Introduction to Computer Organization (ICO, CCIT 4026)

HKU SPACE Community College

(Assignment 04)

Instructions:

1. Fill in the information

		(Insert a photo of your student ID card below)
Student Name:		(No mark would be given if no student ID is presented here)
Student ID:		
Class Number:		

2. **TYPE** all your answers on the space provided with the screen captures of your final assembly source codes (with colors) and the input/output messages on this WORD file, then convert to PDF format for submission.
3. Make sure the source codes are readable and neatly formatted. Unclear, incorrectly-sized or codes of poor readability would NOT be graded.
4. Add adequate amount of program remarks to improve readability.
5. All problems are expected to be answered by using the ***results/instructions*** mentioned in the course materials ONLY. If you have to bring in results/instructions out of this scope, prior consent from the lecturer should be sought and a proper citation has to be provided otherwise the answers would be considered as INCORRECT.
6. Where applicable, it is okay to run (and single-step) your codes on MARS simulator to confirm its functionality before submission.
7. No late submission will be accepted.
8. Please read the Student Handbook for issues related to academic honesty. The College places very high importance on honesty in academic work submitted by students, and adopts a policy of zero tolerance on cheating and plagiarism. Any related offense may be passed to the Guidance Committee. Possible disciplinary action, including termination of study, may follow.
9. By making a submission, you also confirm to have read and understood the above-listed instructions and HKU SPACE Community College's academic honesty policy, and agree to comply with them. Failure to comply with the above instructions may lead to disqualification of your assignment results and possible disciplinary actions.

Q1 [30 marks]

Strings are arrays of characters (bytes) terminated by the null character (byte with value 0). Assuming that $a0 points to a character string, the procedure strlen() returns the length of the string in register $v0 by reading the string one byte at a time until the null character is encountered. Complete the below main program and procedure strlen() by filling in the blanks of the 10 incomplete lines. Write your answers in the table on the following page.

```
1                       .data
2       prompt:         .asciiz "Length of string is: "
3       myString:       .asciiz "this is a random length string."
4
5                       .text
6                       .globl  __start
7       __start:
8                       la      $a0,______              # let $a0 points to myString
9                       jal     ________                # make subroutine call to strlen
10                      add     $t1,$zero,$v0           # save the returned value to $t1
11                      la      $a0,________
12                      li      $v0,4
13                      syscall                         # syscall to print prompt string
14                      add     $a0,$zero,$t1
15                      li      $v0,1
16                      syscall                         # syscall to print the result (length)
17                      li      $v0,10
18                      syscall                         # syscall to terminate program
19
20                      # subroutine to calculate
21                      # string length and
22                      # return in $v0
23
24      strlen:         addi    ___,___,___             # point stack pointer to next word
25                      sw      $a0,______              # push value of $a0 (address pointer) to stack
26                      add     $v0,_____,_____         # initialize $v0 to zero
27      loop:           lb      $t1,0($a0)              # load the next (byte) of string
28                      ___     $t1,$zero,finish        # branch to finish if value is zero (end of string)
29                      ___     $a0,$a0,1               # increment address pointer
30                      addi    ___,___,1               # increment $v0
31                      j       loop                    # loop back for next byte
32      finish:         lw      $a0,0($sp)              # resume the value of $a0 from stack
33                      addi    $sp,$sp,4               # resume stack pointer
34                      jr      ___                     # return from procedure call
```

Answers:

Line 8	
Line 9	
Line 11	
Line 24	
Line 25	
Line 26	
Line 28	
Line 29	
Line 30	
Line 34	

Q2 [40 marks]

Assume the following instruction sequence below runs on a single-cycle processor,

	addi $t1, $zero, 2
L1:	add $t3, $t3, $t1
	sub $t3, $t3, $t2
	beq $t3, $t4, Exit
	j L1
Exit:	add $t4, $t3, $s0

The single-cycle processor's clock cycle time is 200ns. Given the execution of the first instruction starts at time t = 0, and the initial register values are:

$s0 = 10 $t1 = 1 $t2 = 1 $t3 = 2 $t4 = 5

a) What is the total execution time of the single-cycle processor for this sequence (Remarks: consider the given values in the question)? Fill in the register values for each clock cycle of execution on the following table.

