

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI



BÁO CÁO CUỐI KỲ
MÔN THỰC HÀNH KIẾN TRÚC MÁY TÍNH
KỲ 20232

Giảng viên hướng dẫn: TS. Lê Bá Vui

Sinh viên:

Nguyễn Đức Tuân - 20184217

Nguyễn Thành Vinh – 20225779

Hà Nội, tháng ... ngày ... năm 2024

MỤC LỤC

- 1. Phân chia công việc**
- 2. Project 10: Máy tính bỏ túi**
- 3. Project 3: Kiểm tra tốc độ và độ chính xác khi gõ văn bản**

1. Phân chia công việc

- Nguyễn Đức Tuấn - 20184217: Project 10
- Nguyễn Thành Vinh - 20225779: Project 3
- Công cụ sử dụng : Mars4_5

2. Project 10: Máy tính bỏ túi

1.1 Hướng dẫn chạy chương trình

- Mở công cụ Digital Lab Sim, Connect to MIPS và chạy chương trình.
- Nhập các số bằng cách nhập từ các phím trên công cụ Digital Lab Sim và màn hình led sẽ hiển thị 2 chữ số cuối.
- Nhập xong số đầu tiên thì nhập toán tử cộng, trừ, nhân, chia tương ứng với a, b, c, d, e hiện trên phím của công cụ Digital Lab Sim.
- Nhập số thứ hai.
- Nhấn f (tương đương dấu '=') để hiển thị 2 chữ số cuối cùng của kết quả.

1.2 Phân tích cách thức thực hiện

- Nhập từ bàn phím ảo của công cụ Digital Lab Sim và xác định mã của phím vừa nhận.
- Thực hiện kiểm tra:
 - Nếu đó là một số: update lại giá trị tạm thời, và giá trị của số đó sẽ xuất hiện trên led 7 đoạn của công cụ Digital Lab Sim.
 - Nếu đó là một chữ (là một toán tử): Lưu lại toán tử đó (cộng, trừ, nhân, chia lấy phần nguyên, chia lấy phần dư) và chuyển sang nhập số thứ hai.
- Nếu toán tử nhập vào là '=' (dấu bằng) -> hiển thị kết quả ra màn hình trên thanh led 7 đoạn của Digital Lab Sim.

- Những lưu ý:
 - Khi chọn toán tử là ‘/’ (phép chia lấy phần nguyên) hoặc ‘%’ (phép chia lấy phần dư) nếu số hạng thứ 2 bằng 0 thì in ra: **‘Kết quả không tồn tại. Không thể chia cho số 0’**. Đồng thời nhập lại số hạng thứ 2 để thực hiện phép tính.
 - Nếu nhập 1 lúc 2 toán tử mà không nhập số hạng thứ 2 thì chương trình sẽ coi như số hạng thứ 2 bằng 0 và thực hiện phép tính số hạng thứ nhất với toán tử đầu tiên được nhập vào.
 - Không thể nhập cùng một lúc 2 toán tử để tính toán (ví dụ: nhập dấu ‘*’ sau đó nhập luôn dấu ‘-’ thì kết quả sẽ ra như lưu ý thứ 2).
 - Hiện tại thì phép tính chưa thể xử lý nhiều toán hạng cùng một lúc (một phép tính chỉ có thể có một toán tử và 2 số hạng).
 - Trong trường hợp phép trừ, nếu số hạng thứ nhất nhỏ hơn số hạng thứ hai thì kết quả sẽ ra âm. Khi đó kết quả sẽ không hiển thị trên led 7 đoạn của công cụ Digital Lab Sim.

1.3 Code

- Khởi tạo

```

3
4 .eqv SEVENSEG_LEFT 0xFFFF0011 # Dia chi cua den led 7 doan trai.
5 # bit 0 = doan a;
6 # bit 1 = doan b;
7 # bit 7 = dau .
8 .eqv SEVENSEG_RIGHT 0xFFFF0010 # Dia chi cua den led 7 doan phai.
9 .eqv IN_ADDRESS_HEX_KEYBOARD 0xFFFF0012
10 .eqv OUT_ADDRESS_HEX_KEYBOARD 0xFFFF0014
11
12 .data
13 # ma hoa cua Digital Lab Sim
14 zero: .byte 0x3f
15 one: .byte 0x6
16 two: .byte 0x5b
17 three: .byte 0x4f
18 four: .byte 0x66
19 five: .byte 0x6d
20 six: .byte 0x7d
21 seven: .byte 0x7
22 eight: .byte 0x7f
23 nine: .byte 0x6f
24
25 mess1: .ascii "Ket qua khong ton tai.Khong the chia cho so 0 \n"
26
27 .text
28 main:
29
30 Khoi_tao:
31 li $t0, SEVENSEG_LEFT #Bien gia tri so cua den LED trai
32 li $t5, SEVENSEG_RIGHT #Bien gia tri so cua den LED phai
33 li $s0, 0 #Bien kiem tra loai bien nhap vao: (0: so), (1 :toan tu), (2: terminate key)
34 li $s1, 0 #so hien thi o led phai
35 li $s2, 0 #so hien thi o led trai
36 li $s3, 0 #bien kiem tra loai toan tu (1: +, 2: -, 3: *, 4: /, 5: %)
37 li $s4, 0 #so thu nhat
38 li $s5, 0 #so thu hai
39 li $s6, 0 #ket qua phep tinh: '+', '-', '*', '/', '%'
40 li $t9, 0 #gia tri tam thoi
41
42 #gia tri cua led
43 li $t1, IN_ADDRESS_HEX_KEYBOARD #bien dieu khien hang keyboard va enable keyboard interrupt
44 li $t2, OUT_ADDRESS_HEX_KEYBOARD #bien chua vi tri key nhap vao the hang va cot
45 li $t3, 0x80 # bit dung enable keyboard interrupt va enable kiem tra tung hang keyboard
46 sb $t3, 0($t1)
47 li $t7, 0 #gia tri cua so hien tren led
48 li $t4, 0 #byte hien thi len led, zero->nine
49
50 First_value:
51 li $t7, 0 #gia tri cua bit can hien thi ban dau
52 addi $sp, $sp, 4 #day vao stack
53 sb $t7, 0($sp)
54 lb $t4, zero #bit dau tien can hien thi
55 addi $sp, $sp, 4 #day vao stack
56 sb $t4, 0($sp)
57
58 Loop1:
59 nop
60 nop
61 nop
62 b Loop1 #Wait for interrupt
63
64 nop
65 nop
66 nop
67 b Loop1 #Wait for interrupt
68
69 nop
70 nop
71 nop
72 b Loop1 #Wait for interrupt
73
74 end_loop1:
75
76 end_main:
77 li $v0, 10
78 syscall
79

```

○ Kiểm tra xem số vừa được nhập vào

```

79 #-----
80 # Xu ly khi xay ra interupt
81 # Hien thi so vua bam len den led 7 doan
82 # Kiem tra tung hang xem co duoc bam hay khong
83 #-----
84
85 .ktext 0x80000180
86 #-----
87 # Processing
88 # Neu hang co phim duoc nhap -> chuyen toi hang do
89 #-----
90
91     jal    check_row1      #Kiem tra hang 1 xem co phim nao duoc nhap hay khong
92     bnez   $t3, convert_row1  #t3 != 0 -> co phim duoc nhap,kiem tra cac phim trong hang, lay phim do ra
93     nop
94
95     jal    check_row2      #Kiem tra hang 2 xem co phim nao duoc nhap hay khong
96     bnez   $t3, convert_row2  #t3 != 0 -> co phim duoc nhap,kiem tra cac phim trong hang, lay phim do ra
97     nop
98
99     jal    check_row3      #Kiem tra hang 3 xem co phim nao duoc nhap hay khong
100    bnez   $t3, convert_row3  #t3 != 0 -> co phim duoc nhap,kiem tra cac phim trong hang, lay phim do ra
101    nop
102
103    jal    check_row4      #Kiem tra hang 4 xem co phim nao duoc nhap hay khong
104    bnez   $t3, convert_row4  #t4 != 0 -> co phim duoc nhap,kiem tra cac phim trong hang, lay phim do ra
105
106 #-----
107 #Kiem tra tung hang mot xem phim duoc nhap o hang nao
108 #-----
109 #
110 check_row1:
111     addi   $sp, $sp, 4
112     sw     $ra, 0($sp)      #luu ra lai vi ve sau co the doi
113     li     $t3, 0x81        #Kich hoat interrupt, cho phep bam phim o hang 1
114     sb     $t3, 0($t1)
115     jal    Get_value        #lay vi tri cua phim duoc nhap (neu co)
116     lw     $ra, 0($sp)
117     addi   $sp, $sp, -4
118     jr     $ra
119
120 check_row2:
121     addi   $sp, $sp, 4
122     sw     $ra, 0($sp)      #luu ra lai vi ve sau co the doi
123     li     $t3, 0x82        #Kich hoat interrupt, cho phep bam phim o hang 2
124     sb     $t3, 0($t1)
125     jal    Get_value        #lay vi tri cua phim duoc nhap (neu co)
126     lw     $ra, 0($sp)
127     addi   $sp, $sp, -4
128     jr     $ra
129
130 check_row3:
131     addi   $sp, $sp, 4
132     sw     $ra, 0($sp)      #luu ra lai vi ve sau co the doi
133     li     $t3, 0x84        #Kich hoat interrupt, cho phep bam phim o hang 3
134     sb     $t3, 0($t1)
135     jal    Get_value        #lay vi tri cua phim duoc nhap (neu co)
136     lw     $ra, 0($sp)
137     addi   $sp, $sp, -4
138     jr     $ra
139

