**ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA KỸ THUẬT MÁY TÍNH**

🙡★🙣



**BÁO CÁO THỰC HÀNH LAB 04**

**MÔN THỰC HÀNH VI XỬ LÍ –VI ĐIỀU KHIỂN**

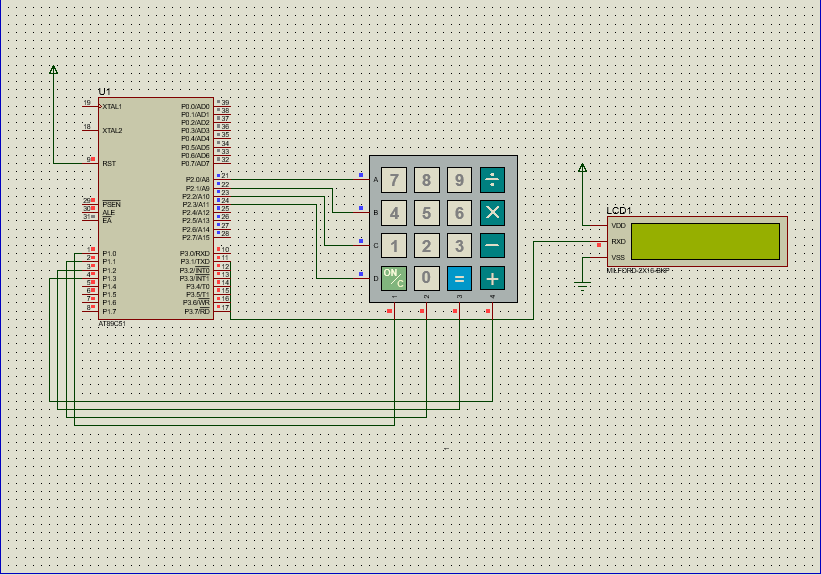
**Họ và tên sinh viên : Nguyễn Hữu Tứ**

**Mã số sinh viên: 19522453**

**GIẢNG VIÊN HƯỚNG DẪN**

**Bùi Phùng Hữu Đức**

***TP. HỒ CHÍ MINH, 5/2021***



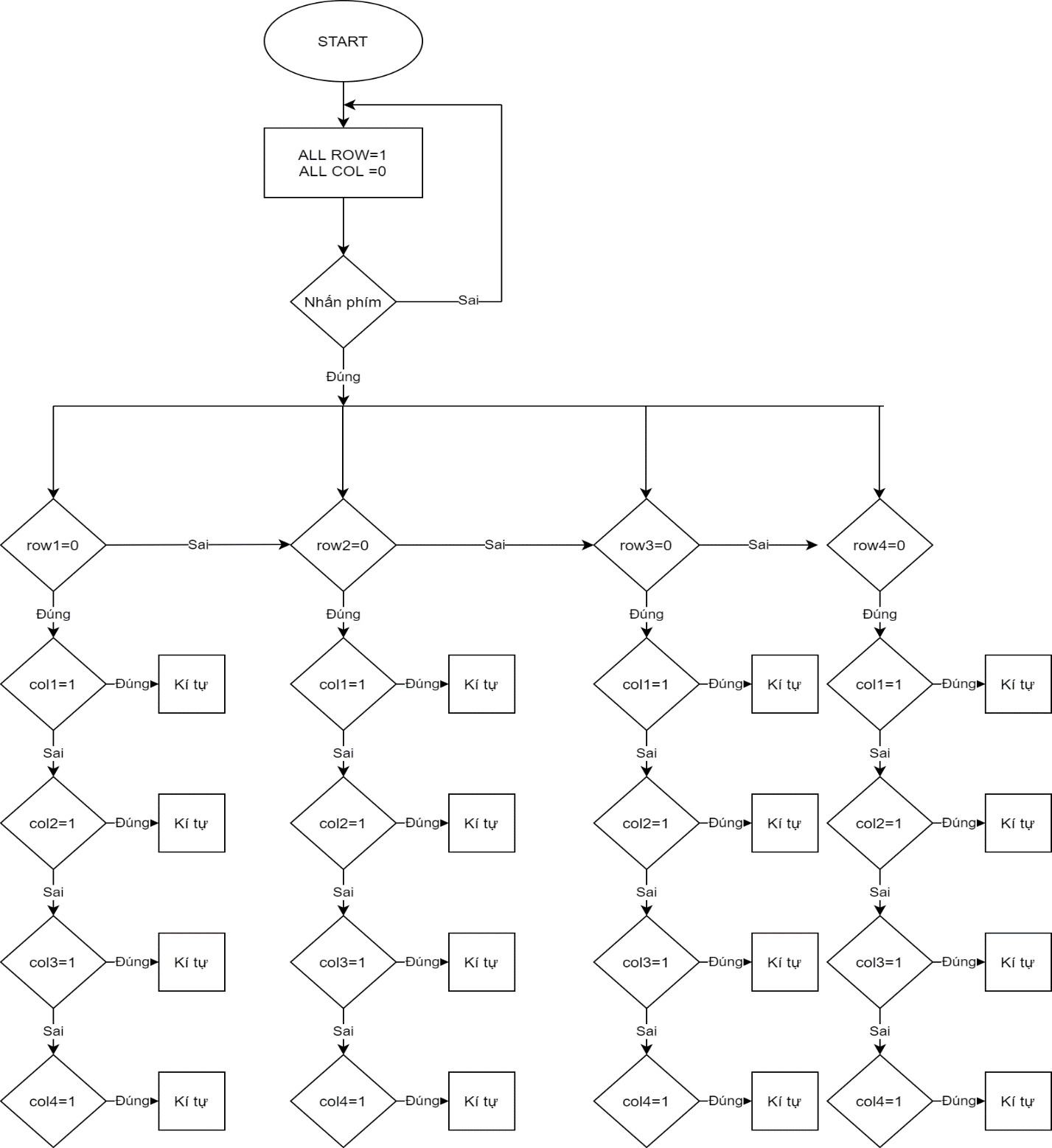
