                                CS220-ASSIGNMENT:7(PROJECT REPORT)

 N.Hemasai 210650

 A.Sai Sreeja 210168

PDS1)

Deciding the registers and the usage protocol:

We have taken an array of 32 registers of size 6 bits

1)reg [0] =0 for li,la

2-3) reg [2], reg [3] =return value from function

             4-7) reg [4]-reg [7] =function arguments

             8-17) reg [1]-reg [8] =saved temporary variables

             18-25) reg [9]-reg [16] =temporary variables

               26-27) reg [26], reg [27] =OS kernel

             28)reg [28] =global pointer

             29)reg [29] =stack pointer

             30)reg [30] =frame pointer

             31)reg [31] =return address

             PDS2)

             Size for instruction memory=27 words of 32 bits (first 27)

             Size for data memory=20 words of 32 bits (next 20)

             PDS3)

             Instruction layout for R, I, J type instructions:

            1)R-type:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Opcode | rs | rt | rd | shamt | funct |
| 6 bits | 5 bits | 5 bits | 5 bits | 5 bits | 6 bits |

The Following Operations are encoded in R-type Instruction format-

* add r0, r1, r2
* sub r0, r1, r2
* addu r0, r1, r2
* subu r0,r1,r2
* and r0,r1,r2
* or r0,r1,r2
* slt r0,r1,r2
* slti 1,2,100

          2)I-type:

|  |  |  |  |
| --- | --- | --- | --- |
| Opcode | rs | rt | Immediate (or address) |
| 6 bits | 5 bits | 5 bits | 16 bits |

The Following Operations are encoded in I-type Instruction format-

* addi r0,r1,1000
* addiu r0,r1, 1000
* andi r0,r1, 1000
* ori r0,r1, 1000
* sll r0, r1, 10
* srl r0, r1, 10
* lw r0,10(r1)
* sw r0,10(r1)
* beq r0,r1,10
* bne r0,r1,10
* bgt r0,r1,10
* bgte r0,r1, 10
* ble r0,r1, 10
* bleq r0,r1, 10
* jr r0

3)J-type:

|  |  |
| --- | --- |
| Opcode | Address |
| 6 bits | 26 bits |

       The Following Operations are encoded in J-type Instruction format-

* j 10
* jal 10

PDS 6,7,8)

For PDS 6,7,8) we used the following format for the opcodes

R-tpe:opcode=1

I-type:opcode>=4 to 10

J-type:opcode =2,3

Addu opcode=0 funct=32

Subu opcode=0 funct=34

And ,andi opcode=1,2

Or,ori opcode=3,4

Xor,xori opcode=5,6

Shift left  opcode=7

Shift right opcode=8

Slt opcode=9