COMP2421 Computer Organization

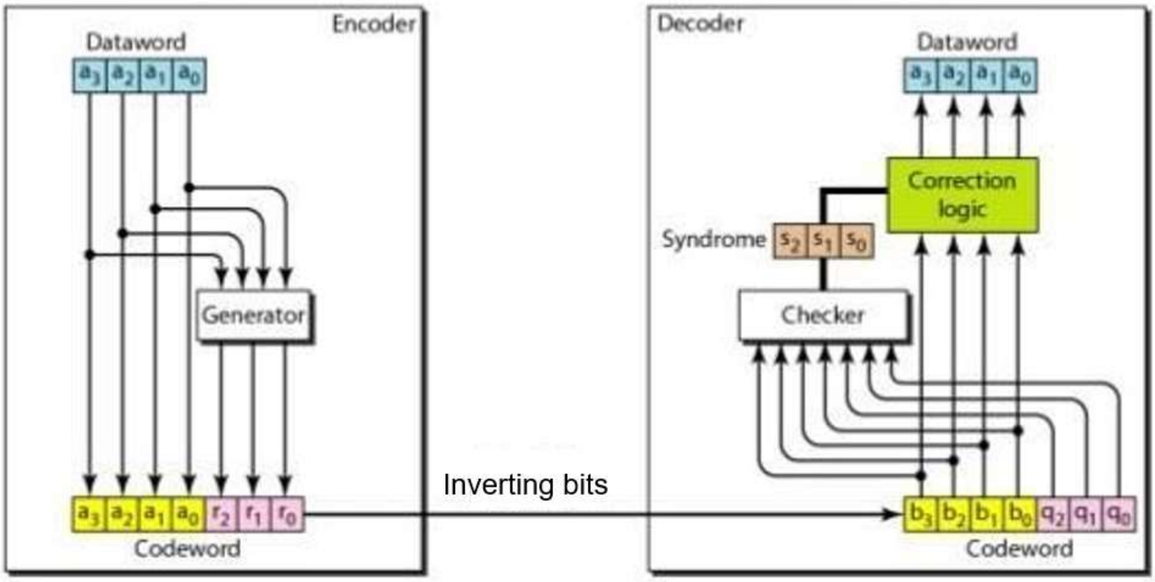
Programming Assignment 2

Due Date and Time: Apr 9, 2023, at 11:59pm

Hamming Code

1. Objectives

The objective of this assignment is to use combinational logic to implement the (7-4) Hamming Code using Logisim. By completing the task, we will implement the Encoder and the Decoder as required in the figure below.



In the Encoder, a dataword (a3,a2,a1,a0) will be provided at the beginning. With the dataword, the corresponding Hamming codeword (a3,a2,a1,a0,r2,r1,r0) will be generated. Part of the bits (r0,r1,a0) will be inverted as (b3,b2,b1,b0,q2,q1,q0). For example, (a3,a2,a1,a0,r2,r1,r0) as 1000111 can be changed (b3,b2,b1,b0,q2,q1,q0) as 1001100.

Taking the (b3,b2,b1,b0,q2,q1,q0) codeword, the Decoder will detect if there is any erroneous bit. When an error occurs, the Decoder will correct the respectively final dataword as (a3, a2, a1,a0).

Your task is thus to implement the combinational logic. This task is further divided to two major sub-tasks and described in the next section.

Note: this task requires you to read the introduction and specifications of some components in Logisim, and learn by yourself based on what you have learned through the Lab sessions. Related references as well as examples are provided in the end of this document.

2. Task Description

The whole task is composed of two sub-tasks: (1) build up the Encoder and (2) implement the Decoder.

2.1 Subtask One

First, you will need to design an Encoder with 16 inputs and 7 outputs. Each input corresponds to one button and the 7 outputs are the seven-bit encoded result (denoted as a3a2a1a0r2r1r0). Specifically, if button i is pressed, the dataword a3a2a1a0 would be the binary encoding of i. Example: if button 5 is pressed, a3a2a1a0 = 0101. You will use the component Button, Priority Encoder, and Splitter from Logisim to implement this function. Play with the Priority Encoder to make sure that it will produce the desired outputs. Especially, the bit width of the output of the Priority Encoder could be larger than 1. Thus, you will need a Splitter to split the output bits. You use a Hex Digital Display to show the values of a3a2a1a0 dataword. Play with this circuit until you got the desired results. The figure shows an example of the Hex Digital display with the value (1010 == A).

Next, you design/implement the Generator with the (7-4) Hamming coding method. The Generator will be resulted with r2r1r0 bit string. You use another Hex Digital Display to show the values of r2r1r0.

2.2 Subtask Two

Second, you are asked to design/implement the Decoder. Before doing the Decoder work, a simple circuit design to perform inversions some of the bits (r0,r1,a0). For example, (a3,a2,a1,a0,r2,r1,r0) as 1000111 can be changed (b3,b2,b1,b0,q2,q1,q0) as 1001100. You use 2 Hex displays to show the values of b3b2b1b0 and q2q1q0.

In the Decoder, there are 2 steps. The initial step is to develop the Checker for detecting if there is any erroneous bit in b3b2b1b0q2q1q0 bit string. The Checker would have a checking result as a syndrome (s2, s1, s0). The following table shows the meanings of the detecting results.

s2s1s0	Erroneous bit
000	None
100	q0
010	q1
110	b0
001	q2
101	b1
011	b2
111	b3

The second step of the Decoder has been developed the Correction Logic circuit. Based on the s2s1s0 result, the final dataword (a3a2a1a0) would be generated. In order to verify the results, 2 Hex displays would show final a3a2a1a0 and s2s1s0.

3. What to submit

1) One circuit. Submit a Hamming Code circuit that implements the desired functions. Name the file as “ HC_circuit_XXXXXXXX.circ”, where “XXXXXXXX” represents your 8-digit student number (the last letter is not needed).

2) An experiment report. The report should include the following contents:
   - Screen shot of the Hamming Code circuit.
   - Three pictures of 3 different buttons pressed. In each picture, you should show that when button i is pressed, the respectively Hex displays will be shown.

4. Assessment

- Subtask One (50 pts)
- Subtask Two (40 pts)
- Report (10 pts)

5. Useful References

1) Specifications of all components in Logisim. http://www.cburch.com/logisim/docs/2.7/en/html/libs/plexers/index.html
2) Example of using Priority Encoder. http://www.cs.unca.edu/brock/classes/Fall2012/csci255/labs/lab03.html
3) Hamming (7,4) Code Discussion and Implementation, https://michaeldipperstein.github.io/hamming.html
4) Logisim: How to use a hex display and a splitter, Logisim: How to use a hex display and a splitter - YouTube