Hình1: Schematic

1. Cách sử dụng KeyPad:

-KeyPad dùng để đọc giá trị của người dùng nhập vào qua thuật toán quét phím .

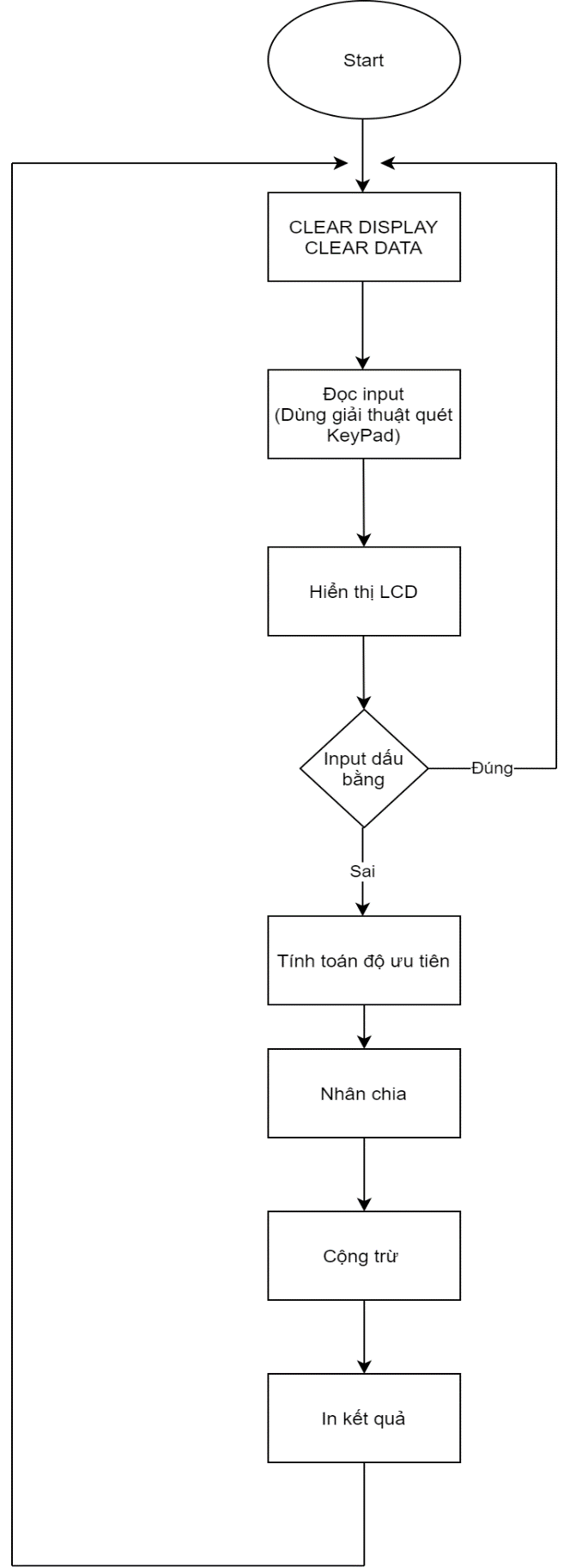
-Cách làm của thuật toán giống với cách tính tọa độ của trên mặt phẳng Oxy sẽ tính xem hàng nào cột nào, rồi sau đó xuất ra kí tự tương ứng với tọa độ đó.

