

THỰC HÀNH VI XỬ LÝ – VI ĐIỀU KHIỂN

GVHD: Trần Hoàng Lộc

Họ và tên sinh viên thực hiện: Đoàn Vũ Phú Minh

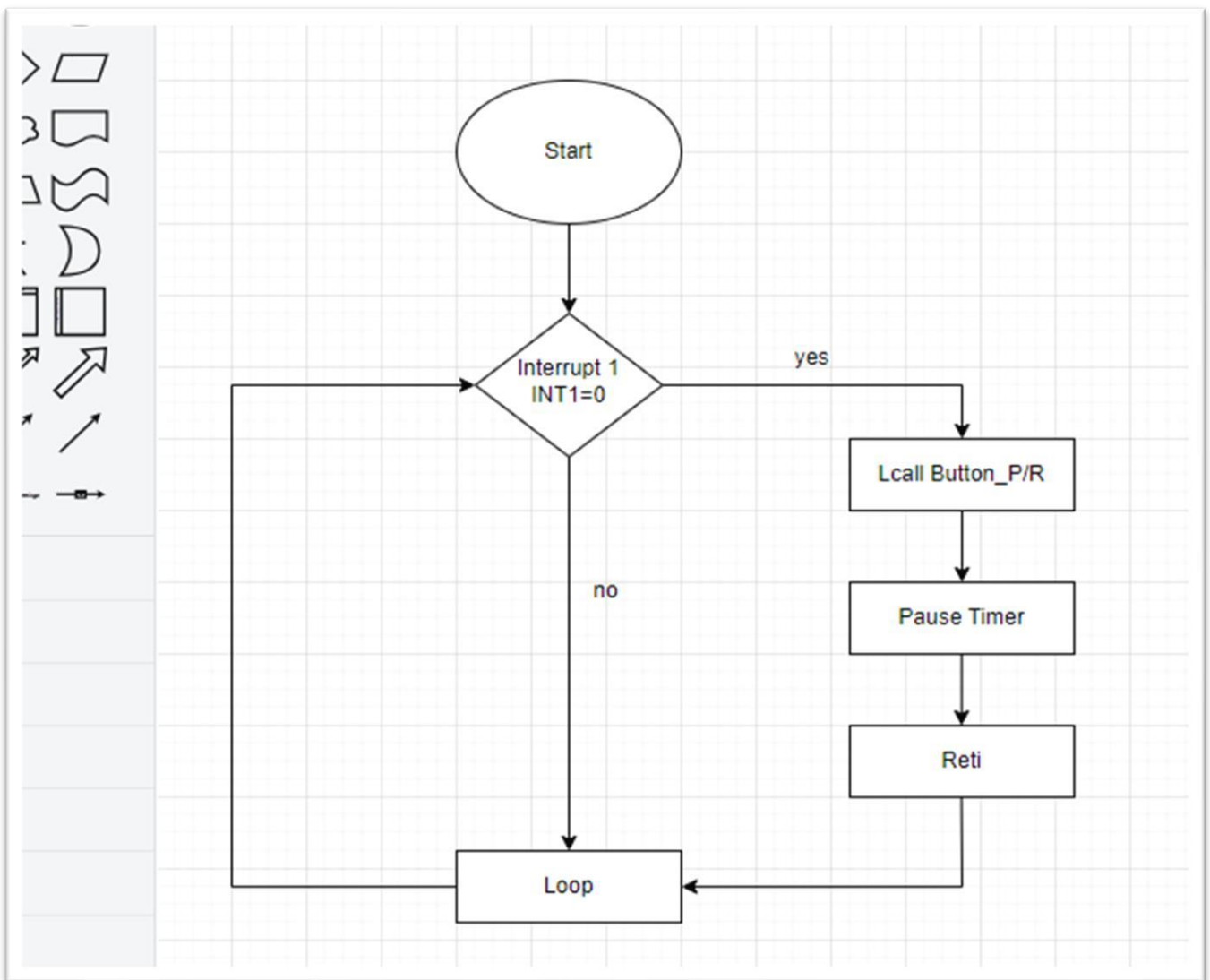
Mã số sinh viên: 22520859

I. PRACTICE CONTENT

1.1 Present and draw a flowchart to handle 2 buttons with the following functions.

BUTTON A

FlowChart Button A – Pause/Resume



Source code – Explain

```

