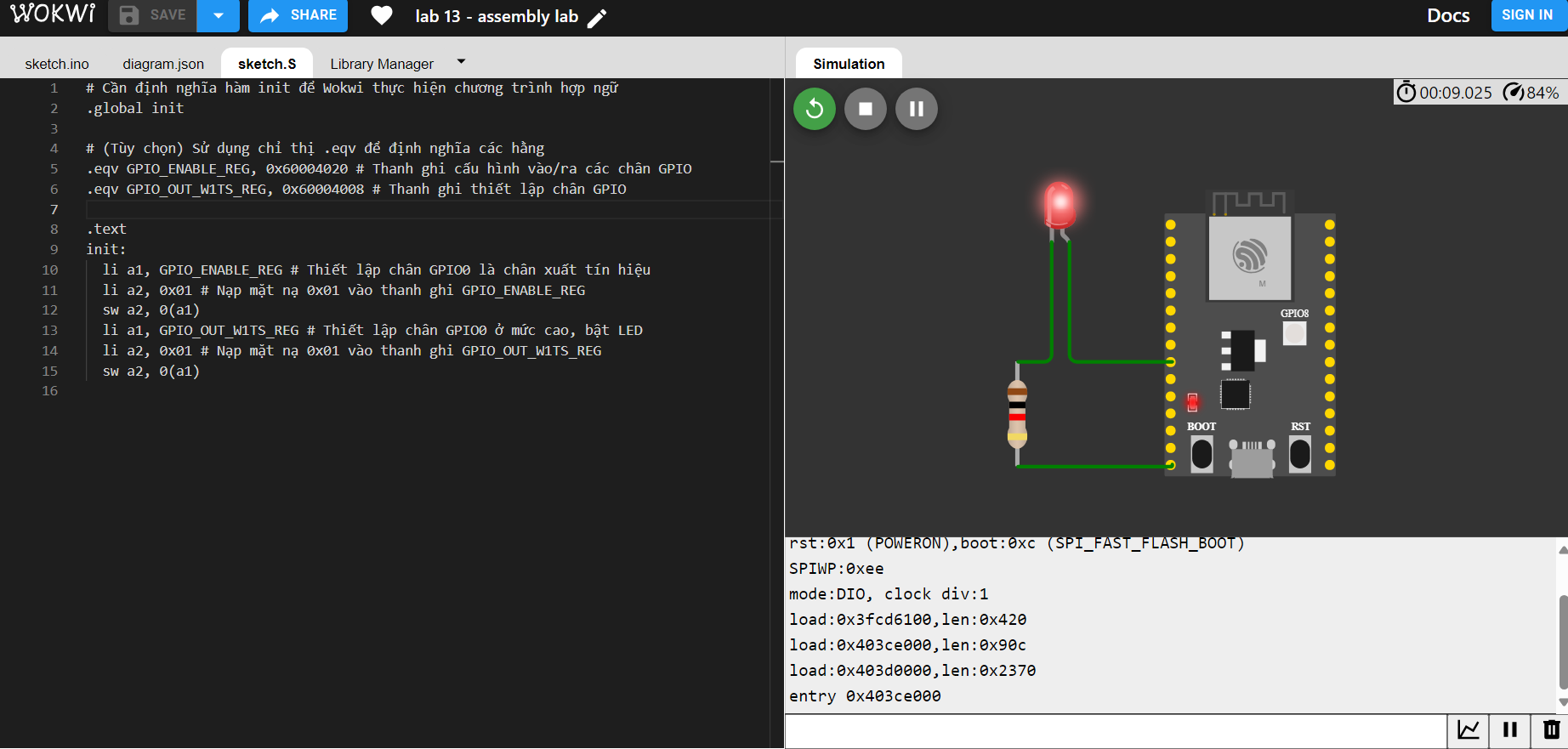
**LAB REPORT**

**IT3280E– 152049– Assembly Language and Computer Architecture Lab**

**Lab 13: Cache memory**

**Assignment 1:**

*Create a project to implement and test Home Assignment 1. Update the source code to test with other GPIO pins (GPIO2, GPIO3, GPIO4)*

