

BÁO CÁO PROJECT 8

NHÓM THỰC HIỆN:

NGUYỄN HỮU KHANG – 18125086

NGUYỄN HOÀNG LONG – 18125096

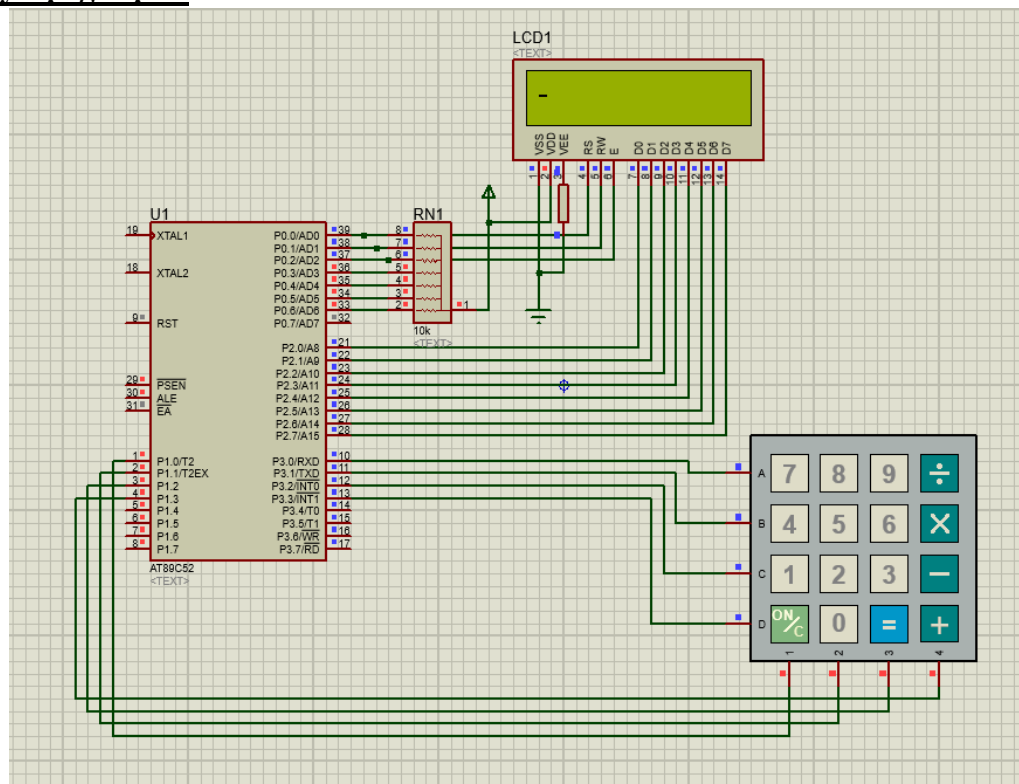
LÊ HUỖNH MINH TRÍ – 18125144

ĐỀ BÀI: Xây dựng một máy tính bỏ túi với các chức năng tính toán tối thiểu gồm các phép toán cộng, trừ, nhân, chia. Giá trị số nhập vào bằng bàn phím ma trận, kết quả hiển thị ra màn hình LCD.

Ý tưởng:

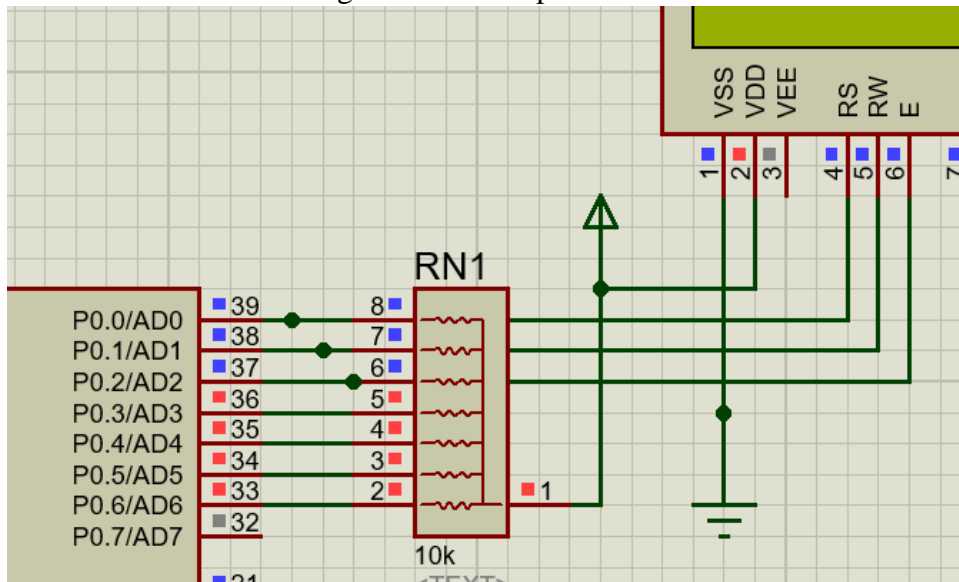
- Tính toán với số dương có một chữ số.
- Sử dụng quét phím để nhận đầu vào chuyển đổi kết quả từ số thập phân sang ký tự để hiển thị LCD
- Khi người dùng nhấn phép tính sau đó nhấn dấu bằng thì hiện ra kết quả.
- Nếu trong quá trình input, người dùng nhập input sai format thì sẽ báo lỗi và quay lại ban đầu.

1. Xây dựng mạch:



Port 2.0->2.7 nối với Cổng D0->D7 của LCD
Port 3.0->3.3 nối với Cổng A->D của bàn phím

Port 1.0->1.3 nối với Cổng 1->4 của bàn phím



- Cổng 37-39 nối với RS, RW, E của LCD
- RN1 là điện trở thanh, dùng để set mức logic của các cổng logic lên 1. Khi bật LCD thì cổng từ 33-36 trên chip đồ lên là nhờ điện trở thanh RN1

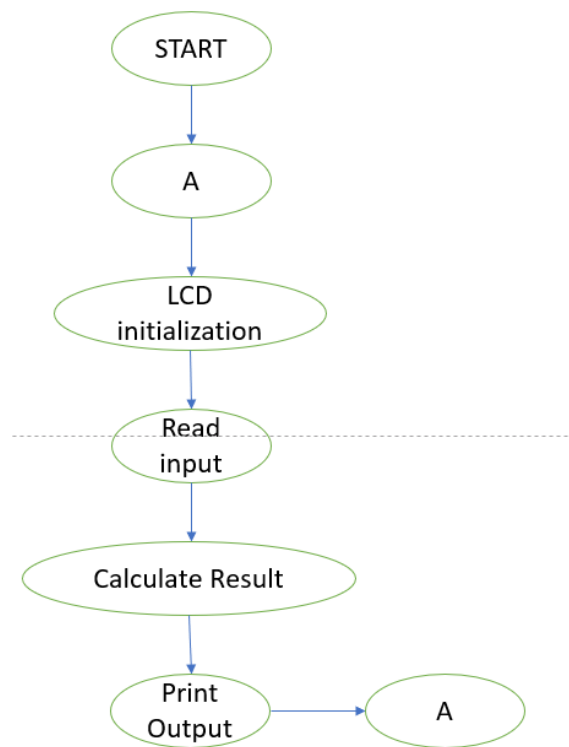
2. Cấu trúc chương trình:

2.1 Hàm main:

Hàm khởi tạo bộ nhớ cho chương trình

```
memoryInit:
    CLR     A                ;#!Clear ACC
    MOV     N1,A             ;#!Clear first operand location in ram
    MOV     N2,A             ;#!Clear second operand location in ram
    MOV     OP,A             ;#!Clear operation location in ram
    MOV     R,A              ;#!Clear result location in ram
    MOV     SIGN,A           ;#!Clear sign location in ram
    MOV     TEMP,A           ;#!Clear result location in ram
    CLR     C                ;#!Clear carry bit
    MOV     DIF,C            ;#!Clear digit flag bit
    MOV     OIF,C            ;#!Clear operator flag bit
    MOV     AIF,C            ;#!Clear assignment operator flag bit
    RET                     ;#!return to MAIN routine
```

