

PRACTICE EXERCISES OF THE MICROPROCESSORS & MICROCONTROLLERS

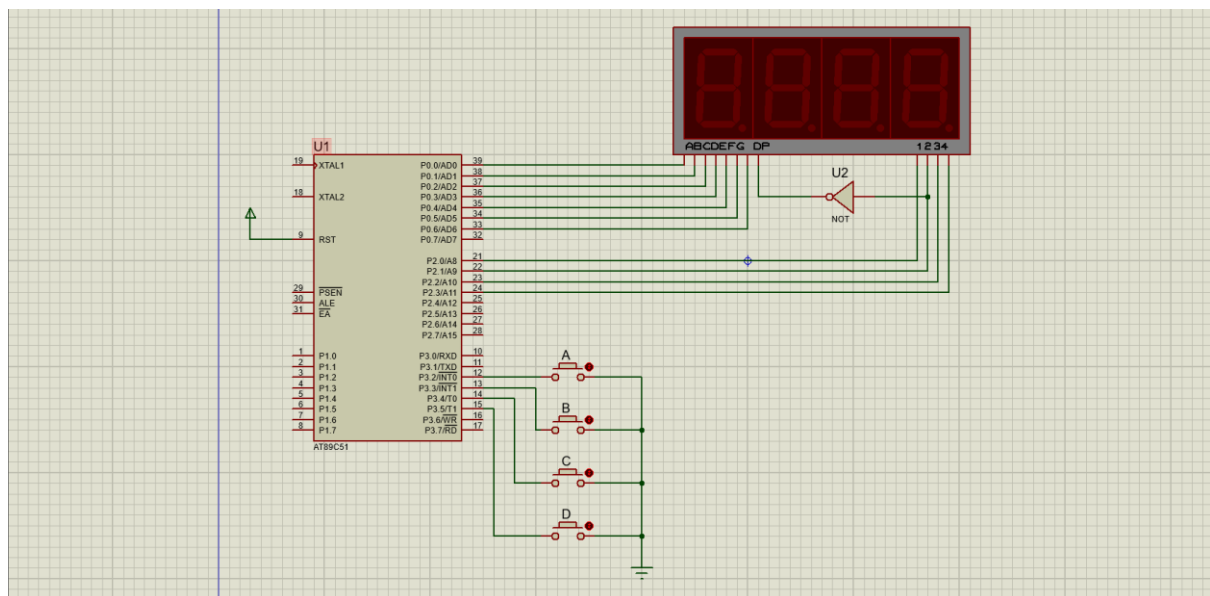
Instructor: The Tung Than

Student's name: Lê Hữu Đạt

Student code: 21520697

PRACTICE REPORT NO #3: USING INTERRUPT

I. Content 1: Design result



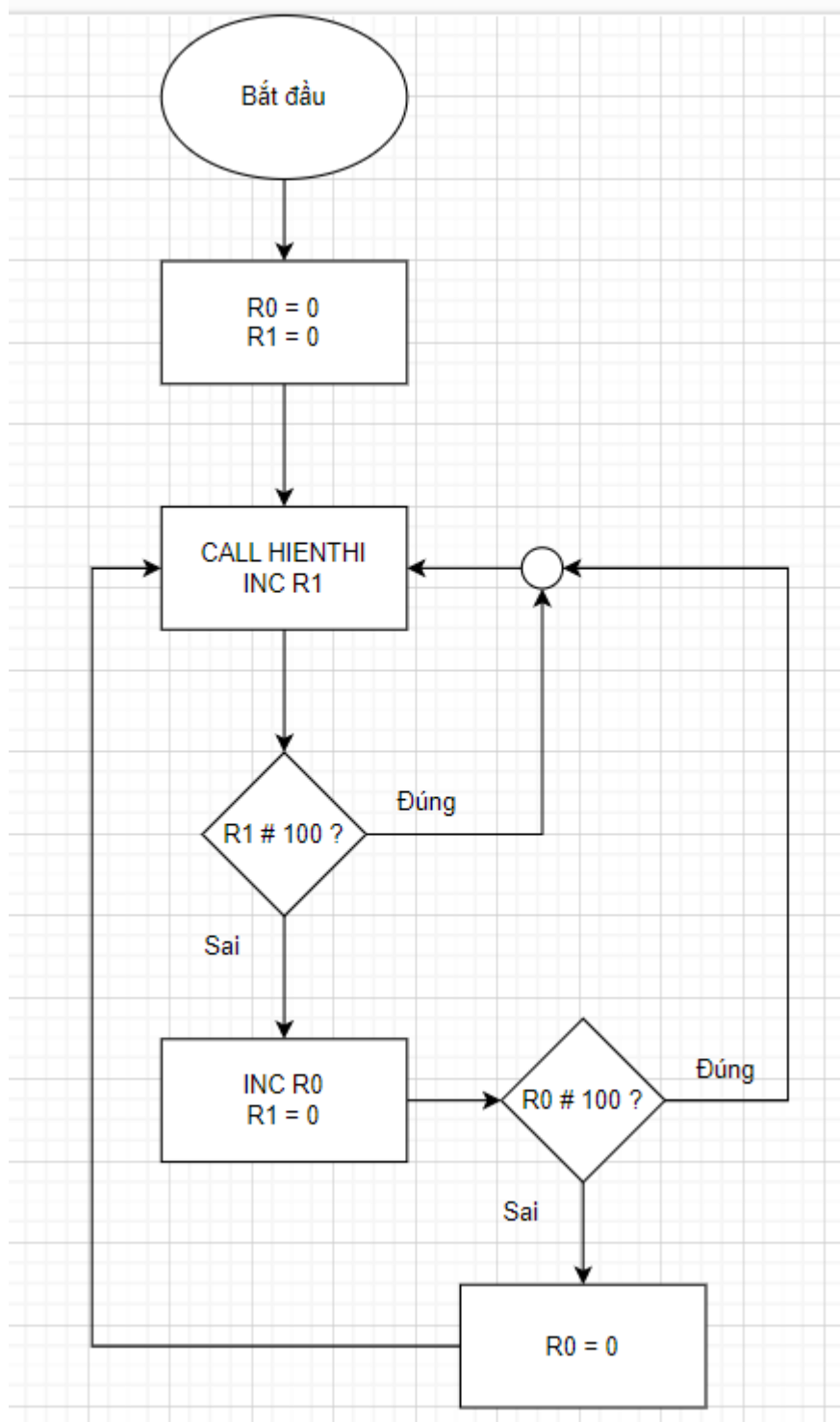
II. Content 2: Explain the operating principle of the effects, accompanied by a video (send a Google Drive link) to demonstrate the circuit operation in case the instructor cannot run the design file.

- Link Google Drive: https://drive.google.com/file/d/1_0c_ysImVH-qVt91ZMV4dtvAPv1psaLg/view?usp=share_link

