E development board

4 Code

4.1 Top-level Module

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

entity hamming is
    Port ( Bin : in  STD_LOGIC_VECTOR (4 downto 1);
           Bout : out  STD_LOGIC_VECTOR (4 downto 1);
       SW : in STD_LOGIC_VECTOR (7 downto 1) );
end hamming;

architecture Behavioral of hamming is

  signal B, BS : STD_LOGIC_VECTOR(7 downto 1);
  signal C : STD_LOGIC_VECTOR(3 downto 1);

  component encoder
  Port ( d : in  STD_LOGIC_VECTOR(4 downto 1);
         b : out  STD_LOGIC_VECTOR(7 downto 1) );
  end component;

  component decoder
  Port ( b : in  STD_LOGIC_VECTOR(7 downto 1);
         c : out  STD_LOGIC_VECTOR(3 downto 1) );
  end component;

  component corrector
  Port ( b : in  STD_LOGIC_VECTOR(7 downto 1);
         c : in  STD_LOGIC_VECTOR(3 downto 1);
         o : out STD_LOGIC_VECTOR(4 downto 1) );
  end component;
begin
  encoder_0: encoder
    port map (
      d => Bin,
      b => B
    );

  BS <= B XOR SW;

  decoder_0: decoder
    port map (
      b => BS,
      c => C
    );
  corrector_0: corrector
    port map (
      b => BS,
      c => C,
      o => Bout
    );
end Behavioral;
```

hamming.vhd

4.2 Hamming Encoder

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

entity encoder is
  Port ( D : in  STD_LOGIC_VECTOR(4 downto 1);
         B : out  STD_LOGIC_VECTOR(7 downto 1)
);
end encoder;

architecture enc_arch of encoder is
begin
  B(1) <= D(1) xor D(2) xor D(4);
  B(2) <= D(1) xor D(3) xor D(4);
  B(3) <= D(1);
  B(4) <= D(2) xor D(3) xor D(4);
  B(5) <= D(2);
  B(6) <= D(3);
  B(7) <= D(4);
end enc_arch;
```

hamming_encoder.vhd

4.3 Hamming Decoder

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

entity decoder is
  Port ( B : in  STD_LOGIC_VECTOR(7 downto 1);
         C : out  STD_LOGIC_VECTOR(3 downto 1));
end decoder;

architecture dec_arch of decoder is
begin
  C(1) <= B(1) xor B(3) xor B(5) xor B(7);
  C(2) <= B(2) xor B(3) xor B(6) xor B(7);
  C(3) <= B(4) xor B(5) xor B(6) xor B(7);
end dec_arch;
```

hamming_decoder.vhd

4.4 Hamming Corrector

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

entity corrector is
  Port ( B : in  STD_LOGIC_VECTOR(7 downto 1);
         C : in  STD_LOGIC_VECTOR(3 downto 1);
         O : out STD_LOGIC_VECTOR(4 downto 1)
);
```

```
9  end corrector;

11 architecture cor_arch of corrector is
   begin
13   process(C, B)
     begin
15     if C = "011" then
         O(1) <= not B(3);
17       O(2) <= B(5);
         O(3) <= B(6);
19       O(4) <= B(7);
       elsif C = "101" then
21       O(1) <= B(3);
         O(2) <= not B(5);
23       O(3) <= B(6);
         O(4) <= B(7);
25     elsif C = "110" then
         O(1) <= B(3);
27       O(2) <= B(5);
         O(3) <= not B(6);
29       O(4) <= B(7);
       elsif C = "111" then
31       O(1) <= B(3);
         O(2) <= B(5);
33       O(3) <= B(6);
         O(4) <= not B(7);
35     else
         O(1) <= B(3);
37       O(2) <= B(5);
         O(3) <= B(6);
39       O(4) <= B(7);
       end if;
41   end process;
   end cor_arch;
```

hamming_corrector.vhd

5 Conclusions

The purpose of this lab was achieved. A Hamming encoder/decoderj was built and tested. Operation was verified through changing each bit transmitted and observing that the output signal remain uncorrupted.