

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

KHOA KỸ THUẬT MÁY TÍNH



BÁO CÁO ĐỒ ÁN GIỮA KÌ

DESIGN A SIMPLE CALCULATOR USING AN 8051 MICROCONTROLLER

THIẾT KẾ MÁY TÍNH CẦM TAY ĐƠN GIẢN SỬ DỤNG VI ĐIỀU KHIỂN 8051

SINH VIÊN THỰC HIỆN:

21521810 – NGUYỄN QUỐC TRƯỜNG AN

21520258 – HỒ TẤN HUY

GIẢNG VIÊN HƯỚNG DẪN

TS. ĐỖ TRÍ NHỰT

THÀNH PHỐ HỒ CHÍ MINH, 26/04/2023

ĐẠI HỌC QUỐC GIA
THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC
CÔNG NGHỆ THÔNG TIN

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
ĐỘC LẬP – TỰ DO – HẠNH PHÚC

CHI TIẾT

TÊN ĐỒ ÁN TIẾNG VIỆT: THIẾT KẾ MÁY TÍNH CẦM TAY ĐƠN GIẢN SỬ DỤNG VI ĐIỀU KHIỂN 8051	
ENGLISH PROJECT NAME: DESIGN A SIMPLE CALCULATOR USING AN 8051 MICROCONTROLLER	
Giảng viên hướng dẫn: TS. Đỗ Trí Nhật, Khoa Kỹ Thuật Máy Tính	
Thời gian thực hiện: Từ ngày: 02/03/2023 Đến ngày: 22/04/2023	
Sinh viên thực hiện: Nguyễn Quốc Trường An, Hồ Tấn Huy	
Phương pháp thực hiện: Mô phỏng Ngôn ngữ lập trình: Ngôn ngữ C Phần mềm mô phỏng: Proteus	
Xác nhận của giảng viên (Ký và ghi rõ họ tên)	Tp. Hồ Chí Minh, 26/04/2023 Sinh viên (Ký và ghi rõ họ tên)

MỤC LỤC

CHƯƠNG 1: GIỚI THIỆU ĐỀ TÀI	5
1.1 Giới thiệu.....	5
1.1.1 Chức năng hệ thống	5
1.1.2 Cách thức hoạt động	5
1.2 Tổng quan các khối chức năng	6
1.2.1 Khối xử lý trung tâm.....	7
1.2.2 Khối dữ liệu ngõ vào	7
1.2.3 Khối hiển thị dữ liệu ngõ ra	7
1.2.4 Khối cấp nguồn.....	7
1.3 Cấu tạo, nguyên tắc hoạt động của các khối thành phần.....	7
1.3.1 Vi điều khiển 8051	8
1.3.2 Module LCD1602	10
1.3.3 Module bàn phím ma trận 4x4.....	13
CHƯƠNG 2 : THIẾT KẾ VÀ THI CÔNG MẠCH.....	15
2.1 Thiết kế.....	15
2.1.1 Khối điều khiển trung tâm	15
2.1.2 Module LCD1602	15
2.1.3 Module bàn phím ma trận 4x4.....	16
2.2 Thi công mạch	16

CHƯƠNG 3: LƯU ĐỒ GIẢI THUẬT VÀ HIỆN THỰC BẰNG NGÔN NGỮ C	17
3.1 Lưu đồ thuật toán chương trình chính	17
3.2 Lưu đồ thuật toán các chương trình con	18
3.2.1 Lưu đồ thuật toán quét phím và xử lý phím	18
3.2.3 Lưu đồ giải thuật kiểm tra biểu thức hợp lệ.....	19
3.2.2 Lưu đồ giải thuật hàm tính toán.....	20
3.2.3 Lưu đồ giải thuật hàm ghi kết quả	21
3.3 Hiện thực giải thuật bằng ngôn ngữ C	22
CHƯƠNG 4 THỰC NGHIỆM VÀ KIỂM TRA	28
4.1 Môi trường thực nghiệm.....	28
4.2 Kết quả thực nghiệm.....	29
CHƯƠNG 5: ĐÁNH GIÁ	31
5.1. Ưu điểm	31
5.2. Một số nhược điểm.....	31
5.3. Hướng phát triển.....	31
TÀI LIỆU THAM KHẢO	32

DANH MỤC HÌNH ẢNH

Hình 1 Sơ đồ khối của hệ thống.....	6
Hình 2 Sơ đồ chân vi điều khiển 8051	8
Hình 3 Sơ đồ chân LCD 16x2.....	11
Hình 4 Sơ đồ schematic của bàn phím ma trận 4x4.....	13
Hình 5 Hình ảnh bàn phím ma trận 4x4 quy ước phím bấm.....	14
Hình 6 Sơ đồ thiết kế mạch của hệ thống	15
Hình 7 Hình ảnh thi công mạch trên phần mềm mô phỏng Proteus	16
Hình 8 Lưu đồ thuật toán chương trình chính.....	17
Hình 9 Lưu đồ thuật toán quét phím	18
Hình 10 Lưu đồ thuật toán kiểm tra biểu thức hợp lệ	19
Hình 11 Lưu đồ thuật toán tính toán	20
Hình 12 Lưu đồ thuật toán hàm ghi kết quả	21

DANH MỤC BẢNG

Bảng 1 Bảng chức năng các chân của LCD 16x2.....	12
Bảng 2 Các mã lệnh của LCD 16x2.....	13
Bảng 3 Bảng các testcase đưa vào thực nghiệm	28
Bảng 4 Bảng kết quả thực nghiệm	29

CHƯƠNG 1: GIỚI THIỆU ĐỀ TÀI

1.1 Giới thiệu

Ngày nay, với sự tiến bộ của khoa học kỹ thuật, các thiết bị hỗ trợ con người ngày càng được cải tiến để phù hợp với nhu cầu đời sống thực tế. Các thiết bị điện tử ngày càng được thu nhỏ và chuyên biệt hóa chức năng, do đó việc sử dụng các thiết bị điện tử tích hợp vi điều khiển như một điều hiển nhiên bởi sự nhỏ gọn, tiện lợi của nó mang lại.

Thêm vào đó, kinh tế ngày càng phát triển, việc mua bán diễn ra khắp mọi nơi, kéo theo nhu cầu tính toán các con số lớn trở nên cần thiết và không thể thiếu. Nhận thấy được điều này, nhóm chúng em chọn chủ đề “Thiết kế máy tính cầm tay đơn giản sử dụng vi điều khiển 8051”.

Do thời gian và kiến thức còn hạn chế nên đề tài của nhóm chúng em còn nhiều thiếu sót, chúng em rất mong nhận được sự chỉ bảo tận tình của thầy để nhóm sửa chữa và hoàn thiện đề tài hơn nữa.