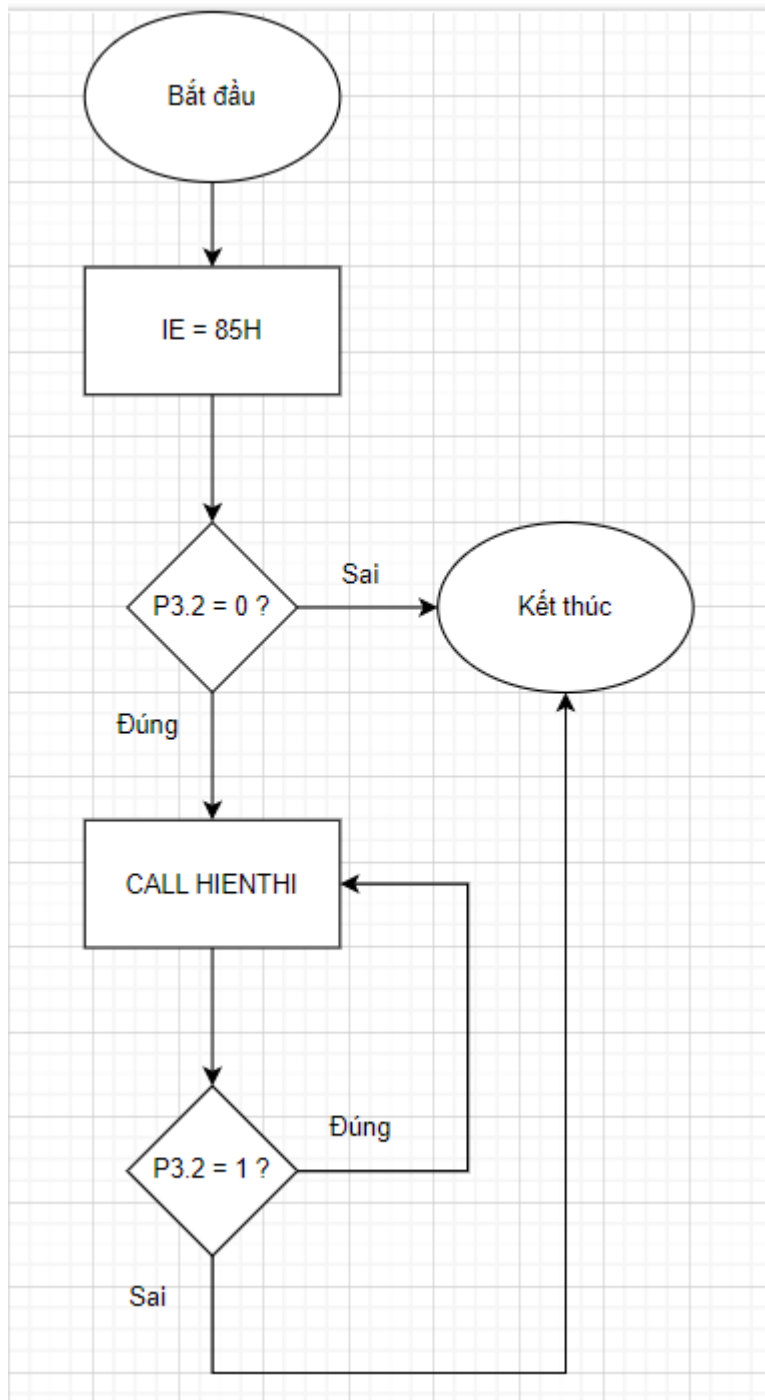
- Nguyên lí hoạt động của mạch:

- + Ban đầu, đồng hồ sẽ đếm từ 00.00 đến 99.99 và quay lại từ đầu
- + Khi bấm nút A đồng hồ sẽ dừng lại và bấm tiếp một lần nữa đồng hồ sẽ tiếp tục đếm.
- + Khi bấm nút B đồng hồ sẽ reset về 00.00 là đếm lại từ đầu