```

```

140 check_row4:
141     addi    $sp, $sp, 4
142     sw      $ra, 0($sp)           #luu ra lai vi ve sau co the doi
143     li      $t3, 0x88           #Kich hoat interrupt, cho phep bam phim o hang 4
144     sb      $t3, 0($t1)
145     jal     Get_value           #lay vi tri cua phim duoc nhap (neu co)
146     lw      $ra, 0($sp)
147     addi    $sp, $sp, -4
148     jr      $ra
149
150
151 #-----
152 # Lay gia tri cua phim vua duoc nhap
153 #-----
154 Get_value:
155     addi    $sp, $sp, 4
156     sw      $ra, 0($sp)
157     li      $t2, OUT_ADDRESS_HEX_KEYBOARD #dia chi chua vi tri phim duoc nhap
158     lb      $t3, 0($t2)         #load vi tri phim duoc nhap
159     lw      $ra, 0($sp)
160     addi    $sp, $sp, -4
161     jr      $ra
162
163 #-----
164 #Convert tu vi tri sang bit
165 #-----
166
167 #-----
168 #Convert hang 1: co 4 gia tri: 0, 1, 2, 3
169 #Ma hoa tuong ung la:  zero: .byte 0x3f
170 #                      one:  .byte 0x6
171 #                      two:  .byte 0x5b
172 #                      three: .byte 0x4f
173 #-----
174
175 convert_row1:
176     beq     $t3, 0x11, case_0     # 0x11 -> so 0
177     beq     $t3, 0x21, case_1     # 0x21 -> so 1
178     beq     $t3, 0x41, case_2     # 0x41 -> so 2
179     beq     $t3, 0xfffff81, case_3 # 0xfffff81 -> so 3
180 case_0:
181     lb      $t4, zero             #t4 = zero (Ma hoa cua '0' tren Digital Lab Sim)
182     li      $t7, 0               #t7= 0
183     j       update_tg
184 case_1:
185     lb      $t4, one              #t4 = one (Ma hoa cua '1' tren Digital Lab Sim)
186     li      $t7, 1               #t7 = 1
187     j       update_tg
188 case_2:
189     lb      $t4, two              #t4 = two (Ma hoa cua '2' tren Digital Lab Sim)
190     li      $t7, 2               #t7 = 2
191     j       update_tg
192 case_3:
193     lb      $t4, three            #t4 = three (Ma hoa cua '3' tren Digital Lab Sim)
194     li      $t7, 3               #t7 = 3
195     j       update_tg
196

```

```

200
201
202
203
204
205 convert_row2:
206     beq    $t3, 0x12, case_4          # 0x12 -> so 4
207     beq    $t3, 0x22, case_5          # 0x22 -> so 5
208     beq    $t3, 0x42, case_6          # 0x42 -> so 6
209     beq    $t3, 0xfffff82, case_7      # 0xfffff82 -> so 7
210 case_4:
211     lb     $t4, four                  #t4 = four (Ma hoa cua '4' tren Digital Lab Sim)
212     li     $t7, 4                     #t7= 4
213     j      update_tg
214 case_5:
215     lb     $t4, five                  #t4 = five (Ma hoa cua '5' tren Digital Lab Sim)
216     li     $t7, 5                     #t7 = 5
217     j      update_tg
218 case_6:
219     lb     $t4, six                   #t4 = six (Ma hoa cua '6' tren Digital Lab Sim)
220     li     $t7, 6                     #t7 = 6
221     j      update_tg
222 case_7:
223     lb     $t4, seven                 #t4 = seven (Ma hoa cua '7' tren Digital Lab Sim)
224     li     $t7, 7                     #t7 = 7
225     j      update_tg
226
227
228 #Convert hang 3: co 4 gia tri: 8, 9, a, b
229 #Ma hoa tuong ung la:  eight: .byte 0x7f
230 #                      nine:  .byte 0x6f
231 #
232
233 convert_row3:
234     beq    $t3, 0x14, case_8          # 0x12 -> so 8
235     beq    $t3, 0x24, case_9          # 0x22 -> so 9
236     beq    $t3, 0x44, case_a          # 0x42 -> phim a
237     beq    $t3, 0xfffff84, case_b     # 0xfffff82 -> phim b
238 case_8:
239     lb     $t4, eight                 #t4 = eight (Ma hoa cua '8' tren Digital Lab Sim)
240     li     $t7, 8                     #t7= 8
241     j      update_tg
242 case_9:
243     lb     $t4, nine                  #t4 = nine (Ma hoa cua '9' tren Digital Lab Sim)
244     li     $t7, 9                     #t7 = 9
245     j      update_tg
246
247 #
248 #Truong hop phim a (phep cong)
249 #
250 case_a:
251     addi   $a3, $zero, 1
252     addi   $s0, $s0, 1                #s0 = 1 -> toan tu duoc nhap vao
253     bne    $s3, 0, set_next_operator
254     addi   $s3, $zero, 1              #s3 = 1 -> phep cong
255
256     j      set_first_number           #chuyen den ham chuyen 2 byte dang hien tren 2 led thanh so de tinh toan
257
258 #
259 #Truong hop phim b (phep tru)
260 #
261 case_b:
262     addi   $a3, $zero, 2
263     addi   $s0, $s0, 1                #s0 = 1 -> toan tu duoc nhap vao
264     bne    $s3, 0, set_next_operator
265     addi   $s3, $zero, 2              #s3 = 2 -> phep tru
266     j      set_first_number
267

```



```

268 #-----
269 #Convert hang 4: co 4 gia tri: c, d, e, f
270 #-----
271
272 convert_row4:
273     beq    $t3, 0x18, case_c        # 0x18 -> phim c
274     beq    $t3, 0x28, case_d        # 0x28 -> phim d
275     beq    $t3, 0x48, case_e        # 0x48 -> phim e
276     beq    $t3, 0xfffff88, case_f    # 0xfffff88 -> phim f
277
278 #-----
279 #Truong hop phim c (phep nhan)
280 #-----
281 case_c:
282     addi   $a3, $zero, 3
283     addi   $s0, $s0, 1              #s0 = 1 -> toan tu duoc nhap vao
284     bne    $s3, 0, set_next_operator
285     addi   $s3, $zero, 3            #s3 = 3 -> phep nhan
286
287     j      set_first_number         #chuyen den ham chuyen 2 byte dang hien tren 2 led thanh so de tinh toan
288
289 #-----
290 #Truong hop phim d (phep chia)
291 #-----
292 case_d:
293     addi   $a3, $zero, 4
294     addi   $s0, $s0, 1              #s0 = 1 -> toan tu duoc nhap vao
295     bne    $s3, 0, set_next_operator
296     addi   $s3, $zero, 4            #s3 = 4 -> phep chia layphan nguyen
297     j      set_first_number
298
299 #-----
300 #Truong hop phim e (phep %)
301 #-----
302 case_e:
303     addi   $a3, $zero, 5
304     addi   $s0, $s0, 1              #s0 = 1 -> toan tu duoc nhap vao
305     bne    $s3, 0, set_next_operator
306     addi   $s3, $zero, 5            #s3 = 5 -> chia layphan du
307     j      set_first_number         #chuyen den ham chuyen 2 byte dang hien tren 2 led thanh so de tinh toan
308
309

```

○ Thực hiện phép tính.

```

310 #-----
311 #Tinh so dau hien thi tren den led
312 #-----
313 set_first_number:
314     addi   $s4, $t9, 0
315     li     $t9, 0
316     j      done
317
318 #-----
319 #Truong hop phim f (=)
320 #-----
321 case_f:
322     addi   $s5, $t9, 0
323
324
325 #-----
326 #Tinh so thu hai hien thi tren den led trong 2 so
327 #-----
328 set_second_number:
329     beq    $s3, 1, phep_cong        # s3 = 1 -> cong
330     beq    $s3, 2, phep_tru        # s3 = 2 -> tru
331     beq    $s3, 3, phep_nhan        # s3 = 3 -> nhan
332     beq    $s3, 4, phep_chia        # s3 = 4 -> chia
333     beq    $s3, 5, phep_lay_du      # s3 = 5 -> lay du
334
335 phep_cong:
336     add    $s6, $s5, $s4
337     li     $s3, 0
338     li     $t9, 0
339     j      in_phep_cong
340     nop
341