Chúng em xin chân thành cảm ơn thầy!

1.1.1 Chức năng hệ thống

- Hệ thống có khả năng thực hiện tính toán số nguyên không dấu hỗ trợ 4 phép tính cơ bản: cộng, trừ, nhân, chia (chia lấy phần nguyên). Chú ý, chỉ hỗ trợ phép tính gồm 2 toán hạng.
- Các toán hạng và kết quả có độ lớn tối đa 16 bit. Chú ý rằng khi kết quả phép toán vượt quá tầm biểu diễn của 16 bit, hệ thống sẽ cho ra kết quả sai.

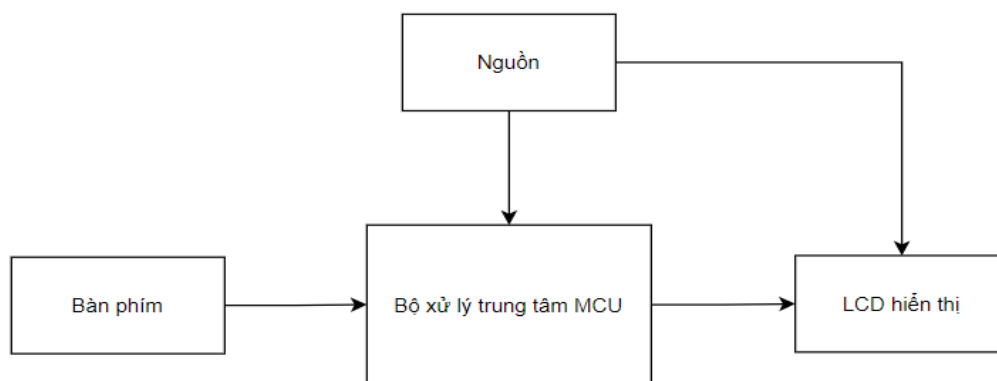
1.1.2 Cách thức hoạt động

- Ban đầu khi cấp nguồn cho hệ thống, ta cần bấm phím ‘ON/C’ để bật hệ thống.

- Các toán hạng và toán tử được nhập từ người dùng thông qua module bàn phím ma trận 4x4.
- Trong quá trình nhập phép tính, toàn bộ toán hạng và toán tử đã nhập đều được hiển thị ở dòng thứ nhất của LCD.
- Khi người dùng bấm phím '=' thì ngay lập tức hệ thống sẽ hiển thị kết quả của phép tính đã nhập ở dòng thứ hai của LCD.
- Nếu bấm phím 'ON/C' thì hệ thống sẽ xóa toàn bộ dữ liệu đã nhập trước đó và trở về lại trạng thái ban đầu.

1.2 Tổng quan các khối chức năng

- Hệ thống gồm các khối:
 - + Khối cấp nguồn 5V.
 - + Khối trung tâm xử lý – vi xử lý 8051.
 - + Khối bàn phím ma trận 4x4.
 - + Khối hiển thị trên LCD1602.



Hình 1 Sơ đồ khối của hệ thống

1.2.1 Khối xử lý trung tâm

- Nhiệm vụ của khối này là tiếp nhận dữ liệu đầu vào từ bàn phím, thực hiện tính toán 4 phép toán cơ bản: cộng, trừ, nhân, chia (lấy nguyên) và truyền dữ liệu và kết quả qua khối hiển thị LCD để hiển thị.
- Thành phần quan trọng nhất của khối này là vi xử lý 8051.

1.2.2 Khối dữ liệu ngõ vào

- Khối dữ liệu ngõ vào là một bàn phím gồm các số từ 0-9, '+', '-', '*', '%', '=' và 'ON/C'.
- Chức năng của khối này dùng để nhập dữ liệu và truyền đến cho khối xử lý trung tâm.

1.2.3 Khối hiển thị dữ liệu ngõ ra

- Khối dữ liệu ngõ ra gồm một màn hình LCD 16x2.
- Chức năng chính của khối này là nhận dữ liệu / lệnh từ khối xử lý trung tâm để hiển thị / thực hiện theo yêu cầu.

1.2.4 Khối cấp nguồn

- Khối cấp nguồn gồm 1 nguồn áp 5V có chức năng cấp nguồn cho hệ thống vi điều khiển và màn hình LCD.

1.3 Cấu tạo, nguyên tắc hoạt động của các khối thành phần

Chúng ta đi vào chi tiết từng khối để rõ hơn về cấu tạo, nguyên tắc hoạt động cũng như các sử dụng các khối của hệ thống.

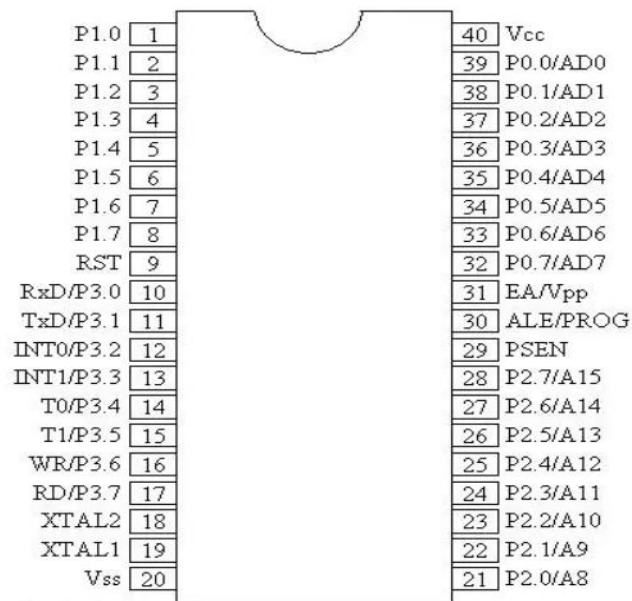
1.3.1 Vi điều khiển 8051

a) Giới thiệu vi điều khiển 8051

Vi điều khiển 8051 là vi điều khiển tổng quát trong họ MCS-51 của Intel, nó chứa những đặc trưng cho một vi điều khiển như:

- 4KB ROM
- 128 Byte RAM
- 4 cổng nhập xuất 8 bit
- 2 bộ định thời 16 bit
- Cổng giao tiếp nối tiếp (UART)
- Có thể giao tiếp với bộ nhớ chương trình ngoài 64KB
- Có thể giao tiếp với bộ nhớ dữ liệu ngoài 64KB
- Có thể xử lý theo bit, bộ nhớ có 210 bit được định địa chỉ theo bit
- Thực hiện phép nhân/chia trong 4 chu kỳ máy

b) Sơ đồ các chân chức năng của vi điều khiển 8051



Hình 2 Sơ đồ chân vi điều khiển 8051

Vi điều khiển 8051 có tất cả 40 chân – tương ứng với 4 cổng, mỗi cổng 10 chân. Các phép toán đọc/ghi dữ liệu được thực hiện thông qua 32 chân trên 4 cổng. 8 chân còn lại là các chân: Vcc, GND, XTAL1, XTAL2, RST, EA(bar), ALE/PROG(bar), PSEN(bar).

