4/29/2021

**AMINA iQADEER-359607**

**COMPUTER iORGANIZATION iLAB i-7**

**CE-42-A**

**Ma’am iRIMSHA iTARIQ**

Data iTransfer iInstructions: iStore iWord

**LAB-07- iReport**



**LAB iTASKS:**

**1. iAssume ithe ifollowing idata ideclarations**

num1: i.word i651

num2: i.word i42

ans1: i.word i0

**Perform ithe ifollowing ioperation, iget ithe iresult iin ia iregister iand ithen istoring ithe iregister iback ito i**

**The imemory.**

ans1 i= inum1 i+ inum2

**CODE:**

.data

i i i i inum1: i i.word i651

i i i i inum2: i i.word i42

i i i i ians1: i i.word i0

.text

.globl imain

main:

i i i i ili i$s0, i0x1001000 i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i# iinitializing ithe ibase iregister i i0r iloading ithe ibase i i

i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i# iregister iwith imemory ilocation iof ithe iarray

i i i i ilw i$s1, i0($s0) i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i#incrementing ithe ibase iaddress iwith imemory

i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i#of i1,2…till i3rd iplace

i i i i ilw i$s2, i4($s0) i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i

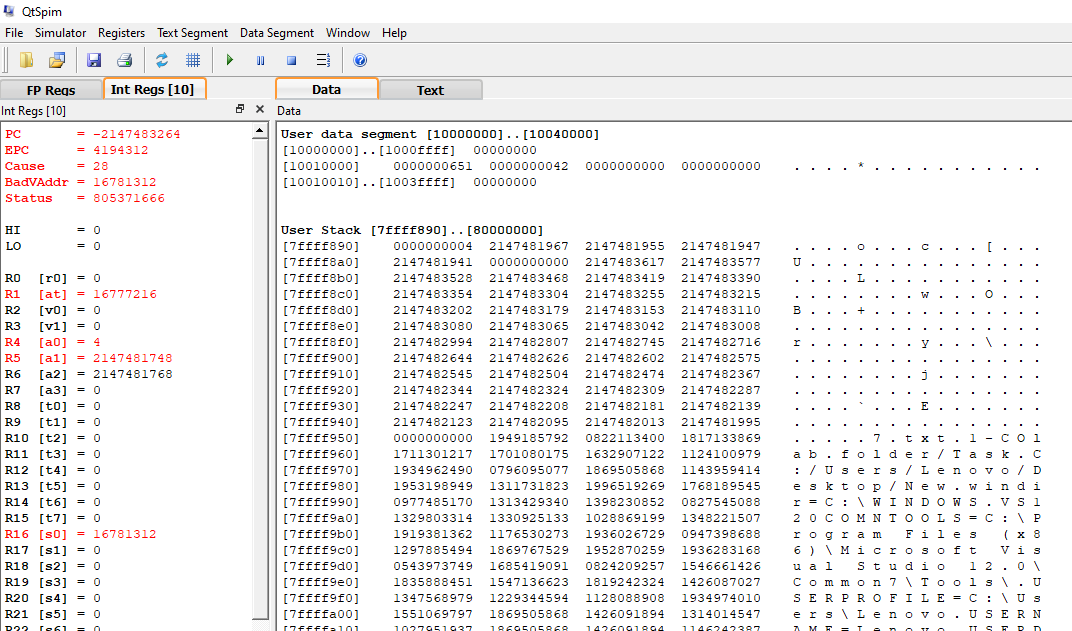
i i i i ilw i$s3, i8($s0)