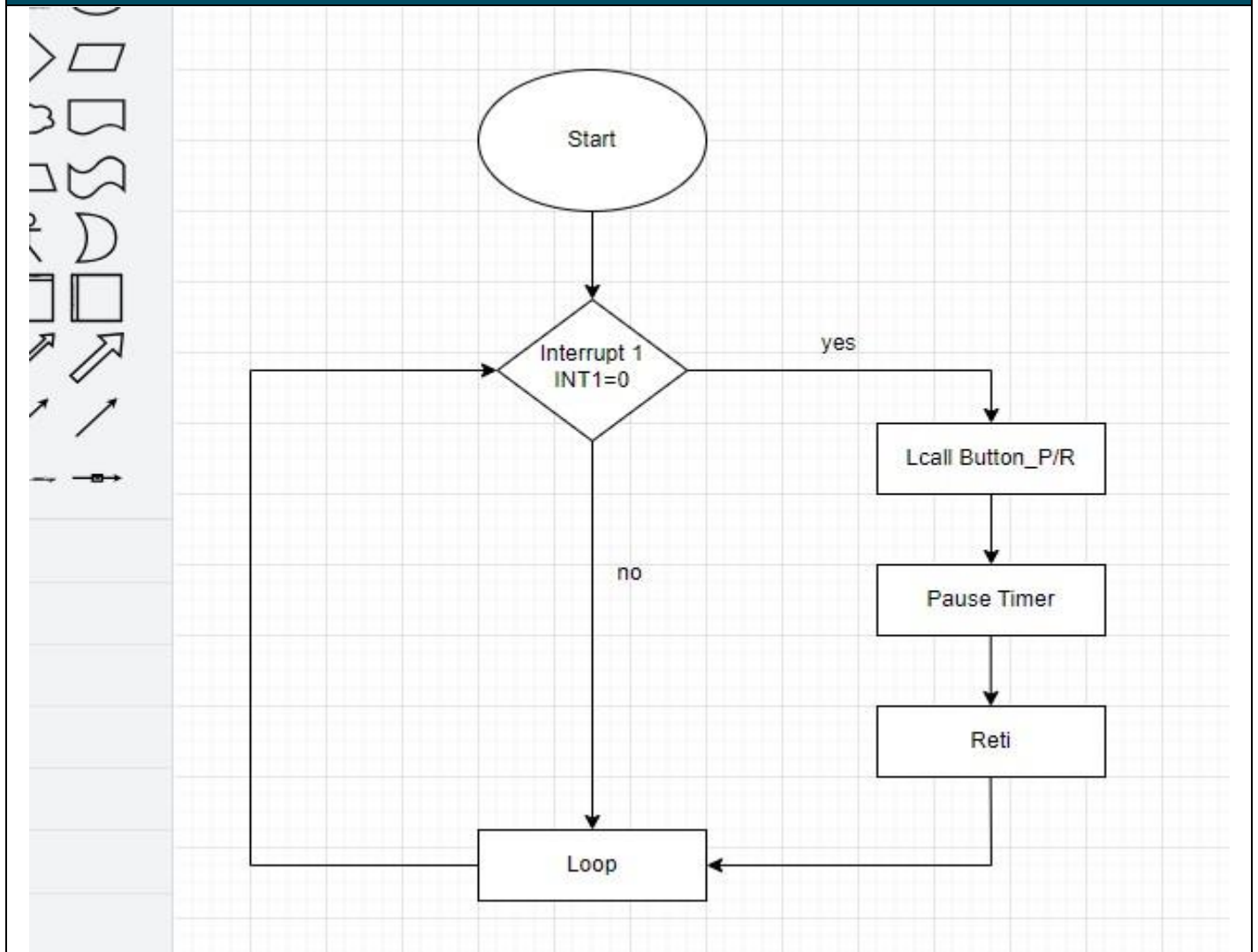
Ngat_INT1:
    inc R4
    cjne R4,#010b,throat
    mov R4,#00d
    mov th0,#0D8h
    mov tl0,#0F0h
    setb TR0
    reti
throat:
    clr TR0
    clr TF0
    acall delayed
    acall delayed
    acall delayed
    reti
30      ljmp ngat_timer ; dùng ng?t timer d? t?o delay
31      org 0013h
32      ljmp Ngat_INT1
33      org 0003h
34      ljmp Ngat_INT0
35

