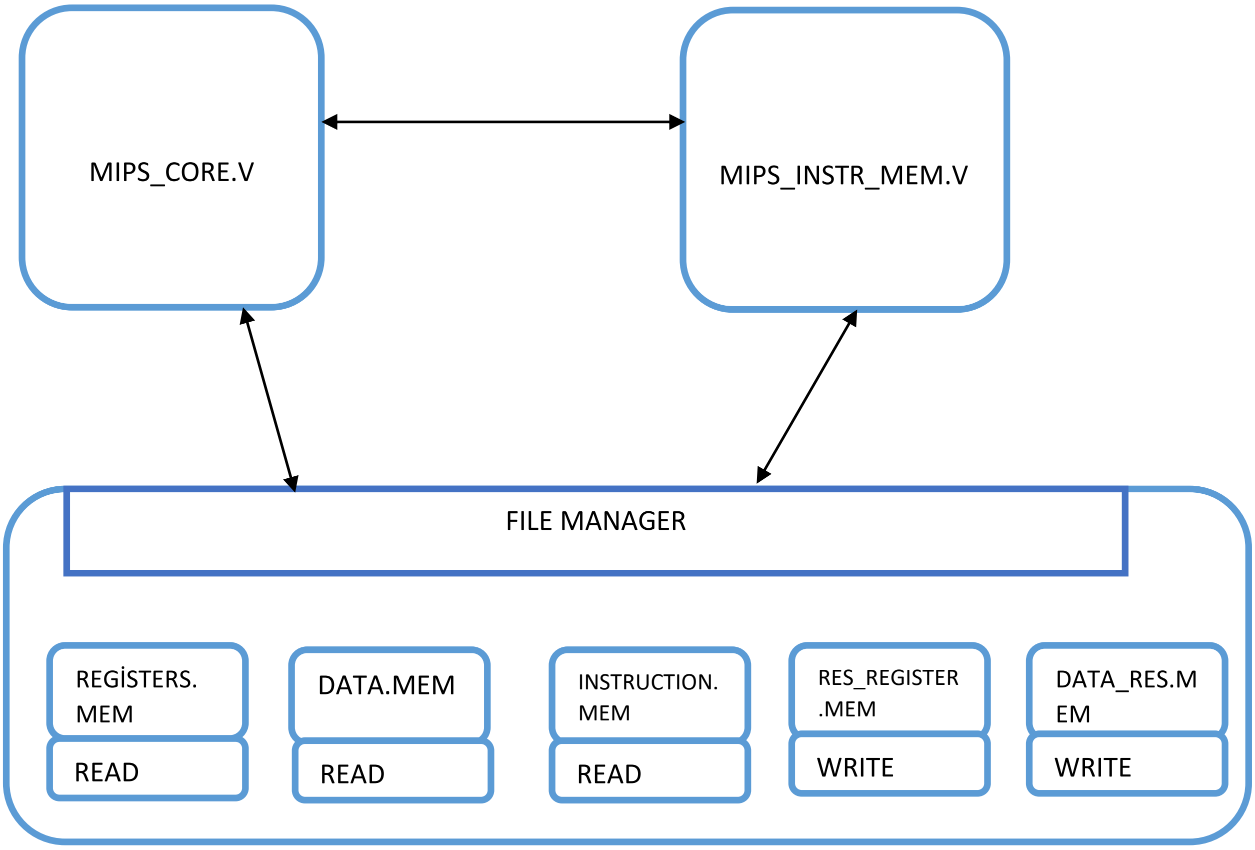
**CSE 331**

# HELLO MIPS

FINAL PROJECT

Rıdvan DEMİRCİ
141044070

# MAIN IDEA



**MIPS_CORE.V : Burada PC ın gösterdiği instruction alınır ve gene burada fetch edilir .registers ve data memory si gene buradadır işlem yapıdımı tekrar registera veya data memory e tekrar yazılır.**

**MIPS_INSTR_MEM.V : Bu modüle ise sadece gelen program counter'a ait olan instructionı geri döndürür okuma işlemini ise INSTRUCTION.MEM dosyasından yapar.**

**File Manager : Buradaki dosyalarla veriler initial begin içersinde alınır ve yazılacak dosyaya program sonunda yazılır.**