Luồng chạy của chương trình:



```
MAIN:  ORG      30H
        CALL    memoryInit
        CALL    lcdInit
        CALL    readInput
        CALL    calculateResult
        CALL    printOutput
        AJMP    MAIN
```

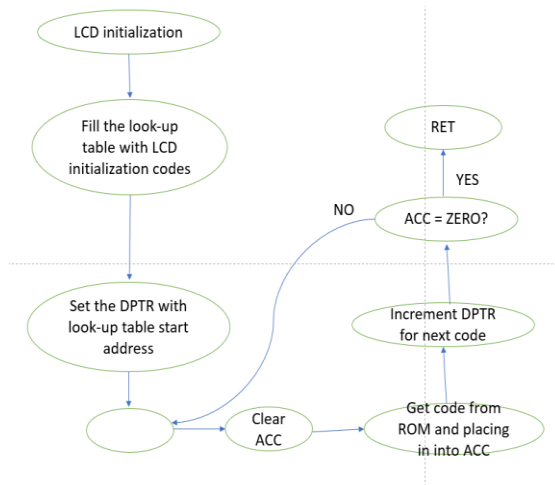
2.2 Hàm khởi tạo màn hình LCD:

Ý nghĩa các lệnh trong hàm lcdInit

```
MOV     DPTR,#MYCOM
CLR     A
MOVC    A,@A+DPTR
ACALL   COMMWRIT
INC     DPTR
JZ      C2
SJMP    C1
RET
```

```
;!Set DPTR to look-up table's address
;!Clear ACC
;!Get code from look-up table and placing it into AC
;!CALL LCD command write subroutine
;!Increment DPTR <DPTR = DPTR + 1>
;!Jump if ACC = ZERO to end of loop
;!If not start the loop again
;!return to MAIN routine
```

Luồng chạy của hàm lcdInit:



```

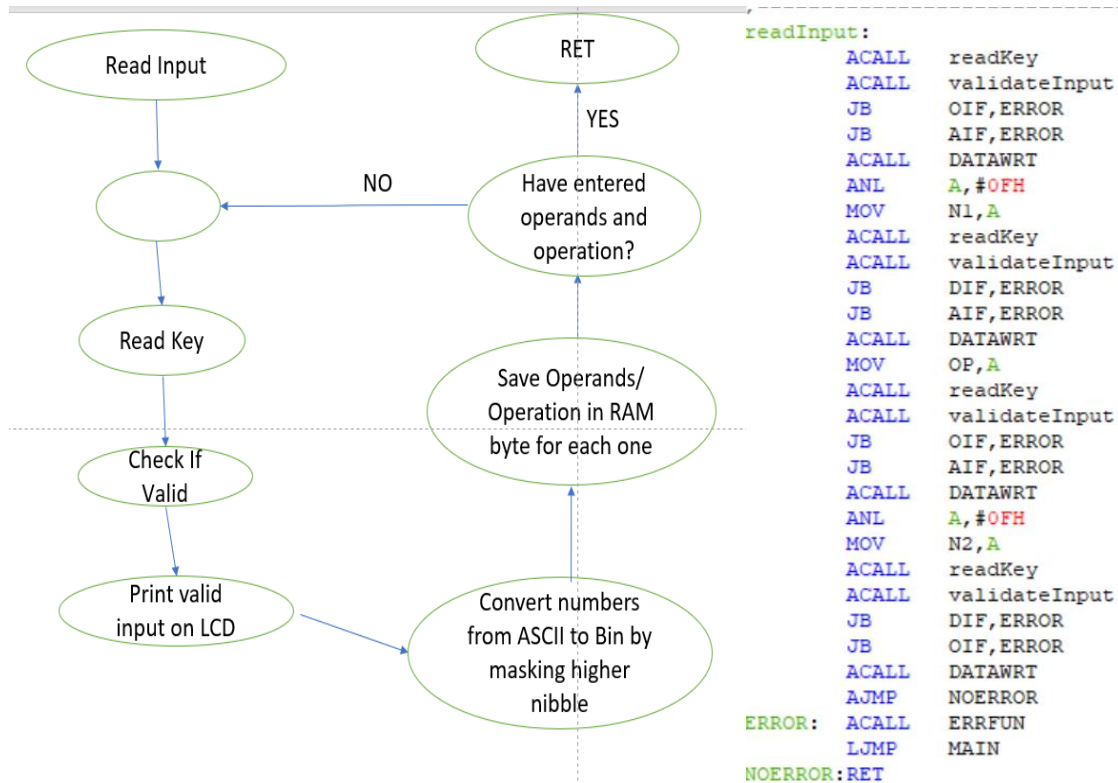
lcdInit:
    MOV     DPTR, #MYCOM
C1:      CLR     A
         MOVC    A, @A+DPTR
         ACALL   COMMWRT
         INC     DPTR
         JZ      C2
         SJMP    C1
C2:      RET
  