* Flowchart để thực hiện thuật toán quét phím của Keypad :



Hình 1: Flowchart thể hiện thuật toán quét Keypad

1. Lưu đồ giải thuật của bài toán:



Hình 2: Lưu đồ giải thuật của bài toán

1. Phần giải thích code:

|  |  |
| --- | --- |
| Phần code | Phần giải thích |
| X0 EQU 30H  X1 EQU 31H  X2 EQU 32H  X3 EQU 33H  X4 EQU 34H  OP0 EQU 35H  OP1 EQU 36H  OP2 EQU 37H  OP3 EQU 38H  KQ EQU 39H  SIGN EQU 40H  TEMP EQU 41H  CHU\_SO BIT 0AH  OPR\_DAU BIT 0BH  DAU\_BANG BIT 0CH  ;-------------------------------  ORG 0000H  JMP MAIN  MAIN:  CALL XOA\_MAN\_HINH  CALL INPUT  CALL MUL\_DIV; nhanchia  CALL ADD\_SUB ; cong tru  CALL OUTPUT  JMP MAIN  ;------------------------------  ;------------------------------  XOA\_MAN\_HINH:  CLR A  MOV X0,A  MOV X1,A  MOV X2,A  MOV X3,A  MOV X4,A  MOV OP0,A  MOV OP1,A  MOV OP2,A  MOV OP3,A  MOV KQ,A  MOV SIGN,A  MOV TEMP,A  CLR C  MOV CHU\_SO,C  MOV OPR\_DAU,C  MOV DAU\_BANG,C  RET  ;--------------------------------  CAL:  CJNE R7,#"+",N1  ADD A,B  MOV KQ,A  MOV SIGN,#"+"  RET  N1:  CJNE R7,#"-",N2  SUBB A,B  JC BU\_2  MOV KQ,A  MOV SIGN,#"+"  RET  BU\_2:  CPL A  INC A  MOV KQ,A  MOV SIGN,#"-"  RET  N2:  CJNE R7,#"\*",N3  MUL AB  MOV KQ,A  MOV SIGN,#"+"  RET  N3:  CJNE R7,#"/",N4  MOV TEMP,B  DIV AB  MOV KQ,A  MOV A,#0AH  MUL AB  MOV B,TEMP  DIV AB  MOV TEMP,A  MOV SIGN,#"+"  N4:  RET  ;------------------------------------  MUL\_DIV:  MOV R7, OP0  CJNE R7, #"/",TINH1  TINH3:  MOV A,X0  MOV B,X1  CALL CAL  MOV X0, KQ  MOV OP0,OP1  MOV X1,X2  MOV OP1,OP2  MOV X2,X3  MOV OP2,OP3  MOV X3,X4  CLR A  MOV X4,A  MOV OP3,A  JMP MUL\_DIV  TINH1:  CJNE R7,#"\*",TINH2  JMP TINH3  TINH2:  MOV R7,OP1  CJNE R7,#"/",TINH4  TINH5:  MOV A,X1  MOV B,X2  CALL CAL  MOV X1,KQ  MOV OP1,OP2  MOV X2,X3  MOV OP2,OP3  MOV X3,X4  CLR A  MOV X4,A  MOV OP3,A  JMP TINH2  TINH4:  CJNE R7,#"\*",TINH6  JMP TINH5  TINH6:  MOV R7,OP2  CJNE R7,#"/",TINH7  TINH9:  MOV A,X2  MOV B,X3  CALL CAL  MOV X2,KQ  MOV OP2,OP3  MOV X3,X4  CLR A  MOV X4,A  MOV OP3,A  JMP TINH6  TINH7:  CJNE R7,#"\*",TINH8  JMP TINH9  TINH8:  MOV R7,OP3  CJNE R7,#"/",TINH10  TINH12:  MOV A,X3  MOV B,X4  CALL CAL  CLR A  MOV X4,A  MOV OP3,A  RET  TINH10:  CJNE R7,#"\*",TINH11  JMP TINH12  TINH11:  RET  ;---------------------------------  ADD\_SUB:  MOV R7,OP0  CJNE R7,#0,TINH15  RET  TINH15:  MOV A,X0  MOV B,X1  CALL CAL  MOV X0,KQ  MOV OP0,OP1  MOV X1,X2  MOV OP1,OP2  MOV X2,X3  MOV OP2,OP3  MOV X3,X4  CLR A  MOV X4,A  MOV OP3,A  JMP