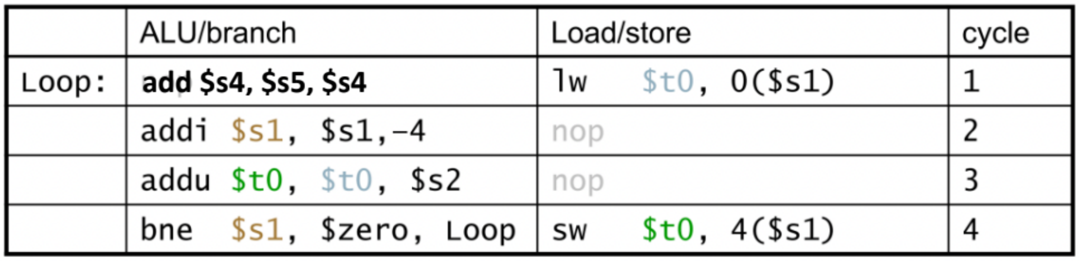
CSE 490/590 Homework 4 Spring 2023

1.

a. Schedule the following instruction sequence for dual-issue MIPS. Consider one ALU/branch instruction and one load/store instruction can be executed in parallel when there is no data dependencies:

```
Loop: lw $t0, 0($s1)       ;$t0=array element
      add $t0, $t0, $s2    ;add scalar in $s2
      sw $t0, 0($s1)       ;store result
      addi $s1, $s1,-4     ;decrement pointer
      add $s4, $s5, $s4    ;update $s4
      bne $s1, $zero, Loop ;branch $s1!=0
```



	ALU/branch	Load/store	cycle
Loop:	**add $s4, $s5, $s4**	lw $t0, 0($s1)	1
	addi $s1, $s1,-4	nop	2
	addu $t0, $t0, $s2	nop	3
	bne $s1, $zero, Loop	sw $t0, 4($s1)	4

b. Compute the IPC in part (a)

IPC = 6/4

2. Show loop unrolling so that there are four copies of the loop body for the following instruction sequence:

```
Loop: lw s0,0(t1)
      add s4,s0,s2
      sw s4,0(t1)
      addi t1,t1,#-8
      bne t1,t2,Loop
```

CSE 490/590 Homework 4 Spring 2023

Assuming `t1` contains the address of the first element and `t2` contains the address of last element. Eliminate any obviously redundant computations and do not reuse any of the registers. Assume you have registers from $s0 to $s16 and $t0 and $t1 available.

```
Loop: lw s0,0(t1)
      add s4,s0,s2
      sw s4,0(t1) // drop addi & bne
      lw s6,-8(t1)
      add s8,s6,s2
      sw s8,-8(t1) // drop addi & bne
      lw s10,-16(t1)
      add s12,s10,s2
      sw s12,-16(t1) // drop addi & bne
      lw s14,-24(t1)
      add s16,s14,s2
      sw s16,-24(t1)
      addi t1,t1,#-32
      bne t1,t2,Loop
```