```

```

342 #-----
343 # In ra dinh dang: a + b = c \n
344 #-----
345 in_phep_cong:
346     li      $v0, 1
347     move    $a0, $s4
348     syscall
349     li      $s4, 0          #reset $s4
350
351     li      $v0, 11
352     li      $a0, '+'
353     syscall
354
355     li      $v0, 1
356     move    $a0, $s5
357     syscall
358     li      $s5, 0          #reset $s5
359
360     li      $v0, 11
361     li      $a0, '='
362     syscall
363
364     li      $v0, 1
365     move    $a0, $s6
366     syscall
367     nop
368
369     li      $v0, 11
370     li      $a0, '\n'
371     syscall
372     li      $s7, 100
373     div     $s6, $s7
374     mfhi    $s6              #Lay 2 gia tri cuoi cua ket qua
375     j show_result_in_led    #Hien thi ket qua tren led
376     nop
377

```

```

422 phep_nhan:
423     mul    $s6, $s4, $s5
424     li     $s3, 0
425     li     $t9, 0
426     j      in_phep_nhan
427     nop
428
429     #-----
430     #In ra dinh dang: a*b = c \n
431     #-----
432 in_phep_nhan:
433     li     $v0, 1
434     move   $a0, $s4
435     syscall
436     li     $s4, 0           #reset $s4
437
438     li     $v0, 11
439     li     $a0, '*'
440     syscall
441
442     li     $v0, 1
443     move   $a0, $s5
444     syscall
445     li     $s5, 0           #reset $s5
446
447
448     li     $v0, 11
449     li     $a0, '='
450     syscall
451
452     li     $v0, 1
453     move   $a0, $s6
454     syscall
455
456     li     $v0, 11
457     li     $a0, '\n'
458     syscall
459
460     li     $s7, 100
461     div    $s6, $s7
462     mfhi   $s6              #Lay 2 chu so sau cung cua ket qua in ra
463     j      show_result_in_led #Hien thi ket qua tren led

```

```

377
378 phep_tru:
379     sub    $s6, $s4, $s5
380     li     $s3, 0
381     li     $t9, 0
382     j      in_phep_tru
383     nop
384
385 #-----
386 #In ra dinh dang: a - b = c \n
387 #-----
388 in_phep_tru:
389     li     $v0, 1
390     move   $a0, $s4
391     syscall
392     li     $s4, 0          #reset $s4
393
394     li     $v0, 11
395     li     $a0, '-'
396     syscall
397
398     li     $v0, 1
399     move   $a0, $s5
400     syscall
401     li     $s5, 0          #reset $s5
402
403
404     li     $v0, 11
405     li     $a0, '='
406     syscall
407
408     li     $v0, 1
409     move   $a0, $s6
410     syscall
411
412     li     $v0, 11
413     li     $a0, '\n'
414     syscall
415
416     li     $s7, 100
417     div    $s6, $s7
418     mfhi   $s6              #lay 2 dia tri cuoi cua ket qua
419
420
421
422
423
424
425
426
427
428
429 phep_chia:
430     beq     $s5, 0, chia_0 # kiem tra xem so chia co phai bang 0 hay khong
431     li     $s3, 0
432     div    $s4, $s5
433     mflo   $s6
434     mfhi   $s7
435     li     $t9, 0
436     j      in_phep_chia
437     nop
438
439 #-----
440 # Neu $s5 = 0 in ra: "Khong the chia cho so 0 \n"
441 #-----
442 chia_0:
443     li     $v0, 55
444     la     $a0, mess1
445     li     $a1, 0
446     syscall
447     j      reset_led
448
449
450

```

```

403
486 in_phep_chia:
487     li      $v0, 1
488     move    $a0, $s4
489     syscall
490     li      $s4, 0          #reset $s4
491
492     li      $v0, 11
493     li      $a0, '/'
494     syscall
495
496     li      $v0, 1
497     move    $a0, $s5
498     syscall
499     li      $s5, 0          #reset $s5
500
501
502     li      $v0, 11
503     li      $a0, '='
504     syscall
505
506     li      $v0, 1
507     move    $a0, $s6
508     syscall
509
510
511     li      $v0, 11
512     li      $a0, '\n'
513     syscall
514
515     li      $s7, 100
516     div     $s6, $s7
517     mfhi    $s6              #Lay 2 gia tri cuoi cua ket qua
518     j       show_result_in_led #Hien thi ket qua tren led
519     nop
520
---
522 phep_lay_du:
523     beq     $s5, 0, chia_lay_du_0 # kiem tra xem so chia co phai bang 0 hay khong
524     li      $s3, 0
525     div     $s4, $s5
526     mflo    $s7
527     mfhi    $s6
528     li      $t9, 0
529     j       in_phep_lay_du
530     nop
531
532 #-----
533 # Neu $s5 = 0 in ra: "Khong the chia cho so 0 \n"
534 #-----
535 chia_lay_du_0:
536     li      $v0, 55
537     la      $a0, mess1
538     li      $a1, 0
539     syscall
540     j       reset_led
541

```

```

541
542 in_phep_lay_du:
543     li    $v0, 1
544     move  $a0, $s4
545     syscall
546     li    $s4, 0           #reset $s4
547
548     li    $v0, 11
549     li    $a0, '%'
550     syscall
551
552     li    $v0, 1
553     move  $a0, $s5
554     syscall
555     li    $s4, 0           #reset $s4
556
557     li    $v0, 11
558     li    $a0, '='
559     syscall
560
561     li    $v0, 1
562     move  $a0, $s6
563     syscall
564
565
566     li    $v0, 11
567     li    $a0, '\n'
568     syscall
569
570     li    $s7, 100
571     div   $s6, $s7
572     mfhi  $s6
573     j     show_result_in_led   #Lay 2 gia tri cuoi cua ket qua
574     nop                                     #Hien thi ket qua tren led
575
576 #-----
577 #Hien thi ket qua tren den led
578 #So 'ab'
579 # Den trai = a = ab div 10
580 # Den phai = b = ab mod 10
581 #-----
582 show_result_in_led:
583     li    $t8, 10           # Gia tri trung gian = 10
584     div   $s6, $t8           # $s6 = a
585     mflo  $t7               # $t7 = result
586     jal   check             #chuyen den ham chuyen t7 thanh bit hien thi len led
587
588     sb    $t4, 0($t0)        # hien thi len led trai
589     add   $sp, $sp, 4
590     sb    $t7, 0($sp)        #day gia tri bit nay vao stack
591     add   $sp, $sp, 4
592     sb    $t4, 0($sp)        #day bit nay vao stack
593     add   $s2, $t7, $zero     #s1 = gia tri bit led phai
594
595     mfhi  $t7               #t7 = remainder
596     jal   check             #convert t7 thanh bit hien thi len led
597     sb    $t4, 0($t5)        #hien thi len led phai
598     add   $sp, $sp, 4
599     sb    $t7, 0($sp)        # day gia tri bit nay vao stack
600     add   $sp, $sp, 4
601     sb    $t4, 0($sp)        # day bit nay vao stack
602     add   $s1, $t7, $zero    # s1 = gia tri bit led phai
603     j     reset_led         # ham reset lai led
604

```

```

605 check:
606     addi    $sp, $sp, 4
607     sw      $ra, 0($sp)
608     beq     $t7, 0, check_0      # t7 = 0 -> Hien thi so 0 tren thanh led
609     beq     $t7, 1, check_1      # t7 = 1 -> Hien thi so 1 tren thanh led
610     beq     $t7, 2, check_2      # t7 = 2 -> Hien thi so 2 tren thanh led
611     beq     $t7, 3, check_3      # t7 = 3 -> Hien thi so 3 tren thanh led
612     beq     $t7, 4, check_4      # t7 = 4 -> Hien thi so 4 tren thanh led
613     beq     $t7, 5, check_5      # t7 = 5 -> Hien thi so 5 tren thanh led
614     beq     $t7, 6, check_6      # t7 = 6 -> Hien thi so 6 tren thanh led
615     beq     $t7, 7, check_7      # t7 = 7 -> Hien thi so 7 tren thanh led
616     beq     $t7, 8, check_8      # t7 = 8 -> Hien thi so 8 tren thanh led
617     beq     $t7, 9, check_9      # t7 = 9 -> Hien thi so 9 tren thanh led
618
619 #-----
620 #Chuyen gia tri thanh bit hien tren thanh led
621 #-----
622 check_0:
623     lb      $t4, zero
624     j       finish_check
625 check_1:
626     lb      $t4, one
627     j       finish_check
628 check_2:
629     lb      $t4, two
630     j       finish_check
631 check_3:
632     lb      $t4, three
633     j       finish_check
634 check_4:
635     lb      $t4, four
636     j       finish_check
637 check_5:
638     lb      $t4, five
639     j       finish_check
640 check_6:
641     lb      $t4, six
642     j       finish_check
643 check_7:
644     lb      $t4, seven
645     j       finish_check
646 check_8:
647     lb      $t4, eight
648     j       finish_check
649 check_9:
650     lb      $t4, nine
651     j       finish_check
652
653 finish_check:
654     lw      $ra, 0($sp)
655     addi    $sp, $sp, -4
656     jr      $ra
657
658 update_tg:
659     mul     $t9, $t9, 10
660     add     $t9, $t9, $t7
661
662 #-----
663 # Hoan thanh xong 1 so -> reset_led
664 #-----
665 done:
666     beq     $s0, 1, reset_led    # s0 = 1 -> toan tu -> chuyen den ham reset led
667     nop
668
669 #-----
670 #ham hien thi bit len led ben trai
671 #-----
672 load_to_left_led:
673     lb      $t6, 0($sp)          #load bit hien thi led tu stack
674     add     $sp, $sp, -4
675     lb      $t8, 0($sp)          #load gia tri cua bit nay
676     add     $sp, $sp, -4
677     add     $s2, $t8, $zero       #s2 = gia tri bit led trai
678     sb      $t6, 0($t0)          # hien thi len led trai
679