ADD\_SUB  ;---------------------------------  INPUT:  CALL LUU\_SO  MOV X0,A  CALL LUU\_DAU  MOV OP0,A  CALL LUU\_SO  MOV X1,A  CALL LUU\_DAU  JB DAU\_BANG,PASS  MOV OP1,A  CALL LUU\_SO  MOV X2,A  CALL LUU\_DAU  JB DAU\_BANG,PASS  MOV OP2,A  CALL LUU\_SO  MOV X3,A  CALL LUU\_DAU  JB DAU\_BANG,PASS  MOV OP3,A  CALL LUU\_SO  MOV X4,A  CALL LUU\_DAU  JMP PASS  LOI:  CALL LOI\_MESS  LJMP MAIN  PASS:  RET  ;-------------------------------  LUU\_SO:  CALL QUET\_KEYPAD  CALL TOAN\_TU  JB OPR\_DAU,LOI  JB DAU\_BANG,LOI  CALL TRUYEN  ANL A,#0FH  RET  ;----------------------------  LUU\_DAU:  CALL QUET\_KEYPAD  CALL TOAN\_TU  JB CHU\_SO,LOI  CALL TRUYEN  RET  ;-----------------------------  ;-----------------------------  OUTPUT:  MOV R7,TEMP  MOV R6,SIGN  CJNE R7,#0,POINTED  RETURN:  MOV A,KQ  MOV B,#100D  DIV AB  JZ LESSTEN  ORL A,#30H  CALL TRUYEN  MOV A,B  MOV B,#0AH  DIV AB  ORL A,#30H  CALL TRUYEN  MOV A,B  ORL A,#30H  CALL TRUYEN  JMP DONE  LESSTEN:  CJNE R6,#"-",DAU  MOV A,#"-"  CALL TRUYEN  DAU:  MOV A,B  MOV B,#0AH  DIV AB  JZ LESSTEN1  ORL A,#30H  CALL TRUYEN  LESSTEN1:  MOV A,B  ORL A,#30H  CALL TRUYEN  CJNE R7,#0,EXIT\_OUT  JMP DONE  EXIT\_OUT:  RET  POINTED:  CALL RETURN  MOV A,#"."  ACALL TRUYEN  MOV A,TEMP  ORL A,#30H  CALL TRUYEN  AJMP DONE  SIGNED:  MOV A,#"-"  CALL TRUYEN  JMP RETURN  DONE:  RET  ;--------------------------------  TOAN\_TU:  CJNE A,#"+", SUB AJMP FOUND\_EXIT  SUB:  CJNE A,#"-", MULTI AJMP FOUND\_EXIT  MULTI:  CJNE A,#"\*", DIVI  AJMP FOUND\_EXIT  DIVI:  CJNE A,#"/", EQUAL  AJMP FOUND\_EXIT  EQUAL:  CJNE A,#"=", EXIT  CLR CHU\_SO  CLR OPR\_DAU  SETB DAU\_BANG  RET  EXIT:  SETB CHU\_SO CLR OPR\_DAU CLR DAU\_BANG RET  FOUND\_EXIT:  CLR CHU\_SO  SETB OPR\_DAU  CLR DAU\_BANG  RET  ;-----------------------------------  QUET\_KEYPAD:  MOV P1,#11111111B  S1: MOV P2,#0  MOV A,P1  ANL A,#00001111B  CJNE A,#00001111B,S2  SJMP S1    S2: CALL DELAY\_20MS  MOV A,P1  ANL A,#00001111B  CJNE A,#00001111B,OUT  SJMP S1  OUT:CALL DELAY\_20MS  MOV A,P1  ANL A,#00001111B  CJNE A,#00001111B,OUT1  OUT1:MOV P2,#11111110B  MOV A,P1  ANL A,#00001111B  CJNE A,#00001111B,ROW\_0  MOV P2,#11111101B  MOV A,P1  ANL A,#00001111B  CJNE A,#00001111B,ROW\_1  MOV P2,#11111011B  MOV A,P1  ANL A,#00001111B  CJNE A,#00001111B,ROW\_2  MOV P2,#11110111B  MOV A,P1  ANL A,#00001111B  CJNE A,#00001111B,ROW\_3  LJMP S2  ROW\_0:  MOV DPTR,#ROW0  JMP KT  ROW\_1:  MOV DPTR,#ROW1  JMP KT  ROW\_2:  MOV DPTR,#ROW2  JMP KT  