Cụ thể chức năng các chân như sau:

- Chân 1-8: Là các chân của Port 1, các chân này có điện trở kéo lên và chỉ có chức năng nhập xuất.
- Chân 9: Là chân RST (Reset) dùng để reset hệ thống về trạng thái ban đầu
- Chân 10-17 là các chân của Port 3, có điện trở kéo lên, ngoài chức năng nhập xuất, các chân còn có các chức năng riêng biệt:
 - + Chân 10 – RxD: Ngõ vào cổng nối tiếp
 - + Chân 11 – TxD: Ngõ ra cổng nối tiếp
 - + Chân 12 – INT0: Ngắt ngoài 0
 - + Chân 13 – INT1: Ngắt ngoài 1
 - + Chân 14 – T0: Ngõ vào bộ định thời 0
 - + Chân 15 – T1: Ngõ vào bộ định thời 1
 - + Chân 16 – WR: Tín hiệu điều khiển ghi dữ liệu lên bộ nhớ ngoài
 - + Chân 17 – RD: Tín hiệu điều khiển đọc dữ liệu từ bộ nhớ ngoài
- Chân 18-19: Thường được kết nối với bộ tạo tần số ngoài thạch anh
- Chân 20 và chân 40: Lần lượt là các chân nối đất và nối nguồn 5V
- Chân 21-28: Là các chân có 2 chức năng của Port 2 là nhập xuất và 8 bit địa chỉ cao hoặc dùng làm tín hiệu điều khiển trong quá trình nạp và kiểm tra chip
- Chân 29: Là chân PSEN cho phép đọc bộ nhớ chương trình mở rộng đối với các ứng dụng sử dụng ROM ngoài.
- Chân 30: Là chân ALE cho phép xuất tín hiệu chốt địa chỉ để giải đa hợp bus dữ liệu và bus địa chỉ ở Port 0 trong quá trình giao tiếp với bộ nhớ ngoài.

- Chân 31: Là chân EA cho phép thực thi chương trình từ ROM ngoài.
- Chân 32-39: Là các chân của Port 0, không có điện trở kéo lên. Có chức năng nhập xuất hoặc dùng bus dữ liệu và bus địa chỉ đa hợp

c) Ứng dụng của vi điều khiển 8051

Vi điều khiển 8051 được ứng dụng rất rộng rãi:

- Điều khiển các thiết bị như đèn LED, motor, cảm biến...
- Thiết kế các hệ thống đo lường và điều khiển.
- Ứng dụng trong các thiết bị điện tử tiêu dùng như đồng hồ điện tử, điều khiển từ xa, máy tính cá nhân.

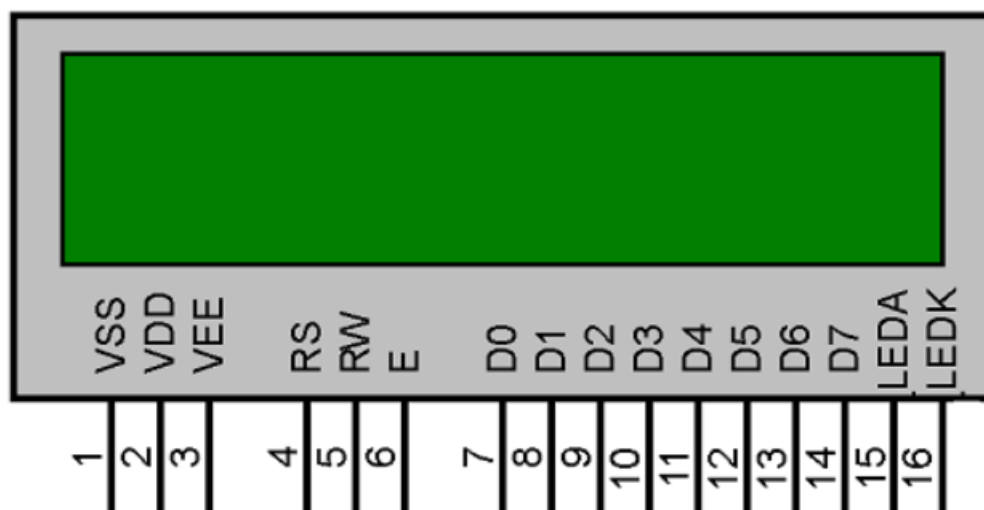
1.3.2 Module LCD1602

a) Giới thiệu về công nghệ màn LCD

Màn hình LCD chứa một ma trận các điểm ảnh hiển thị hình ảnh lên màn hình. Màn hình LCD thời kì đầu có màn hình ma trận thụ động, điều khiển các pixel riêng lẻ bằng cách gửi điện tích đến hàng và cột của chúng. Vì một số lượng điện tích hạn chế có thể được gửi đi mỗi giây, nên màn hình ma trận thụ động được biết đến là xuất hiện vết mờ khi hình ảnh di chuyển nhanh trên màn hình. Màn hình LCD hiện đại thường sử dụng công nghệ ma trận chủ động, chứa bóng bán dẫn màng mỏng là TFT. Các bóng bán dẫn này bao gồm các tụ điện cho phép các pixel riêng lẻ “tích cực” giữ lại điện tích của chúng. Do đó, màn hình LCD ma trận chủ động hiệu quả hơn và có vẻ nhạy hơn so với màn hình ma trận thụ động.

b) Tổng quan về module LCD1602

LCD 16x2 là loại màn hình được sử dụng nhiều nhất trong các thiết bị điện tử và các hệ thống nhúng. 16x2 LCD có 2 hàng, mỗi hàng hiển thị được 16 ký tự. Mỗi ký tự được biểu diễn bằng một ma trận 5x7 điểm ảnh.



