**Nguyễn Đình Dương – 20225966**

**Assignment 1:**

|  |
| --- |
| .global init  .eqv GPIO\_ENABLE\_REG, 0x60004020 # Thanh ghi cấu hình vào/ra các chân GPIO  .eqv GPIO\_OUT\_W1TS\_REG, 0x60004008 # Thanh ghi thiết lập chân GPIO  init:  # Thiết lập GPIO2 là Output  li a1, GPIO\_ENABLE\_REG  li a2, 0x04 # Mặt nạ cho GPIO2 (bit 2)  sw a2, 0(a1)  # Thiết lập GPIO3 là Output  li a1, GPIO\_ENABLE\_REG  li a2, 0x08 # Mặt nạ cho GPIO3 (bit 3)  sw a2, 0(a1)  # Thiết lập GPIO4 là Output  li a1, GPIO\_ENABLE\_REG  li a2, 0x10 # Mặt nạ cho GPIO4 (bit 4)  sw a2, 0(a1)  # Bật LED trên GPIO2  li a1, GPIO\_OUT\_W1TS\_REG  li a2, 0x04 # Mặt nạ cho GPIO2  sw a2, 0(a1)  # Bật LED trên GPIO3  li a1, GPIO\_OUT\_W1TS\_REG  li a2, 0x08 # Mặt nạ cho GPIO3  sw a2, 0(a1)  # Bật LED trên GPIO4  li a1, GPIO\_OUT\_W1TS\_REG  li a2, 0x10 # Mặt nạ cho GPIO4  sw a2, 0(a1) |

**Lưu ý:** Tùy với mỗi chân thì sẽ chỉ lưu và chạy GPIO\_ENABLE\_REG và GPIO\_OUT\_W1TS\_REG của chân đó.

Output:

|  |  |
| --- | --- |
| GPI02 |  |
| GPI03 |  |
| GPI04 |  |

**Assignment 2**

Code GPI02

|  |
| --- |
| .global init  .eqv GPIO\_ENABLE\_REG, 0x60004020 # Thanh ghi cấu hình vào/ra các chân GPIO  .eqv GPIO\_OUT\_W1TS\_REG, 0x60004008 # Thanh ghi thiết lập chân GPIO  .eqv GPIO\_OUT\_W1TC\_REG, 0x6000400C # Thanh ghi xóa chân GPIO  .text  init:  # Thiết lập GPIO2 là Output  li a1, GPIO\_ENABLE\_REG  li a2, 0x04 # Mặt nạ cho GPIO2 (bit 2)  sw a2, 0(a1)  main\_loop:  # Bật LED trên GPIO2  li a1, GPIO\_OUT\_W1TS\_REG  li a2, 0x04 # Mặt nạ cho GPIO2  sw a2, 0(a1)  # Gọi hàm delay  call delay\_asm  # Tắt LED trên GPIO2  li a1, GPIO\_OUT\_W1TC\_REG  li a2, 0x04 # Mặt nạ cho GPIO2  sw a2, 0(a1)  # Gọi hàm delay  call delay\_asm  # Quay lại vòng lặp chính  j main\_loop  # Hàm delay: chờ một khoảng thời gian  delay\_asm:  li a3, 0 # Giá trị biến đếm  li a4, 10000000 # Thời gian chờ (số lần đếm)  loop\_delay\_gpio2:  addi a3, a3, 1  blt a3, a4, loop\_delay\_gpio2  ret |