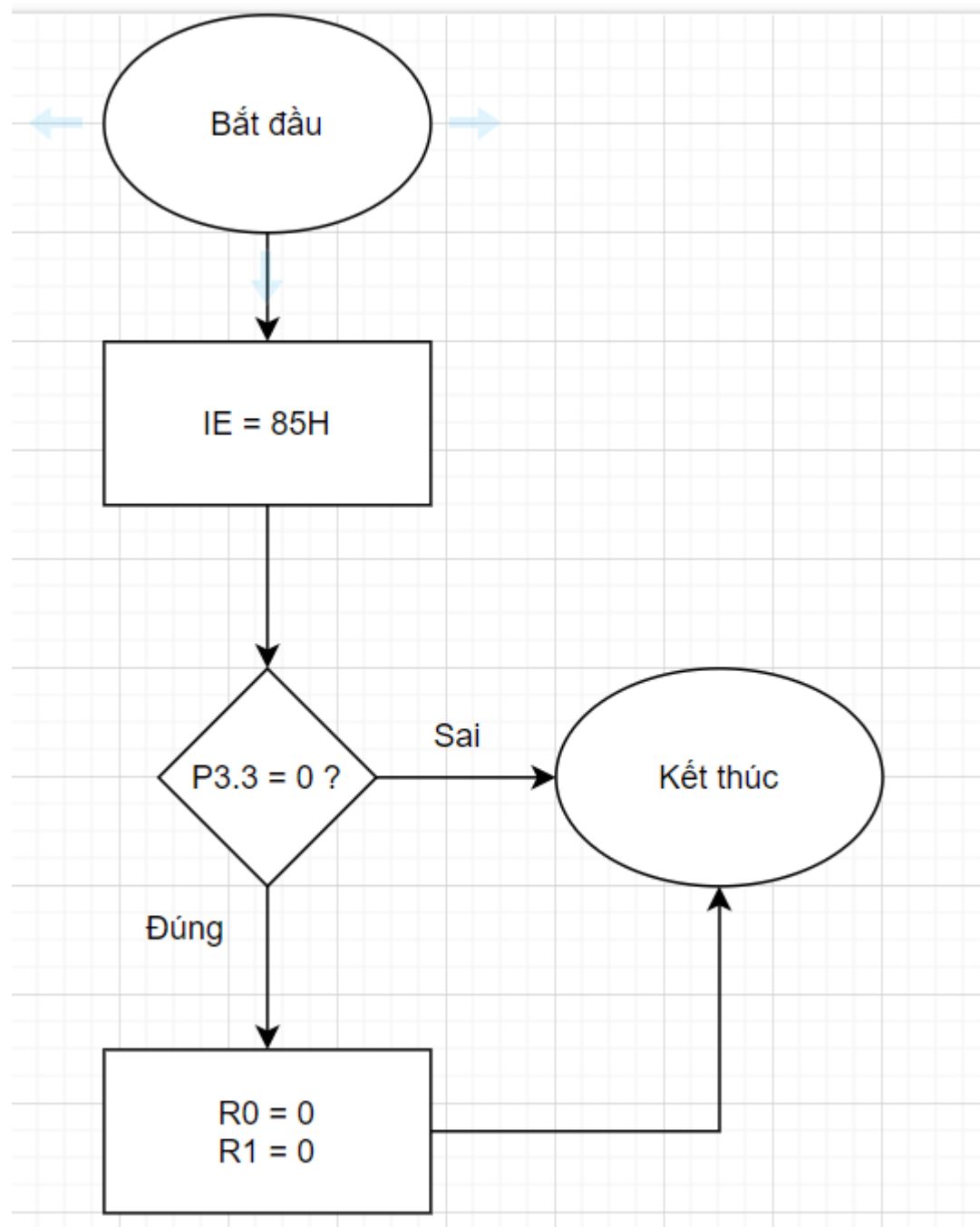
- Lưu đồ thuật toán của mạch:



- Lưu đồ thuật toán ngắt ngoài khi bấm nút A:



- Lưu đồ thuật toán nút B:



Code	Giải thích
ORG 0000H	Địa chỉ vector ngắt Reset
SJMP MAIN	Nhảy đến hàm MAIN
ORG 0003H	Địa chỉ vector ngắt INT0
SJMP BUTTON_A	Nhảy đến hàm BUTTON_A
ORG 0013H	Địa chỉ vector ngắt INT1
SJMP BUTTON_B	Nhảy đến hàm BUTTON_B
MAIN:	Hàm MAIN
MOV R0, #0	Gán thanh ghi R0 = 0
MOV R1, #0	Gán thanh ghi R1 = 0
MOV IE, #85H	Gán thanh ghi IE = 85H để cho phép ngắt toàn bộ, ngắt từ INT0 và INT1
CALL CLOCK	Gọi hàm CLOCK
JMP MAIN	Nhảy đến hàm MAIN
CLOCK:	Hàm CLOCK
CALL HIENTHI	Gọi hàm HIENTHI
INC R1	Tăng giá trị thanh ghi R1 lên 1 đơn vị
CJNE R1, #100, CLOCK	Nhảy đến hàm CLOCK nếu thanh ghi R1 không bằng 100
INC R0	Tăng thanh ghi R0 lên 1 đơn vị
MOV R1, #0	Gán thanh ghi R1 = 0
CJNE R0, #100, CLOCK	Nhảy đến hàm CLOCK nếu thanh ghi R1 không bằng 100
MOV R0, #0	Gán thanh ghi R0 = 0
RET	Thoát khỏi hàm CLOCK
HIENTHI:	Hàm HIENTHI
MOV DPTR, #MALED	Gán thanh ghi DPTR = địa chỉ mảng MALED
MOV A, R0	Gán A = R0
MOV B, #10	Gán B = 10
DIV AB	A = A/B và B = A%B
MOVC A, @A + DPTR	Gán A = A + DPTR