Hình 3 Sơ đồ chân LCD 16x2

Các chân của LCD 16x2 được chia thành 5 loại, chi tiết thể hiện dưới bản sau:

Loại	Số chân	Tên chân	Chức năng
Chân nguồn	1	VSS	Chân nối đất
	2	VDD / VCC	Chân nối nguồn 5V
Chân cài đặt tương phản	3	VEE	Có thể kết nối với nguồn và biến trở để điều khiển độ tương phản
	4	RS	Chế độ Command: RS = 0 Chế độ Data: RS = 1

Chân điều khiển	5	RW	Chế độ Write: RW = 0 Chế độ Read: RW = 1
	6	E	Chân Enable, kích từ 1 xuống 0 để kích hoạt LCD
Chân dữ liệu	7-14	D0-D7	Truyền dữ liệu hoặc lệnh tới LCD
Chân điều khiển đèn nền	15	LED A / A	Nguồn dương cho đèn nền
	16	LED K / K	Nguồn âm cho đèn nền

Bảng 1 Bảng chức năng các chân của LCD 16x2

Để giao tiếp với LCD 16x2 ở chế độ thực thi lệnh (command), ta sử dụng mã hex quy định sẵn như bảng dưới đây:

Mã lệnh	Chức năng
0F	Bật LCD, bật con trỏ
01	Xóa toàn bộ màn hình
04	Di chuyển con trỏ sang trái
06	Di chuyển con trỏ sang phải

0E	Bật màn, nhấp nháy con trỏ
38	Sử dụng 2 hàng và ma trận 5x7
0C	Bật màn, tắt con trỏ
C0	Đưa con trỏ xuống vị trí đầu dòng 2

Bảng 2 Các mã lệnh của LCD 16x2

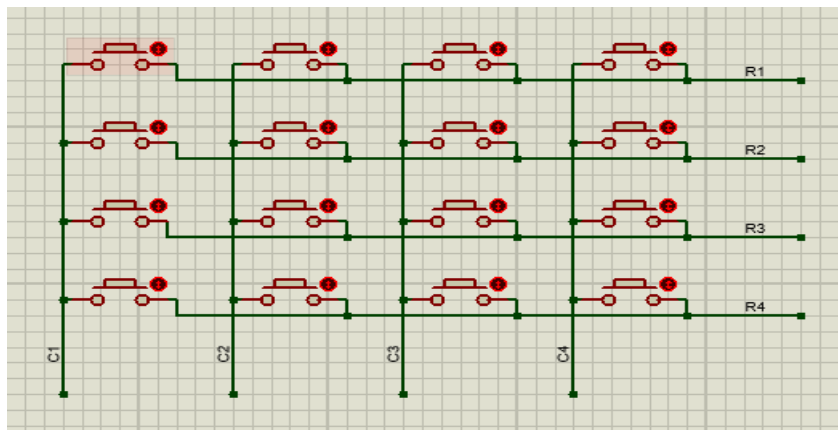
c) Ứng dụng

LCD 16x2 là khối hiển thị dữ liệu đầu ra không thể thiếu trong hầu hết các hệ thống nhúng.

1.3.3 Module bàn phím ma trận 4x4

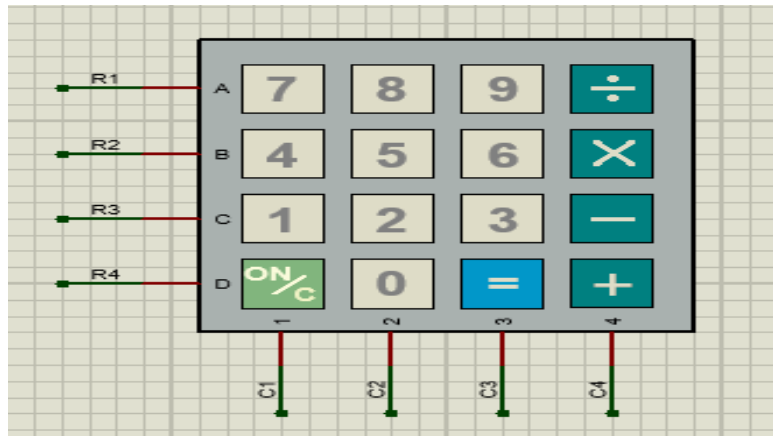
a) Tổng quan module bàn phím ma trận 4x4

Bàn phím ma trận 4x4 là tổ hợp của 16 nút nhấn đơn, được kết nối với nhau theo các hàng và các cột. Chi tiết cấu tạo và sơ đồ nguyên lý bàn phím 4x4 như hình bên dưới:



Hình 4 Sơ đồ schematic của bàn phím ma trận 4x4

Với việc các phím hoạt động độc lập, nên ta có thể quy ước từng ký tự ứng với từng nút bấm tùy vào mục đích sử dụng. Trong phạm vi đồ án này, ta quy ước các phím bấm như sau:



Hình 5 Hình ảnh bàn phím ma trận 4x4 quy ước phím bấm

b) Ứng dụng

Bàn phím ma trận 4x4 được sử dụng rộng rãi trong các mạch điện tử, có chức năng chính là nhập dữ liệu.

- Chân RW được nối vào chân P3.3 của vi điều khiển 8051
- Chân E được nối vào chân P3.2 của vi điều khiển 8051
- Các chân D0-D7 tương nối với các chân P2.0-P2.7 của vi điều khiển 8051

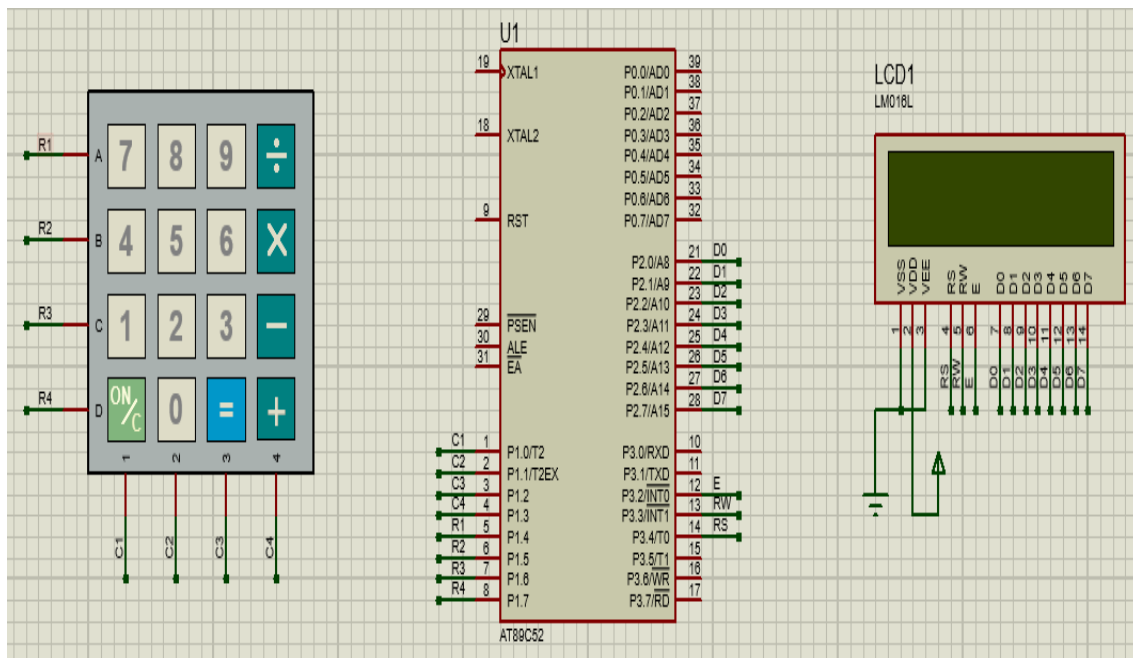
2.1.3 Module bàn phím ma trận 4x4

Bàn phím ma trận 4x4 gồm 8 chân ứng với 4 hàng và 4 cột. Các chân của bàn phím được kết nối với vi điều khiển 8051 như sau:

- 4 cột C1-C4 tương ứng được nối với P1.0-P1.3 của vi điều khiển 8051
- 4 hàng R1-R4 tương ứng được nối với P1.4-P1.7 của vi điều khiển 8051

2.2 Thi công mạch

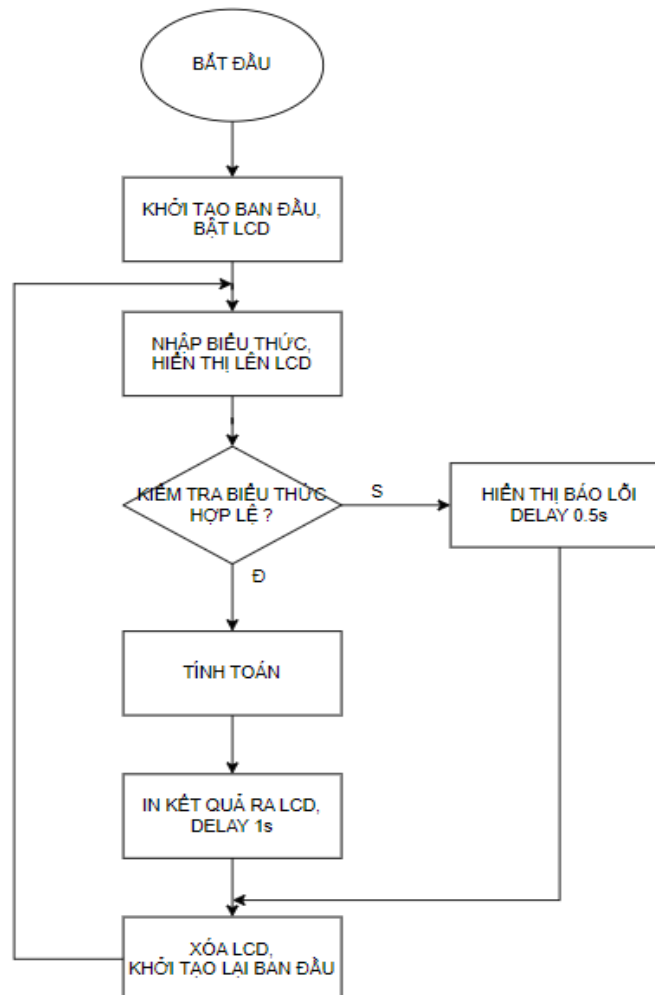
Hệ thống được thi công theo phương pháp mô phỏng trên phần mềm mô phỏng Proteus.



Hình 7 Hình ảnh thi công mạch trên phần mềm mô phỏng Proteus

CHƯƠNG 3: LƯU ĐỒ GIẢI THUẬT VÀ HIỆN THỰC BẰNG NGÔN NGỮ C

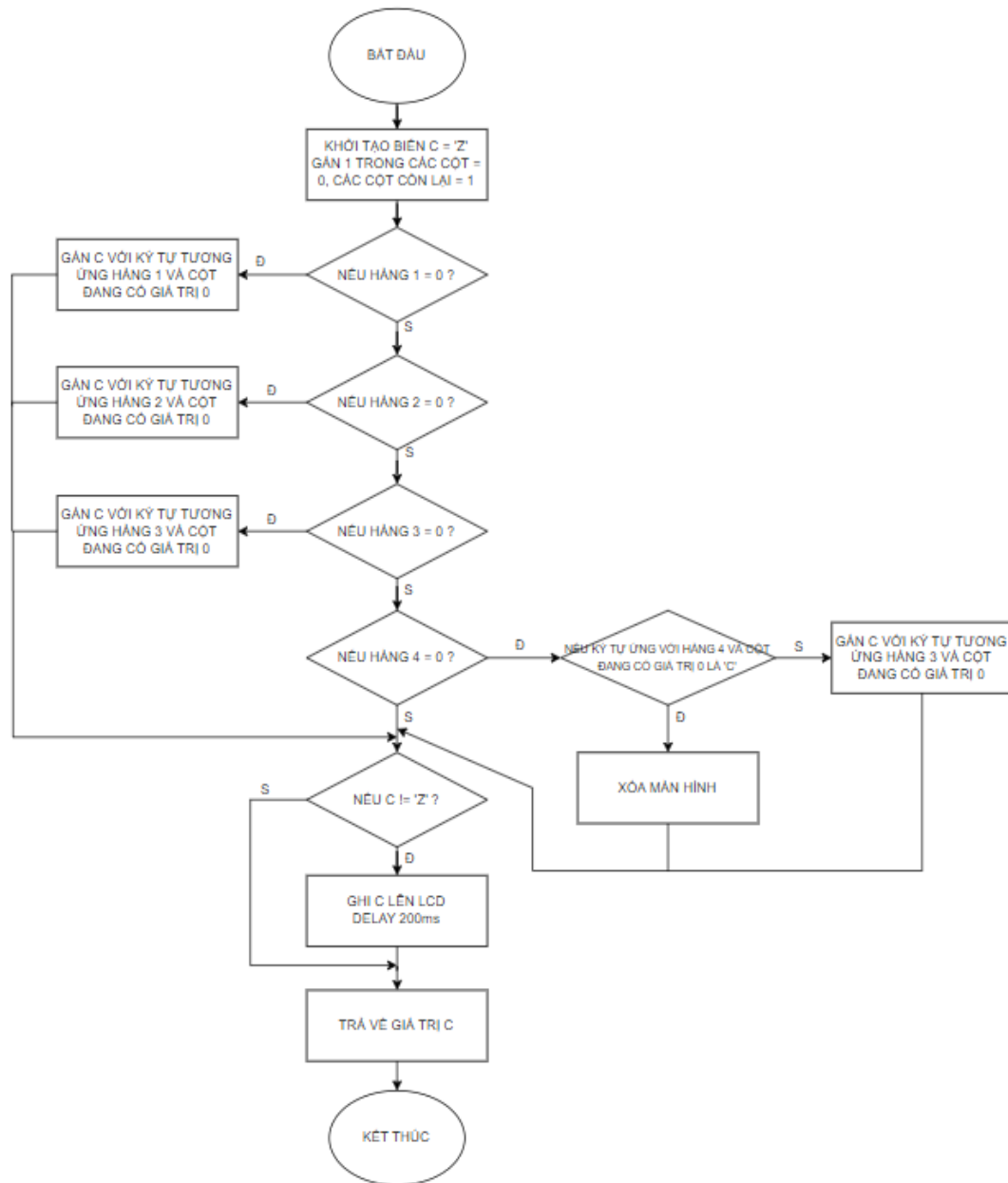
3.1 Lưu đồ thuật toán chương trình chính



