Question 1

What is the speed up that could be achieved according to **Amdahl's Law** for infinite number of processors if **5%** of a program is sequential and the remaining part is ideally parallel ?

According to Amdahl's law speed up will be limited by the serial portion of the program even if infinite number of processors are used for parallelizing the execution task:

S = 1/ (1-P)

where p is the parallel part/portion of the program

Given, sequential part of program is 5%. So parallel part of the program (P)

= 1 - sequential part

= 1 - 0.05 (or 5%)

= 0.95 (or 95%)

Now S = 1 / (1-P)

ie S = 1 / (1-0.95)=1 / 0.05

= 20

Question 2

You have a system that contains a special processor for performing floating-point operations. You have determined that 50% of your computations can use the floating-point processor . Given the speedup of the floating point processor is 8.

a) Compute the overall speedup achieved by using the floating-point processor.

b) Compute the overall speedup achieved if you modify the compiler so that 62% of the computations can use the floating-point processor.

a) Given ,

$Fraction_{enhanced} = 0.5$, $speedup_{enhanced} = 8$.

overall speedup = ?

overall speedup = 1/((1-0.5) + 0.5/8) = 1/(0.5 + 0.0625) = 1.78.

b) Given ,

$Fraction_{enhanced} = 0.62$, $speedup_{enhanced} = 8$.

overall speedup = ?

overall speedup = 1/((1-0.62) + 0.62/8) = 1/(0.38 + 0.0775) = 2.186.

Tutorial - 4

Arithmetic and Logic Instructions

Question-6

Explain the result of the following set of instructions ?

a)
```
li  $s0, -2147483648
li $s1 , 2147483647
div $s2,$s0,$s1
div $s2 $s0,$s1
rem $s3,$s0,$s1
```

b)
```
li  $s0, -2147483648
li  $s1 , -2147483647
rem $s3,$s0,$s1
```

c)
```
lui  $s0, 32768
lui  $s1 ,1
divu $s2 $s0,$s1
remu $s3,$s0,$s1
```

Question-6

Q6 . Sol :

a) This is signed number operation.o/p quotient -1 and Remainder -1. NO overflow actual o/p and expected o/p are same.

b) This is signed number operation.o/p quotient 1 and Remainder -1. NO overflow actual o/p and expected o/p are same.

c) o/p quotient 32768 and remainder 0.”No carry as the o/p is within the range of numbers that can be represented by using the unsigned number representation. actual o/p and expected o/p are same.

Tutorial - 3

Question - 4

Q4. Convert the following C statements to equivalent MIPS assembly language. Assume that the variables f, g, h are assigned to registers $s0, $s1, $s2 respectively.

Assume that the base address of the array A and B are in registers $s6 and $s7 respectively.

a) f = g + h + B[4]

b) f = h + A[B[4]]

Solution -4

a) Given : f = g + h + B[4]

f =$s0 ,g=$s1 ,h=$s2

A=$s6,B=$s7

sol: MIPS code : lw $t0,16($s7)

add $s0,$s1,$s2

add $s0,$s0,$t0

Solution -4

Contd…

b) Given : Given : f = h + a[B[4]]

f =$s0 ,g=$1 ,h=$s2

A=$s6,B=$s7

sol : MIPS code : lw $t0,16($s7)

```
sll $t1,$t0,2
add $t2, $t1, $s6
lw $t3, 0($t2)
add $s0, $s2, $t3
```

Question - 5

Q5. If the instruction set of the MIPS processor is modified, the instruction format must also be changed. For each of the suggested changes, show the size of the bit fields of an R-type format instruction. What is the total number of bits needed for each instruction?

a) 8 registers.
b) 10 bit immediate constants
c) 128 Registers.

Solution -5

a) Given , 8 registers

sol : bits req for representing reg is $\log_2 8 = 3$

R-type Ins format :

Opcode(6)	Rs(3)	Rt(3)	Rd(3)	Shmt(5)	Funct(6)

Total bits req : 26 bits

b) Given , 10 bit immediate constants

R-type Ins format remains same as immediate constants are not used directly in R-type instruction formats .

Solution -5

C) Given 128 registers,

sol : bits req to represent regs : $\log_2 128 = 7$.

R-type ins format :

Opcode(6)	Rs(7)	Rt(7)	Rd(7)	Shmt(5)	Funct(6)

Total bits required : 38