MOV P0, A	Gán P0 = A
SETB P2.0	Gán P2.0 = 1
CALL DELAY	Gọi hàm DELAY
CLR P2.0	Gán P2.0 = 0
MOV A, B	Gán A = B
MOVC A, @A + DPTR	Gán A = A + DPTR
MOV P0, A	Gán P0 = A
SETB P2.1	Gán P2.1 = 1
CALL DELAY	Gọi hàm DELAY
CLR P2.1	Gán P2.1 = 0
MOV A, R1	Gán A = R1
MOV B, #10	Gán B = 10
DIV AB	$A = A/B$ và $B = A \% B$
MOVC A, @A + DPTR	Gán A = A + DPTR
MOV P0, A	Gán P0 = A
SETB P2.2	Gán P2.2 = 1
CALL DELAY	Gọi hàm DELAY
CLR P2.2	Gán P2.2 = 0
MOV A, B	Gán A = B
MOVC A, @A + DPTR	Gán A = A + DPTR
MOV P0, A	Gán P0 = A
SETB P2.3	Gán P2.3 = 1
CALL DELAY	Gọi hàm DELAY
CLR P2.3	Gán P2.3 = 0
RET	Thoát khỏi hàm HIENTHI
BUTTON_A:	Hàm BUTTON_A
JNB P3.2, \$	Nếu P3.2 = 0 thì nhảy tại chỗ
LOOP:	Hàm LOOP
CALL HIENTHI	Gọi HIENTHI
JB P3.2, LOOP	Nếu P3.2 = 1 thì nhảy lại hàm LOOP

JNB P3.2, \$	Nếu P3.2 = 0 thì nhảy tại chỗ
RETI	Thoát khỏi chương trình phục vụ ngắt
BUTTON_B:	Hàm BUTTON_B
MOV R0, #0	Gán thanh ghi R0 = 0
MOV R1, #0	Gán thanh ghi R1 = 0
RETI	Thoát khỏi chương trình phục vụ ngắt
DELAY:	Hàm DELAY
MOV R2, #5	Gán thanh ghi R2 = 5
DELAY1:	Hàm DELAY1
MOV R3, #247	Gán thanh ghi R3 = 247
DJNZ R3, \$	Giảm R3 đi 1 đơn vị và nhảy tại chỗ nếu R3 khác 0
DJNZ R2, DELAY1	Giảm R2 đi 1 đơn vị và nhảy đến DELAY1 nếu R2 khác 0
RET	Thoát khỏi chương trình DELAY
MALED: DB 0XC0, 0XF9, 0XA4, 0XB0, 0X99, 0X92, 0X82, 0XF8, 0X80, 0X90	Mảng 1 chiều chứa mã LED 7 đoạn chung anode
END	Kết thúc chương trình

