



Practical Report - 6 IE1024 - Computer Organization and Architecture

1st year 2nd semester – 2024

Student Name: Tharukshan S

Student ID: IT23750456

Task-01

```
1  .data
2  .global __do_copy_data
3
4  source_str: .byte 'H', 'e', 'l', 'l', 'o', 0x00
5  dest_str: .byte 'W', 'o', 'r', 'l', 'd', 0x00
6
7  .text
8  .global main
9
10 main:
11     ldi ZH, hi8(source_str)
12     ldi ZL, lo8(source_str)
13
14     ldi YH, hi8(dest_str)
15     ldi YL, lo8(dest_str)
16
17 copy_loop:
18     ld r16, Z+
19     cp r16, 0x00
20     breq done
21
22     st Y+, r16
23     rjmp copy_loop
24
25 done:
26     rjmp .
27
```

Variables

Call Stack

Breakpoints

Output

SRAM Data Memory x

Address

Symbol

Hex

Decimal

Binary

Char

SRAM Memory

4000

source_str

0x48

72

01001000

'H'

4001

0x65

101

01100101

'e'

4002

0x6C

108

01101100

'l'

4003

0x6C

108

01101100

'l'

4004

0x6F

111

01101111

'o'

4005

0x00

0

00000000

','

4006

dest_str

0x48

72

01001000

'H'

4007

0x65

101

01100101

'e'

4008

0x6C

108

01101100

'l'

4009

0x6C

108

01101100

'l'

400A

0x6F

111

01101111

'o'

400B

0x00

0

00000000

','

Memory

SRAM Data Memory

Format

Symbol

Filter

Address

Assembly_lab_T1 (Build, Load, ...)

4

22:19

INS

Task-02

```

1  .data
2  number: .byte 7 ; Example number (can be modified)
3
4  .text
5  .global main
6
7  main:
8      ldi ZH, hi8(number) ; Load high byte of number address into ZH
9      ldi ZL, lo8(number) ; Load low byte of number address into ZL
10
11     ld r16, Z            ; Load the number into r16
12
13     andi r16, 1          ; Mask the least significant bit (check even or odd)
14     breq is_even         ; If the zero flag is set, number is even
15
16     ; Odd number case: Multiply by 2
17     lsl r16              ; Logical shift left (multiply by 2)
18     rjmp store_result    ; Jump to store the result
19
20 is_even:
21     ; Even number case: Divide by 2
22     lsr r16              ; Logical shift right (divide by 2)
23
24 store_result:
25     st Z, r16            ; Store the result back to memory
26
27     nop                  ; No operation (end of program)

```

Variables		Notifications		SRAM Data Memory x								Program Memory				Output		Call Stack		Breakpoints		
Address		00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	ASCII				
																		SRAM Memory				
4000		00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00				
4010		00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00				
4020		00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00				
4030		00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00				
4040		00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00				
4050		00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00				
4060		00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00				
4070		00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00				
4080		00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00				

Memory SRAM Data Memory Format Hex