****

* For GPIO2:

+ Source code:

|  |
| --- |
| # Cần định nghĩa hàm init để Wokwi thực hiện chương trình hợp ngữ  .global init  # (Tùy chọn) Sử dụng chỉ thị .eqv để định nghĩa các hằng  .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  .text  init:  li a1, GPIO\_ENABLE\_REG  li a2, 0x04  sw a2, 0(a1)  li a1, GPIO\_OUT\_W1TS\_REG  li a2, 0x04  sw a2, 0(a1) |

+ Result:

|  |  |  |
| --- | --- | --- |
| GPIO | Source code | Result |
| GPIO2 | .text  init:  li a1, GPIO\_ENABLE\_REG  li a2, 0x04  sw a2, 0(a1)  li a1, GPIO\_OUT\_W1TS\_REG  li a2, 0x04  sw a2, 0(a1) |  |
| GPIO3 | .text  init:  li a1, GPIO\_ENABLE\_REG  li a2, 0x08  sw a2, 0(a1)  li a1, GPIO\_OUT\_W1TS\_REG  li a2, 0x08  sw a2, 0(a1) |  |
| GPIO4 | .text  init:  li a1, GPIO\_ENABLE\_REG  li a2, 0x10  sw a2, 0(a1)  li a1, GPIO\_OUT\_W1TS\_REG  li a2, 0x10  sw a2, 0(a1) |  |

**Assignment 2:**

*Create a project to implement and test Home Assignment 2. Update the source code to test with other GPIO pins (GPIO2, GPIO3, GPIO4) and adjust the LED blinking duration.*

* Source code update:

|  |  |
| --- | --- |
| GPIO | Source code |
| GPIO2 | init:  li a1, GPIO\_ENABLE\_REG # Setup GPIO0 as output  li a2, 0x04  sw a2, 0(a1)  main\_loop:  li a1, GPIO\_OUT\_W1TS\_REG # GPIO0 -> HIGH  li a2, 0x04  sw a2, 0(a1)  call delay\_asm # Delay    li a1, GPIO\_OUT\_W1TC\_REG # Clear GPIO0  li a2, 0x04  sw a2, 0(a1)  call delay\_asm # Delay    j main\_loop # Loop |
| GPIO3 | init:  li a1, GPIO\_ENABLE\_REG # Setup GPIO0 as output  li a2, 0x08  sw a2, 0(a1)  main\_loop:  li a1, GPIO\_OUT\_W1TS\_REG # GPIO0 -> HIGH  li a2, 0x08  sw a2, 0(a1)  call delay\_asm # Delay    li a1, GPIO\_OUT\_W1TC\_REG # Clear GPIO0  li a2, 0x08  sw a2, 0(a1)  call delay\_asm # Delay    j main\_loop # Loop |
| GPIO4 | init:  li a1, GPIO\_ENABLE\_REG # Setup GPIO0 as output  li a2, 0x10  sw a2, 0(a1)  main\_loop:  li a1, GPIO\_OUT\_W1TS\_REG # GPIO0 -> HIGH  li a2, 0x10  sw a2, 0(a1)  call delay\_asm # Delay    li a1, GPIO\_OUT\_W1TC\_REG # Clear GPIO0  li a2, 0x10  sw a2, 0(a1)  call delay\_asm # Delay    j main\_loop # Loop |

# Assignment 3:

|  |  |  |
| --- | --- | --- |
| Number | 7-digit hex equivalent | Result |
| 0 | 0xC0 | A computer chip with wires and a red light  Description automatically generated with medium confidence |
| 1 | 0xF9 |  |
| 2 | 0x24 |  |
| 3 | 0x30 |  |
| 4 | 0x19 |  |
| 5 | 0x12 |  |
| 6 | 0x02 |  |
| 7 | 0xF8 |  |
| 8 | 0x0 |  |
| 9 | 0x10 |  |

# Assignment 4:

*Create a project to implement and test Home Assignment 4. Update the source code to use other GPIO pins (GPIO2, GPIO3, GPIO4) as signal input pins*