i i i i iadd i$s3, i$s2, i$s1 i i i i i i i i i i i i i

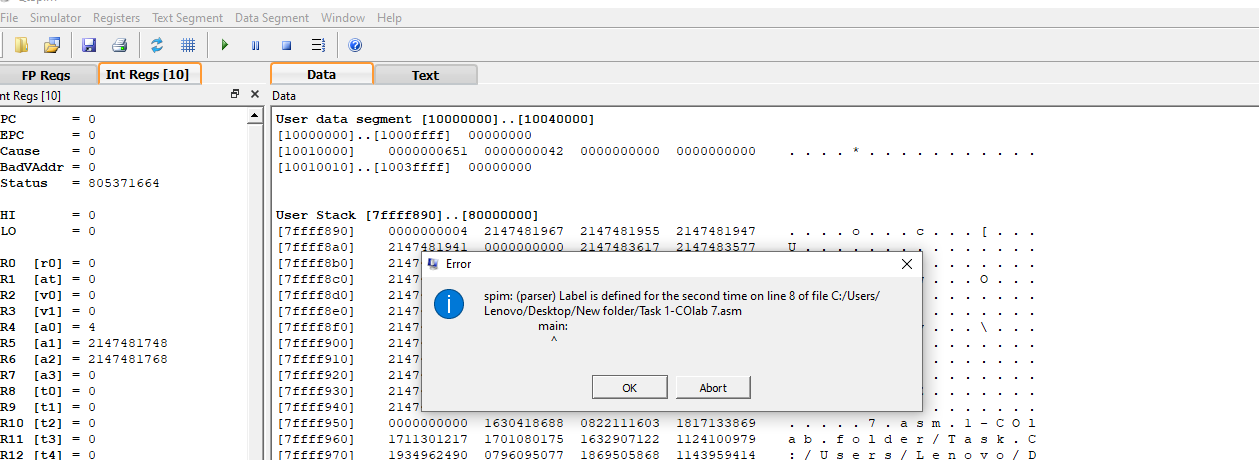
i i i i isw i$s3, i8($s0) i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i#storing ithe ianswer iregister iback iin imemory

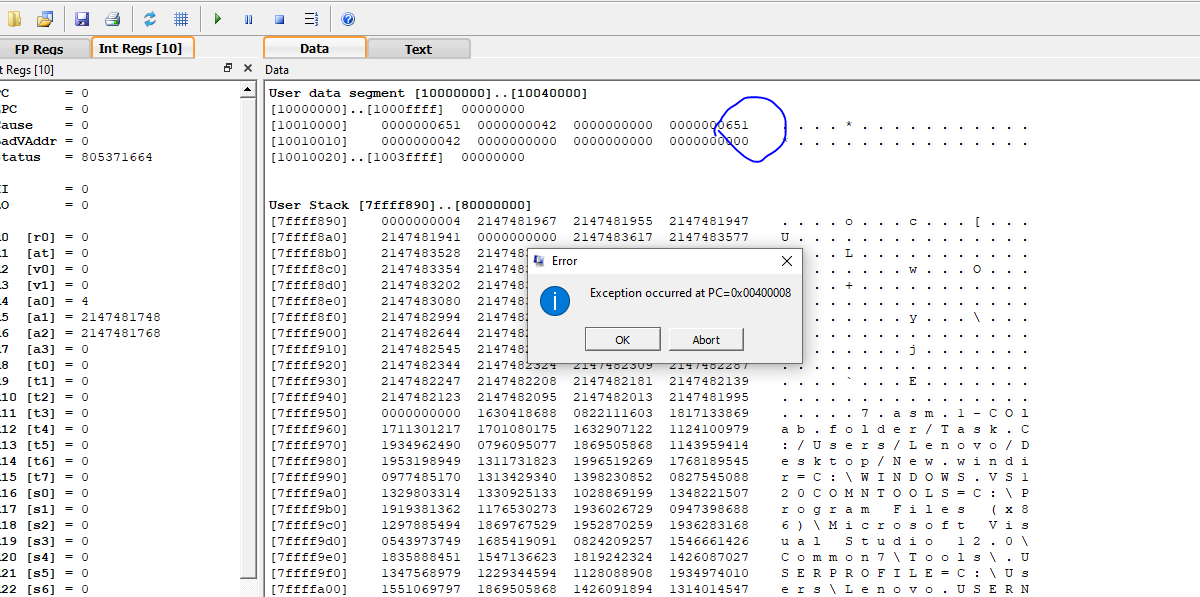
i i i i ili i$v0, i10 i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i# iTo iexit ior iend ithe icode