```

2.3 Hàm đọc input từ người dùng

Ý nghĩa các lệnh trong hàm readInput:

<pre> ACALL readKey ACALL validateInput JB OIF,ERROR JB AIF,ERROR ACALL DATAWRT ANL A,#0FH MOV N1,A ACALL readKey ACALL validateInput JB DIF,ERROR JB AIF,ERROR ACALL DATAWRT MOV OP,A ACALL readKey ACALL validateInput JB OIF,ERROR JB AIF,ERROR ACALL DATAWRT ANL A,#0FH MOV N2,A ACALL readKey ACALL validateInput JB DIF,ERROR JB OIF,ERROR ACALL DATAWRT AJMP NOERROR ERROR: ACALL ERRFUN LJMP MAIN NOERROR:RET </pre>	<pre> ;#!CALL keypad subroutine to read one number ;#!CALL subroutine to confirm of valid number ;#!Jump to ERROR routine if OIF = 1 ;#!Jump to ERROR routine if AIF = 1 ;#!CALL subroutine so pass data to LCD ;#!Masking to convert number form ASCII to BCD ;#!Save number at RAM ;#!CALL keypad subroutine to read operation(+,-,*,/) ;#!CALL subroutine to confirm of valid operation ;#!Jump to ERROR routine if DIF = 1 ;#!Jump to ERROR routine if AIF = 1 ;#!CALL subroutine so pass data to LCD ;#!Save operation at RAM ;#!CALL keypad subroutine to read one number ;#!CALL subroutine to confirm of valid number ;#!Jump to ERROR routine if OIF = 1 ;#!Jump to ERROR routine if AIF = 1 ;#!CALL subroutine so pass data to LCD ;#!Masking to convert number form ASCII to BCD ;#!Save number at RAM ;#!CALL keypad subroutine to read assignemt operation(=) ;#!CALL subroutine to confirm of valid input ;#!Jump to ERROR routine if OIF = 1 ;#!Jump to ERROR routine if OIF = 1 ;#!CALL subroutine so pass data to LCD ;#!Jump to END of subroutine if NO ERROR ;#!CALL ERROR subroutine if input was not valid ;#!return if ERROR to start of program again ;#!return if NO ERROR normally to MAIN </pre>
---	--