[10 marks]

Write ONLY the value when the content of the register is changed

Register value	(t=0)	Clock Cycle														
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$t1 =																
$t2 =																
$t3 =																
$t4 =																
Instruction:																

Total number of instructions executed =	
Total execution time =	

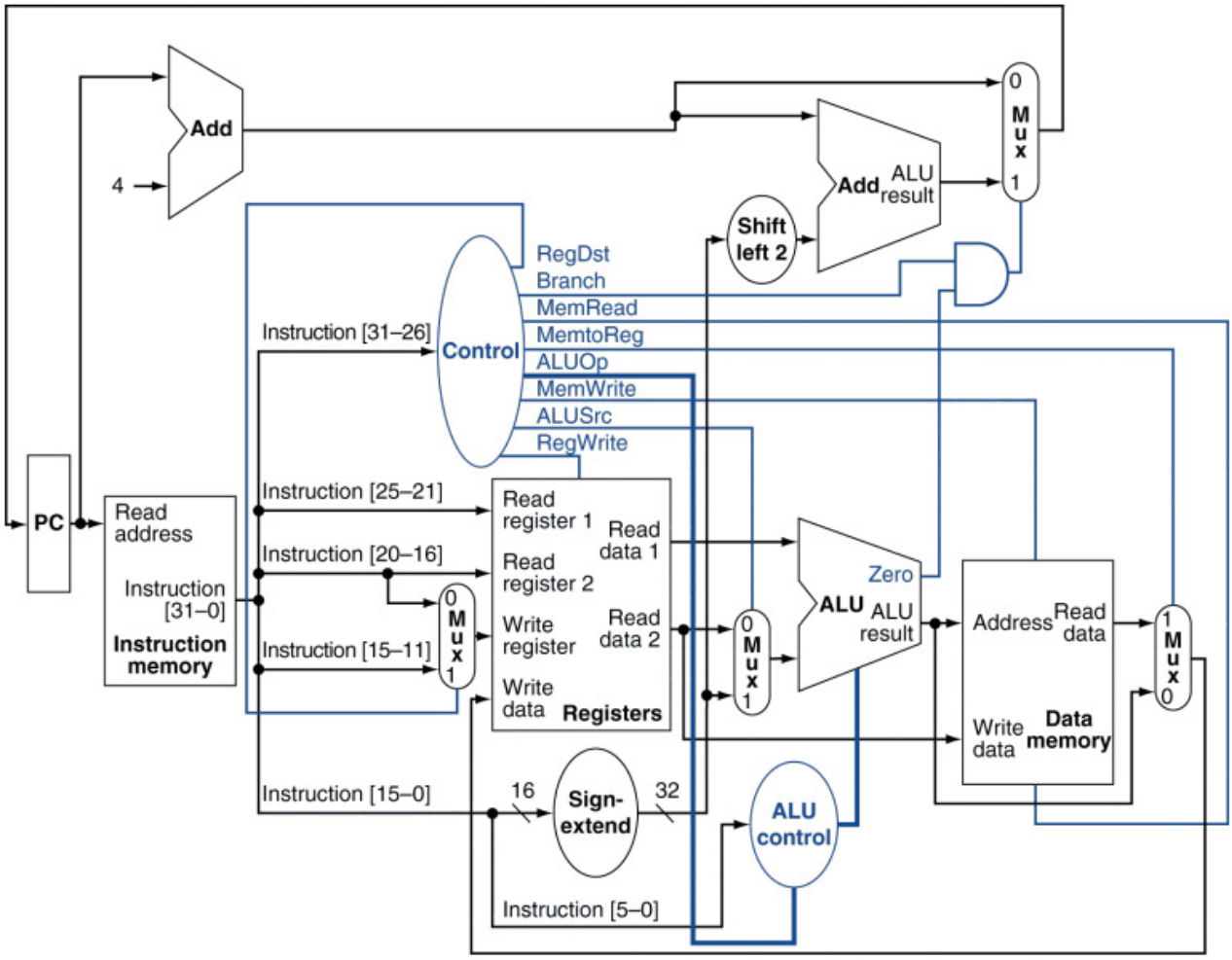
b) What are the expected register values (in decimal) of $t3 and $t4 after executing the instruction sequence?

$t3 =	
$t4 =	

[6 marks]

c) For the single-cycle processor, what are the values (in binary) of instruction[31-26], instruction[25-21], instruction[20-16], instruction[15-11], instruction[5-0], Read data 1, Read data 2, ALU result, and all the control signals when the time is at 2050 ns. The below single-cycle processor diagram can be used as an aid towards finding the answers.