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

* Source code update:

|  |  |  |
| --- | --- | --- |
| GPIO | Source code (only parts that were changed) | Result |
| GPIO2 | .eqv IO\_MUX\_GPIO0\_REG, 0x60009008 # function register GPIO2  # Read status of GPI2  lw a2, 0(a1)  andi a3, a2, 0x04 |  |
| GPIO3 | .eqv IO\_MUX\_GPIO0\_REG, 0x6000900C # function register GPIO3  # Read status of GPI3  lw a2, 0(a1)  andi a3, a2, 0x08 |  |
| GPIO4 | .eqv IO\_MUX\_GPIO0\_REG, 0x60009014 # function register GPIO4  # Read status of GPI4  lw a2, 0(a1)  andi a3, a2, 0x10 |  |

# Assignment 5:

*Create a project to implement a circuit that counts from 0 to 9 on a 7-segment LED display*

* Source code:

|  |
| --- |
| .global init  .eqv GPIO\_ENABLE\_REG, 0x60004020 # Enable output GPIO  .eqv GPIO\_OUT\_REG, 0x60004004 # Output register for GPIO  .eqv IO\_MUX\_GPIO4\_REG, 0x60009014 # Setup function for GPIO4  .eqv IO\_MUX\_GPIO5\_REG, 0x60009018 # Setup function for GPIO5  .eqv IO\_MUX\_GPIO6\_REG, 0x6000901C # Setup function for GPIO6  .eqv IO\_MUX\_GPIO7\_REG, 0x60009020 # Setup function for GPIO7  # Hex values for digits 0-9 (7-segment codes)  # 0xC0 = 0, 0xF9 = 1, 0x24 = 2, 0x30 = 3, 0x19 = 4, 0x12 = 5, 0x02 =6, 0xF8 = 7, 0x0 = 8, 0x10 = 9  .eqv DIGIT\_0, 0xC0  .eqv DIGIT\_1, 0xF9  .eqv DIGIT\_2, 0x24  .eqv DIGIT\_3, 0x30  .eqv DIGIT\_4, 0x19  .eqv DIGIT\_5, 0x12  .eqv DIGIT\_6, 0x02  .eqv DIGIT\_7, 0xF8  .eqv DIGIT\_8, 0x0  .eqv DIGIT\_9, 0x10  .text  init:  # Set GPIO output pins (GPIO0 to GPIO7) as outputs  li a1, GPIO\_ENABLE\_REG  li a2, 0xFF # Set GPIO0 to GPIO7 as output  sw a2, 0(a1)  # Configure GPIO4, GPIO5, GPIO6, GPIO7 for GPIO function  li a2, 0x1000  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)  # Main loop to count from 0 to 9  count\_loop:  li a1, GPIO\_OUT\_REG  # Display 0  li a2, DIGIT\_0  sw a2, 0(a1)  call delay\_asm  # Display 1  li a2, DIGIT\_1  sw a2, 0(a1)  call delay\_asm  # Display 2  li a2, DIGIT\_2  sw a2, 0(a1)  call delay\_asm  # Display 3  li a2, DIGIT\_3  sw a2, 0(a1)  call delay\_asm  # Display 4  li a2, DIGIT\_4  sw a2, 0(a1)  call delay\_asm  # Display 5  li a2, DIGIT\_5  sw a2, 0(a1)  call delay\_asm  # Display 6  li a2, DIGIT\_6  sw a2, 0(a1)  call delay\_asm  # Display 7  li a2, DIGIT\_7  sw a2, 0(a1)  call delay\_asm  # Display 8  li a2, DIGIT\_8  sw a2, 0(a1)  call delay\_asm  # Display 9  li a2, DIGIT\_9  sw a2, 0(a1)  call delay\_asm  # Loop again  j count\_loop  # New delay function  delay\_asm:  li a3, 0 # Counter  li a4, 5000000 # Wait time (counting times)  loop\_delay:  addi a3, a3, 1 # Increment counter  blt a3, a4, loop\_delay # Loop until counter reaches a4  ret |