Hình 8 Lưu đồ thuật toán chương trình chính

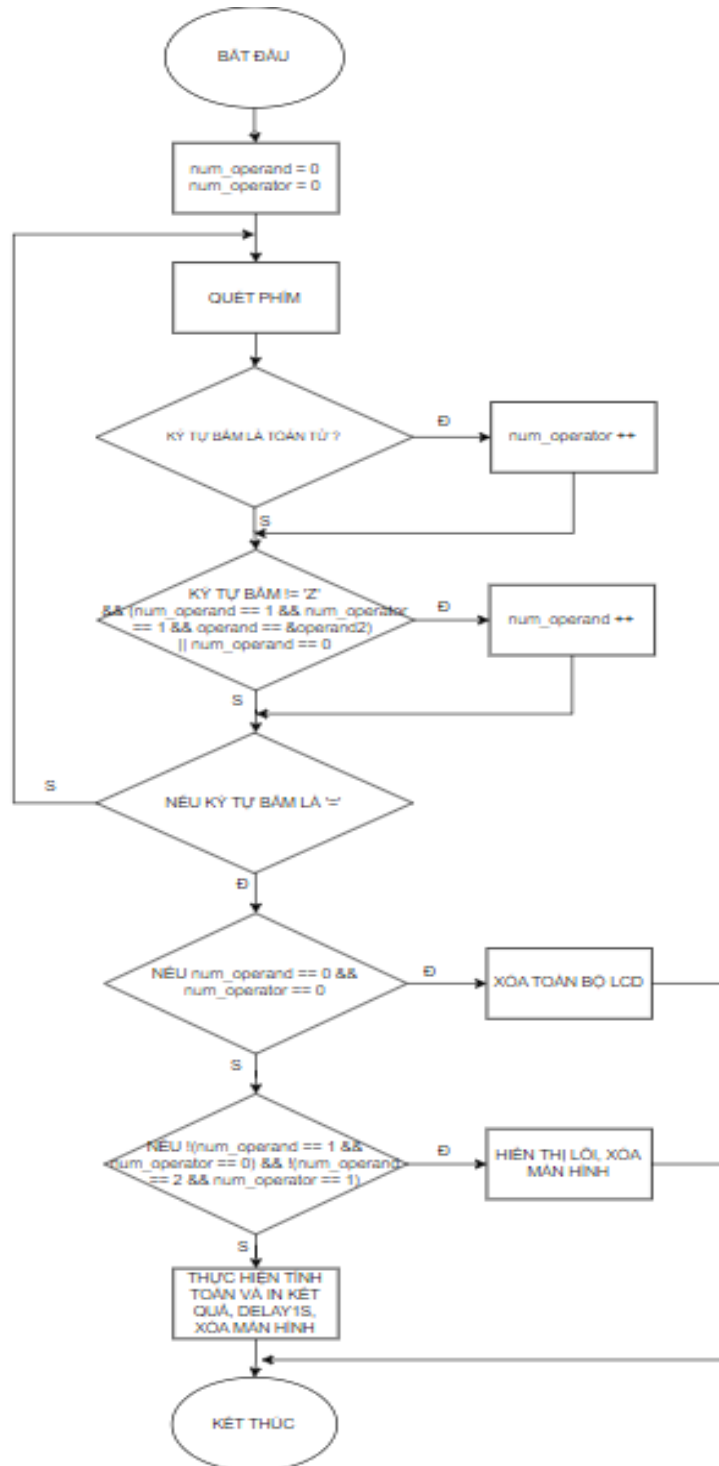
3.2 Lưu đồ thuật toán các chương trình con