```

- Sử dụng cổng INT1 để thực hiện ngắt cho chức năng Pause và Resume, khi Button A được nhấn thì hàm thực hiện ngắt sẽ được thực thi. Hàm thực thi ngắt được khai báo thông qua (org 0013h). Nếu như button A được nhấn thì đồng hồ sẽ tạm ngưng hoạt động.
- Sử dụng thanh ghi R4 để xác định button nhấn được dừng lại hay tiếp tục chạy.
- Thực hiện hàm delay thời gian bằng cách thiết lập Timer0 với các thông số thực hiện Timer với thời gian là 1% của 1s.
- Nếu như button được nhấn là A thì nó sẽ chạy vào lable throat. Ta thực hiện xóa các cờ của Timer để tạm ngưng Timer, sau đó dùng 1 vài lệnh delayed nhỏ để tránh giật.

BUTTON B

FlowChart Button B – Reset



Source code – Explain

```

165
166
167 Ngat_INT0:
168     mov mgiay,#0
169     mov giay,#0
170     reti
171
172
30     ljmp ngat_timer ; dùng ng?t timer d? t?o delay
31     org 0013h
32     ljmp Ngat_INT1
33     org 0003h
34     ljmp Ngat_INT0
35

```

- Sử dụng cổng INT0 để thực hiện ngắt khi Reset xảy ra. Hàm ngắt được khai báo sau câu lệnh (org 0003h).
- Ta thực hiện cài đặt giá trị cho mgiay và giay trở về 0 tương tự như việc reset chương trình.

1.2 Using AT89C51/AT89C52 in combination with 4 7Segment LED modules and 2 buttons above.

Design a Sport clock circuit with the ability to count accurately to 1% of seconds, counting range from 00.00 seconds to 99.99 seconds and has 2 buttons to control Pause/Resume and Reset.

Source code – Explain

```

40     org 0100h
41 Start:
42     ; Write your code here
43     ; Thiết lập các tham số cần thiết cho INTO INT1 và Timer0
44     mov TMOD,#01h
45     setb TCON.2
46     setb TCON.3 ; dùng cho INT1
47     setb TCON.0
48     setb TCON.1 ; dùng cho INT0
49     mov ie,#10000111b
50     mov th0,#008h
51     mov tl0,#0F0h
52     ;=====
53     ; Cài đặt thanh ghi R4 để xác định điều kiện cho các button
54     mov R4,#0d
55     setb TR0
56     setb P1.0
57

```

- Dùng Timer 0 để thực hiện cài đặt giá trị đếm cho Timer tương ứng là 1ms.
- Ta thực hiện cài đặt các giá trị của thanh ghi TMOD, có M0 và M1 tương ứng là 1 và 0 tương ứng chế độ 1. Sau đó cài đặt các cờ tương ứng như Setb TR0 để chạy Timer. Những lệnh khác nằm ngoài việc cài đặt Timer sẽ được giải thích phía dưới.