A screenshot of a computer

Description automatically generated

Code GPI03

|  |
| --- |
| .global init  .eqv GPIO\_ENABLE\_REG, 0x60004020 # Thanh ghi cấu hình vào/ra các chân GPIO  .eqv GPIO\_OUT\_W1TS\_REG, 0x60004008 # Thanh ghi thiết lập chân GPIO  .eqv GPIO\_OUT\_W1TC\_REG, 0x6000400C # Thanh ghi xóa chân GPIO  .text  init:  # Thiết lập GPIO3 là Output  li a1, GPIO\_ENABLE\_REG  li a2, 0x08 # Mặt nạ cho GPIO3 (bit 3)  sw a2, 0(a1)  main\_loop:  # Bật LED trên GPIO3  li a1, GPIO\_OUT\_W1TS\_REG  li a2, 0x08 # Mặt nạ cho GPIO3  sw a2, 0(a1)  # Gọi hàm delay  call delay\_asm  # Tắt LED trên GPIO3  li a1, GPIO\_OUT\_W1TC\_REG  li a2, 0x08 # Mặt nạ cho GPIO3  sw a2, 0(a1)  # Gọi hàm delay  call delay\_asm  # Quay lại vòng lặp chính  j main\_loop  # Hàm delay: chờ một khoảng thời gian  delay\_asm:  li a3, 0 # Giá trị biến đếm  li a4, 5000000 # Thời gian chờ (số lần đếm)  loop\_delay\_gpio3:  addi a3, a3, 1  blt a3, a4, loop\_delay\_gpio3  ret |

A computer screen shot of a computer

Description automatically generated

Code GPIO4

|  |
| --- |
| .global init  .eqv GPIO\_ENABLE\_REG, 0x60004020 # Thanh ghi cấu hình vào/ra các chân GPIO  .eqv GPIO\_OUT\_W1TS\_REG, 0x60004008 # Thanh ghi thiết lập chân GPIO  .eqv GPIO\_OUT\_W1TC\_REG, 0x6000400C # Thanh ghi xóa chân GPIO  .text  init:  # Thiết lập GPIO4 là Output  li a1, GPIO\_ENABLE\_REG  li a2, 0x10 # Mặt nạ cho GPIO4 (bit 4)  sw a2, 0(a1)  main\_loop:  # Bật LED trên GPIO4  li a1, GPIO\_OUT\_W1TS\_REG  li a2, 0x10 # Mặt nạ cho GPIO4  sw a2, 0(a1)  # Gọi hàm delay  call delay\_asm  # Tắt LED trên GPIO4  li a1, GPIO\_OUT\_W1TC\_REG  li a2, 0x10 # Mặt nạ cho GPIO4  sw a2, 0(a1)  # Gọi hàm delay  call delay\_asm  # Quay lại vòng lặp chính  j main\_loop  # Hàm delay: chờ một khoảng thời gian  delay\_asm:  li a3, 0 # Giá trị biến đếm  li a4, 5000000 # Thời gian chờ (số lần đếm)  loop\_delay\_gpio4:  addi a3, a3, 1  blt a3, a4, loop\_delay\_gpio4  ret |

A screenshot of a computer

Description automatically generated

**Assignment 3:**

Code:

|  |
| --- |
| .global init  # GPIO Register Definitions  .eqv GPIO\_ENABLE\_REG, 0x60004020 # GPIO output enable register  .eqv GPIO\_OUT\_REG, 0x60004004 # GPIO output register  .eqv IO\_MUX\_GPIO4\_REG, 0x60009014 # GPIO4 function configuration  .eqv IO\_MUX\_GPIO5\_REG, 0x60009018 # GPIO5 function configuration  .eqv IO\_MUX\_GPIO6\_REG, 0x6000901C # GPIO6 function configuration  .eqv IO\_MUX\_GPIO7\_REG, 0x60009020 # GPIO7 function configuration  # 7-segment display patterns for digits 0-9 (Common Anode - active low)  # Segment mapping: GPIO0-6 -> segments a-g  .data  digits:  .word 0xC0 # 0: 1100 0000 - segments a,b,c,d,e,f on (0 = on)  .word 0xF9 # 1: 1111 1001 - segments b,c on  .word 0xA4 # 2: 1010 0100 - segments a,b,d,e,g on  .word 0xB0 # 3: 1011 0000 - segments a,b,c,d,g on  .word 0x99 # 4: 1001 1001 - segments b,c,f,g on  .word 0x92 # 5: 1001 0010 - segments a,c,d,f,g on  .word 0x82 # 6: 1000 0010 - segments a,c,d,e,f,g on  .word 0xF8 # 7: 1111 1000 - segments a,b,c on  .word 0x80 # 8: 1000 0000 - all segments on  .word 0x90 # 9: 1001 0000 - segments a,b,c,f,g on  .text  init:  # Configure GPIO0-7 as outputs  li a1, GPIO\_ENABLE\_REG  li a2, 0xFF # Enable GPIO0-7 as outputs  sw a2, 0(a1)  # Configure GPIO4-7 function as GPIO  li a2, 0x1000 # Set GPIO function  li a1, IO\_MUX\_GPIO4\_REG  sw a2, 0(a1)  li a1, IO\_MUX\_GPIO5\_REG  sw a2, 0(a1)  li a1, IO\_MUX\_GPIO6\_REG  sw a2, 0(a1)  li a1, IO\_MUX\_GPIO7\_REG  sw a2, 0(a1)  display\_loop:  la a3, digits # Load address of digit patterns  li a4, 0 # Counter for digits 0-9    next\_digit:  # Load and display the current digit pattern  lw a2, 0(a3) # Load pattern for current digit  li a1, GPIO\_OUT\_REG  sw a2, 0(a1) # Output pattern to GPIO    # Delay  call delay    # Move to next digit  addi a3, a3, 4 # Next pattern address  addi a4, a4, 1 # Increment counter  li t0, 10  blt a4, t0, next\_digit # If counter < 10, continue    j display\_loop # Repeat from 0  # Delay subroutine  delay:  li t0, 0 # Counter  li t1, 10000000 # Delay duration  delay\_loop:  addi t0, t0, 1  blt t0, t1, delay\_loop  ret |

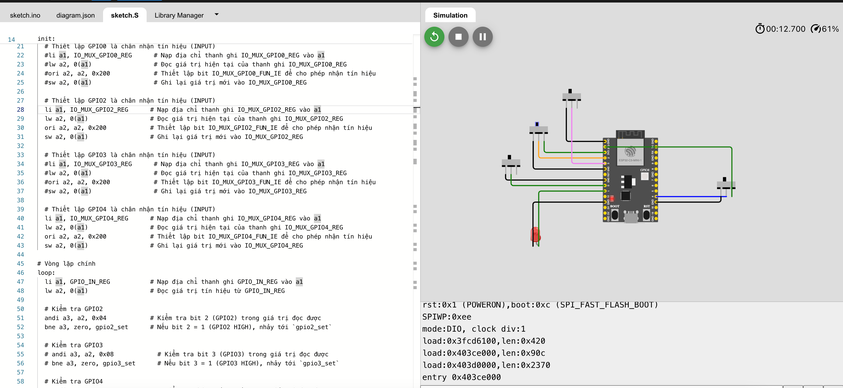
Output:

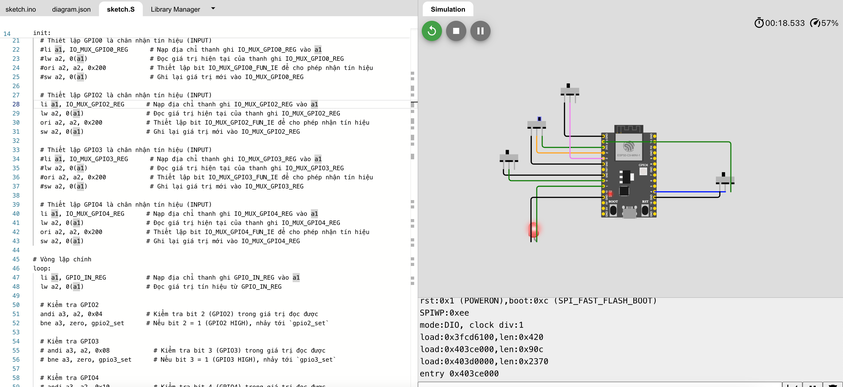
|  |  |
| --- | --- |
| 1 |  |
| 2 |  |
| 3 |  |
| 4 |  |
| 5 |  |
| 6 |  |
| 7 |  |
| 8 |  |
| 9 |  |