```



```

680 #-----
681 #ham hien thi bit len led ben phai
682 #-----
683 load_to_right_led:
684     sb      $t4, 0($t5)      # hien thi bit len led phai
685     add     $sp, $sp, 4
686     sb      $t7, 0($sp)      #day gia tri bit nay vao stack
687     add     $sp, $sp, 4
688     sb      $t4, 0($sp)      #day bit nay vao stack
689     add     $s1, $t7, $zero   #s1 = gia tri bit led phai
690     j       finish
691
692 reset_led:
693     li      $s0, 0           #s0 = 0 -> doi nhap so tiep theo trong 2 so
694     li      $t8, 0
695     addi    $sp, $sp, 4
696     sb      $t8, 0($sp)
697     lb      $t6, zero        # day bit zero vao stack
698     addi    $sp, $sp, 4
699     sb      $t6, 0($sp)
700 finish:
701     j       return
702     nop
703
704 #-----
705 #-----
706 return:
707     la      $a3, Loop1
708     mtc0    $a3, $14
709     eret
710
711 set_next_operator:
712 #-----
713 #tinh so thu 2
714 #-----
715 set_second_number1:
716     addi    $s5, $t9, 0
717     beq     $s3, 1, phep_cong_1      #s3 = 1 -> phep cong
718     beq     $s3, 2, phep_tru_1       #s3 = 2 -> phep tru
719     beq     $s3, 3, phep_nhan_1      #s3 = 3 -> phep nhan
720     beq     $s3, 4, phep_chia_1      #s3 = 4 -> phep chia
721     beq     $s3, 5, phep_lay_du_1    #s3 = 5 -> phep lay du
722
723 phep_cong_1:
724     add     $s6, $s5, $s4
725     li      $s3, 0
726     li      $t9, 0
727     j       in_phep_cong_1
728     nop
729

```



```

729
730 #-----
731 # In ra dinh dang: a + b = c \n
732 #-----
733 in_phep_cong_1:
734     li    $v0, 1
735     move  $a0, $s4
736     syscall
737     li    $s4, 0           #reset $s4
738
739     li    $v0, 11
740     li    $a0, '+'
741     syscall
742
743     li    $v0, 1
744     move  $a0, $s5
745     syscall
746     li    $s5, 0           #reset $s5
747
748     li    $v0, 11
749     li    $a0, '='
750     syscall
751
752     li    $v0, 1
753     move  $a0, $s6
754     syscall
755     nop
756
757     li    $v0, 11
758     li    $a0, '\n'
759     syscall
760     li    $s7, 100
761     div   $s6, $s7
762     mfhi  $s6               #Lay 2 gia tri cuoi cua ket qua
763     j show_result_in_led_1  #Hien thi ket qua tren led
764     nop
765

```

```

766 phep_tru_1:
767     sub    $s6, $s4, $s5
768     li     $s3, 0
769     li     $t9, 0
770     j      in_phep_tru_1
771     nop
772
773     #-----
774     #In ra dinh dang: a - b = c \n
775     #-----
776 in_phep_tru_1:
777     li     $v0, 1
778     move   $a0, $s4
779     syscall
780     li     $s4, 0           #reset $s4
781
782     li     $v0, 11
783     li     $a0, '-'
784     syscall
785     li     $s5, 0           #reset $s5
786
787     li     $v0, 1
788     move   $a0, $s5
789     syscall
790
791     li     $v0, 11
792     li     $a0, '='
793     syscall
794
795     li     $v0, 1
796     move   $a0, $s6
797     syscall
798
799     li     $v0, 55
800     li     $a0, '\n'
801     syscall
802     li     $s7, 100
803     div    $s6, $s7
804     mfhi   $s6              #Lay 2 gia tri cuoi cua ket qua
805     j      show_result_in_led_1 #Hien thi ket qua tren led
806     nop

```

```

808 phep_nhan_1:
809     mul    $s6, $s4, $s5
810     li     $s3, 0
811     li     $t9, 0
812     j      in_phep_nhan_1
813     nop
814
815     #-----
816     #In ra dinh dang: a*b = c \n
817     #-----
818 in_phep_nhan_1:
819     li     $v0, 1
820     move   $a0, $s4
821     syscall
822     li     $s4, 0           #reset $s4
823
824     li     $v0, 11
825     li     $a0, '*'
826     syscall
827
828     li     $v0, 1
829     move   $a0, $s5
830     syscall
831     li     $s5, 0           #reset $s5
832
833     li     $v0, 11
834     li     $a0, '='
835     syscall
836
837     li     $v0, 1
838     move   $a0, $s6
839     syscall
840
841     li     $v0, 11
842     li     $a0, '\n'
843     syscall
844
845     li     $s7, 100
846     div    $s6, $s7
847     mfhi   $s6              #Lay 2 chu so sau cung cua ket qua in ra
848     j      show_result_in_led_1  #Hien thi ket qua tren led
849     nop
850
851 phep_chia_1:
852     beq    $s5, 0, chia_01 # kiem tra xem so chia co phai bang 0 hay khong
853     li     $s3, 0
854     div    $s4, $s5         # s6=s4/s5
855     mflo   $s6
856     mfhi   $s7
857     li     $t9, 0
858     j      in_phep_chia_1
859     nop
860
861     #-----
862     # Neu $s5 = 0 in ra: "Khong the chia cho so 0 \n"
863     #-----
864 chia_01:
865     li     $v0, 55
866     la     $a0, mess1
867     li     $a1, 0
868     syscall
869     j      reset_led1
870

```

```

870
871 in_phep_chia_1:
872     li    $v0, 1
873     move  $a0, $s4
874     syscall
875     li    $s4, 0          #reset $s4
876
877     li    $v0, 11
878     li    $a0, '/'
879     syscall
880
881     li    $v0, 1
882     move  $a0, $s5
883     syscall
884     li    $s5, 0          #reset $s5
885
886     li    $v0, 11
887     li    $a0, '='
888     syscall
889
890     li    $v0, 1
891     move  $a0, $s6
892     syscall
893
894     li    $v0, 11
895     li    $a0, '\n'
896     syscall
897
898     li    $s7, 100
899     div   $s6, $s7
900     mfhi  $s6              #Lay 2 chu so sau cung cua ket qua in ra
901     j     show_result_in_led_1 #Hien thi ket qua tren led
902     nop
903

```

```

904 phep_lay_du_1:
905     beq   $s5, 0, chia_lay_du_01 # kiem tra xem so chia co phai bang 0 hay khong
906     li    $s3, 0
907     div   $s4, $s5
908     mflo  $s7
909     mfhi  $s6
910     li    $t9, 0
911     j     in_phep_lay_du_1
912     nop
913
914 #-----
915 # Neu $s5 = 0 in ra: "Khong the chia cho so 0 \n"
916 #-----
917 chia_lay_du_01:
918     li    $v0, 55
919     la    $a0, mess1
920     li    $a1, 0
921     syscall
922     j     reset_led1
923

```

```

923
924 in_phep_lay_du_1:
925     li      $v0, 1
926     move    $a0, $s4
927     syscall
928     li      $s4, 0          #reset $s4
929
930     li      $v0, 11
931     li      $a0, '%'
932     syscall
933
934     li      $v0, 1
935     move    $a0, $s5
936     syscall
937     li      $s4, 0          #reset $s4
938
939     li      $v0, 11
940     li      $a0, '='
941     syscall
942
943     li      $v0, 1
944     move    $a0, $s6
945     syscall
946
947
948     li      $v0, 11
949     li      $a0, '\n'
950     syscall
951
952     li      $s7, 100
953     div     $s6, $s7
954     mfhi    $s6             #Lay 2 gia tri cuoi cua ket qua
955     j       show_result_in_led_1    #Hien thi ket qua tren led
956     nop
957

```

```

957
958 #-----
959 #Hien thi ket qua tren den led
960 #So 'ab'
961 # Den trai = a = ab div 10
962 # Den phai = b = ab mod 10
963 #-----
964 show_result_in_led1:
965     li      $t8, 10      # Gia tri trung gian = 10
966     div     $s6, $t8     # $s6 = a
967     mflo    $t7          # $t7 = result
968     jal     check_l1     #chuyen den ham chuyen t7 thanh bit hien thi len led
969
970     sb      $t4, 0($t0)   # hien thi len led trai
971     add     $sp, $sp, 4
972     sb      $t7, 0($sp)   #day gia tri bit nay vao stack
973     add     $sp, $sp, 4
974     sb      $t4, 0($sp)   #day bit nay vao stack
975     add     $s2, $t7, $zero #s1 = gia tri bit led phai
976
977     mfhi    $t7          #t7= remainder
978     jal     check_l1     #convert t7 thanh bit hien thi len led
979     sb      $t4, 0($t5)   #hien thi len led phai
980     add     $sp, $sp, 4
981     sb      $t7, 0($sp)   # day gia tri bit nay vao stack
982     add     $sp, $sp, 4
983     sb      $t4, 0($sp)   # day bit nay vao stack
984     add     $s1, $t7, $zero # s1 = gia tri bit led phai
985     j       reset_led1   # ham reset lai led
986
987