Luồng chạy của hàm readInput



2.4 Hàm tính toán kết quả:

Ý nghĩa các lệnh trong hàm calculateResult

```

MOV     A,N1                ;!Copy first number to ACC
MOV     B,N2                ;!Copy second number to B
MOV     R7,OP               ;!Copy operation to R7
CJNE    R7,#"+",NEXT11      ;!Check firstly if the operation was "+", if not jump
ADD     A,B                 ;!If so, Add the two operands
MOV     R,A                 ;!Save result at RAM
MOV     SIGN,#"+"           ;!Set result sign to positive
RET                                ;!return to MAIN

NEXT11: CJNE    R7,#"-",NEXT22 ;!Check secondly if the operation was "-", if not jump
SUBB    A,B                 ;!If so, Sub the two operands
JC      NIGATIV             ;!Check on carry for negative result
MOV     R,A                 ;!Save result at RAM
MOV     SIGN,#"+"           ;!Set result sign to positive
RET                                ;!return to MAIN

NIGATIV: CPL     A           ;!Convert to 1's complement if negativ
INC     A                 ;!Increment to get 2's complement
MOV     R,A                 ;!Save result at RAM
MOV     SIGN,#"- "         ;!Set result sign to negative
RET                                ;!return to MAIN

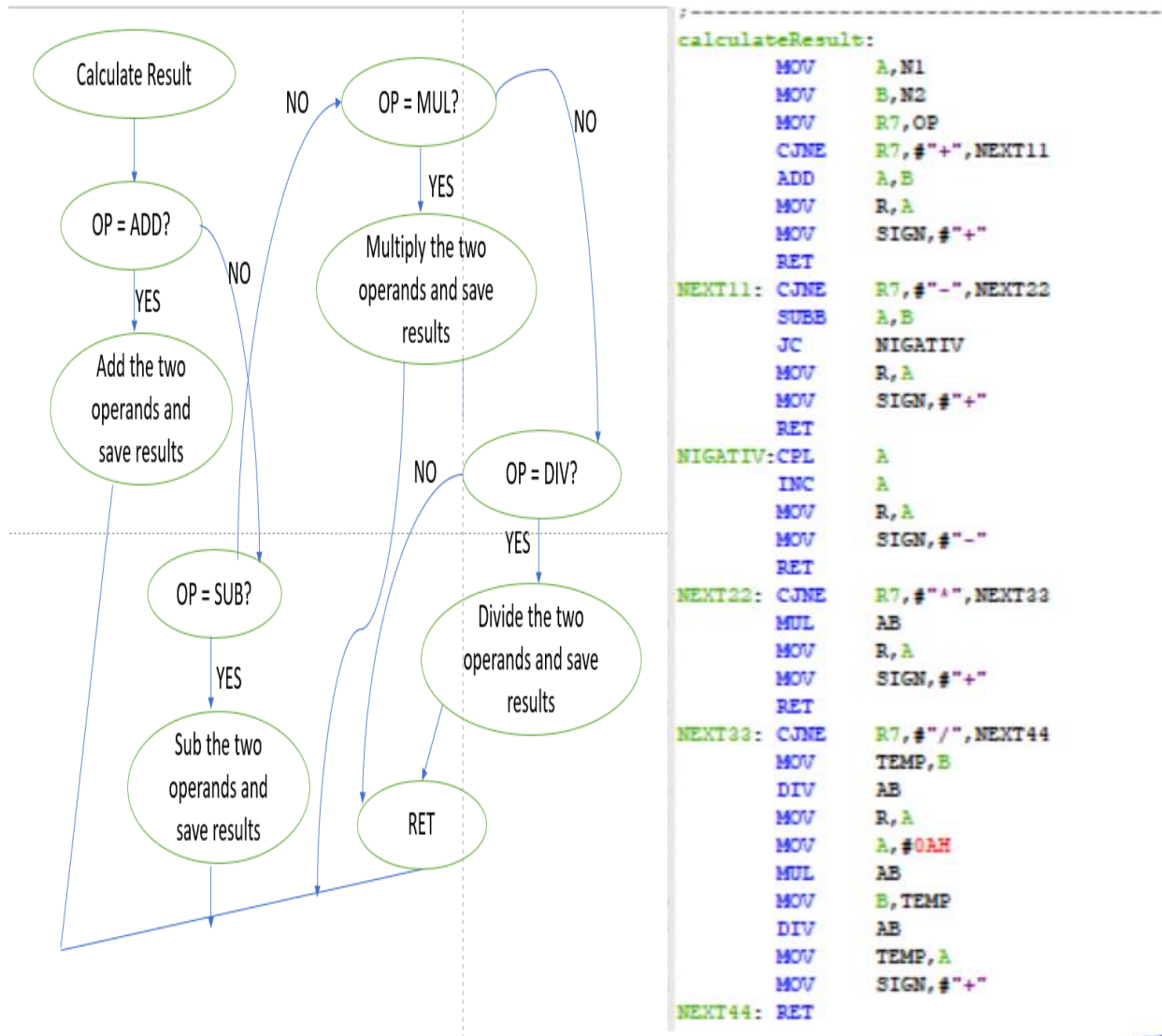
NEXT22: CJNE    R7,#"^",NEXT33 ;!Check thirdly if the operation was "^", if not jump
MUL     AB                 ;!If so, Multibly the oeprands
MOV     R,A                 ;!Save result in RAM
MOV     SIGN,#"+"           ;!Set result sign to positive
RET                                ;!return to MAIN

NEXT33: CJNE    R7,#"/",NEXT44 ;!Check finally if the operation was "/", if not jump
MOV     TEMP,B             ;!Copy to TEMP location
DIV     AB                 ;!Carry out division to the two operands
MOV     R,A                 ;!Save the quotient in RAM
MOV     A,#0AH             ;!Fill ACC with 10
MUL     AB                 ;!Multibly reminder by 10
MOV     B,TEMP             ;!Get the second number again
DIV     AB                 ;!Divide again
MOV     TEMP,A             ;!Save the result into RAM
MOV     SIGN,#"+"           ;!Set result sign to positive
NEXT44: RET                ;!return to MAIN