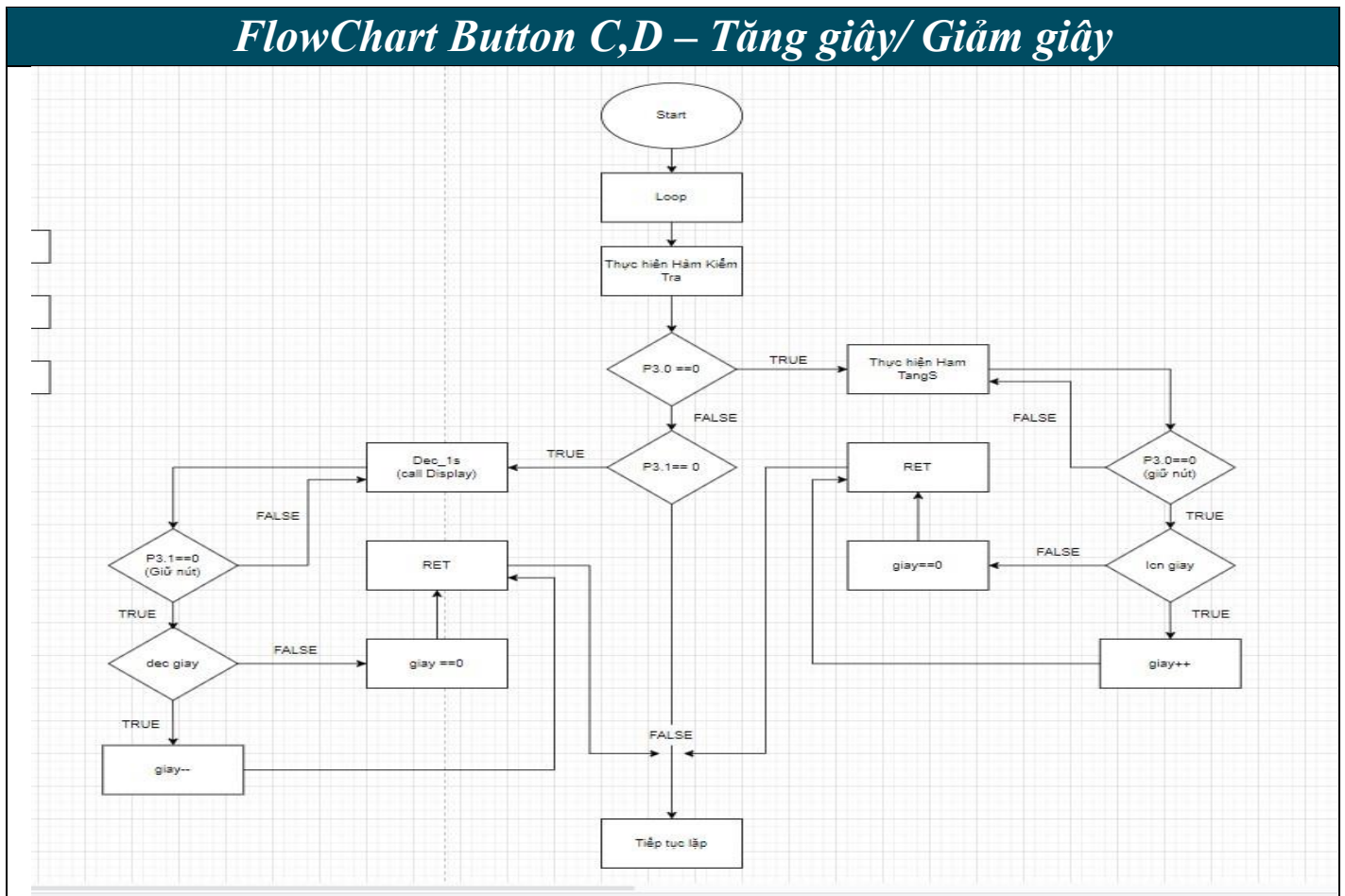
II. EXCERCIES

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

BUTTON C

FlowChart Button C,D – Tăng giây/ Giảm giây



Source code – Explain

```

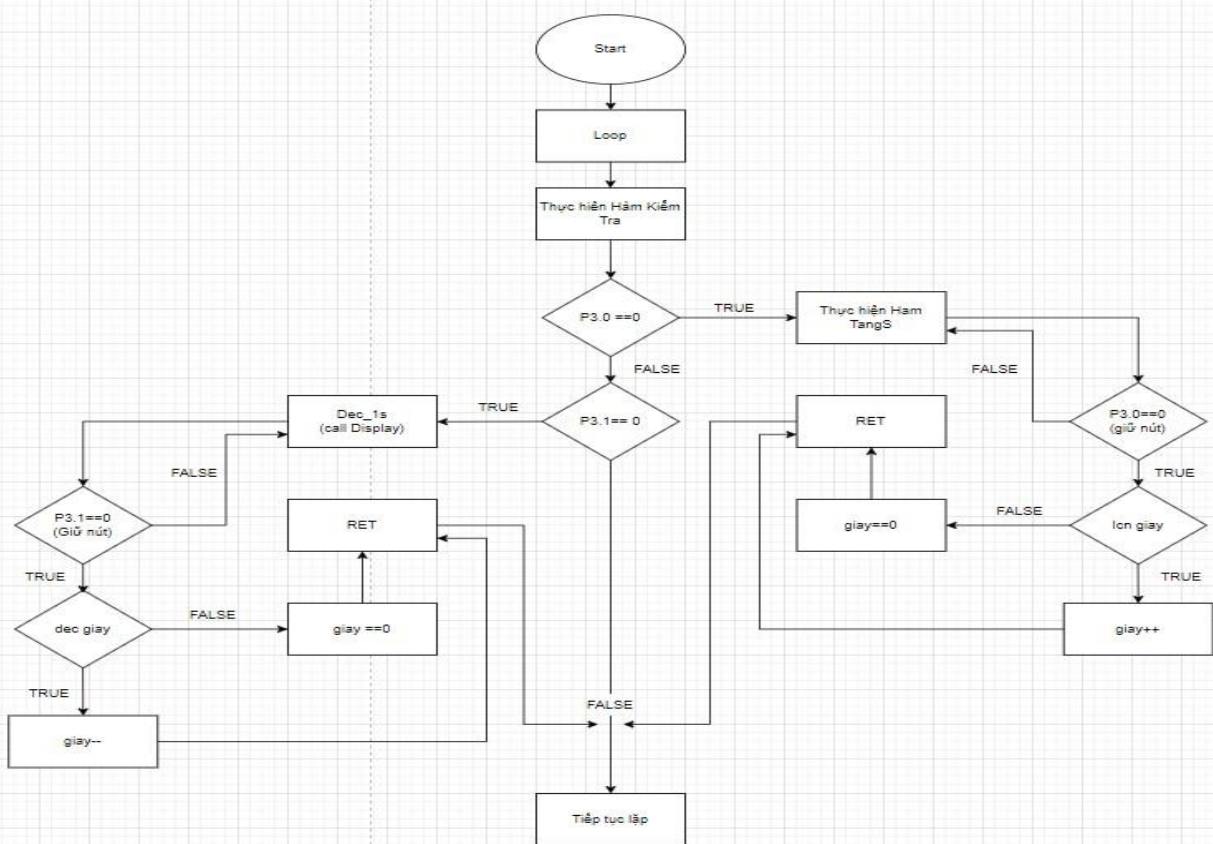
73         reti
74
75 HamKiemTra:
76     jnb P3.0,TangS
77     jnb P3.1,GiamS
78     reti
79
145     reti
146
147 TangS:
148     acall Hamhienthi
149     jnb P3.0,TangS
150     inc giay
151     cjne giay,#60,throat1
152     mov giay,#0
153 throat1:
154     reti
155
156
157 GiamS:
158     acall Hamhienthi
159     jnb P3.1,GiamS
160     dec giay
161     cjne giay,#0,throat2
162     mov giay,#0
163 throat2:
164     reti
165

```

- Sử dụng button để cài giá trị vào cổng P3.0 và P3.1 tương ứng là Tăng giây và giảm giây - TangS: sẽ thực hiện tăng biến giây lên 1 nếu như P3.0 bằng giá trị là 0. Ta tránh việc giữ button quá lâu để biến giây tăng không kiểm soát. Thực hiện dùng lệnh jnb kiểm tra liên tục nếu như chưa nhả ra thì vẫn chỉ hiển thị khi nhả ra rồi mới thực hiện tăng giá trị giây lên .
- Thực hiện các hàm kiểm tra để đảm bảo giá trị giây không vượt quá 60.