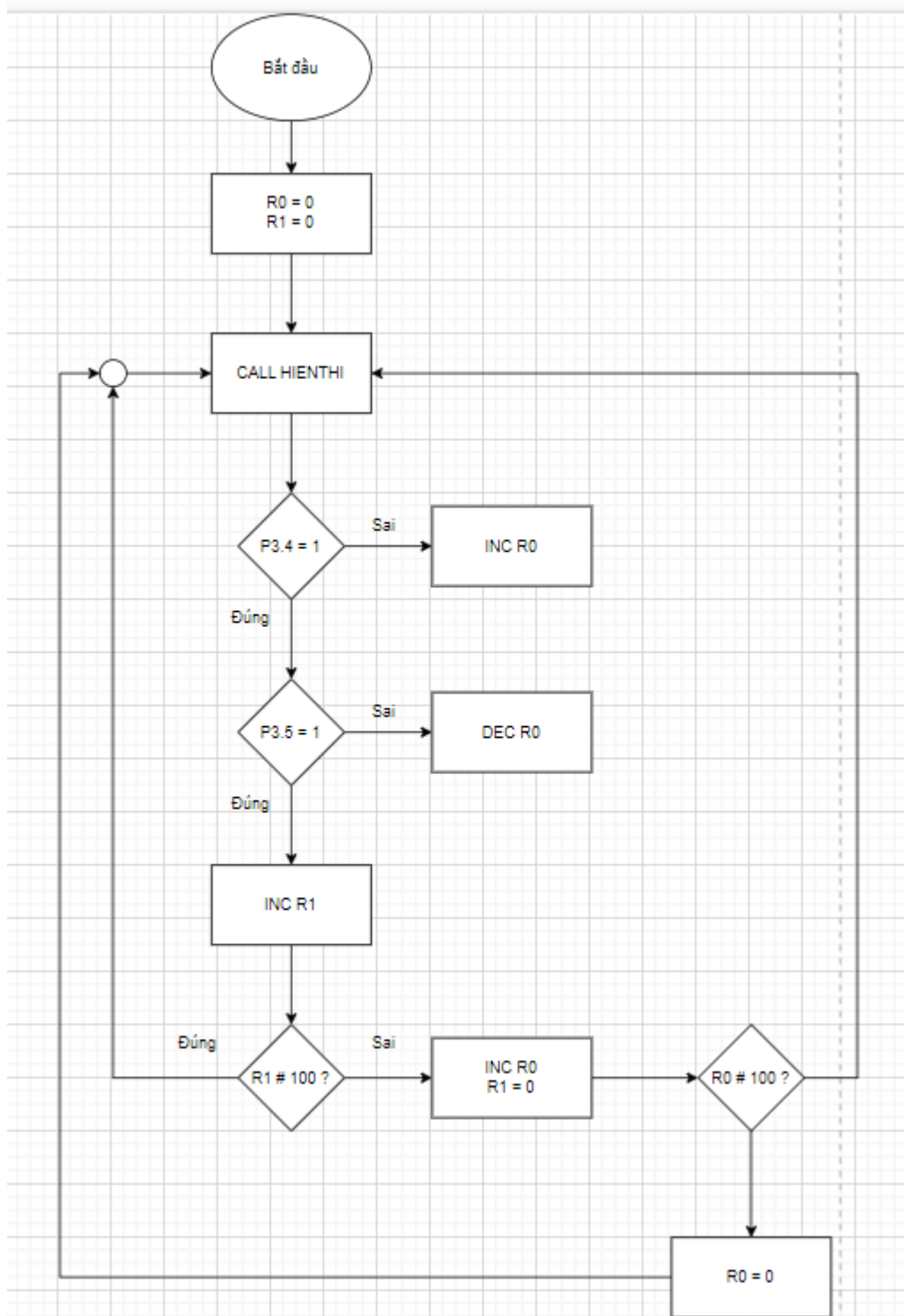
III. Content 3: Add 2 buttons to the Sport watch with the following function:

- **Button C: Increase the number of seconds counting to 1 second.**
- **Button D: Decrease the number of seconds to 1 second.**

- Nguyên lí hoạt động của mạch:

- + Ban đầu, đồng hồ sẽ đếm từ 00.00 đến 99.99 và quay lại từ đầu.
- + Khi bấm nút C đồng hồ sẽ tăng giây lên 1 đơn vị.
- + Khi bấm nút D đồng hồ sẽ giảm giây đi 1 đơn vị.

- Lưu đồ thuật toán của mạch:



CODE	Giải thích
ORG 0000H	Địa chỉ vector ngắt Reset
SJMP MAIN	Nhảy đến hàm MAIN
MAIN:	Hàm MAIN
MOV R0, #0	Gán thanh ghi R0 = 0
MOV R1, #0	Gán thanh ghi R1 = 0
CALL CLOCK	Gọi hàm CLOCK
JMP MAIN	Nhảy đến hàm MAIN
CLOCK:	Hàm CLOCK
CALL HIENTHI	Gọi hàm HIENTHI
JB P3.4, NEXT	Nếu P3.4 = 1 thì sẽ nhảy đến NEXT
JNB P3.4, \$	Nếu P3.4 = 0 thì sẽ nhảy tại chỗ
INC R0	Tăng R0 lên 1 đơn vị
NEXT:	Hàm NEXT
JB P3.5, NEXT1	Nếu P3.5 = 1 thì sẽ nhảy đến NEXT1
JNB P3.5, \$	Nếu P3.5 = 0 thì sẽ nhảy tại chỗ
DEC R0	Giảm R0 lên 1 đơn vị
NEXT1:	Hàm NEXT1
INC R1	Tăng giá trị thanh ghi R1 lên 1 đơn vị
CJNE R1, #100, CLOCK	Nhảy đến hàm CLOCK nếu thanh ghi R1 không bằng 100
INC R0	Tăng thanh ghi R0 lên 1 đơn vị
MOV R1, #0	Gán thanh ghi R1 = 0
CJNE R0, #100, CLOCK	Nhảy đến hàm CLOCK nếu thanh ghi R1 không bằng 100
MOV R0, #0	Gán thanh ghi R0 = 0
RET	Thoát khỏi hàm CLOCK
HIENTHI:	Hàm HIENTHI
MOV DPTR, #MALED	Gán thanh ghi DPTR = địa chỉ mảng MALED
MOV A, R0	Gán A = R0