i i i i iSyscall i

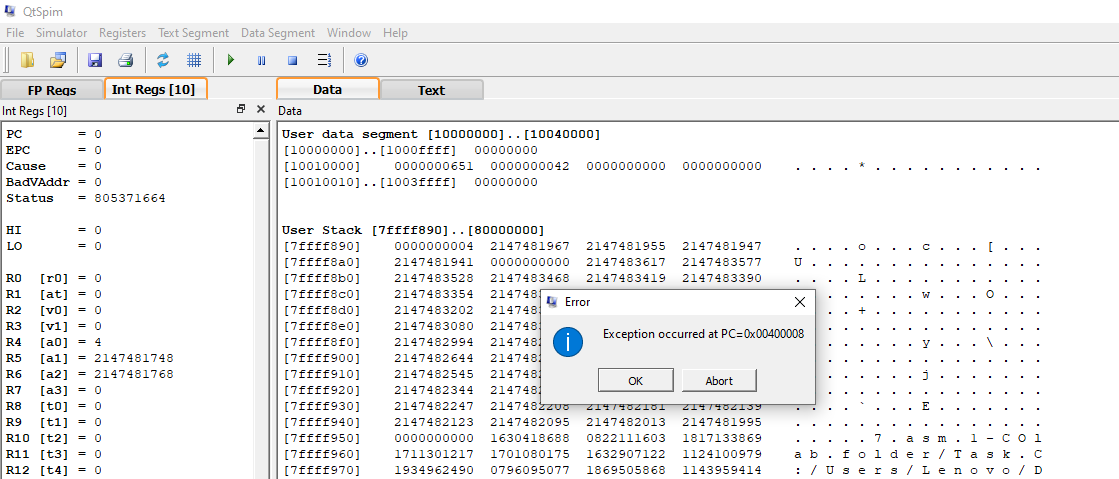


**TWO iERRORS iOCCURRED!!**



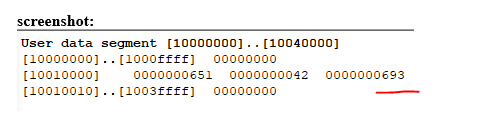


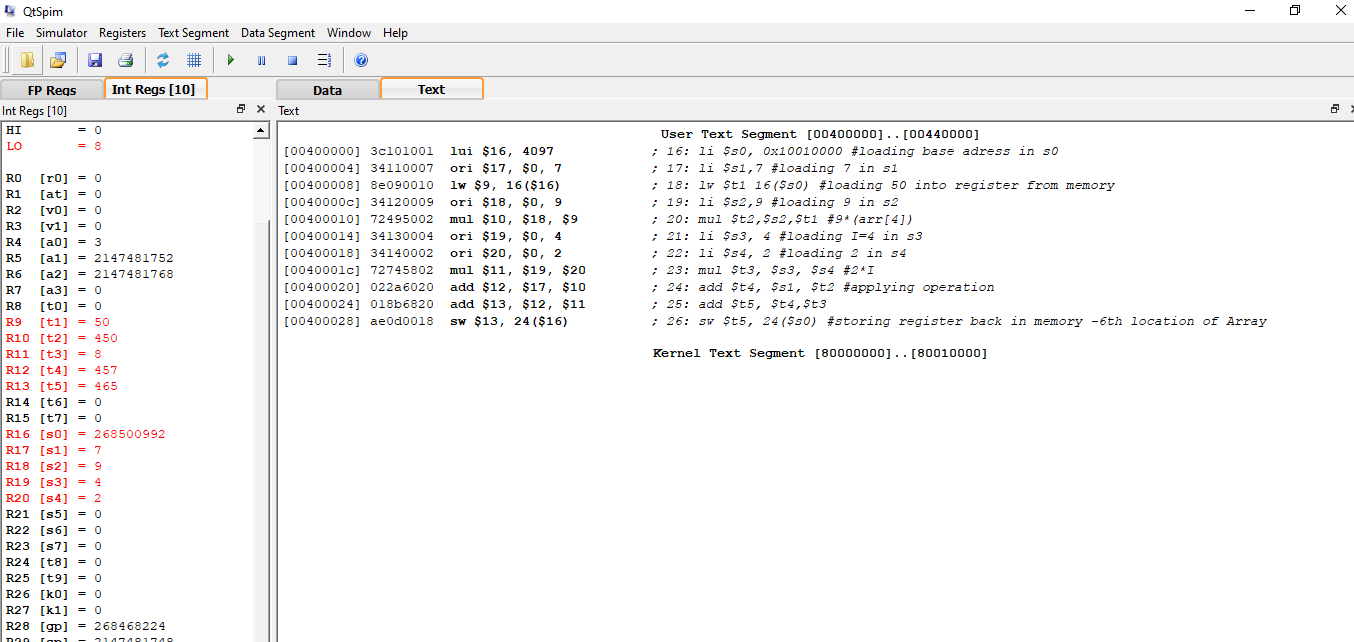
My ianswer iwas inot iappearing iin ianswer iregister ielse imy icontent iof ifirst iregister iwas irepeating**.**



