CSE340: Computer Architecture

Summer 2021

Assignment 1

Q1. Consider the following code is a part of a C program. Now Write a MIPS code for the following C code (15 points):

```
for (int i = 0; i < 4; i++) {
  if (i != A[B[i + 1]]) {
    x = (5 * y) + (6 * z);  // You cannot use multiplication
    y = (6 * x) + A[i];     // or consecutive additions.
    z = B[A[i]] + A[B[i]] - 4;
  }
}
```

where i, x, y, and z are in register $s0, $s1, $s2, $s3, respectively. Base addresses of A and B are in register $s6 and $s7, respectively.

Solution:

```
      # For loop initialization
      addi $s0, $zero, 0    # s0 = i = 0;
      addi $t0, $zero, 4    # t0 = N = 4;

LOOP: slt $t1, $s0, $t0
      beq $t1, $zero, END

      # check i != A[B[i + 1]]
      addi $t1, $s0, 1      # t1 = i + 1
      sll $t1, $t1, 2       # t1 = t1 x 4
      add $t1, $s7, $t1     # t1 = Base addr of B + offset i + 1
      lw $t1, 0($t1)        # t1 = B[i + 1]
      sll $t1, $t1, 2       # t1 = t1 x 4
      add $t1, $s6, $t1     # t1 = Base addr of A + offset B[i + 1]
```

```
      lw $t1, 0($t1)          # t1 = A[B[i + 1]]

      beq $s0, $t1, ELSE      # branch if i == A[B[i + 1]]

      # x = 5y + 6z
      sll $t1, $s2, 2         # t1 = 4y
      add $t1, $t1, $s2       # t1 = 4y + y = 5y

      sll $t2, $s3, 2         # t2 = 4z
      sll $t3, $s3, 1         # t3 = 2z
      add $t2, $t2, $t3       # t2 = 4z + 2z = 6z

      add $s1, $t1, $t2       # s1 = x = 5y + 6z

      # y = 6x + A[i]
      sll $t1, $s1, 2         # t1 = 4x
      sll $t2, $s1, 1         # t2 = 2x
      add $t1, $t1, $t2       # t1 = 4x + 2x = 6x

      sll $t7, $s0, 2         # t7 = i x 4
      add $t2, $s6, $t7       # t2 = Base addr of A + offset i
      lw $t2, 0($t2)          # t2 = A[i]

      add $s2, $t1, $t2       # s2 = y = 6x + A[i]

      # z = B[A[i]] + A[B[i]] - 4
      sll $t2, $t2, 2         # t2 = A[i] x 4
      add $t1, $s7, $t2       # t1 = Base addr of B + offset A[i]
      lw $t1, 0($t1)          # t1 = B[A[i]]

      add $t2, $s7, $t7       # t2 = Base addr of B + offset i
      lw $t2, 0($t2)          # t2 = B[i]
      sll $t2, $t2, 2         # t2 = B[i] x 4
      add $t2, $t2, $s6       # t2 = Base addr of A + offset B[i]
      lw $t2, 0($t2)          # t2 = A[B[i]]

      add $t1, $t2, $t1       # t1 = B[A[i]] + A[B[i]]
      addi $s3, $t1, -4       # s3 = B[A[i]] + A[B[i]] - 4

ELSE: addi $s0, $s0, 1        # i = i + 1
      j LOOP
END:
```

Q2. Write a MIPS code that uses a loop to calculate the 10^{th} Fibonacci number and store that number in register $s1. You are not allowed not use any pseudo-instruction. (6 points)

Solution:

```
      addi $t0, $zero, 0    # t0 = 0
      addi $t1, $zero, 1    # t1 = 1

      # loop initialization
      addi $s0, $zero, 9    # n = 9
      addi $t3, $zero, 0    # i = 0

LOOP: slt $t4, $t3, $s0     # check i < n
      beq $t4, $zero, END

      # Fib calculation
      add $s1, $t1, $t2     # s1 = t0 + t1
      add $t1, $t2, $zero   # t1 = t2
      add $t2, $s1, $zero   # t2 = s1
      addi $t3, $t3, 1      # i = i + 1
      j LOOP
END:
```

Q3. Assume following instruction is executed:

```
bne $10, $11, -1024
```

If the PC has the value 0x00001240 then calculate the branch target address. You must show all calculation steps in binary following the branch target address calculation diagram.(4 points)

Solution:

offset in decimal $= -1024 \times 4 = -4096$

offset in hex $= -0x1000$

branch target address $= 0x00001240 + 4 - 0x1000 = 0x00000244$