# Life cycle of 1 instruction & METHOD

**Instruction fetch için MIPS_INSTR_MEM.V modulune gidilir burada initial durumunda tüm instructionlar instruction.mem den alınmış oluyo ve indis olarakda PC gösterdiği değer alınır,tekrar mips_core da ise instructionlar tüm parçalarına aytılır(rs,rt,rd,immadiate,jump,opcode,function field) daha sonra opcode 'a bakılır ona göre instruction belirlenir ve işlemleri yapılır.**

# Testbench Results

## İnstruction.mem 'deki yaptığım testin kodu kısaca şöyledir.

**İf($30 == 0) then**

**0'dan $2 deki sayıları toplar $3'e yazar.**

**Else if($30 == 1) then**

**0'dan MEM[2] deki sayıya kadar toplar MEM[3]'e yazar**

**Else veya ($30 == 3)**

**Procedure gide $2 = 100 atar ve 0-100 arası sayıları toplar $3 e atar**

**Ama ilk başta register daki veriler ilk 10'a kadar sıralı olmalı,verildiği gibi.**

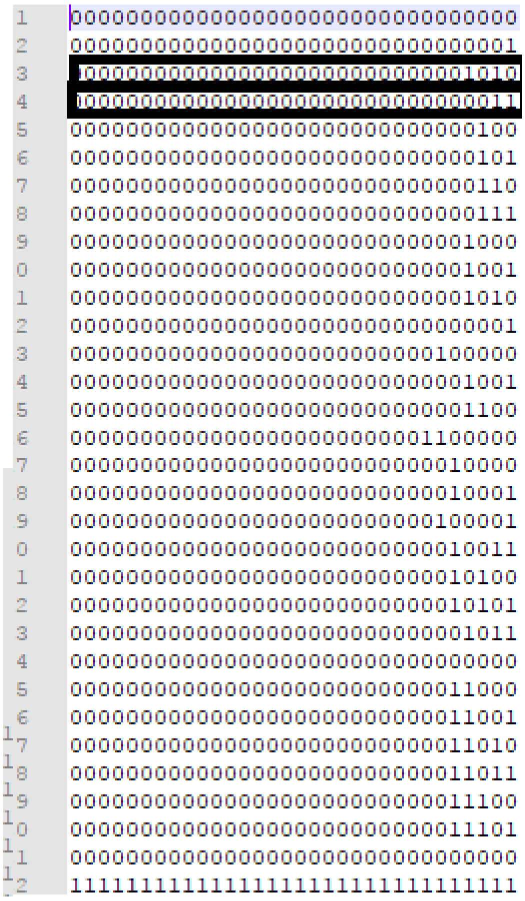
## Assmble kodu:

```
00010011110000010000000000001001      // beq $30,$1 , +9 mem sayilari topla
00010111110000110000000000000001                  // bne $30,$3 , +1 $30 0 ise
00001100000000000000000000010011                  // jal 19.adressp
//// $ 30 == 0 ise $2 deki sayilari toplar /////////
00100000000000110000000000000000                  //$3 = $0 +0        //baslangic $2 ye kadar olan sayilari toplar $3 e yazar
00100000000001000000000000000000                  //$4 = $0 +0
00000000100000110001100000100000                  //$3 = $3 +$4
00100000100001000000000000000001                  //$4 = $4 +1
00010100100000101111111111111101                  // bne $4,$2,-3
00000000100000110001100000100000                  //$3 = $3 +$4      // end toplama
00001010000000000000000000000000                  //exit jump uzak
/////////////2. ornek  $30 == 1 ise  mem 2 ye kadar toplar  mem[3] e yazar///////////////////////////////////
10001100000000100000000000000010                  // $2 = M[2]
00100000000000110000000000000000                  //$3 = $0 +0   /2 $2 ye kadar olan sayilari toplar $3 e yazar
00100000000001000000000000000000                  //$4 = $0 +0
00000000100000110001100000100000                  //$3 = $3 +$4
00100000100001000000000000000001                  //$4 = $4 +1
00010100100000101111111111111101                  // bne $4,$2,-3
00000000100000110001100000100000                  //$3 = $3 +$4      // end toplama
10101100000000110000000000000011                  // mem[3] = $3
00001010000000000000000000000000                  //exit jump uzak
/////// jal kullanilmistir $30 = 3 ise gelir ve $2 yi 100 yapar ve geri gider///
00100000000000100000000001100100                  // $2 = 100
00000011111000000000000000001000                  // jr $31 kaldigi yere gider
```