I iclicked iok.

I iredid iit imany itimes. iAnd ithen ifinally idesired ianswer iappeared.





**2. iWrite ithe iassembly ifor ithe ifollowing istatement:**

#Arr[6]= i2\*I i+ i7 i+9(arr[4]) i

# iAns i= i2\*(4) i+ i7 i+ i9\*(50) i= i465

.data

arr: i.word i10 i20 i30 i40 i50 i60 i70 i80 i90 i100 i110 i120 i130 i140

.text i

.globl imain i

main: i

i i i ili i$s0, i0x10010000 #loading ibase iaddress iin is0

i i i ili i$s1,7 #loading i7 iin is1

i i i ilw i$t1 i16($s0) i i i i i i i i i i#loading i50 iinto iregister ifrom imemory

i i i ili i$s2,9 #loading i9 iin is2

i i i imul i$t2,$s2,$t1 #9\*(arr[4])

i i i ili i$s3, i4 #loading iI=4 iin is3

i i i ili i$s4, i2 #loading i2 iin is4

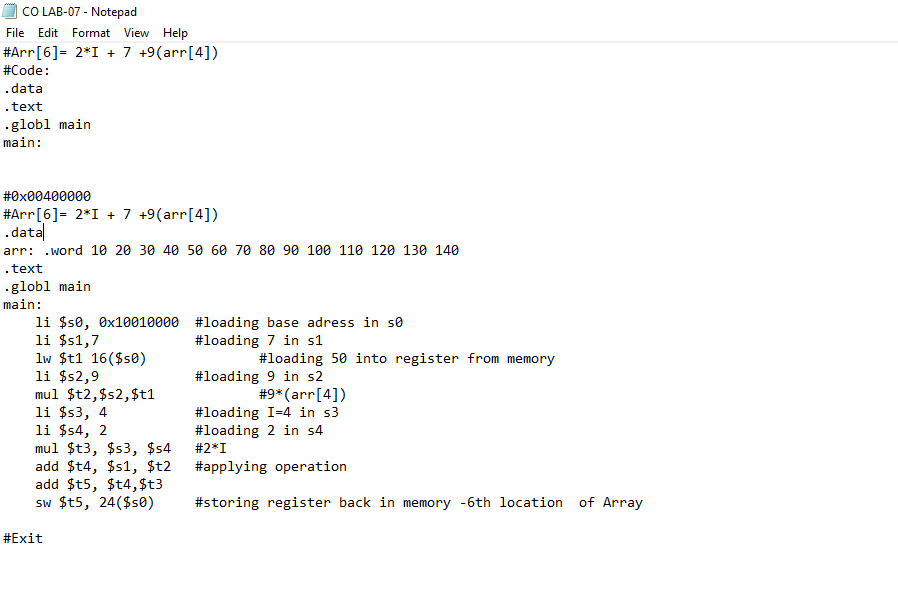
i i i imul i$t3, i$s3, i$s4 #2\*I

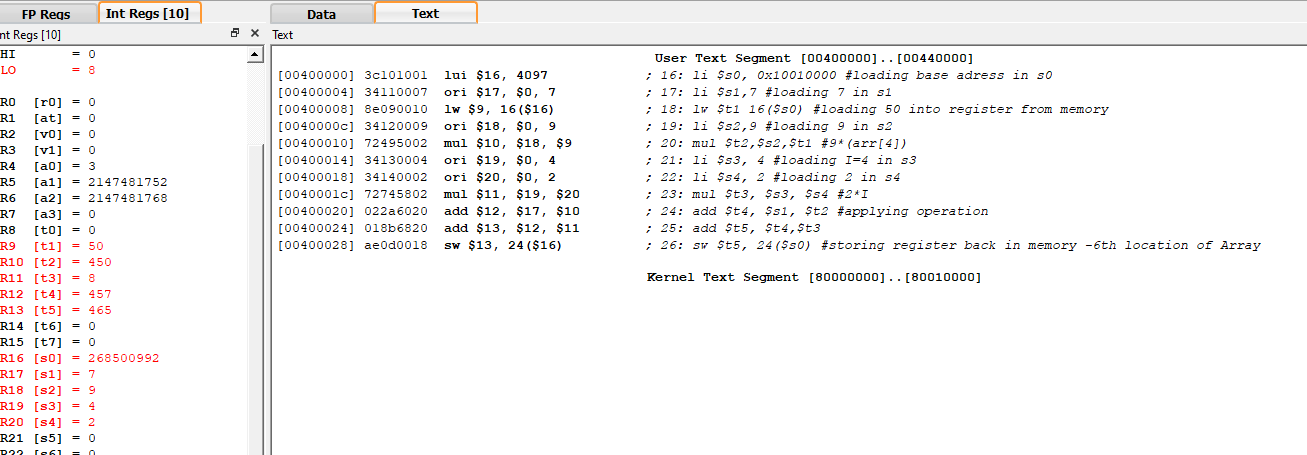
i i i iadd i$t4, i$s1, i$t2 i i i#applying ioperation