Assignment 4

|  |
| --- |
| .global init  # Định nghĩa các thanh ghi quan trọng  .eqv GPIO\_OUT\_W1TS\_REG, 0x60004008 # Thanh ghi thiết lập (SET register)  .eqv GPIO\_OUT\_W1TC\_REG, 0x6000400C # Thanh ghi xóa (CLEAR register)  .eqv GPIO\_ENABLE\_REG, 0x60004020 # Thanh ghi cho phép xuất tín hiệu  .eqv GPIO\_IN\_REG, 0x6000403C # Thanh ghi đọc trạng thái GPIO  .eqv IO\_MUX\_GPIO0\_REG, 0x60009004 # Thanh ghi thiết lập chức năng GPIO0  .eqv IO\_MUX\_GPIO2\_REG, 0x60009008 # Thanh ghi thiết lập chức năng GPIO2  .eqv IO\_MUX\_GPIO3\_REG, 0x6000900C # Thanh ghi thiết lập chức năng GPIO3  .eqv IO\_MUX\_GPIO4\_REG, 0x60009010 # Thanh ghi thiết lập chức năng GPIO4  # Hàm khởi tạo  init:  # Thiết lập GPIO1 là chân xuất tín hiệu (OUTPUT)  li a1, GPIO\_ENABLE\_REG # Nạp địa chỉ thanh ghi GPIO\_ENABLE\_REG vào a1  li a2, 0x02 # Giá trị 0x02 tương ứng với việc kích hoạt GPIO1  sw a2, 0(a1) # Ghi giá trị 0x02 vào thanh ghi GPIO\_ENABLE\_REG  # Thiết lập GPIO0 là chân nhận tín hiệu (INPUT)  li a1, IO\_MUX\_GPIO0\_REG # Nạp địa chỉ thanh ghi IO\_MUX\_GPIO0\_REG vào a1  lw a2, 0(a1) # Đọc giá trị hiện tại của thanh ghi IO\_MUX\_GPIO0\_REG  ori a2, a2, 0x200 # Thiết lập bit IO\_MUX\_GPIO0\_FUN\_IE để cho phép nhận tín hiệu  sw a2, 0(a1) # Ghi lại giá trị mới vào IO\_MUX\_GPIO0\_REG  # Thiết lập GPIO2 là chân nhận tín hiệu (INPUT)  #li a1, IO\_MUX\_GPIO2\_REG # Nạp địa chỉ thanh ghi IO\_MUX\_GPIO2\_REG vào a1  #lw a2, 0(a1) # Đọc giá trị hiện tại của thanh ghi IO\_MUX\_GPIO2\_REG  #ori a2, a2, 0x200 # Thiết lập bit IO\_MUX\_GPIO2\_FUN\_IE để cho phép nhận tín hiệu  #sw a2, 0(a1) # Ghi lại giá trị mới vào IO\_MUX\_GPIO2\_REG  # Thiết lập GPIO3 là chân nhận tín hiệu (INPUT)  #li a1, IO\_MUX\_GPIO3\_REG # Nạp địa chỉ thanh ghi IO\_MUX\_GPIO3\_REG vào a1  #lw a2, 0(a1) # Đọc giá trị hiện tại của thanh ghi IO\_MUX\_GPIO3\_REG  #ori a2, a2, 0x200 # Thiết lập bit IO\_MUX\_GPIO3\_FUN\_IE để cho phép nhận tín hiệu  #sw a2, 0(a1) # Ghi lại giá trị mới vào