MOV B, #10	Gán B =10
DIV AB	$A = A/B$ và $B = A \% B$
MOVC A, @A + DPTR	Gán A = A + DPTR
MOV P0, A	Gán P0 = A
SETB P2.0	Gán P2.0 = 1
CALL DELAY	Gọi hàm DELAY
CLR P2.0	Gán P2.0 = 0
MOV A, B	Gán A = B
MOVC A, @A + DPTR	Gán A = A + DPTR
MOV P0, A	Gán P0 = A
SETB P2.1	Gán P2.1 = 1
CALL DELAY	Gọi hàm DELAY
CLR P2.1	Gán P2.1 = 0
MOV A, R1	Gán A = R1
MOV B, #10	Gán B =10
DIV AB	$A = A/B$ và $B = A \% B$
MOVC A, @A + DPTR	Gán A = A + DPTR
MOV P0, A	Gán P0 = A
SETB P2.2	Gán P2.2 = 1
CALL DELAY	Gọi hàm DELAY
CLR P2.2	Gán P2.2 = 0
MOV A, B	Gán A = B
MOVC A, @A + DPTR	Gán A = A + DPTR
MOV P0, A	Gán P0 = A
SETB P2.3	Gán P2.3 = 1
CALL DELAY	Gọi hàm DELAY
CLR P2.3	Gán P2.3 = 0
RET	Thoát khỏi hàm HIENTHI
DELAY:	Hàm DELAY
MOV R2, #5	Gán thanh ghi R2 = 5

<p>DELAY1:</p> <p>MOV R3, #247</p> <p>DJNZ R3, \$</p> <p>DJNZ R2, DELAY1</p> <p>RET</p> <p>MALED: DB 0XC0, 0XF9, 0XA4, 0XB0, 0X99, 0X92, 0X82, 0XF8, 0X80, 0X90</p> <p>END</p>	<p>Hàm DELAY1</p> <p>Gán thanh ghi R3 = 247</p> <p>Giảm R3 đi 1 đơn vị và nhảy tại chỗ nếu R3 khác 0</p> <p>Giảm R2 đi 1 đơn vị và nhảy đến DELAY1 nếu R2 khác 0</p> <p>Thoát khỏi chương trình DELAY</p> <p>Mảng 1 chiều chứa mã LED 7 đoạn chung anode</p> <p>Kết thúc chương trình</p>
--	---

*** So sánh sự khác nhau giữa ngắt theo cạnh (Edge) và ngắt theo mức (Level)**

Ngắt theo cạnh (Edge)	Ngắt theo mức (Level)
<ul style="list-style-type: none"> - Ngắt theo cạnh xảy ra khi tín hiệu đầu vào thay đổi từ mức thấp (Low) sang mức cao (High) hoặc ngược lại. Điều này cho phép ngắt được kích hoạt khi có sự kiện thay đổi trạng thái xảy ra. - Ngắt theo cạnh thường được xử lý nhanh hơn so với ngắt theo mức 	<ul style="list-style-type: none"> - Ngắt theo mức xảy ra khi tín hiệu đầu vào được giữ ở mức cao hoặc mức thấp trong một khoảng thời gian dài. Điều này cho phép ngắt được kích hoạt khi một tín hiệu ở mức nhất định được giữ trong khoảng thời gian xác định. - Ngắt theo mức có thể giảm thiểu sự động thái không mong muốn của ngắt do tín hiệu đầu vào có thể bị nhiễu hoặc dao động.