# SCREENSHOT

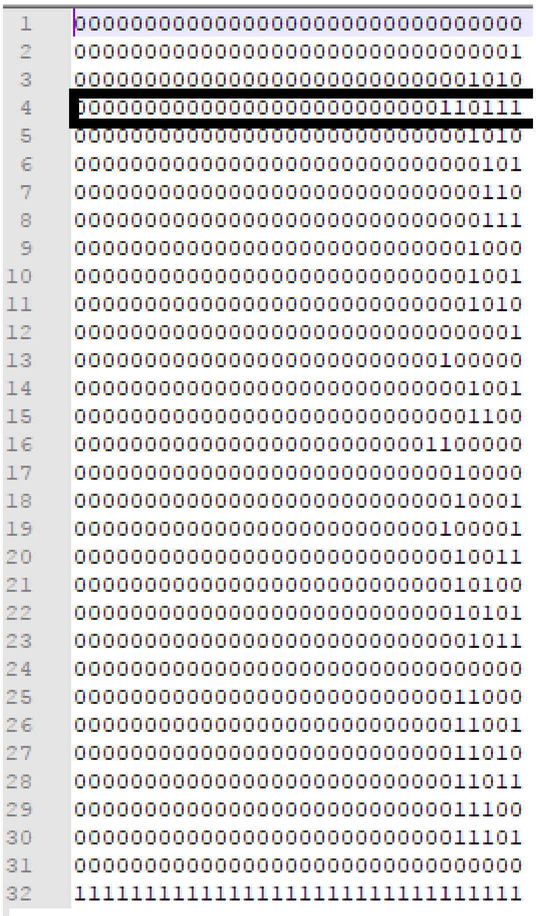
**REGİSTER DEGERLERİ 0-10 ARASI SIRALI İSE YANİ $0 = 0, $1 = 1…$10=10 İSE, $2 = İSTEDİGİNİZ DEGERİ KOYABİLİRSİNİZ , BEN DENEME İÇİN 10 KOYDUM ,VE $30 = 0 İSE 0-10 ARASI SAYILARI TOPLAR VE REGİSTERA TEKRAR YAZAR.**

$30 = 0,$2 = 10 ,,**REGİSTERS.MEM**



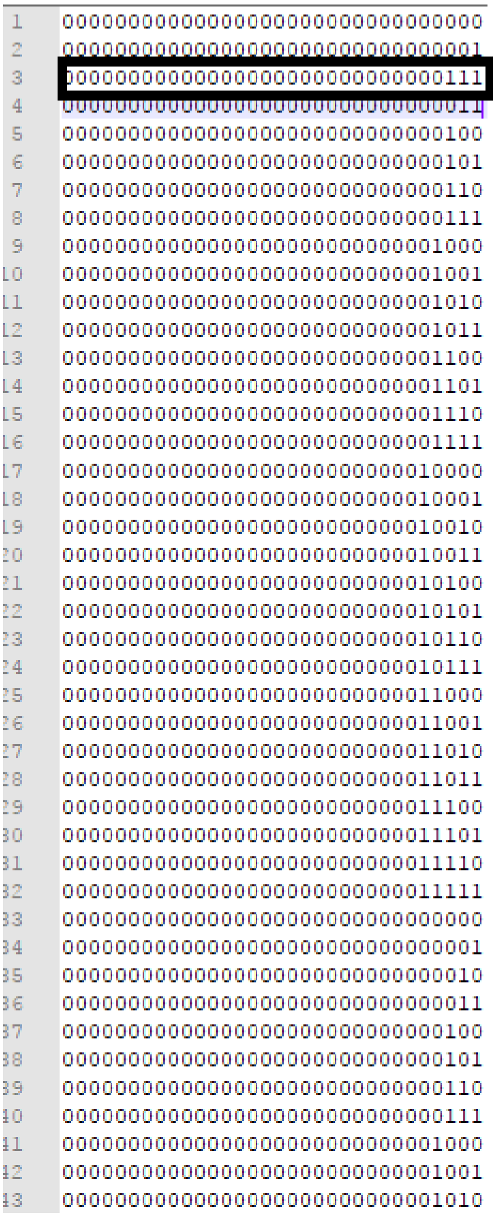
```
00000000000000000000000000000000
00000000000000000000000000000001
00000000000000000000000000001010
00000000000000000000000000000011
00000000000000000000000000000100
00000000000000000000000000000101
00000000000000000000000000000110
00000000000000000000000000000111
00000000000000000000000000001000
00000000000000000000000000001001
00000000000000000000000000001010
00000000000000000000000000000001
00000000000000000000000000100000
00000000000000000000000000001001
00000000000000000000000000001100
00000000000000000000000001100000
00000000000000000000000000010000
00000000000000000000000000010001
00000000000000000000000000100001
00000000000000000000000000010011
00000000000000000000000000010100
00000000000000000000000000010101
00000000000000000000000000001011
00000000000000000000000000000000
00000000000000000000000000011000
00000000000000000000000000011001
00000000000000000000000000011010
00000000000000000000000000011011
00000000000000000000000000011100
00000000000000000000000000011101
00000000000000000000000000000000
11111111111111111111111111111111
```

