SUHAS JAIN 19CS30048

1) (a) 80% of the process will be sped up by multi core processor.

$$\text{Maximum Sped up} = \frac{Te}{Te\left(0.2 + \frac{0.8}{80}\right)}$$

$$= \frac{Te}{0.21\,Te}$$

$$= \boxed{4.761}$$

(b) Maximal sped up $= (0.2 + 0.8) \times 80$
(Gustavson - bassis law)

$$= \boxed{64.2}$$

Reason: Amdahl's law is based on assumption of a fixed problem size, ie. workload does not change when performance is increased. Gustavson's law says size of problem changes to best exploit the computing powers available. Therefore the formula and hence values of maximum sped up is different.

SUHAS JAIN 19CS30048

2) Loop will be executed only once.

Total clock cycles = 5+3+3+3+3+3+3+2+4
= 29 clock cycles.

Time for 1 clock cycle = $\frac{1}{3.5}$ ns.

Total time = $\frac{29}{3.5}$ = 8.28 ns

3)

(a) C statement

```
int f = (x==0) ? z : y;
```

(b)

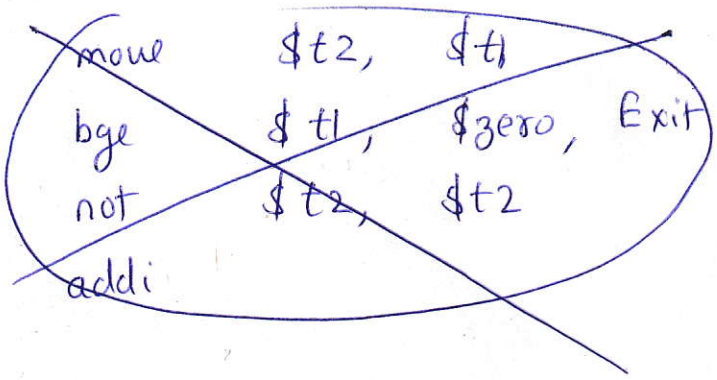
```
beq   $t1, $zero, L2
L1 :  add  $t4, $t2, $zero
L2 :  add  $t4, $t3, $zero
      add  $t0, $t4, $zero
```

```
beq   $t1, $zero, L1
move  $t0, $t2
j exit
L1:   move $t0, $t3
exit: ----
```

SUHAS JAIN 19CS30048

4) Ans: (d) More instruction count, less CPI, higher clip yeild, better performance

5)



~~sra $t2, $t1, 31~~
~~xor $t2, $t1, $t2~~

```
sra  $t3, $t1, 31    #$t3 = -1 if N<0
                          else 0.
xor  $t2, $t3, $t1
sub  $t2, $t2, $t3
```

6) (a) Program computes GCD of 2 numbes stored in registers $a0 and $a1.

(b) Output:

The value is: 3

SUHAS JAIN 19CS30048

7) The number 0xdcba in its binary representation has its MSB as 1. That means sign bit is 1. So during sign extension it will have all 1's in top 16 MSB's before adding. Instead for correct answer they should be 0.

Final value stored in $t0. =

0x1233dcba

Instead we should have written ori instead of addi.

8)
```
lui  $t1, 1
ori  $t1, $t1, 5698
```

9) Out of the 4 inputs atleast 1 should be 1, others can be anything.

Full adder:

both 0 for 1 case

atleast 1 bit 1 for 7 cases.

NOR

0 for 3 cases

1 for 1 case

NAND

0 for 1 case

1 for 7 cases.

SUHAS JAIN 1MS20048

Output will be 0 if all input bits are 0 else 1.

Cases when output will be 0.

$= 3 \times 1 \times 1 = 3$

Cases when it will be 1

$= 256 - 3$

$= \boxed{253}$

10)

```
toupper:
    la   $t0, $a0

loop:
    lb   $t1, ($a0)
    beqz $t1, exit
    blt  $t1, 'a', no_change
    bgt  $t1, 'z', no_change
    addiu $t1, $t1, -32
    sb   $t1, ($a0)

no_change:
    addiu $a0, $a0, 1
    j loop

exit:
    la $a0, $t0
    jr $ra.
```