BUTTON D

FlowChart Button C,D – Tăng giây/ Giảm giây



Source code – Explain


```

73     reti
74
75 HamKiemTra:
76     jnb P3.0,TangS
77     jnb P3.1,GiamS
78     reti
79
145     reti
146
147 TangS:
148     acall Hamhienthi
149     jnb P3.0,TangS
150     inc giay
151     cjne giay,#60,throat1
152     mov giay,#0
153 throat1:
154     reti
155
156
157 GiamS:
158     acall Hamhienthi
159     jnb P3.1,GiamS
160     dec giay
161     cjne giay,#0,throat2
162     mov giay,#0
163 throat2:
164     reti
165

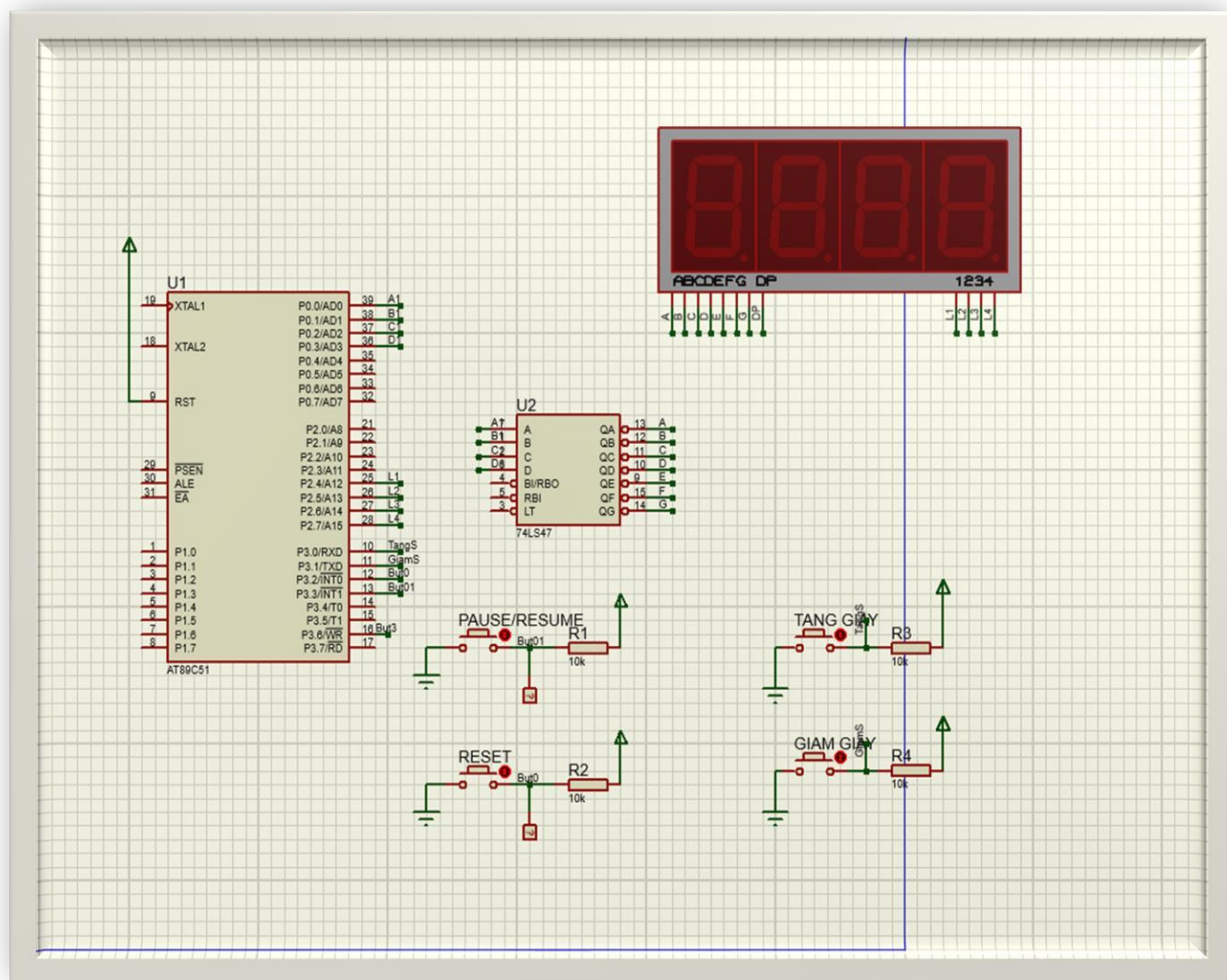
```