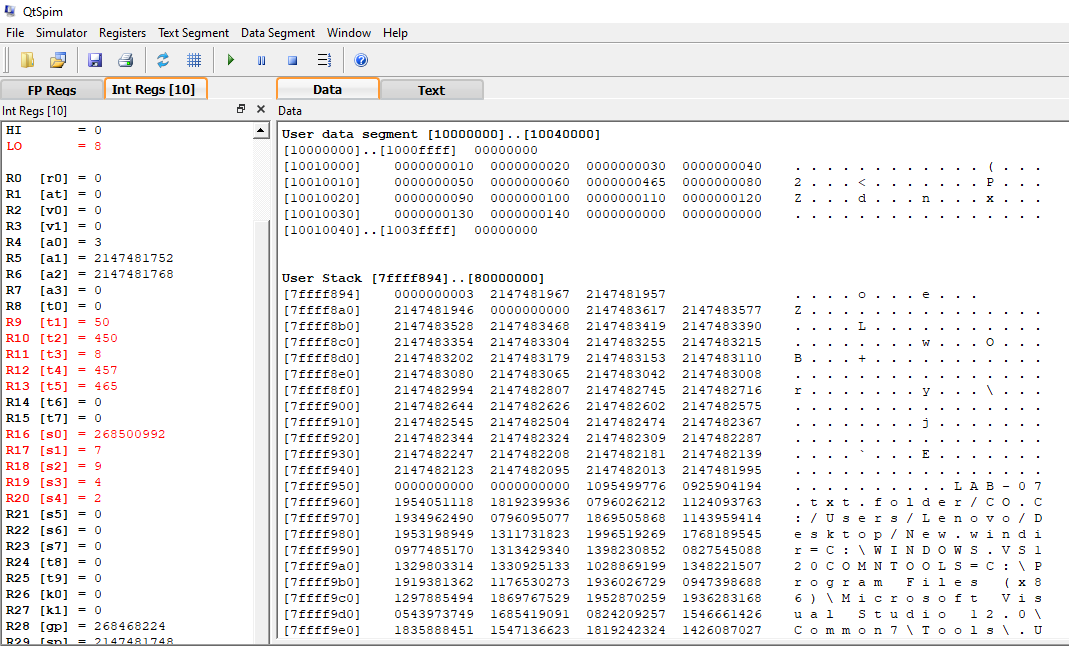
i i i iadd i$t5, i$t4,$t3

i i i isw i$t5, i24($s0) i i i i i#storing iregister iback iin imemory i-6th ilocation i iof iArray

#Exit







**Question i#1: i iWhat iis idifference ibetween iload iword iand istore iword?**

In icomputer iengineering, ia iload–store iarchitecture iis ian iinstruction iset iarchitecture ithat idivides iinstructions iinto itwo icategories: imemory iaccess i(load iand istore ibetween imemory iand iregisters) iand iALU ioperations i(which ionly ioccur ibetween iregisters).

A i**word** igenerally imeans ithe inumber iof ibits ithat ican ibe itransferred iat ione itime ion ithe idata ibus, iand istored iin ia iregister. iIn ithe icase iof i**MIPS**, ia i**word** iis i32 ibits, ithat iis, i4 ibytes. i**Words** iare ialways istored iin iconsecutive ibytes, istarting iwith ian iaddress ithat iis idivisible iby i4.

i**When ia iword i(4 ibytes) iis iloaded ior istored ithe imemory iaddress imust ibe ia imultiple iof ifour.**

Now,

**LOAD iWORD: i**A i**load** ioperation icopies idata ifrom imain imemory iinto ia iregister. i

It imeans, i**load** iinto iregister iReg iDest ithe i**word** icontained iin ithe iaddress iresulting ifrom iadding ithe icontents iof iregister iResource iand ithe iOffset ispecified.

**STORE iWORD:** i

The iSW iinstruction istores idata ito ia ispecified iaddress ion ithe idata imemory iwith ia ipossible ioffset, ifrom ia  