```

```

988 check_l1:
989     addi    $sp, $sp, 4
990     sw      $ra, 0($sp)
991     beq     $t7, 0, check_01
992     beq     $t7, 1, check_11
993     beq     $t7, 2, check_21
994     beq     $t7, 3, check_31
995     beq     $t7, 4, check_41
996     beq     $t7, 5, check_51
997     beq     $t7, 6, check_61
998     beq     $t7, 7, check_71
999     beq     $t7, 8, check_81
1000    beq     $t7, 9, check_91
1001
1002    #-----
1003    #Chuyen gia tri thanh bit hien tren thanh led
1004    #-----
1005    check_01:
1006        lb      $t4, zero
1007        j        finish_check_1
1008    check_11:
1009        lb      $t4, one
1010        j        finish_check_1
1011    check_21:
1012        lb      $t4, two
1013        j        finish_check_1
1014    check_31:
1015        lb      $t4, three
1016        j        finish_check_1
1017    check_41:
1018        lb      $t4, four
1019        j        finish_check_1
1020    check_51:
1021        lb      $t4, five
1022        j        finish_check_1
1023    check_61:
1024        lb      $t4, six
1025        j        finish_check_1
1026    check_71:
1027        lb      $t4, seven
1028        j        finish_check_1
1029    check_81:
1030        lb      $t4, eight

```

t7 = 0 -> Hien thi so 0 tren thanh led
t7 = 1 -> Hien thi so 1 tren thanh led
t7 = 2 -> Hien thi so 2 tren thanh led
t7 = 3 -> Hien thi so 3 tren thanh led
t7 = 4 -> Hien thi so 4 tren thanh led
t7 = 5 -> Hien thi so 5 tren thanh led
t7 = 6 -> Hien thi so 6 tren thanh led
t7 = 7 -> Hien thi so 7 tren thanh led
t7 = 8 -> Hien thi so 8 tren thanh led
t7 = 9 -> Hien thi so 9 tren thanh led

```

1036
1037 finish_check_1:
1038     lw     $ra, 0($sp)
1039     addi   $sp, $sp, -4
1040     jr     $ra
1041
1042 done_1:
1043     beq    $s0, 1, reset_led1    # s0 = 1 -> toan tu -> chuyen den ham reset led
1044     nop
1045
1046 reset_led1:
1047     li     $s0, 0                #s0 = 0 -> doi nhap so tiep theo trong 2 so
1048     li     $t8, 0
1049     addi   $sp, $sp, 4
1050     sb     $t8, 0($sp)
1051     lb     $t6, zero            # day bit zero vao stack
1052     addi   $sp, $sp, 4
1053     sb     $t6, 0($sp)
1054
1055     beq    $a3, 1, set_add
1056     nop
1057
1058     beq    $a3, 2, set_sub
1059     nop
1060
1061     beq    $a3, 3, set_mul
1062     nop
1063
1064     beq    $a3, 4, set_div
1065     nop
1066
1067     beq    $a3, 5, set_mod
1068     nop
1069

```



```

1069
1070 set_add:
1071     addi    $s3, $zero, 1
1072     j       finish1
1073     nop
1074
1075 set_sub:
1076     addi    $s3, $zero, 2
1077     j       finish1
1078     nop
1079
1080 set_mul:
1081     addi    $s3, $zero, 3
1082     j       finish1
1083     nop
1084
1085 set_div:
1086     addi    $s3, $zero, 4
1087     j       finish1
1088     nop
1089
1090 set_mod:
1091     addi    $s3, $zero, 5
1092     j       finish1
1093     nop
1094
1095 finish1:
1096     j       return_1
1097     nop
1098
1099 #-----
1100 #-----
1101 return_1:
1102     la $a3, Khoi_tao
1103     mtc0 $a3, $14
1104     eret
1105

```

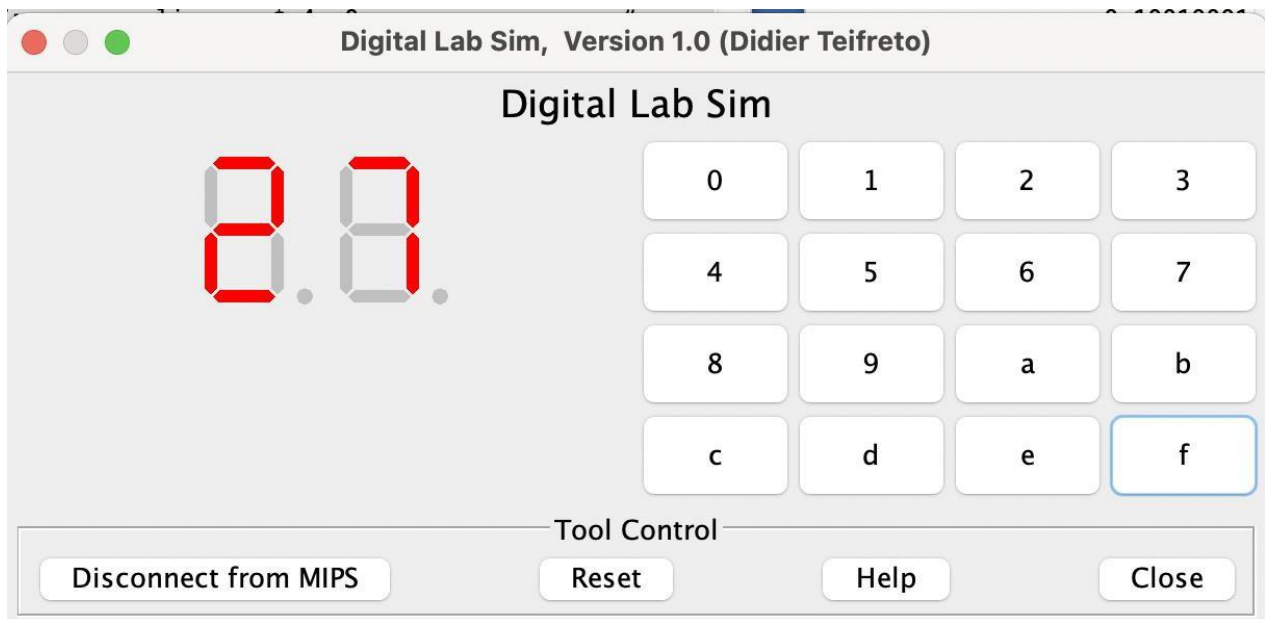
1.4 Chạy thử chương trình

Thực hiện các phép toán:

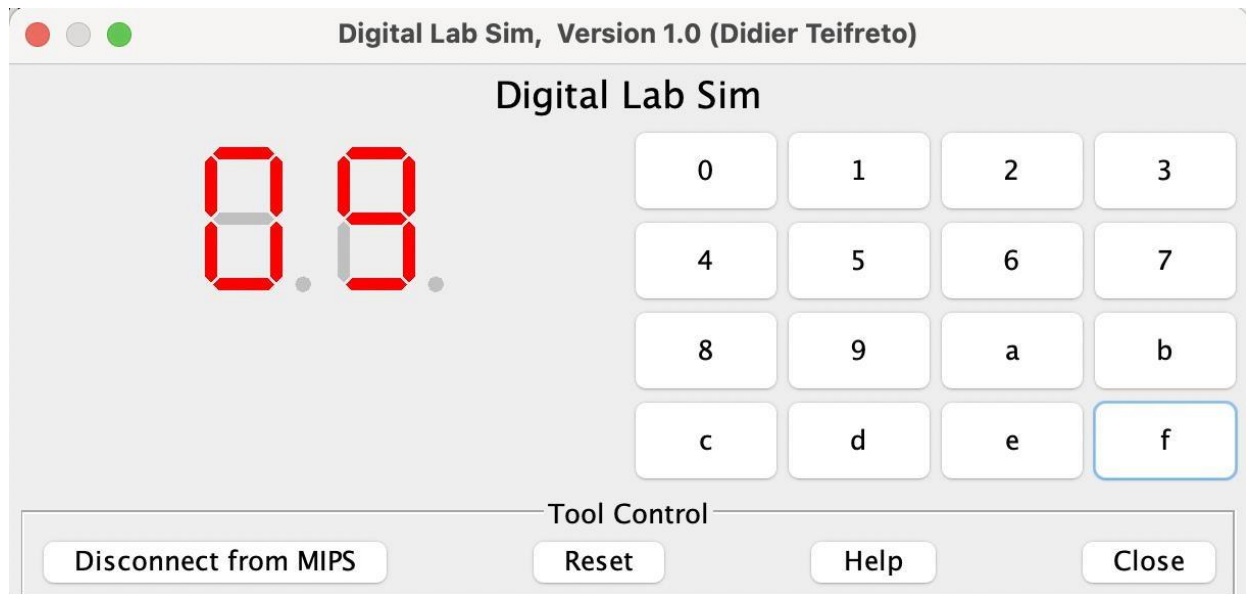


Hiện thị ra ngoài màn hình:

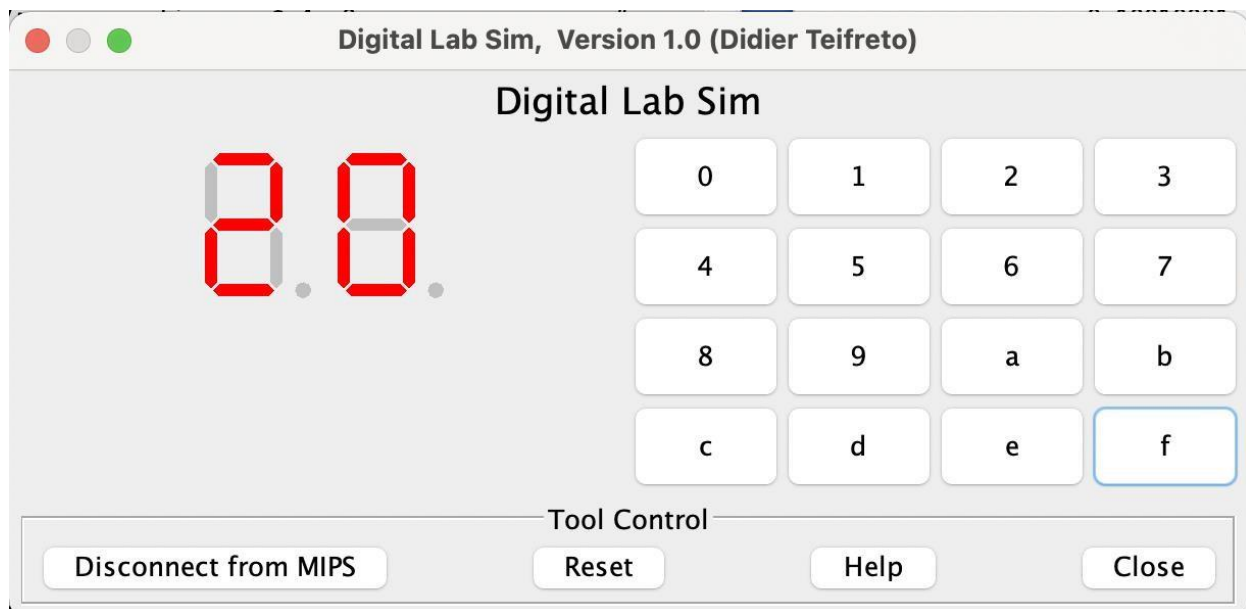
$$15+112 = 127$$



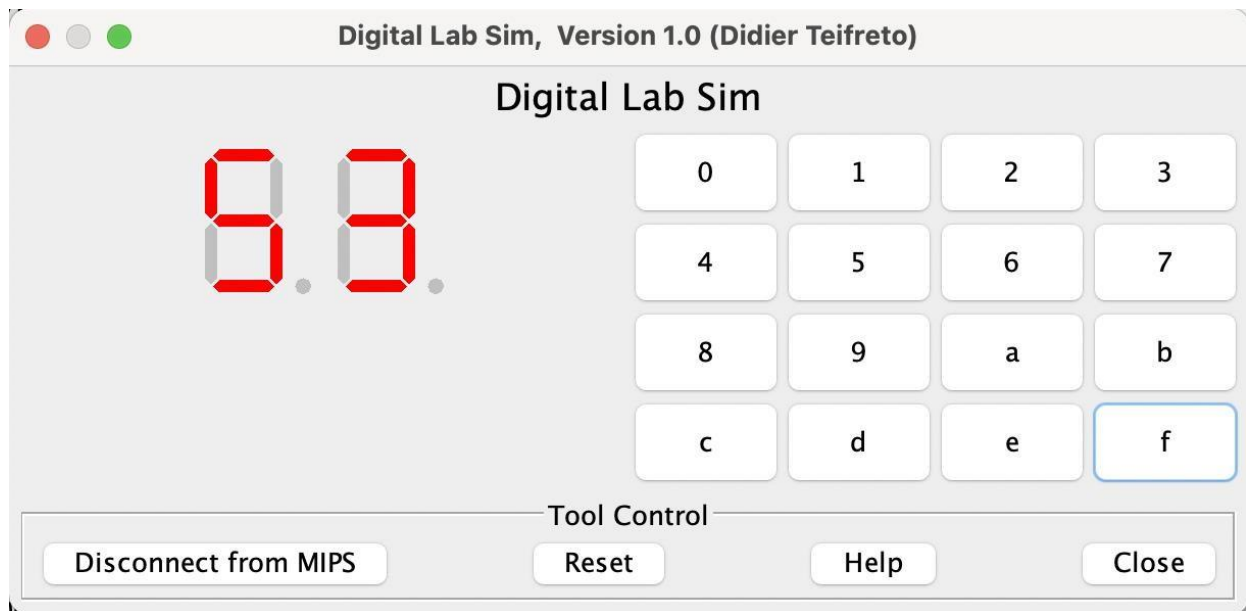
$$15 - 6 = 9$$



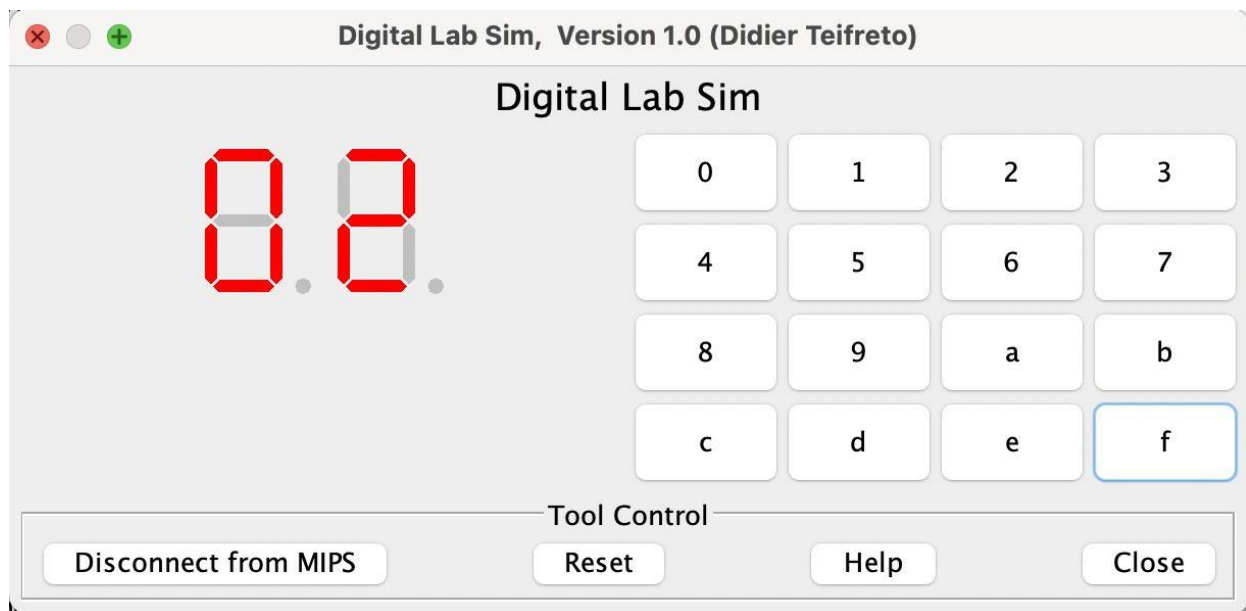
$$15 * 8 = 120$$



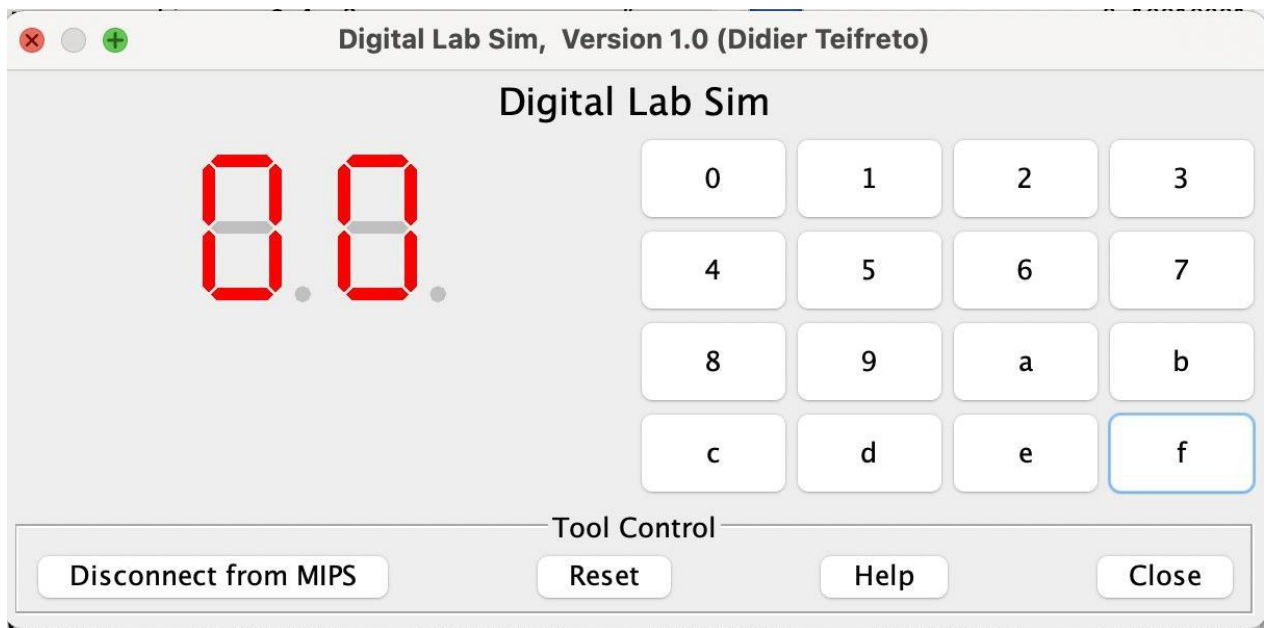
$$159 / 3 = 53$$



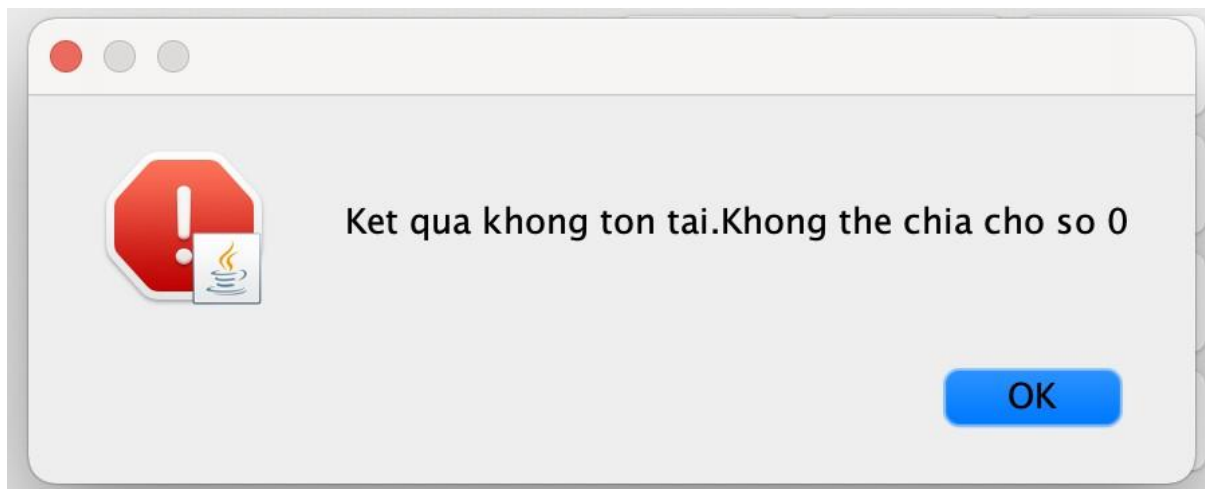
$$17 \% 3 = 2$$



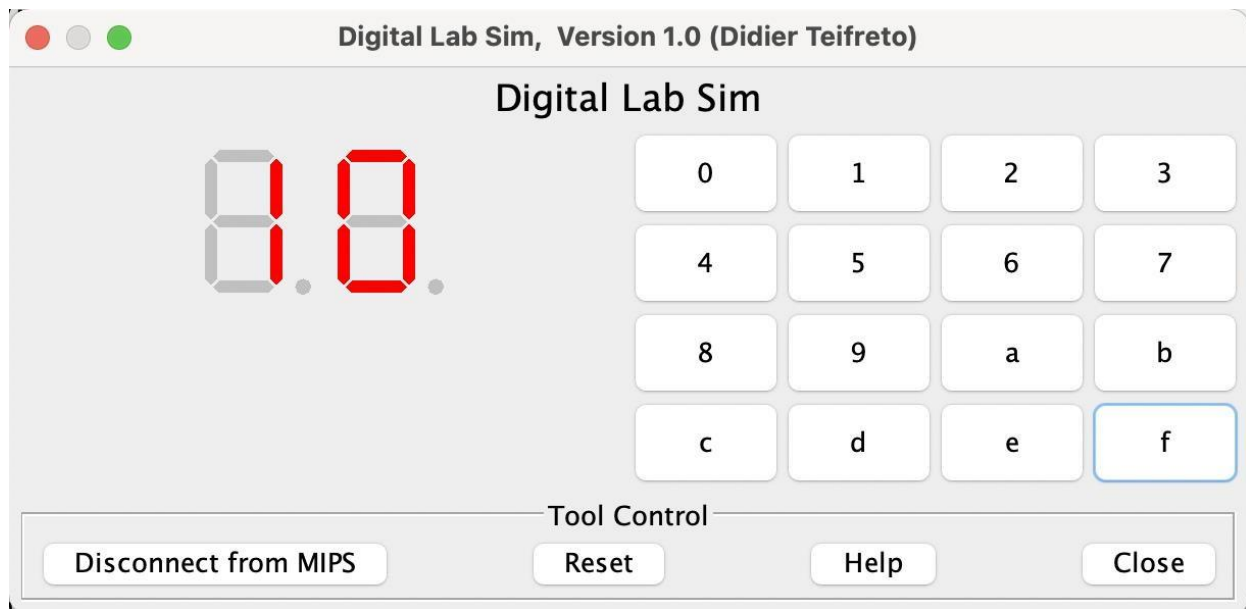
$12 - 45 = -33$ -> không in được kết quả trên Digital Lab Sim.



51 / 0 -> Không thể thực hiện được.



51 / 5 = 10



3. Project 3: Kiểm tra tốc độ và độ chính xác khi gõ văn bản

a) phân tích cách làm , thuật toán

- Sử dụng vòng lặp vô hạn. Trong vòng lặp có kiểm tra giá trị tại địa chỉ KEY_READY nếu khác 0 thì có kí tự nhập từ bàn phím. Nếu có kí tự nhập từ bàn phím sẽ nhảy đến trạng thái MakeIntR để xử lý trạng thái ngắt.

- Sau đó ta kiểm tra xem đó là loại ngắt nào bằng cách dùng phép AND :

- Nếu nó là ngắt từ bộ đếm thì sẽ thực hiện chương trình đếm số lần ngắt nếu số lần ngắt lớn hơn 40 (tức là 1s) thì sẽ làm mới số lần ngắt và cộng thêm 1 giây vào biến đếm giây đã khai báo
- Nếu ngắt từ bàn phím thì ta sẽ thực hiện so sánh kí tự nhập vào với kí tự trong đoạn văn bản nếu giống nhau thì sẽ tăng biến đếm số kí tự đúng thêm 1 còn không giống nhau thì ta bỏ qua.

- Sau đó quay trở về chương trình chính, nếu so sánh hết các ký tự trong đoạn văn bản hoặc khi người dùng ấn nút enter thì sẽ dừng vòng lặp

- Cuối cùng ta in ra kết quả :

- Để in ra số ký tự đúng ra đèn LED ta dùng biến lưu số ký tự đúng chuyển sang 1 số khác để các thanh LED in ra đúng số
- Để tính số ký tự trên 1 đơn vị thời gian : ta lấy số ký tự được nhập vào nhân với 60 rồi chia cho thời gian nhập ký tự sẽ ra được số ký tự trên 1 phút

b) Mã Nguồn

```
.eqv SEVENSEG_LEFT 0xFFFF0011      #dia chi led 7 doan trai
.eqv SEVENSEG_RIGHT 0xFFFF0010     #dia chi led 7 doan phai