-Sử dụng button để cài giá trị vào cổng P3.0 và P3.1 tương ứng là Tăng giây và giảm giây - GiamS: thực hiện kiểm tra giá trị P3.1 để thực hiện giảm giá trị biến giây đi 1 đơn vị. Cũng tương tự với việc TangS thì ta vẫn phải kiểm tra việc giữ nút bấm của người dùng bằng lệnh jnb để đảm bảo khi nhả nút thì ta sẽ thực hiện giảm giá trị giây đi 1 đơn vị.

- Thực hiện các lệnh kiểm tra tính hợp lệ của giá trị giây khi giây đạt giá trị 0 thì trả về Reset.

III. REPORT

1.1 Design result (screenshot and pasted in the report).



GIẢI THÍCH CHI TIẾT TOÀN SOURCE CODE

Source code – Explain

```

21 ;=====
22 ; RESET and INTERRUPT VECTORS
23 ;=====
24 ; Reset Vector
25 org 0000h
26 mov mgiay,#0
27 mov giay,#0
28 jmp Start
29 org 000bh
30 ljmp ngat_timer ; dùng ngắt timer để tạo delay
31 org 0013h
32 ljmp Ngat_INT1
33 org 0003h
34 ljmp Ngat_INT0
35

```

- Cài đặt giá trị ban đầu cho mgiay và giay là 0.
- Thực hiện khai báo hàm ngắt Timer 0 bằng lệnh org 000bh và ljmp ngat_timer
- Thực hiện khai báo hàm ngắt ngoài INT0 bằng lệnh org 0003h và ljmp Ngat_INT0
- Thực hiện khai báo hàm ngắt ngoài INT1 bằng lệnh org 0013h và ljmp Ngat_INT1

Source code – Explain

```

40      org    0100h
41  Start:
42      ; Write your code here
43      ; Thiết lập các tham số cần thiết cho INTO INT1 và Timer0
44      mov    TMOD,#01h
45      setb   TCON.2
46      setb   TCON.3 ; dùng cho INT1
47      setb   TCON.0
48      setb   TCON.1 ; dùng cho INT0
49      mov    ie,#10000111b
50      mov    th0,#0D8h
51      mov    tl0,#0F0h
52      ;=====
53      ; Cài đặt thanh ghi R4 để xác định điều kiện cho các button
54      mov    R4,#0d
55      setb   TR0
56      setb   P1.0

```

-Đoạn chương trình main thực hiện:

- Gán giá trị cho thanh ghi TMOD #01h tương ứng thực hiện Timer 0 chế độ 1.
- Thực hiện setb cho các bit địa chỉ trong thanh ghi TCON với TCON.0, TCON.1 tương ứng là IT0 và IE0 để cho phép dùng ngắt ngoài cổng INT0. Tương tự là TCON.2, TCON.3 là IT1 và IE1 để cho phép thực hiện ngắt ngoài cổng INT1.
- Thực hiện gán giá trị cho thanh ghi IE để cho phép ngắt bao gồm set các bit tương ứng là EA: cho phép toàn bộ ngắt, EX1,EX0 cho phép ngắt ngoài từ INT1 và INT0 và ET0 cho phép ngắt Timer0.
- Cài đặt TL0 và TH0 các giá trị tương ứng để thực hiện Timer 0 với thời gian đếm là 1ms.
- Gán giá trị cho thanh ghi R4 để thực hiện điều hướng việc lựa chọn Pause hay Resume.
- Thực hiện setb TR0 để bắt đầu đếm Timer.