$3 = 55 ,,**res_register.mem**



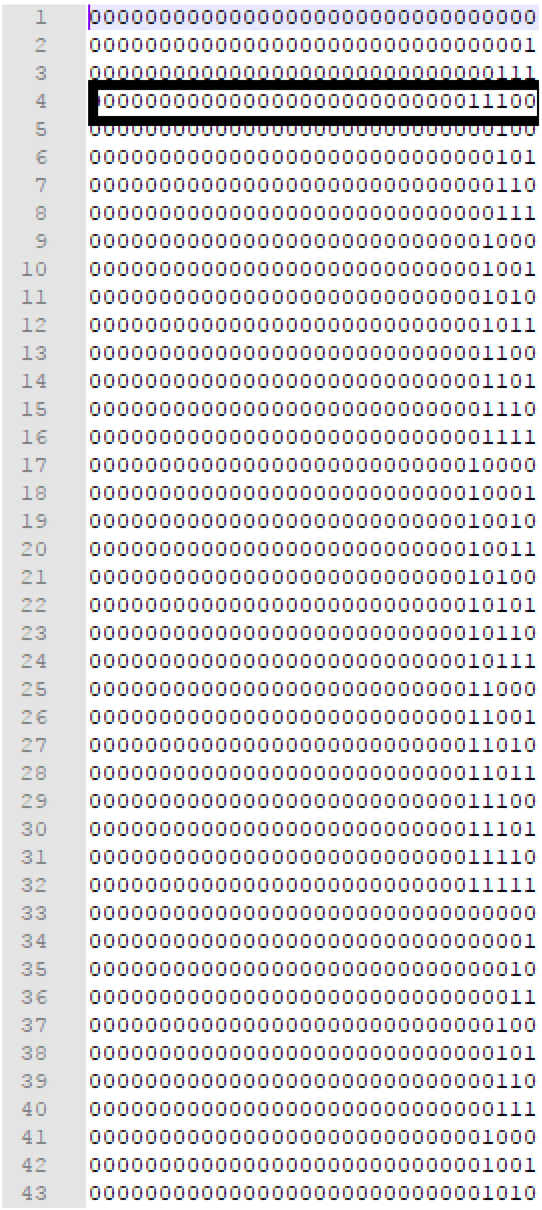
```
00000000000000000000000000000000
00000000000000000000000000000001
00000000000000000000000000001010
00000000000000000000000000110111
00000000000000000000000000001010
00000000000000000000000000000101
00000000000000000000000000000110
00000000000000000000000000000111
00000000000000000000000000001000
00000000000000000000000000001001
00000000000000000000000000001010
00000000000000000000000000000001
00000000000000000000000000100000
00000000000000000000000000001001
00000000000000000000000000001100
00000000000000000000000001100000
00000000000000000000000000010000
00000000000000000000000000010001
00000000000000000000000000100001
00000000000000000000000000010011
00000000000000000000000000010100
00000000000000000000000000010101
00000000000000000000000000001011
00000000000000000000000000000000
00000000000000000000000000011000
00000000000000000000000000011001
00000000000000000000000000011010
00000000000000000000000000011011
00000000000000000000000000011100
00000000000000000000000000011101
00000000000000000000000000000000
11111111111111111111111111111111
```