ROW\_3:  MOV DPTR,#ROW3  JMP KT  KT:  RRC A  JNC FOUND  INC DPTR  JMP KT  FOUND:  CLR A  MOVC A,@A+DPTR  CJNE A,#"X",F1  CALL XOA\_MAN\_HINH\_CMD  F1:  RET  ;-----------------------------------  LOI\_MESS:  MOV DPTR,#LOIOR  E1:  CLR A  MOVC A,@A+DPTR  CALL TRUYEN  CALL DELAY\_20MS  INC DPTR  JZ E2  SJMP E1  E2:  CALL DELAY\_20MS  CALL XOA\_MAN\_HINH\_CMD  RET  ;-------------------------------------  XOA\_MAN\_HINH\_CMD:  MOV A,#254D  CALL TRUYEN  MOV A,#1D  CALL TRUYEN  LJMP MAIN  RET  ;-------------------------------------  DELAY\_20MS:  MOV R0,#4  LOOP:  MOV TMOD,#01H  MOV TH0,#HIGH(-5000)  MOV TL0,#LOW(-5000)  SETB TR0  HERE1:  JNB TF0,HERE1  CLR TF0  CLR TR0  DJNZ R0,LOOP  RET  ;--------------------------------------  TRUYEN:  MOV TMOD,#20H  MOV TH1,#0FDH  MOV SCON,#50H  SETB TR1  MOV SBUF,A  HERE:  JNB TI,HERE  CLR TI  RET  ;----------------------------------  ORG 300H  LOIOR: DB "LOIOR",0  XOA\_MAN\_HINH\_LCD: DB 254D,01D  ROW0: DB "7","8","9","/"  ROW1: DB "4","5","6","\*"  ROW2: DB "1","2","3","-"  ROW3: DB "X","0","=","+"  END  ;;--------------------------- | Ta định nghĩa các toán tử,toán hạng vào các địa chỉ trực tiếp trong ram  Các Xi là lưu chữ số  Các Opi là lưu dấu  Lưu kết quả tính toán  Lưu dấu  Lưu giá trị tính toán tạm  Bit để phát hiện đó là chữ số  Bit để phát hiện đó là dấu  Bit để phát hiện đó là dấu bằng  Gọi hàm xóa màn hình  Gọi hàm nhập  Gọi hàm nhân chia  Gọi hàm cộng trừ  Gọi hàm xuất  Nhảy lại hàm Main để thực hiện vòng lặp  Hàm xóa màn hình thì ta gán tất cả các số,dấu,..  Gán tất cả các biến bằng 0.  Xóa cờ A  Các biến được gán cho A mà A=0  Hàm Cal dùng để phân loại xem đây toán tử gì  So sánh R7 với + nếu đúng thì thực hiện phép cộng, không thì nhảy tới nhãn N2  A+B  Lưu kết quả từ A vào KQ  Lưu dấu + vào SIGN  So sánh R7 với - nếu đúng thì thực hiện phép cộng, không thì nhảy tới nhãn N2  A-B  Nhảy tới BU\_2 nếu A=0, nghĩa là phép tính âm  Lưu kết quả vào KQ  Lưu dấu + vào SIGN  Hàm tính bù 2  Đảo bit của A  Tiến hành +1  Lưu kết quả vào KQ  Lưu dấu – vào SIGN  So sánh R7 với \* nếu đúng thì thực hiện phép cộng, không thì nhảy tới nhãn N3  A nhân B  Lưu kết quả vào KQ  Lưu dấu + vào SIGN  So sánh R7 với / nếu đúng thì thực hiện phép cộng, không thì nhảy tới nhãn N4  Lưu B vào biến tạm TEMP  A/B  Phần nguyên lưu vào KQ  Cho A=10 mục đích tính tiếp phần sau phần dư  A\*B  Biến temp gán giá trị cho B  A/B  Gán giá trị A vào temp  Sign dc gán bằng dấu +  Thoát khi tới nhãn N4  Các hàm ưu tiên tính toán  Ý tưởng: Thực hiện nhân chia trước  Gán giá trị OP0 là dấu đầu tiên cho R7  Sao sánh R7 với dấu / nếu không = nhảy tới TINH1, bằng thì nhảy tới TINH3 để tính toán  Hàm TINH3 thực hiện tính toán và dời các toàn tử từ sau lên trước, bản chất chỉ thực hiện 1 phép tính đầu với 2 số và 1 phép tính.  