```

..

Luồng chạy của hàm calculateResult:

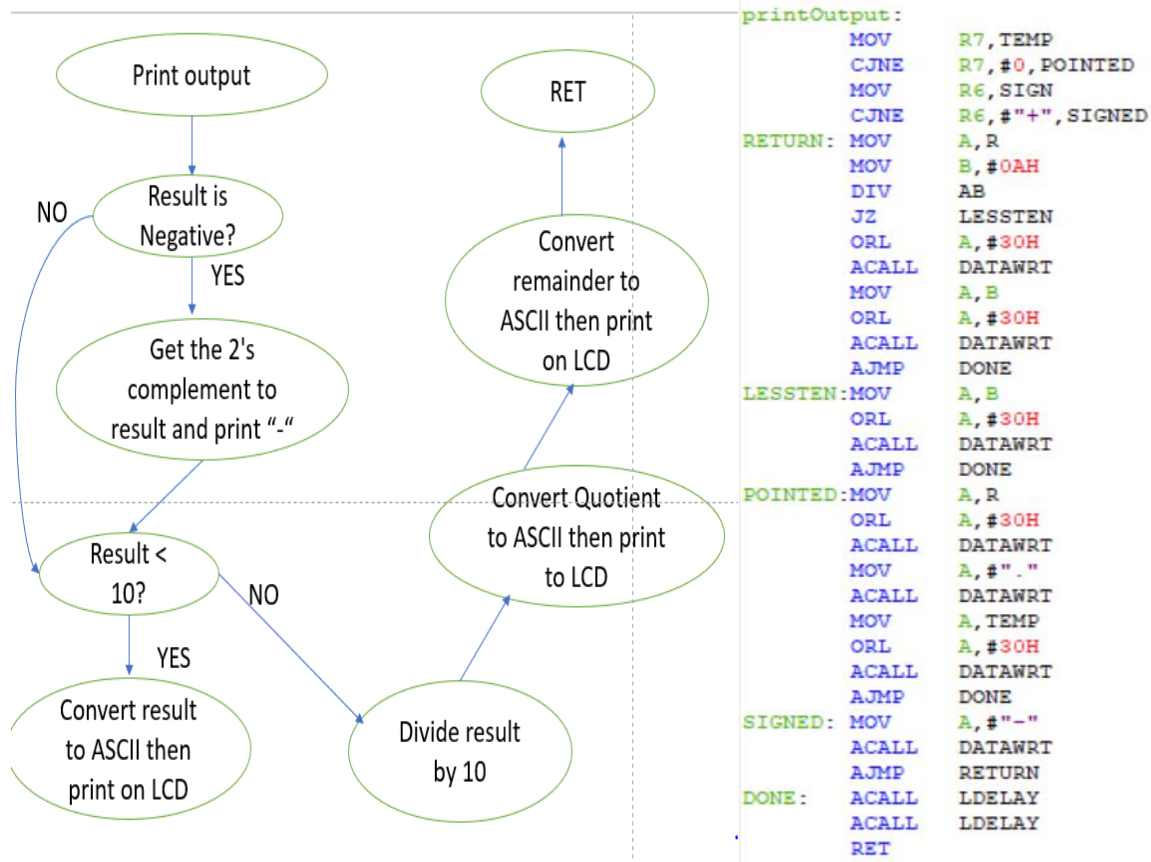


2.5 Hàm in kết quả ra màn hình LCD:

Ý nghĩa các lệnh trong hàm printOutput

```
MOV     R7,TEMP                ;#!Copy TEMP contents to R7
CJNE    R7,#0,POINTED         ;#!Check if the result was decimal point number
MOV     R6,SIGN                ;#!Copy SIGN contents to R6
CJNE    R6,#"+",SIGNED        ;#!Check on sign if positive or negative
MOV     A,R                    ;#!If positive, Copy R contents to ACC
MOV     B,#0AH                ;#!Fill B register with 10
DIV     AB                    ;#!Divide result by 100
JZ      LESSTEN                ;#!jump if reminder equals 0 (result<10)
ORL     A,#30H                ;#!If not convert first digit from BIN to ASCII
ACALL   DATAWRT              ;#!CALL subroutine, so pass data to LCD
MOV     A,B                    ;#!Get the second digit
ORL     A,#30H                ;#!Convert it from BIN to ASCII
ACALL   DATAWRT              ;#!CALL subroutine, so pass data to LCD
AJMP    DONE                  ;#!Done!!
:MOV     A,B                    ;#!If the result was one digit
ORL     A,#30H                ;#!Convert from Bin to ASCII
ACALL   DATAWRT              ;#!CALL subroutine, so print data on LCD
AJMP    DONE                  ;#!Done!!
:MOV     A,R                    ;#!For decimal point numbers
ORL     A,#30H                ;#!Convert from Bin to ASCII
ACALL   DATAWRT              ;#!CALL subroutine, so print data on LCD
MOV     A,#"."                ;#!MOV dot ascii code to ACC
ACALL   DATAWRT              ;#!CALL subroutine, so print data on LCD
MOV     A,TEMP                ;#!Get the digit that after point
ORL     A,#30H                ;#!Convert it from BIN to ASCII
ACALL   DATAWRT              ;#!CALL subroutine, so print data on LCD