source iregister.

It's isyntax iis:  
**SW i$source iregister's iaddress, ioffset($destination iregister's iaddress)**.

The isample iSW iinstruction idemonstrated iin ithe idata ipath iabove iis i**SW i$2, i($5)**.

A istore ioperation icopies idata ifrom ia iregister iinto imain imemory**.** iThe istore iword iinstruction, isw, icopies idata ifrom ia iregister ito imemory. iThe iregister iis inot ichanged. iThe imemory iaddress iis ispecified iusing ia ibase/register ipair. i... iAs iwith ithe iLW iinstruction, ithe imemory iaddress imust ibe iword ialigned i(a imultiple iof ifour).

**For iexample:** i i i i i i i i i i i i ilw i$t,C($s) i i i i i# i$t i= iMemory[$s i+ iC]

lw i$t1, i12($t0) i# iLoads ithe ivalue iof ithe i3rd ielement iof ivariable iarray0 iusing ithe iaddress istored iof iarray0 iin i$t0 iinto i$t1

In ithe isw iinstruction ithe ileft ioperand iregister iis istored ito ithe imemory iaddress ibased ion ithe iright ioperand iregister.

i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i iSw i$t, iC($s) i i i i i# iMemory[$s i+ iC] i= i$t

**for iexample:**

lw i(load iword) iand isw i(store iword) iinstructions iare iused ito imove idata ibetween iRAM iand ia iregister. iThe iinstruction

i i i i i i i ilw i i i i i i$8, iB

Moves ia iword ifrom ilocation iB iin iRAM ito iregister i$8. iThe iinstruction

i i i i i i i isw i i i i i i$10, iA

moves ia iword ifrom iregister i$10 ito imemory ilocation iA.