Kết quả tính toán được lưu vào X0  Di chuyển giá trị của OP1->OP0  Di chuyển giá trị của X2->X1  Di chuyển giá trị của OP2->OP1  Di chuyển giá trị của X3->X2  Di chuyển giá trị của OP3->OP2  Di chuyển giá trị của X4->X3  Xóa A  Di chuyển giá trị A->X4,xóa toán tử thứ 5  Di chuyển xóa dấu thứ 4  Thực hiện lại vòng lặp để tiếp tục quét  Với ý tưởng trên ta thực hiện tương tự như dấu \*  Kết quả tính toán được lưu vào X1  Di chuyển giá trị của OP1->OP0  Di chuyển giá trị của X2->X1  Di chuyển giá trị của OP2->OP1  Di chuyển giá trị của X3->X2  Di chuyển giá trị của OP3->OP2  Di chuyển giá trị của X4->X3  Xóa A  Di chuyển giá trị A->X4,xóa toán tử thứ 5  Di chuyển xóa dấu thứ 4  Thực hiện lại vòng lặp nhảy tới TINH2  Thực hiện tương tự cho đến khi nào hết toán tử thứ 5 và dấu thứ 4 thì dừng  Hàm ưu tiên tính dấu + và –  Với ý tưởng tương tự hàm trên  OP0 được đưa vào R7  So sánh R7 với 0 nếu khác thì nhảy tới TINH15  Thoát  Hàm TINH15  Giá trị của X0->A  Giá trị của X0->A  Hàm tính toán  Lưu KQ vào X0,thực hiện dịch các toán hạng tới trước  Di chuyển giá trị OP1->OP0  Di chuyển giá trị X2->X1  Di chuyển giá trị OP2->OP1  Di chuyển giá trị X3->X2  Di chuyển giá trị OP3->OP2  Di chuyển giá trị X4->X3  Xóa A  Xóa phần tử đứng cuối  Xóa dấu đứng cuối  Nhảy lại hàm  Hàm nhập  Hàm lưu số  Lưu số vào X0  Hàm lưu dấu  Lưu dấu vào OP0  Hàm lưu số  Lưu số vào X1  Hàm lưu dấu  Lưu dấu vào OP1  Hàm lưu số  Lưu số vào X2  Hàm lưu dấu  Lưu dấu vào OP2  Nếu nhấn dấu = thì nhảy tới hàm PASS  Lưu dấu vào X2  Hàm lưu dấu  Lưu số vào X3  Tương tự  Nếu nhấn dấu = thì nhảy tới hàm PASS  Lưu dấu vào OP3  Hàm lưu số  Lưu số vào X4  Gọi hàm lưu dấu  Nhảy tới hàm PASS  Hàm lỗi error  Gọi hàm hiển thị lỗi  Nhảy lại tới hàm main  Hàm Pass là thoát  Hàm lưu số  Đầu tiên quét KeyPad để nhận giá trị  Gọi hàm toán tử để kiểm tra và lưu số  Kiểm tra bit dấu có phải hay không, nếu đúng thì lỗi  Gọi hàm kiển tra dấu bằng  Gọi hàm để truyền UART để hiện thị lên LCD  Xóa giá trị thừa ở A , ta xóa 4 bit có trọng số cao nhất  Thoát  Hàm lưu dấu  Quét keypad nhận giá trị  Gọi hàm toán tử để lưu và kiểm tra số  