IO\_MUX\_GPIO3\_REG  # Thiết lập GPIO4 là chân nhận tín hiệu (INPUT)  #li a1, IO\_MUX\_GPIO4\_REG # Nạp địa chỉ thanh ghi IO\_MUX\_GPIO4\_REG vào a1  #lw a2, 0(a1) # Đọc giá trị hiện tại của thanh ghi IO\_MUX\_GPIO4\_REG  #ori a2, a2, 0x200 # Thiết lập bit IO\_MUX\_GPIO4\_FUN\_IE để cho phép nhận tín hiệu  #sw a2, 0(a1) # Ghi lại giá trị mới vào IO\_MUX\_GPIO4\_REG  # Vòng lặp chính  loop:  li a1, GPIO\_IN\_REG # Nạp địa chỉ thanh ghi GPIO\_IN\_REG vào a1  lw a2, 0(a1) # Đọc giá trị tín hiệu từ GPIO\_IN\_REG  # Kiểm tra GPIO0  andi a3, a2, 0x01 # Kiểm tra mức tín hiệu GPIO0  bne a3, zero, gpio0\_set  # Kiểm tra GPIO2  #andi a3, a2, 0x04 # Kiểm tra bit 2 (GPIO2) trong giá trị đọc được  #bne a3, zero, gpio2\_set # Nếu bit 2 = 1 (GPIO2 HIGH), nhảy tới gpio2\_set  # Kiểm tra GPIO3  #andi a3, a2, 0x08 # Kiểm tra bit 3 (GPIO3) trong giá trị đọc được  #bne a3, zero, gpio3\_set # Nếu bit 3 = 1 (GPIO3 HIGH), nhảy tới gpio3\_set  # Kiểm tra GPIO4  #andi a3, a2, 0x10 # Kiểm tra bit 4 (GPIO4) trong giá trị đọc được  #bne a3, zero, gpio4\_set # Nếu bit 4 = 1 (GPIO4 HIGH), nhảy tới gpio4\_set  clear:  # Tắt LED (GPIO1 = 0)  li a1, GPIO\_OUT\_W1TC\_REG # Nạp địa chỉ thanh ghi GPIO\_OUT\_W1TC\_REG  li a2, 0x02 # Giá trị 0x02 tương ứng với việc xóa GPIO1  sw a2, 0(a1) # Ghi giá trị vào thanh ghi để tắt LED  j loop # Quay lại đầu vòng lặp  gpio0\_set:  # Bật LED (GPIO1 = 1) nếu GPIO0 = HIGH  li a1, GPIO\_OUT\_W1TS\_REG # Nạp địa chỉ thanh ghi GPIO\_OUT\_W1TS\_REG  li a2, 0x02 # Giá trị 0x02 tương ứng với việc thiết lập GPIO1  sw a2, 0(a1) # Ghi giá trị vào thanh ghi để bật LED  j loop # Nhảy tới nhãn loop  gpio2\_set:  # Bật LED (GPIO1 = 1) nếu GPIO2 = HIGH  li a1, GPIO\_OUT\_W1TS\_REG  li a2, 0x02  sw a2, 0(a1)  j loop  gpio3\_set:  # Bật LED (GPIO1 = 1) nếu GPIO3 = HIGH  li a1, GPIO\_OUT\_W1TS\_REG  li a2, 0x02  sw a2, 0(a1)  j loop  gpio4\_set:  # Bật LED (GPIO1 = 1) nếu GPIO4 = HIGH  li a1, GPIO\_OUT\_W1TS\_REG  li a2, 0x02  sw a2, 0(a1)  j loop |