.eqv MASK_CAUSE_COUNTER 0x00000400 #Bit 10: Counter interrupt
.eqv COUNTER 0xFFFF0013            #Time Counter
.eqv KEY_CODE 0xFFFF0004           # ASCII code from keyboard, 1 byte
.eqv KEY_READY 0xFFFF0000          # =1 if has a new keycode ?
                                   # Auto clear after lw

.data
arr: .byte      63, 6, 91, 79, 102, 109, 125, 7, 127, 111      #tu 0 den 9 de doi sang
lam den led sang
str: .asciiiz "bo mon ki thuat may tinh"
msg1: .asciiiz "thoi gian hoan thanh : "
msg2: .asciiiz "(s)\nso tu tren mot don vi thoi gian : "
msg3: .asciiiz "tu/phut\n"

#~~~~~
~~
# MAIN Procedure
#~~~~~
~~

.text

main:
li $k0,KEY_CODE
li $k1,KEY_READY
# Enable the interrupt of TimeCounter of Digital Lab Sim
li $t1, COUNTER
sb $t1, 0($t1)
addi $s0,$zero,0 # $s0 dem so ky tu nhap vao
addi $s1,$zero,0 # $s1 dem tong so ky tu dung
addi $s2,$zero,0 # dem so lan ngat
addi $s3,$zero,0 # dem thoi gian
la $a1, str
```

```
#-----  
# nhap dau vao  
#-----
```

```
loop:  
lw $t1,0($k1)      # $t1 = [$k1] = KEY_READY  
bne $t1,0,MakeIntR # $t1 != 0 hay co phim duoc nhap thi chuyen den MakeIntR  
li $v0,32          # BUG: must sleep to wait for Time  
li $a0,5           # sleep 5ms  
syscall  
j loop
```

```
MakeIntR:  
teqi $t1,1          # neu $t1=1 co phim duoc nhap thi chuyen den chuong trinh ngat  
j loop              # quay lai vong lap de nhap ki tu moi
```

```
#-----  
# Interrupt subroutine  
#-----
```

```
.ktext 0x80000180  
IntSR: #-----  
# Temporary disable interrupt  
#-----
```

```
dis_int:li $t1, COUNTER # BUG: must disable with Time Counter  
sb $zero, 0($t1)  
#-----  
# Processing  
#-----
```

```
#kiem tra ly do ngat
```

```
get_caus:  
mfc0 $t1,$13        # $t1 la ly do ngat
```

```
IsCount:  
li $t2, MASK_CAUSE_COUNTER  
and $at,$t2,$t1  
bne $at,$t2,Keyboard_IntR
```


