Problem 4

I.

Answer the following questions on logic circuits. Let us design a 2-bit comparator (CMP) that takes 2-bit positive input values $A = a_2a_1$ and $B = b_2b_1$. The CMP outputs c, where c is 1 when $A \geq B$, and 0 otherwise. The symbol of the 2-bit CMP is shown in Fig. 1.

(1) Write the truth table of the 1-bit comparator CMP_1 that outputs c_1 as shown below, where a_1 and b_1 are the input values.

$$c_1 = \begin{cases} 1, & a_1 \geq b_1 \\ 0, & a_1 < b_1 \end{cases}$$

(2) Write the truth table of the 1-bit comparator CMP_2 that outputs c_2 as shown below, where a_2, b_2, and c_1 are the input values. Note that c_1 is the output described in Question (1).

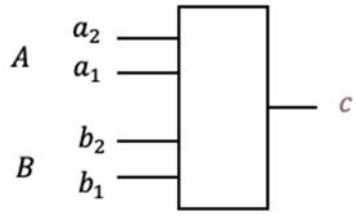
$$c_2 = \begin{cases} 1, & a_2 > b_2 \\ c_1, & a_2 = b_2 \\ 0, & a_2 < b_2 \end{cases}$$

(3) You can design the 2-bit CMP by combining the circuits of CMP_1 and CMP_2, described in Questions (1) and (2). Draw a circuit of the 2-bit CMP using the symbols shown in Fig. 2.

Next, let us design a sequential circuit MAX using the 2-bit CMP. When 2-bit positive values $X_1, X_2, \cdots$ are input sequentially, the MAX outputs the maximum value of the inputs. Design the MAX according to the following procedure.

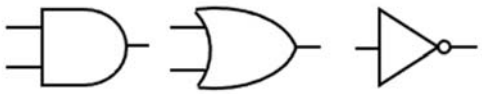
(4) Using the 2-bit CMP and a 2-bit D flip-flop (D-FF), design a circuit which stores $\max(A, B)$ in the 2-bit D-FF. Draw the circuit using the symbols shown in Figs. 1, 2, and 3.

(5) You can construct the sequential circuit MAX by assigning X_i to the input A, and $\max(X_1, X_2, \cdots, X_{i-1})$ to the input B of the circuit designed in Question (4). Draw the sequential circuit MAX using the symbols shown in Figs. 1, 2 and 3. You can assume that the initial input value of B is 00.



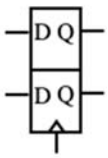
2-bit CMP

Fig. 1



AND OR NOT

Fig. 2



2-bit D-FF

Fig. 3

II.

Answer the following questions on storage and management of data using hash tables. Let us consider a method of managing positive integers in a hash table `table[N]` with N elements. Here, a hash function $H(x) = \mathrm{mod}(x, N)$ is used to indicate the index to store a positive integer x in the hash table. When positive integers have the same hash value, they are managed by the linked list of data structure `node` which is described in `Program 1`.

(1) Show the contents of `table[N]`, when $N = 11$ and integers {15, 53, 22, 59, 15, 41, 20} are stored using the hash function in sequence.

(2) `Program 2` describes `insert(x)` function that stores positive integer x in `table[N]`. Fill in the blanks of `Program 2` to complete the program in C language.

(3) Assume that `search(x)` function returns 1 if a given positive integer x is stored in `table[N]`, and it returns 0 otherwise. Write the `search(x)` function in C language.

(4) Describe in a few lines the point(s) to pay attention to, when you write a function that deletes a positive integer x stored in `table[N]`.

```
/* Program 1 */
struct node {
      int  value;
      struct node *next;
};
struct node *table[N];

/* Program 2 */
int H(int x) { return(x % N); }
void insert(int x) {
       struct node *new, *check;

       new = (struct node *)malloc(sizeof (struct node));
       new -> value = x;
       new -> next = NULL;
       check = table[H(x)];
       [                                    ]
}
```