Output:





Assignment 5

Code:

|  |
| --- |
| .global init  # GPIO Register Definitions  .eqv GPIO\_ENABLE\_REG, 0x60004020 # GPIO output enable register  .eqv GPIO\_OUT\_REG, 0x60004004 # GPIO output register  .eqv IO\_MUX\_GPIO4\_REG, 0x60009014 # GPIO4 function configuration  .eqv IO\_MUX\_GPIO5\_REG, 0x60009018 # GPIO5 function configuration  .eqv IO\_MUX\_GPIO6\_REG, 0x6000901C # GPIO6 function configuration  .eqv IO\_MUX\_GPIO7\_REG, 0x60009020 # GPIO7 function configuration  # 7-segment display patterns for digits 0-9 (Common Anode - active low)  # Segment mapping: GPIO0-6 -> segments a-g  .data  digits:  .word 0xC0 # 0: 1100 0000 - segments a,b,c,d,e,f on (0 = on)  .word 0xF9 # 1: 1111 1001 - segments b,c on  .word 0xA4 # 2: 1010 0100 - segments a,b,d,e,g on  .word 0xB0 # 3: 1011 0000 - segments a,b,c,d,g on  .word 0x99 # 4: 1001 1001 - segments b,c,f,g on  .word 0x92 # 5: 1001 0010 - segments a,c,d,f,g on  .word 0x82 # 6: 1000 0010 - segments a,c,d,e,f,g on  .word 0xF8 # 7: 1111 1000 - segments a,b,c on  .word 0x80 # 8: 1000 0000 - all segments on  .word 0x90 # 9: 1001 0000 - segments a,b,c,f,g on  .text  init:  # Configure GPIO0-7 as outputs  li a1, GPIO\_ENABLE\_REG  li a2, 0xFF # Enable GPIO0-7 as outputs  sw a2, 0(a1)  # Configure GPIO4-7 function as GPIO  li a2, 0x1000 # Set GPIO function  li a1, IO\_MUX\_GPIO4\_REG  sw a2, 0(a1)  li a1, IO\_MUX\_GPIO5\_REG  sw a2, 0(a1)  li a1, IO\_MUX\_GPIO6\_REG  sw a2, 0(a1)  li a1, IO\_MUX\_GPIO7\_REG  sw a2, 0(a1)  display\_loop:  la a3, digits # Load address of digit patterns  li a4, 0 # Counter for digits 0-9    next\_digit:  # Load and display the current digit pattern  lw a2, 0(a3) # Load pattern for current digit  li a1, GPIO\_OUT\_REG  sw a2, 0(a1) # Output pattern to GPIO    # Delay  call delay    # Move to next digit  addi a3, a3, 4 # Next pattern address  addi a4, a4, 1 # Increment counter  li t0, 10  blt a4, t0, next\_digit # If counter < 10, continue    j display\_loop # Repeat from 0  # Delay subroutine  delay:  li t0, 0 # Counter  li t1, 10000000 # Delay duration  delay\_loop:  addi t0, t0, 1  blt t0, t1, delay\_loop  ret |

Output:

|  |  |
| --- | --- |
| 1 | A screenshot of a computer  Description automatically generated |
| 2 | A screenshot of a computer  Description automatically generated |
| 3 | A circuit board with wires and numbers  Description automatically generated |
| 4 | A circuit board with wires connected to it  Description automatically generated |
| 5 | A circuit board with a number and a number on it  Description automatically generated |
| 6 | A circuit board with a number and a number on it  Description automatically generated |
| 7 | A circuit board with many wires  Description automatically generated |
| 8 | A circuit board with a number and wires  Description automatically generated |
| 9 | A circuit board with wires and numbers  Description automatically generated |

1. **Khởi tạo (init):**
   * Cấu hình các chân GPIO0-7 làm đầu ra (output).
   * Cấu hình chức năng GPIO cho các chân GPIO4-7.
2. **Vòng lặp hiển thị (display\_loop):**
   * Tải các giá trị số từ bảng digits.
   * Lặp qua các giá trị số (0-9) và hiển thị lên các chân GPIO.
   * Sau mỗi lần hiển thị, gọi hàm **delay** để tạo độ trễ.
3. **Chuyển sang số tiếp theo (next\_digit):**
   * Cập nhật địa chỉ và chỉ số số tiếp theo.
   * Nếu đã hiển thị đủ 10 số (0-9), quay lại hiển thị từ số 0.
4. **Hàm delay (delay):**
   * Tạo một độ trễ bằng cách sử dụng một vòng lặp không làm gì ngoài việc tăng biến đếm.