|  |
| --- |
| PRACTICE EXERCISES OF THE MICROPROCESSORS & MICROCONTROLLERS  Instructor: The Tung Than  Student's name: Lê Hữu Đạt  Student code: 21520697 |

**PRACTICE REPORT NO #2:**

COMMUNICATION WITH 7-SEGMENT LED AND TIMER

1. **Content 1:** **Design result**

A screenshot of a computer

Description automatically generated with low confidence

1. **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/1XpJhpWYC59Ek2WYLrkRhrCSM-ytdwdIi/view?usp=sharing

- Lưu đồ thuật toán quét LED 7 đoạn:

Diagram, schematic

Description automatically generated

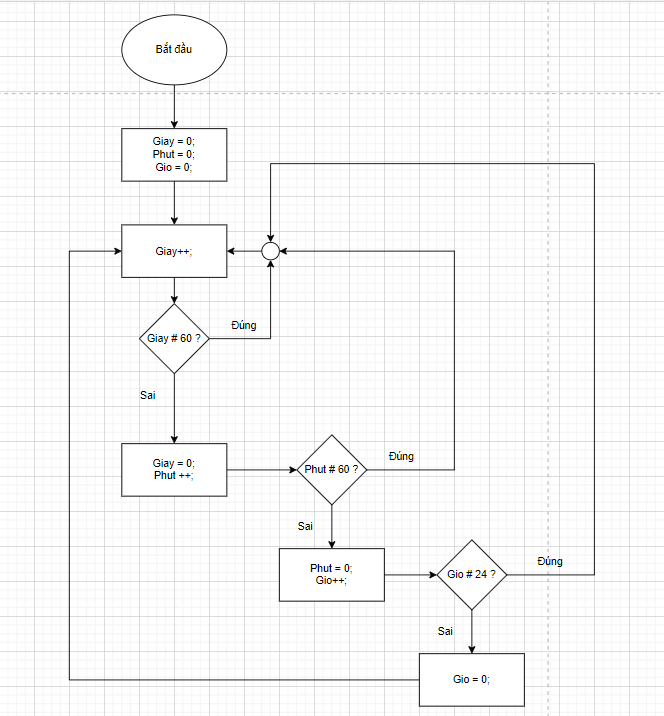
- Nguyên lí hoạt động của mạch:

+ Giây (thanh ghi R2) đếm từ 0 đến 59, khi quá 59 thì phút sẽ cộng thêm 1 đơn vị, giây sẽ reset về 0

+ Phút (thanh ghi R1) đếm từ 0 đến 59, khi quá 59 thì giờ sẽ cộng thêm 1 đơn vị, giây và phút sẽ reset về 0

+ Giờ (thanh ghi R0) đếm từ 0 đến 23 khi quá 23 thì giờ, phút và giây sẽ reset về 0

- Lưu đồ thuật toán của mạch:



|  |  |
| --- | --- |
| **Code** | **Giải thích** |
| ORG 00H  MOV R0, #0  MOV R1, #0  MOV R2, #0  MAIN:  CALL CLOCK  JMP MAIN  CLOCK:  CALL HIENTHI  INC R2  CJNE R2, #60, CLOCK    INC R1  MOV R2, #0  CJNE R1, #60, CLOCK    INC R0  MOV R1, #0  CJNE R0, #24, CLOCK    MOV R0, #0  RET  HIENTHI:  MOV DPTR, #MALED  MOV R3, #50  HT1:  MOV A, R0  MOV B, #10  DIV AB  MOVC A, @A + DPTR  MOV P0, A  SETB P3.0  CALL TIMER  CLR P3.0  MOV A, B  MOVC A, @A + DPTR  MOV P0, A  CLR P0.7  SETB P3.1  CALL TIMER  CLR P3.1  SETB P0.7  MOV A, R1  MOV B, #10  DIV AB  MOVC A, @A + DPTR  MOV P0, A  SETB P3.2  CALL TIMER  CLR P3.2  MOV A, B  MOVC A, @A + DPTR  MOV P0, A  SETB P3.3  CALL TIMER  CLR P3.3  DJNZ R3, HT1  RET  TIMER:  MOV TMOD, #01H  MOV TH0, #HIGH(-5000)  MOV TL0, #LOW(-5000)  SETB TR0  JNB TF0, $  CLR TR0  CLR TF0  RET  MALED: DB 0XC0, 0XF9, 0XA4, 0XB0, 0X99, 0X92, 0X82, 0XF8, 0X80, 0X90  END | Gán giá trị thanh ghi R0 = 0  Gán giá trị thanh ghi R1 = 0  Gán giá trị thanh ghi R2 = 0  Hàm MAIN  Gọi hàm CLOCK  Nhảy trở lại hàm MAIN  Hàm CLOCK  Gọi hàm HIENTHI  Tăng giá trị thanh ghi R2 lên 1 đơn vị  So sánh nếu thanh ghi R2 khác 60 thì quay lại hàm CLOCK  Tăng thanh ghi R1 lên 1 đơn vị  Gán thanh ghi R2 = 0  So sánh nếu thanh ghi R1 khác 60 thì quay lại hàm CLOCK  Tăng thanh ghi R0 lên 1 đơn vị  Gán thanh ghi R1 = 0  So sánh nếu thanh ghi R0 khác 24 thì quay lại hàm CLOCK  Gán thanh ghi R0 = 0  Thoát khỏi hàm CLOCK  Hàm HIENTHI  Gán thanh ghi DPTR = địa chỉ mảng MALED  Gán R3 = 50  Hàm HT1  Gán A = R0  Gán B =10  A = A/B và B = A%B  Gán A = A + DPTR  Gán P0 = A  Gán P3.0 = 1 để sáng LED hàng chục của giờ  Gọi hàm TIMER  Gán P3.0 = 0 để tắt LED hàng chục của giờ  Gán A = B  Gán A = A + DPTR  Gán P0 = A  Gán P0.7 = 0 để sáng dấu chấm phân biệt giờ và phút  Gán P3.1 = 1 để sáng LED hàng đơn vị của giờ  Gọi hàm TIMER  Gán P3.1 = 0 để tắt LED hàng đơn vị của giờ  Gán P0.7 = 1 để tắt dấu chấm phân biệt giờ và phút  Gán A = R1  Gán B =10  A = A/B và B = A%B  Gán A = A + DPTR  Gán P0 = A  Gán P3.2 = 1 để sáng LED hàng chục của phút  Gọi hàm TIMER  Gán P3.2 = 0 để tắt LED hàng chục của phút  Gán A = B  Gán A = A + DPTR  Gán P0 = A  Gán P3.3 = 1 để sáng LED hàng đơn vị của phút  Gọi hàm TIMER  Gán P3.3 = 0 để tắt LED hàng đơn vị của phút  Giảm R3 xuống 1 đơn vị và nhảy đển HT1 nếu R3#0  Thoát khỏi hàm HIENTHI  Hàm TIMER  Chọn chế độ TIMER 0 chế độ 1  Nạp giá trị cho TH0  Nạp giá trị cho TL0  Gán TR0 = 1  Nhảy tại chỗ nếu TF0 = 0  Gán TR0 = 0  Gán TF0 = 0  Thoát khỏi hàm TIMER  Mảng 1 chiều chứa mã LED 7 đoạn chung anode  Kết thúc |

1. **Content 3:** **With the above clock design, use a loop to create a delay instead of Timer. State the advantages and disadvantages of the two methods.**

|  |  |  |
| --- | --- | --- |
|  | **Ưu điểm** | **Nhược điểm** |
| **Sử dụng Timer** | - Tính chính xác cao trong việc tạo độ trễ.  - Có thể thực hiện các tác vụ khác trong khi Timer đang chạy. | - Cần sử dụng thư viện bổ sung.  - Phụ thuộc vào hệ thống.  - Không dễ hiểu và triển khai bằng với sử dụng vòng lặp. |
| **Sử dụng Delay** | - Dễ dàng hiểu và triển khai.  - Không cần thư viện bổ sung.  - Không phụ thuộc vào hệ thống. | - Không đảm bảo tính chính xác trong việc tạo ra độ trễ vì tốc độ thực hiện của vòng lặp có thể khác nhau trên các hệ thống khác nhau hoặc trong các điều kiện khác nhau.  - Không thể thực hiện các tác vụ khác trong khi vòng lặp đang chạy. |