AJMP    DONE                  ;#!Done!!
MOV     A,#"-"                ;#!IF signed mov "-" ASCII to ACC
ACALL   DATAWRT              ;#!CALL subroutine, so print data on LCD
AJMP    RETURN                ;#!Return to carryout the remain process
ACALL   LDELAY                ;#!Give data some time on LCD
ACALL   LDELAY                ;#!More time again
RET                             ;#!return to MIAN routine
```

Luồng chạy của hàm in kết quả



2.6 Các hàm phụ trợ và ý nghĩa:

Hàm kiểm tra input hợp lệ

```

validateInput:
    CJNE    A,#"+", next1
    AJMP    found
next1:    CJNE    A,#"-", next2
    AJMP    found
next2:    CJNE    A,#"*", next3
    AJMP    found
next3:    CJNE    A,#"/", next4
    AJMP    found
next4:    CJNE    A,#"=", next5
    CLR     DIF
    CLR     OIF
    SETB    AIF
    RET
next5:    SETB    DIF
    CLR     OIF
    CLR     AIF
    RET
found:    CLR     DIF
    SETB    OIF
    CLR     AIF
    RET
  
```

```

;#!Determine and set flags for each kind of input
;#!Check if input is add operator,
;#!Jump to found if input match with (+,-,*,/)
;#!Check if input is sub operator,
;#!Jump to found if input match with (+,-,*,/)
;#!Check if input is multiply operator
;#!Jump to found if input match with (+,-,*,/)
;#!Check if input is add operator,
;#!Jump to found if input match with (+,-,*,/)
;#!Check if input is assignment operator
;#!CLR digit input flag for non numbers
;#!CLR operation input flag for non operation
;#!SET assignment input flag for (=)
;#!return to MAIN
;#!SET digit input flag for numbers
;#!CLR operation input flag for non operation
;#!CLR assignment input flag if not (=)
;#!return to MAIN
;#!CLR digit input flag for non numbers
;#!SET operation input flag for any of operations
;#!CLR assignment input flag if not (=)
;#!return
  