3.2.1 Lưu đồ thuật toán quét phím và xử lý phím



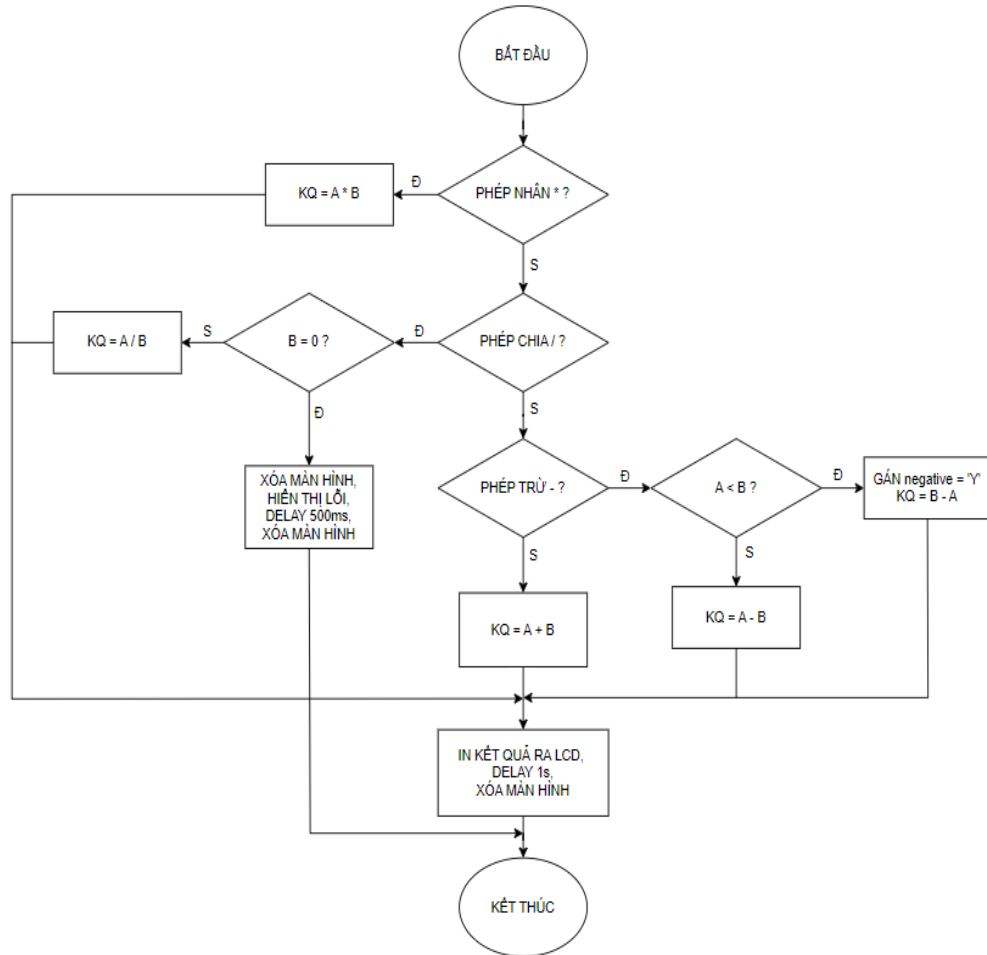
Hình 9 Lưu đồ thuật toán quét phím

3.2.3 Lưu đồ giải thuật kiểm tra biểu thức hợp lệ



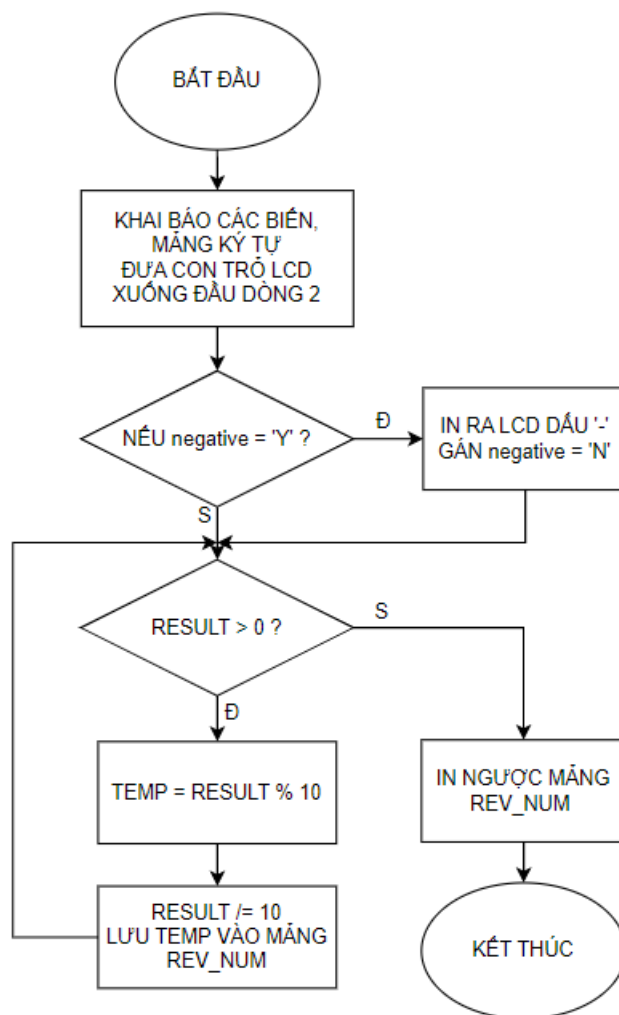
Hình 10 Lưu đồ thuật toán kiểm tra biểu thức hợp lệ

3.2.2 Lưu đồ giải thuật hàm tính toán



Hình 11 Lưu đồ thuật toán tính toán

3.2.3 Lưu đồ giải thuật hàm ghi kết quả



Hình 12 Lưu đồ thuật toán hàm ghi kết quả

3.3 Hiện thực giải thuật bằng ngôn ngữ C

```
#include <stdio.h>
#include <regx52.h>

void init();
void command(unsigned int);
void write_data(unsigned char);
void ms_delay(unsigned int);
char get_key(int, int, int, int, char, char, char, char);
void num_generator(char, unsigned int*);
void write_result(unsigned int, char*);

sbit EN = P3^2;      // Enable LCD to display data: HIGH -> LOW
sbit RW = P3^3;      // WRITE = 0; READ = 1
sbit RS = P3^4;      // Register Select: RS = 0 -> Command mode, RS = 1 -> Data mode

unsigned int *operand;
unsigned int operand1 = 0;
unsigned int operand2 = 0;
unsigned int result = 0;
int num_operand, num_operator;

void main(){

    int i, j;
    int en_arr[4][4] = {{0,1,1,1},
                        {1,0,1,1},
                        {1,1,0,1},
                        {1,1,1,0}};

    char c_arr[4][4] = {{'7','4','1','C'},
                        {'8','5','2','0'},
                        {'9','6','3','='},
                        {'/','*','-','+'}};

    char error[10] = {'M','A','T','H',' ','E','R','R','O','R'};
    char pressed_key, operator1, negative = 'N';

    P1 = 0xFF;
    operand = &operand1;
```



```

init();
num_operand = 0;
num_operator = 0;
while(1){
    for(i = 0; i < 4; i++){

        pressed_key = get_key(en_arr[i][0], en_arr[i][1], en_arr[i][2], en_arr[i][3], c_arr[i][0],
c_arr[i][1], c_arr[i][2], c_arr[i][3]);

        if(pressed_key=='/' || pressed_key=='*' || pressed_key=='-' || pressed_key=='+'){
            if(num_operand == 1) operand = &operand2;
            operator1 = pressed_key;
            num_operator++;
        }

        if((pressed_key != 'Z') && (pressed_key != '=') && (pressed_key != '/') && (pressed_key !=
'*') && (pressed_key != '-') && (pressed_key != '+')){
            num_generator(pressed_key, operand);
            if((num_operand == 1 && num_operator == 1 && operand == &operand2) ||
(num_operand == 0))
                num_operand++;
        }

        if(pressed_key == '='){

            if(num_operand == 0 && num_operator == 0){
                init();
            }

            else
                if(!(num_operand == 1 && num_operator == 0) && !(num_operand == 2 &&
num_operator == 1)){
                    init();
                    for(j = 0; j < 10; j++)
                        write_data(error[j]);
                    ms_delay(500);
                    init();
                }

            init();
            num_operand = 0;
            num_operator = 0;
            while(1){

```

```

else{
    if(operator1 == '*')
        result = operand1 * operand2;

    else
    if(operator1 == '/'){
        if(operand2 == 0){
            init();
            for(j = 0; j < 10; j++)
                write_data(error[j]);
            ms_delay(500);
            init();
            continue;

        } else
            result = operand1 / operand2;
    }

    else
    if(operator1 == '-'){
        if(operand1 < operand2){
            negative = 'Y';
            result = operand2 - operand1;
        } else
            result = operand1 - operand2;
        }
    else
        result = operand1 + operand2;

    write_result(result, &negative);

    ms_delay(1000);
    init();
}

}

}

}

else{
    if(operator1 == '*')

```

```

void init(){

    operand = &operand1;
    operand1 = 0;
    operand2 = 0;
    result = 0;
    num_operand = 0;
    num_operator = 0;

    command(0x0F);

    command(0x38);

    command(0x01);

    ms_delay(50);
}

void command(unsigned int comm){

    RW = 0;
    RS = 0;
    P2 = comm;
    EN = 1;
    ms_delay(50);
    EN = 0;
}

void write_data(unsigned char c){

    RW = 0;
    RS = 1;
    P2 = c;
    EN = 1;
    ms_delay(50);
    EN = 0;
}

void init(){

    operand = &operand1;