[24 marks]



Determine the values (0, 1 or X) or in binary format of the below signals:

instruction[31-26] =

instruction[25-21] =

instruction[20-16] =

instruction[15-11] =

instruction[5-0] =

Read data 1 output =

Read data 2 output =

ALU result =

RegDst =

ALUSrc =

MemtoReg =

RegWrite =

MemRead =

MemWrite =

Branch =

ALUop[1:0] =

Q3 [30 marks]

A “stuck at 1 or 0 fault” refers to the hardware failure that a control signal is physically stuck at logic high (1) or logic low at all time, regardless of what it should be. In the below cases (from (a) to (e)), comment on the situation about what type of instructions won’t work correctly if that failure happens on the processor shown in Figure 1. Also, give explanations of instruction(s) which would still perform correctly under such failure.

(a) RegWrite = 0

(b) ALUOp[1] = 0

(c) ALUsrc = 0

(d) RegDst = 1

(e) MemWrite = 1

[30 marks]

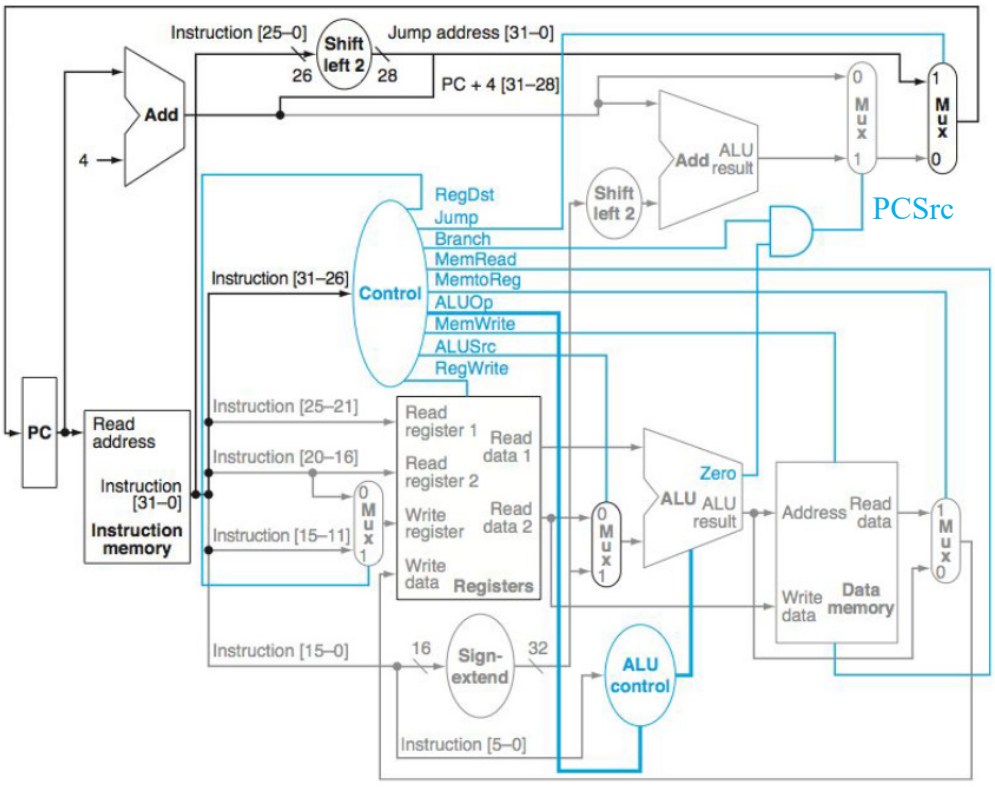


Figure 1 The simple data path processor with the control unit.

For example,

PCSrc = 1:

Instruction Class	Working Y/N?	Explanations
R-type	N	The address generated by the branch target address generator would be updated to PC which causes an incorrect branch and it is not executing the next instruction.
Load Word	N	
Store Word	N	
Branch	N	When the condition is NOT satisfied, the branch target address will still replace the PC instead of PC+4. It works only when the branch condition is satisfied.
Jump	Y	The jump address is still able to update the PC regardless of PCSrc.

Answers:

a. RegWrite = 0:

Instruction Class	Working Y/N?	Explanations
R-type		
Load Word		
Store Word		
Branch		
Jump		

b. ALUOp[1] = 0:

Instruction Class	Working Y/N?	Explanations
R-type		
Load Word		
Store Word		
Branch		
Jump		

c. ALUsrc = 0:

Instruction Class	Working Y/N?	Explanations
R-type		
Load Word		
Store Word		
Branch		
Jump		

d. RegDst = 1:

Instruction Class	Working Y/N?	Explanations
R-type		
Load Word		
Store Word		
Branch		
Jump		

e. MemWrite = 1:

Instruction Class	Working Y/N?	Explanations
R-type		
Load Word		
Store Word		
Branch		
Jump		

--- END OF ASSIGNMENT 4 ---