Source code – Explain

```

58
59  nhay:
60      acall HamKiemTra
61      acall Hamhienthi
62      cjne R4,#0d,nhay
63      cjne mgiay,#99,nhay ; so sanh coi giay co bang 60 chua ; Luc nay la Lay R0 ne
64      inc giay
65      mov mgiay,#0 ; sau khi tang phut thi cai dat lai giay
66      cjne giay,#60,nhay
67      mov giay,#0
68      sjmp $
69
70  Hamhienthi:
71      acall chuyenbcd ; co the thuc hien bang 1 IC 747
72      acall hienthi
73      reti
74
75  HamKiemTra:
76      jnb P3.0,TangS
77      jnb P3.1,GiamS
78      reti

```

```

81 hienthi:
82 mov R3, #10000000b ; 80h bang 0100 0000b
83 mov R0, #30h ; hien thi ham don vi cua giay ; o the gan truc tiep bang 1 gia tri
84 ht: mov dptr, #maled ; lay con tro dau tien dua vao dptr
85 mov a, @R0 ; lay gia tri cua R0 luu tai 30h gui vao a
86 movc a, @a+dptr ; vi du gia tri cua a la 5 thi dptr se lay gia tri thu 5 trong mang ung
87 mov P0, a
88 mov P2, R3
89 acall delayed ; cai nay de quet led
90 mov p2, #00h;
91 inc R0
92 mov a, R3 ; dua gia tri R3 vao thanh ghi a
93 rr a ; tuc la xoay phai a tuc la dich sang phai 1 bit tat ca ; luc nay a = 0010 0000b
94 mov R3, a
95 cjne a, #00001000b, ht
96 ret
97
98
99 chuyenbcd: ; chia lay du neu giay lon hon 10
100 push acc
101 mov a, mgiay
102 mov b, #10
103 div ab
104 mov 31h, a
105 mov 30h, b
106
107 mov a, giay
108 mov b, #10
109 div ab
110 mov 33h, a
111 mov 32h, b
112 pop acc
113 ret

```

-Thực hiện HamKiemTra để xem có button C hoặc D đã được ấn chưa. Nếu như button Pause được bấm thì R4 sẽ tăng giá trị lên 1 lúc đó hàm chỉ có thể gọi đến HamKiemTra và Hamhienthi và ngưng thực hiện việc tăng giá trị thông qua Timer0.

- Thực hiện kiểm tra giá trị ngưỡng cho giay và mgiay nếu như vượt ngưỡng sẽ cài đặt về giá trị mặc định là 0.

- Trong Hamhienthi ta thực hiện giá trị đơn vị tính được gán trong thanh ghi #30h để gán vào R0 sau đó thực hiện trả tương ứng vào mảng để lấy ra giá trị tương ứng thể hiện (Tương tự bài thực hành Lab02).

Source code – Explain

```
130
131 Ngat_INT1:
132     inc R4
133     cjne R4,#010b,throat
134     mov R4,#00d
135     mov th0,#0D8h
136     mov tl0,#0F0h
137     setb TR0
138     reti
139 throat:
140     clr TR0
141     clr TF0
142     acall delayed
143     acall delayed
144     acall delayed
145     reti
146
147 TangS:
148     acall Hamhienthi
149     jnb P3.0,TangS
150     inc giay
151     cjne giay,#60,throat1
152     mov giay,#0
153 throat1:
154     reti
155
156
157 GiamS:
158     acall Hamhienthi
159     jnb P3.1,GiamS
160     dec giay
161     cjne giay,#0,throat2
162     mov giay,#0
163 throat2:
164     reti
165
166
167 Ngat_INT0:
168     mov mgiay,#0
169     mov giay,#0
170     reti
```

-Hàm Ngat_INT1: tăng giá trị R4 lên 1 để kiểm tra xem đó là thực hiện Button A hay là B. Nếu là A thì nhảy vào throat thực hiện dùng đếm Timer0.

- Nếu là Button B thì thực hiện cài đặt lại giá trị cho R4 là 0 thực hiện cài lại giá trị cho thanh ghi TL0 và TH0 để thực hiện tiếp tục tăng giá trị cho đồng hồ.

- Ham TangS: thực hiện Button C khi được nhấn sẽ kiểm tra P3.0 và tăng giay lên 1 đơn vị.

- Ham GiamS: thực hiện Button C khi được nhấn sẽ kiểm tra P3.0 và tăng giay lên 1 đơn vị.