ly do ngat la bo dem

Counter_IntR:

ble \$s2,40,demtiiep # neu chua du 1s thi dem tiep

addi \$s2,\$zero,0 # tao lai so lan ngat

addi \$s3,\$s3,1 # tang len 1s

j en_int # nhay den xu ly chuoi

demtiiep:

addi \$s2,\$s2,1

j en_int # nhay den xu ly chuoi

xu ly chuoi

Keyboard_IntR:

lb \$t0,0(\$a1) # lay ki tu thi i trong doan van ban

beq \$t0,\$0,end_program

lw \$t1,0(\$k0) # lay ki tu nhap vao tu ban phim

bne \$t0,\$t1,end_process # neu 2 ki tu ko bang nhau thi nhay den end_process

nop

addi \$s1,\$s1,1 # tang so ki tu dung len 1

end_process:

addi \$s0,\$s0,1 # tang so ki tu nhap vao len 1

addi \$a1,\$a1,1 # tang i 1 don vi trong doan van ban

en_int:

Must clear cause reg

#-----

Re-enable interrupt

#-----

li \$t1, COUNTER

sb \$t1, 0(\$t1)

mtc0 \$zero, \$13

#-----

Evaluate the return address of main routine

epc <= epc + 4

#-----

next_pc:mfc0 \$at, \$14 # \$at <= Coproc0.\$14 = Coproc0.epc

addi \$at, \$at, 4 # \$at = \$at + 4 (next instruction)

mtc0 \$at, \$14 # Coproc0.\$14 = Coproc0.epc <= \$at

return: eret # Return from exception

#-----

end_program:

in ra thoi gian hoan thanh

addi \$v0, \$0, 4

la \$a0, msg1

syscall

addi \$v0, \$0, 1

addi \$a0, \$s3, 0

syscall

addi \$v0, \$0, 4

la \$a0, msg2

syscall

in ra so tu tren 1 don vi thoi gian

addi \$v0, \$0, 1

addi \$a0, \$0, 60

mult \$s0, \$a0 # nhan so ky tu nhap vao voi 60

mflo \$s0

div \$s0, \$s3 # so ky tu trong 1p

mflo \$a0

syscall

addi \$v0, \$0, 4

la \$a0, msg3

syscall

addi \$s0, \$s1, 0

jal show

end_main:

li \$v0, 10

syscall

#-----

hien thi ket qua so ky tu dung tren led

show:

addi \$sp, \$sp, -4

sw \$ra, 0(\$sp)

addi \$t0, \$zero, 10

div \$s1, \$t0

mflo \$v0 # \$v0 = so hang chuc cua so ky tu dung

mfhi \$v1 # \$v1 = so hang don vi cua so ky tu dung

la \$a0, arr

add \$a0, \$a0, \$v0 # \$a0 la so de hien thi ra \$v0

lb \$a0, 0(\$a0) #Set value for segments

jal SHOW_7SEG_LEFT

la \$a0, arr

add \$a0, \$a0, \$v1 # \$a0 la so de hien thi ra \$v1

lb \$a0, 0(\$a0) #Set value for segments

```

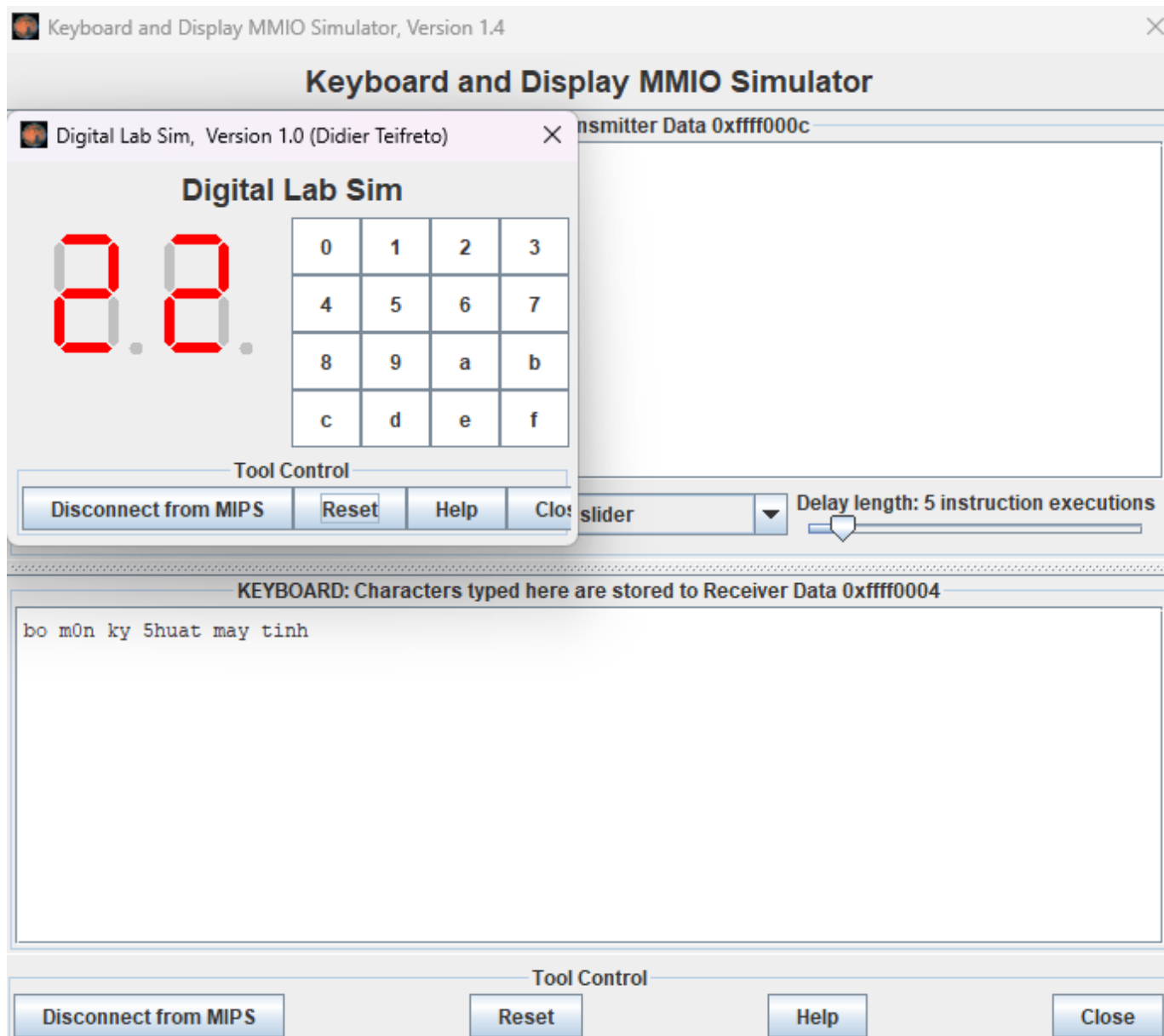
jal SHOW_7SEG_RIGHT
lw $ra, ($sp)
addi $sp, $sp, 4
jr $ra
SHOW_7SEG_LEFT:
li $t0, SEVENSEG_LEFT      #Assign port's address
sb $a0, 0($t0)              #Assign new value
jr $ra
SHOW_7SEG_RIGHT:
li $t0, SEVENSEG_RIGHT      #Assign port's address
sb $a0, 0($t0)              #Assign new value
jr $ra
nop

#-----

```

c) Kết quả chạy mô phỏng

Khi ta nhập “bom0n ki 5huat may tinh”



Nhập sai chữ o và t nên sẽ là 22

Reset: reset completed.

thoi gian hoan thanh : 9 (s)

so tu tren mot don vi thoi gian : 160 tu/phut

-- program is finished running --

Vì số từ nhập là 24 và thời gian chạy nên số từ trên 1 phút là 160

BÀI BÁO CÁO ĐẾN ĐÂY LÀ KẾT THÚC



Cảm ơn thầy đã đọc, chúc thầy một ngày làm việc hiệu quả!