Gọi hàm chứ số và kiểm tra lỗi  Gọi hàm truyền đi  Hàm xuất dữ liệu ra LCD  Di chuyển giá trị temp -> R7  Di chuyển giá trị SIGN -> R6  So sánh R7 với 0  Hàm trả về  A=KQ  B=100  A/B  A=0 thì nhảy  Or A với 30H  Gọi để truyền  A=B  B=10  A/B  Or A với 30H  Gọi truyền dữ liệu hiển thị  A=B  Or A với #30h  Gọi truyền dữ liệu hiển thị  Hàm thực hiện in ra số float  Hàm toán tử check các dấu,chữ số và dấu bằng để thực hiện in ra nếu có lỗi.  So sanh với dấu +  So sanh với dấu -  Set bit dấu = lên 1 báo đây là dấu bằng  Set bit của chữ số lên 1 báo đấy là chữ số  Set bit của dấu lên 1 báo đấy là dấu  Thực hiện giải thuật quét KeyPad xác định tọa đó của phím nhấn.  Cho các cột đều bằng 1  Cho các hàng =0  Di chuyển giá trị của P1->A  Xóa 4 bit có trọng số cao nhất vì 4 bit này không dùng tới  So sánh A với 00001111B, nếu khác thì có nghĩa có phím đã nhấn thì nhảy tới S2, không quay lại S1  Gọi hàm delay 20ms  Di chuyển dữ liệu từ P1->A  Xóa bỏ bớt 4 bit đầu do không cần sử dụng tới  So sánh A với 00001111H nếu khác nhảy tới OUT  Nếu bằng thì nhảy lại S1  Gọi hàm Delay 20ms  Di chuyển dữ liệu từ P1->A  Xóa bỏ bớt 4 bit đầu do không cần sử dụng tới  So sánh A với 00001111H nếu khác nhảy tới OUT1  Hàm OUT1 kiểm tra các hàng  Di chuyển dữ liệu từ P1->A  Xóa 4 bit không cần thiết  So sánh nếu A khác với 00001111H thì nhảy tới ROW\_0  Nếu bằng qua check hàng tiếp theo  Di chuyển dữ liệu từ P1->A  Xóa 4 bit không cần thiết  So sánh nếu A khác với 00001111H thì nhảy tới ROW\_1  Nếu bằng qua check hàng tiếp theo  Di chuyển dữ liệu từ P1->A  Xóa 4 bit không cần thiết  So sánh nếu A khác với 00001111H thì nhảy tới ROW\_2  Nếu bằng qua check hàng tiếp theo  Di chuyển dữ liệu từ P1->A  Xóa 4 bit không cần thiết  So sánh nếu A khác với 00001111H thì nhảy tới ROW\_3  Nhảy tới lại hàm S2  Hàm ROW\_0  Gán địa chỉ bảng đầu của mảng ROW0 cho DPTR  Nhảy tới hàm KT  Hàm ROW\_1  Gán địa chỉ bảng đầu của mảng ROW1 cho DPTR  Nhảy tới hàm KT  Hàm ROW\_2  Gán địa chỉ bảng đầu của mảng ROW2 cho DPTR  Nhảy tới hàm KT  Hàm ROW\_3  Gán địa chỉ bảng đầu của mảng ROW3 cho DPTR  Nhảy tới hàm KT  Hàm KT  Quay phải A với Cy là bit nhỏ nhất  Nếu C=0 thì nhảy tới hàm FOUND  Không thì tăng DPTR+1  Nhảy lại tới hàm KT  Hàm FOUND  Xóa A  Lấy giá trị từ mảng sang cho A  Nếu là phím AC thì xóa màn hình  Hàm xóa màn hình  Hàm F1  Thoát  Hàm in ra lỗi  Hàm xóa màn hình  Hàm tính Delay 20ms  Hàm truyền sử dụng UART  Phần khai báo mảng và chuỗi | |