$30 = 1,MEM[2] = 7,,**data.MEM**



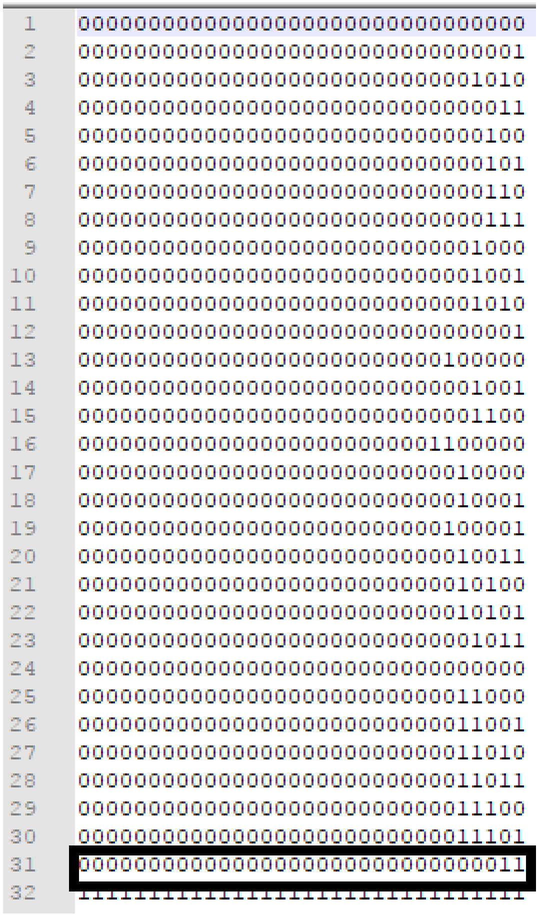
```
00000000000000000000000000000000
00000000000000000000000000000001
00000000000000000000000000000111
00000000000000000000000000000011
00000000000000000000000000000100
00000000000000000000000000000101
00000000000000000000000000000110
00000000000000000000000000000111
00000000000000000000000000001000
00000000000000000000000000001001
00000000000000000000000000001010
00000000000000000000000000001011
00000000000000000000000000001100
00000000000000000000000000001101
00000000000000000000000000001110
00000000000000000000000000001111
00000000000000000000000000010000
00000000000000000000000000010001
00000000000000000000000000010010
00000000000000000000000000010011
00000000000000000000000000010100
00000000000000000000000000010101
00000000000000000000000000010110
00000000000000000000000000010111
00000000000000000000000000011000
00000000000000000000000000011001
00000000000000000000000000011010
00000000000000000000000000011011
00000000000000000000000000011100
00000000000000000000000000011101
00000000000000000000000000011110
00000000000000000000000000011111
00000000000000000000000000000000
00000000000000000000000000000001
00000000000000000000000000000010
00000000000000000000000000000011
00000000000000000000000000000100
00000000000000000000000000000101
00000000000000000000000000000110
00000000000000000000000000000111
00000000000000000000000000001000
00000000000000000000000000001001
00000000000000000000000000001010
```

