1. Work the following problems based on material in Chapter 2 of text by A. Tanenbaum:
   a. A bit-map terminal has a 1024 $\times$ 768 display. The display is redrawn 75 times a second. How long is the pulse corresponding to one pixel?
   b. A manufacturer advertises that its color bit-map terminal can display $2^{24}$ different colors. Yet the hardware has only one byte for each pixel. How can this be done?
   c. When even-parity ASCII text is transmitted asynchronously at a rate of 2880 characters/second over a 28,800 bps modem, what percent of the received bits actually contain data (as opposed to overhead)?

2. From Chapter 2 and sections 5.5.7, 5.6.4, and 5.6.5 of Chapter 5 of the text by A. Tanenbaum, *Structured Computer Organization*, define what is meant by the following terms:
   a. Interrupt
   b. DMA
   c. Trap
   d. Cycle Stealing
   e. Bus Arbiter

3. Read Sections 3.2, 3.3, and 3.7 in Chapter 3 of text by A. Tanenbaum, *Structured Computer Organization*, Prentice-Hall, 2006, and work the following problems from Chapter 3:
   a. Problem 3-15 in 5th ed. (Problem 3-16 in 6th ed.)
   b. Problem 3-22 in 5th ed. (Problem 3-23 in 6th ed.)

4. A 64-bit static random access memory (RAM) integrated circuit (IC) chip is organized as 16 words with 4 bits per word with internal bi-directional data bus and tri-state data bus connection to the outside world. In addition to address and data lines, it has two control inputs (an enable (EN) input and a read/write (R/W) input); address decoding logic is included on the chip to select one of 16 words, and the function table for the control inputs is shown as follows:

<table>
<thead>
<tr>
<th>Enable</th>
<th>Read/Write</th>
<th>Operation</th>
<th>Condition of Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Write</td>
<td>(Data bus used as input)</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>Read</td>
<td>Value of selected word</td>
</tr>
<tr>
<td>1</td>
<td>X</td>
<td>Inhibit</td>
<td>High Impedance state</td>
</tr>
</tbody>
</table>

   a. Draw a (box) block diagram labelling the data input lines, data output lines, address selection lines, and control inputs for this IC. Note that the enable input (EN) on this chip is also called the “chip select (CS)” input on other similar RAM chips. (The low order bit is $A_0$ and the high order bit is $A_n$ for the address, etc. for the data lines.)
   b. Show the external connections of eight of these RAM chips that will produce a 64 word by 8 bit RAM. You need not draw all of the wires or label each and every pin so long as your labeling makes clear what is connected to what. Assume a 2 to 4 decoder package is available if appropriate; and if you use it, show it as a box with inputs and outputs clearly labeled and show where its outputs are used as inputs. (There must be no ambiguity among the connections.)
5. Consider a 8-bit data word with 4 check bits that uses a Hamming single error correcting code for the 12-bit codeword. By adding another parity bit to the coded word, the Hamming code can be used to correct a single error and detect double errors. If we include this additional parity bit, then a previous 12-bit coded word becomes 001110010100P_{13}, where P_{13} is evaluated from the exclusive-OR of the other 12 bits. This produces the 13-bit word 0011100101001 (even parity). When the 13-bit word is read from memory the check bits are evaluated and so is the parity P over the entire 13 bits. If P = 0, the parity is correct (even parity), but if P = 1, then the parity over the 13 bits is incorrect (odd parity). Letting S = s_3s_2s_1s_0 be the binary representation of the syndrome resulting from evaluation of the Hamming single error correcting parity check equations, then the following four cases can occur:

- If S = 0 and P = 0: No error occurred.
- If S ≠ 0 and P = 1: A single error occurred, which can be corrected.
- If S ≠ 0 and P = 0: A double error occurred, which is detected but cannot be corrected.
- If S = 0 and P = 1: An error occurred in the P_{13} bit.

Note that this scheme cannot detect more than two errors.

a. It is necessary to formulate the Hamming code for four data bits, d_3, d_5, d_6, d_7, together with three parity bits p_1, p_2, and p_4.
   - (1) Assign the parity bits and show the resulting 7-bit composite codeword for the data word 0010.
   - (2) Evaluate three parity check equations for the syndrome s_2s_1s_0 assuming no error has occurred.
   - (3) Assume an error in bit d_5 occurred during a write into memory. Show how the error in the bit is detected and corrected.
   - (4) Add parity bit P_8 to include a double-error detection capability in the code. Assume that errors occurred in bits p_2 and d_5. Show how the double error is detected.

b. Given the 8-bit data word 01011011, generate the 13-bit codeword for the Hamming code that corrects single errors and detects double errors.

c. Given the 11-bit data word 11001001010, generate the 16-bit codeword for the Hamming code that corrects single errors and detects double errors.

d. How many parity check bits must be included with the data word to achieve single error correction and double error detection when the data word contains:
   - (1) 16 bits;
   - (2) 32 bits;
   - (3) 48 bits;
   - (4) 64 bits;

6. A 12-bit Hamming-code word containing 8 bits of data and 4 parity bits is read from memory. What was the original 8-bit data word that was written into memory if the 12-bit word read out is as follows:

   (a) 000011101010
   (b) 101110001010
   (c) 101111101010

7. Design a synchronous binary counter using three JK negative-edge-triggered flipflops (having active high direct clear inputs) whose combinational circuit outputs (T1, T2, T3, and T4) produce a repeating sequence of four nonoverlapping clock pulses. (Hint: Use a modulo-8 binary counter each of whose outputs (T1, T2, T3, and T4) takes the value 1 for exactly one odd numbered state and takes the value 0 otherwise. Also note that the direct clear inputs on the flip-flops can be driven by a power-on reset circuit to clear all three flip-flops and initialize them to the zero state before starting the repeating count sequence by driving their clock inputs with a synchronizing clock oscillator signal.)
8. Redesign the 1-bit ALU in Figure 1 below so that the functions performed, as specified by the 2-bit control inputs $F_1F_0$ are those shown in the following table:

<table>
<thead>
<tr>
<th>$F_1$</th>
<th>$F_0$</th>
<th>ALU output</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>A+B addition</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>A .and. B logical AND</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>A straight-through</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>A logical complement</td>
</tr>
</tbody>
</table>

![Diagram of 1-bit ALU](image)

Figure 1: 1-bit ALU

9. Redesign the combinational shift circuit in $5^{th}$ ed. textbook Fig. 3-16, p. 154 (or $6^{th}$ ed. textbook Fig. 3-15, p. 164) to provide the following functionality with two shift control bits $H_1H_0$ replacing the single control line $C$ so that

<table>
<thead>
<tr>
<th>$H_1$</th>
<th>$H_0$</th>
<th>ALU output</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>No Shift (i.e. straight-through)</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>Logical right shift by 1 bit</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>Logical left shift by 1 bit</td>
</tr>
</tbody>
</table>

10. Design a 1 out of 16 decoder circuit with additional inputs $ENC$ and $T4$ (from problem 7) that decodes the four bit number $C_3C_2C_1C_0$ and produces an output pulse coincident with $T4$ when the enable input $ENC = 1$; the outputs remain all zeros if $ENC = 0$. 

![Diagram of 1 out of 16 decoder](image)