```

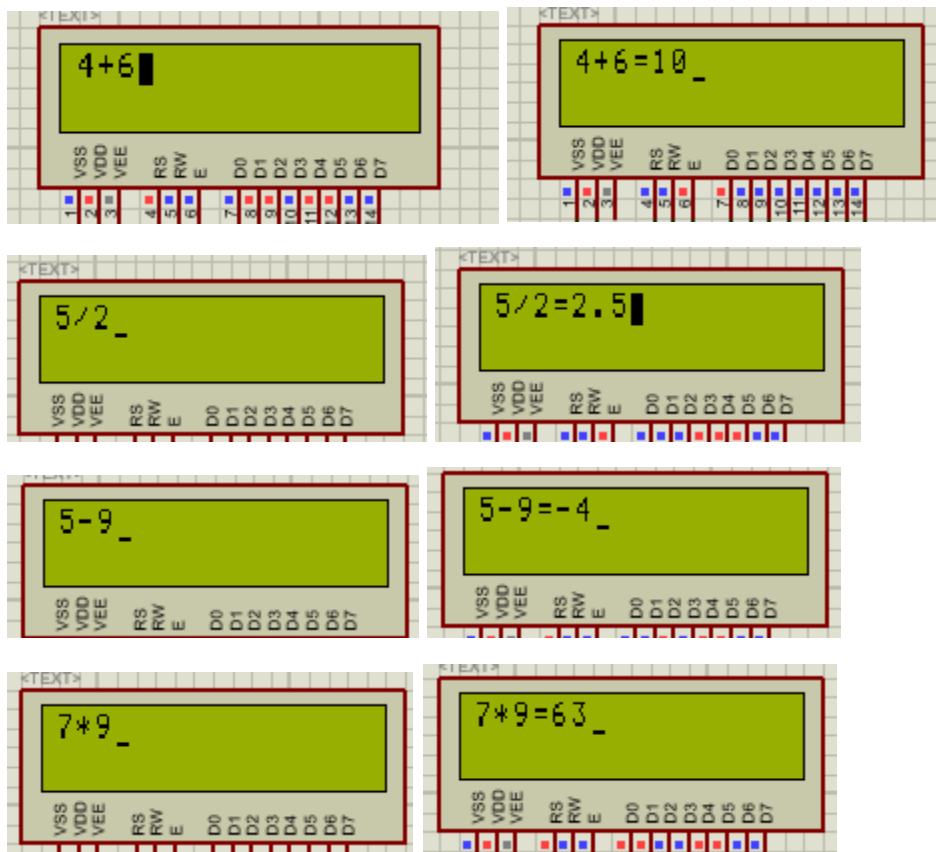
Delay

```

SDELAY:                                ;#!Short Delay subroutine
        MOV     R0,#50                ;#!R0 = 50
S11:    MOV     R1,#255                ;#!R1 = 255
S21:    DJNZ    R1,S21                ;#!stay here until R1 becomes ZERO
        DJNZ    R0,S11                ;#!
        RET                             ;#!return to caller
;-----
LDELAY:                                ;#!Long Delay Subroutine
        MOV     R0,#7                 ;#!R0 = 7
L33:    MOV     R1,#190                ;#!R1 = 190
L22:    MOV     R2,#255                ;#!R2 = 255
L11:    DJNZ    R2,L11                ;#!stay here until R2 becomes ZERO
        DJNZ    R1,L22                ;#!
        DJNZ    R0,L33                ;#!
        RET                             ;#!return to caller

```

3. Demo chương trình



4. Hạn chế:

- Chỉ có thể tính toán với số dương có một chữ số
- Chỉ có thể thực hiện 1 phép tính một lần.
- Không thể lưu kết quả của phép tính vừa thực hiện