```

```

void ms_delay(unsigned int time){
    TMOD = 0x10;
    TL1 = 0x18;
    TH1 = 0xFC;
    TR1 = 1;
    while(time--){

        while(TF1 == 0);
        TF1 = 0;
        TL1 = 0x18;
        TH1 = 0xFC;

    }
    TR1 = 0;
}

char get_key(int c1, int c2, int c3, int c4, char data1, char data2, char data3, char data4){
    char c = 'Z';
    P1_0 = c1;
    P1_1 = c2;
    P1_2 = c3;
    P1_3 = c4;

    if(P1_4 == 0)
        c = data1;
    else
        if(P1_5 == 0)
            c = data2;
        else
            if(P1_6 == 0)
                c = data3;
            else
                if(P1_7 == 0)
                    if(data4 == 'C')
                        init();
                    else
                        c = data4;
                if(c != 'Z'){
                    write_data(c);
                    ms_delay(200);
                }
    return c;
}

```

```

void num_generator(char c, unsigned int* operand){

    int digit = 0;
    digit = c - '0';
    *operand = digit + (*operand * 10);

}

```

```

void write_result(unsigned int result, char *neg){

    int i = 0, j = 0, tmp = 0;
    char rev_num[20];

    command(0xC0);

    if(*neg == 'Y'){
        write_data('-');
        *neg = 'N';
    }

    do{
        tmp = result % 10;
        result /= 10;
        rev_num[i++] = tmp + '0';
    } while(result > 0);

    for(j = i-1; j >= 0; j--)
        write_data(rev_num[j]);

}

```

```

void num_generator(char c, unsigned int* operand){

    int digit = 0;
    digit = c - '0';
    *operand = digit + (*operand * 10);

}

```

```

void write_result(unsigned int result, char *neg){

    int i = 0, j = 0, tmp = 0;
    char rev_num[20];

```

CHƯƠNG 4 THỰC NGHIỆM VÀ KIỂM TRA

4.1 Môi trường thực nghiệm

Hệ thống được thực nghiệm trên phần mềm mô phỏng proteus.

Các testcase được đưa vào thực nghiệm:

8x5	7-2	4/2	1/0	0x0	0/0	1+2	25x10	45-7	45/9
70x2	-69	x23	2x-45	/15	/0	103-9	100x1	100/0	1+2+3
1+10-	-10+5	112-2	15+5	45/5	45/0	2+3+	2x3+	2-3++	7/5
10-50	40-60	90/10	20/50	20+90	68x3	75/25	75/3	75x2	80-10
0-190	0x99	0/178	123x3	45-54	45/54	10/0	x0	-0	75-19
19/19	54/27	3x150	50x50	99-99	15/0	89-20	49x7	81/9	21x3
113x3	101-7	79x5	65x15	50-60	10-10	1-100	1-10/0	23-32	45x54
15+50	25x45	8x4	32/8	91/10	35-50	100+4	110x3	101-0	45/0
34x5	69-96	75+45	83x45	23x32	45/15	1999/0	25/3	15x2	101-4
199/0	4x0-0	5x6x7	12/3/2	1+3+7	9-7-5	0/0/0	0x0x0	2/2/2	10/10

Bảng 3 Bảng các testcase đưa vào thực nghiệm

4.2 Kết quả thực nghiệm

Kết quả thực nghiệm thu được như sau:

40	5	2	Error	0	Error	3	250	38	5
140	Error	Error	Error	Error	Error	94	100	Error	Error
Error	Error	110	20	9	Error	Error	Error	Error	1
-40	-20	9	0	110	204	3	25	150	70
-190	0	Error	369	-9	0	Error	Error	Error	56
1	2	450	2500	0	Error	69	343	9	63
339	94	395	975	-10	0	-99	Error	-9	2430
65	1125	32	4	9	-15	104	330	101	Error
170	-27	120	3735	736	3	Error	8	30	97
Error	Error	Error	Error	Error	Error	Error	Error	Error	1

Bảng 4 Bảng kết quả thực nghiệm

Qua kết quả chạy thực nghiệm ta thấy:

- Độ chính xác = $\frac{\text{Số testcase cho kết quả đúng kì vọng}}{\text{Tổng số testcase}} \cdot 100 = \frac{100}{100} \cdot 100 = 100\%$.
- Hệ thống cho ta kết quả đúng như chức năng, yêu cầu ban đầu đặt ra.
- Tốc độ phản hồi nhanh, tốc độ tính toán cực nhanh.
- Không xuất hiện hiện tượng đơ, lỗi trong quá trình test.

CHƯƠNG 5: ĐÁNH GIÁ

5.1. Ưu điểm

- Thời gian đáp ứng nhanh, không có hiện tượng lỗi phần cứng.
- Kết quả cho ra với sự chính xác 100%.
- Tính toán được phép tính có 2 toán hạng với độ lớn lên tới $2^{16} - 1$.
- Thao tác với hệ thống đơn giản, dễ sử dụng.

5.2. Một số nhược điểm

- Chỉ có thể thực hiện phép toán có 2 toán hạng, không thực hiện được các phép toán phức tạp có nhiều hơn 1 toán tử và 2 toán hạng.
- Kết quả của phép chia không thể lấy số thập phân.

5.3. Hướng phát triển

Để phục vụ nhu cầu tự nghiên cứu, nhóm chúng em dự định sẽ tối ưu giải thuật để thực hiện tính toán phép tính kết quả ở dạng thập phân và hỗ trợ tính toán phép tính có nhiều toán hạng, toán tử. Để làm được điều này, nhóm dự kiến sử dụng cấu trúc dữ liệu Stack, ký pháp Balan (tiền tố, trung tố, hậu tố) để thực hiện giải thuật tính toán và sử dụng thêm bộ nhớ ngoài để có không gian lưu trữ chương trình tốt hơn.

TÀI LIỆU THAM KHẢO

- [1] Tống Văn On – Hoàng Đức Hải, **HỌ VI ĐIỀU KHIỂN 8051**, Nhà xuất bản Lao Động – Xã Hội
- [2] Vũ Đức Lung – Lê Quang Minh – Phan Đình Duy, **VI ĐIỀU KHIỂN**, Nhà xuất bản Đại Học Quốc Gia TP. Hồ Chí Minh
- [3] <https://circuitdigest.com/microcontroller-projects/lcd-interfacing-with-8051-microcontroller-89s52>
- [4] <https://dientutuonglai.com/giao-tiep-giua-ban-phim-16-phim-voi-vi-dieu-khien-8051.html>