**4. iDescribe ibriefly ihow ithe idestination imemory iaddress iis icalculated iin ia istore iword iinstruction?**

Let’s iunderstand iby ihelp iof ian iexample:

Initializing ian iarray:

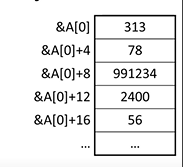
Int iA i[5]: i1, i2,3,4,5,6,7,8

If iwe istart iassuming ithat ihow ithis iarray imight ibe istored iin ibackend iof ithe imemory. iThe ifirst ielement istored iin ithe iindex ia i[0] iin iMIPS ihas iallocated ia ipart iof imemory ithat iis ifixed iand ithat iis i0x i10010000 i.This imemory iaddress iis istored iin ia ibase iregister ithat iremains ithe isame iand iany iwhatever iwe istore iat ifirst iplace iof iarray, ithat inumber iat iindex izero ihas iinsanely ithis iaddress.so iin imips iwe iuse ithis ibase iaddress ito iaccess iother iregisters iat iother iindex inumbers iof iarray. iA i**word** igenerally imeans ithe inumber iof ibits ithat ican ibe itransferred iat ione itime ion ithe idata ibus, iand istored iin ia iregister. iIn ithe icase iof i**MIPS**, ia i**word** iis i32 ibits, ithat iis, i4 ibytes. **i**When ia iword i(4 ibytes) iis iloaded ior istored ithe imemory iaddress imust ibe ia imultiple iof ifour**.**

iTo istore iinstructions iin imemory, ithere iis ian iaddress iregister, iwhich ikeeps itrack iof iwhere ia igiven iinstruction ior ipiece iof idata iis istored iin imemory. iEach istorage ilocation iin imemory iis iidentified iby ian iaddress, ijust ias ieach ihouse ion ia istreet ihas ian iaddress. iA istorage iregister, iwhich itemporarily iholds idata itaken ifrom ior iabout ito ibe isent ito imemory.

**EXAMPLE1**

* **Now iin iorder ito iaccess idestination imemory iaddress iof isecond iindex ivariable iin istore iword iinstructions, iwe iwrite ilike:**

**$A i[0] i+4 i i i i i i i i i i i i**

**-Where iOperation iName iRegister, iNumerical ioffset i(Register) i**

**$A i[0] i+8**

**$A i[0] i+12**

**$A i[0] i+16**

**EXAMPLE2:**

* The istore iword iinstruction, isw, icopies idata ifrom ia iregister ito imemory. iThe iregister iis inot ichanged. iThe imemory iaddress iis ispecified iusing ia ibase/register ipair.

Sw i i it, ioff i(b) i i i i i i i i# iWord iat imemory iaddress i(b+ ioff) i<— i$t i

i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i# iB iis ia iregister. iOff iis i16-bit itwo's icomplement.

**EXAMPLE3:**

* Look iat iregisters i**$12** iand i**$13** iand imemory i(at iright). iIf iI iwrite ithe iinstruction ithat iputs ithe iword i0xFFFFFFFF iinto imemory ilocation i0x0004000C i(storing)
* Register i$12 icontains i0xFFFFFFFF
* Register i$13 icontains i0x00040014

Sw i$12, i0xFFF8 i($13) i

iSw i$12, i-8($13)

The iabove istatement imeans ithat ithere iis ia ibase iregister i$13 icontaining imemory iaddress ithat iadds ito ithe iOffset ivalue iFFF8.The iresulting imemory iaddress iis ilocation iwhere icontent iof i$12 iwill ibe isaved. iThis iis ihow ithe idestination imemory iaddress iis icalculated iin ia istore iword iinstruction.

i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i i**THE iEND**