

LỜI GIẢI THAM KHẢO ĐỀ THI CUỐI KÌ HTMT&NNLT HK201

Lưu ý: Nếu sinh viên có cách làm **đúng** khác với đáp án thì **vẫn được điểm**.

Câu 1 (b4): (1,0 điểm)

Viết chương trình hợp ngữ LC-3 thực hiện phép tính XOR 16 bit với đầu vào là 2 ô nhớ được lưu tại địa chỉ x4000 và x4001, kết quả được lưu vào ô nhớ x4010. Chương trình bắt đầu từ ô nhớ có địa chỉ x3000.

Lời giải:

Lưu ý: LC-3 không hỗ trợ phép tính logic OR nên phải phối hợp phép AND và luật De Morgan, tránh nhầm lẫn giữa phép cộng đại số (ADD) với phép OR logic. Trong một số trường hợp nếu đảm bảo 2 đầu vào bit không cùng bằng 1 thì cho phép dùng phép ADD thay cho việc OR.

.ORIG x3000 LDI R0, M_IN1 LDI R1, M_IN2	Lấy dữ liệu từ ô nhớ
NOT R2, R0 NOT R3, R1 AND R2, R2, R1 AND R3, R3, R0	Tính \overline{AB} và $A\overline{B}$
NOT R2, R2 NOT R3, R3 AND R2, R2, R3 NOT R2, R2	OR hai giá trị vừa tính được với nhau. Trong trường hợp này cho phép dùng phép ADD thay thế vì các bit tương ứng ở 2 giá trị đầu vào không thể cùng bằng 1 được.
STI R2, M_OUT HALT M_IN1 .FILL x4000 M_IN2 .FILL x4001 M_OUT .FILL x4010 .END	Lưu giá trị ra

Kết quả chạy thử

▪ x4000	0011001100110011	x3333	ST	R1, x3F34
▪ x4001	0101010101010101	x5555	AND	R2, R5, R5
▪ x4002	0000000000000000	x0000	NOP	
▪ x4003	0000000000000000	x0000	NOP	
▪ x4004	0000000000000000	x0000	NOP	
▪ x4005	0000000000000000	x0000	NOP	
▪ x4006	0000000000000000	x0000	NOP	
▪ x4007	0000000000000000	x0000	NOP	
▪ x4008	0000000000000000	x0000	NOP	
▪ x4009	0000000000000000	x0000	NOP	
▪ x400A	0000000000000000	x0000	NOP	
▪ x400B	0000000000000000	x0000	NOP	
▪ x400C	0000000000000000	x0000	NOP	
▪ x400D	0000000000000000	x0000	NOP	
▪ x400E	0000000000000000	x0000	NOP	
▪ x400F	0000000000000000	x0000	NOP	
▪ x4010	0110011001100110	x6666	LDR	R3, R1, #-26

Câu 2 (b4): (3,0 điểm)

Viết chương trình thực hiện các yêu cầu sau (có thể xây dựng các hàm con để bài làm được gọn và dễ hiểu hơn):

- (1) Yêu cầu người dùng nhập vào một chuỗi kí tự chỉ bao gồm các kí tự số (0 – 9), các kí tự chữ (từ a – z và từ A – Z), không được có các kí tự khác (kể cả khoảng trắng). Kiểm tra điều kiện và yêu cầu nhập lại **cho đến khi** thỏa mãn điều kiện (1 điểm).
- (2) Sắp xếp các kí tự trong chuỗi vừa được nhập theo thứ tự mã ASCII tăng dần. In ra chuỗi vừa được sắp xếp lại (1 điểm).
- (3) Xóa đi những kí tự bị lặp lại trong chuỗi vừa được sắp xếp. In ra chuỗi đã được loại những kí tự thừa (1 điểm).

Lời giải:

```
#include <stdio.h>
#include <conio.h>
#include <string.h>
char kiem_tra(char* s);
void sap_xep(char* s);
void xoa_ki_tu_trung(char* s);
char chuoi[256];
int main()
{
    // Nhập chuỗi
    do
    {
        puts("- Moi nhap chuoi:");
        gets(chuoi);
    } while (kiem_tra(chuoi));
    // Sắp xếp chuỗi
    sap_xep(chuoi);
    puts("- Chuoi da sap xep:"); puts(chuoi);
    // Xóa kí tự trùng
    xoa_ki_tu_trung(chuoi);
    puts("- Chuoi sau khi xoa ki tu trung:"); puts(chuoi);
}
char kiem_tra(char* s)
{
    int len = strlen(s);
    for (int i = 0; i < len; i++)
    {
        if ((s[i] >= '0') && (s[i] <= '9')) continue;
        if ((s[i] >= 'A') && (s[i] <= 'Z')) continue;
        if ((s[i] >= 'a') && (s[i] <= 'z')) continue;
        return 1;
    }
    return 0;
}
void sap_xep(char* s)
{
    int len = strlen(s);
    for (int i = 1; i < len; i++)
        for(int j = i; j > 0; j--)
            if (s[j] < s[j - 1]) {char temp = s[j]; s[j] = s[j-1]; s[j-1] = temp;}
}
void xoa_ki_tu_trung(char* s)
{
    // Lưu ý: Hàm dưới chỉ chạy trong trường hợp chuỗi đã được sắp thứ tự trước
    int len = strlen(s);
    int i = 0;
    while (i != len)
    {
        if (s[i] == s[i + 1])
        {
            for (int j = i + 1; j < len; j++) s[j] = s[j + 1];
            len--;
        }
        else i++;
    }
}
```

Chạy thử:

```
- Moi nhap chuoi:  
he thong may tinh  
- Moi nhap chuoi:  
hethongmaytinh&&  
- Moi nhap chuoi:  
Hethongmaytinh20202021  
- Chuoi da sap xep:  
00012222Haeghimnotty  
- Chuoi sau khi xoa ki tu trung:  
012Haeghimnoty
```