$30 = 1,MEM[3] = 28,,**res_data.MEM**



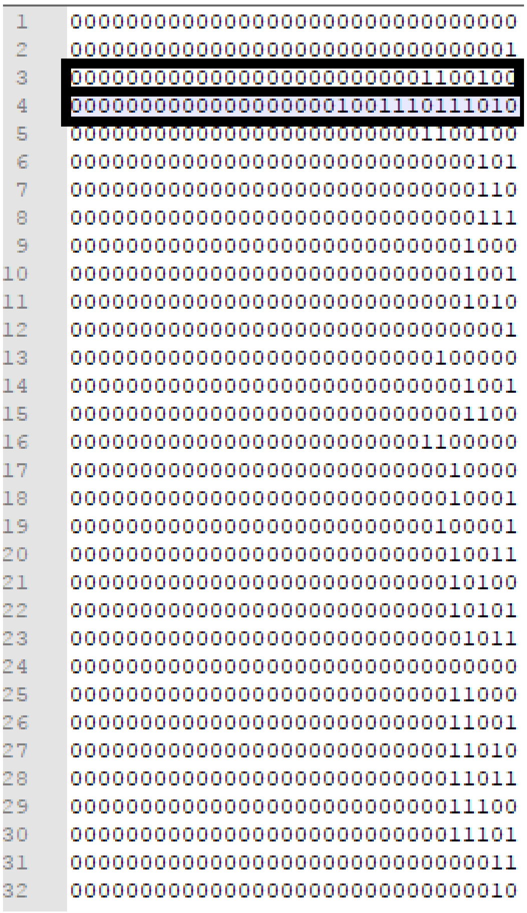
```
00000000000000000000000000000000
00000000000000000000000000000001
00000000000000000000000000000111
00000000000000000000000000011100
00000000000000000000000000000100
00000000000000000000000000000101
00000000000000000000000000000110
00000000000000000000000000000111
00000000000000000000000000001000
00000000000000000000000000001001
00000000000000000000000000001010
00000000000000000000000000001011
00000000000000000000000000001100
00000000000000000000000000001101
00000000000000000000000000001110
00000000000000000000000000001111
00000000000000000000000000010000
00000000000000000000000000010001
00000000000000000000000000010010
00000000000000000000000000010011
00000000000000000000000000010100
00000000000000000000000000010101
00000000000000000000000000010110
00000000000000000000000000010111
00000000000000000000000000011000
00000000000000000000000000011001
00000000000000000000000000011010
00000000000000000000000000011011
00000000000000000000000000011100
00000000000000000000000000011101
00000000000000000000000000011110
00000000000000000000000000011111
00000000000000000000000000000000
00000000000000000000000000000001
00000000000000000000000000000010
00000000000000000000000000000011
00000000000000000000000000000100
00000000000000000000000000000101
00000000000000000000000000000110
00000000000000000000000000000111
00000000000000000000000000001000
00000000000000000000000000001001
00000000000000000000000000001010
```

$30 = 3,jal işlemi $2’ye 100, **registers.mem**



```
00000000000000000000000000000000
00000000000000000000000000000001
00000000000000000000000000001010
00000000000000000000000000000011
00000000000000000000000000000100
00000000000000000000000000000101
00000000000000000000000000000110
00000000000000000000000000000111
00000000000000000000000000001000
00000000000000000000000000001001
00000000000000000000000000001010
00000000000000000000000000000001
00000000000000000000000000100000
00000000000000000000000000001001
00000000000000000000000000001100
00000000000000000000000001100000
00000000000000000000000000010000
00000000000000000000000000010001
00000000000000000000000000100001
00000000000000000000000000010011
00000000000000000000000000010100
00000000000000000000000000010101
00000000000000000000000000001011
00000000000000000000000000000000
00000000000000000000000000011000
00000000000000000000000000011001
00000000000000000000000000011010
00000000000000000000000000011011
00000000000000000000000000011100
00000000000000000000000000011101
00000000000000000000000000000011
11111111111111111111111111111111
```

$30 = 3,jal işlemi $2=100,$3 = 5050, **res_ registers.mem**



```
00000000000000000000000000000000
00000000000000000000000000000001
00000000000000000000000001100100
00000000000000000001001110111010
00000000000000000000000001100100
00000000000000000000000000000101
00000000000000000000000000000110
00000000000000000000000000000111
00000000000000000000000000001000
00000000000000000000000000001001
00000000000000000000000000001010
00000000000000000000000000000001
00000000000000000000000000100000
00000000000000000000000000001001
00000000000000000000000000001100
00000000000000000000000001100000
00000000000000000000000000010000
00000000000000000000000000010001
00000000000000000000000000100001
00000000000000000000000000010011
00000000000000000000000000010100
00000000000000000000000000010101
00000000000000000000000000001011
00000000000000000000000000000000
00000000000000000000000000011000
00000000000000000000000000011001
00000000000000000000000000011010
00000000000000000000000000011011
00000000000000000000000000011100
00000000000000000000000000011101
00000000000000000000000000000011
00000000000000000000000000000010
```

## Notes :

**Projede dosya yolları verilirken tam yol verilmeli..**

**$readmemb("C:/Users/Asus/Desktop/organizasyon/odevler/odev3/project/templateForProject03_restored/registers.mem", registers);**

**Model_Simde test banch sadece run edilirse bir şey yapılmadan çalışacaktır , ve ayrıca düzgün çalışaması için modelsim in işini bitirmesi beklenmelidir.**