1986

### EASTERN MEDITERRANEAN UNIVERSITY

**COMPUTER ENGINEERING DEPARTMENT**

**Güz 2019-20**

# CMPE 325 – Bilgisayar Mimarisi ve Düzenlenmesi DENEY 1

**PCSpim'e Giriş (MIPS R2000 Simülatörü)**

**1.**Giriş PCSpim, MIPS R2000 / R3000 RISC bilgisayarları için programlar çalıştıran bir simülatördür. SPIM, montaj dili veya MIPS çalıştırılabilir dosyaları içeren dosyaları okuyabilir ve hemen çalıştırabilir. SPIM, MIPS programlarını çalıştırmak için kullanılan bir sistemdir. Deneylerde, MIPS işlemcili bir iş istasyonu yerine bir simülatör kullanacağız, çünkü bir simülatör bize hata ayıklamanın yanı sıra komut setini anlamada çeşitli özellikler sunar. Ayrıca, bir MIPS simülatörü hemen hemen her bilgisayar ve işletim sistemi için kullanılabilir. Ayrıca, simülatör işlemcinin sonraki sürümlerinde geliştirilen yeni özellikleri, talimatları veya sözde talimatları neredeyse hiçbir ek ücret ödemeden içerecek şekilde güncellenebilir. MIPS simülatörüne SPIM denir ve www.mkp.com/cod2e.htm adresinden ücretsiz olarak temin edilebilir. SPIM, SUN, VAX, IBM-PC uyumlu ve Windows gibi yaygın olarak kullanılan iş istasyonlarının çoğu için yazılmıştır. PC-SPIM, MIPS-2000 için birleştirilmiş birleştirme-yükleyici-işlemci belleği ve G / Ç simülasyonudur. Çeşitli komut satırı seçenekleri ve satır içi assembler komutları ile sofistike bir programdır. Windows 32 bit işletim sistemlerinde kurulumu kolaydır.

### MIPS ASSEMBLER SYNTAX

Assembler dosyalarındaki yorumlar keskin bir "#" işaretiyle başlar. Keskin işaretten çizginin sonuna kadar olan her şey göz ardı edilir.

Tanımlayıcılar, bir sayı ile başlamayan alfanümerik karakterlerin, "\_" ve altı "." Noktalarının bir dizisidir. Öğretim kodları, tanımlayıcı olarak kullanılamayan sözcüklerdir. Etiketler, satırın başına konularak, ardından iki nokta üst üste konularak bildirilir, örneğin:

**.data item: .word 1**

**.text**

**.globl main # must be global.**

**main: lw $t01,item # loads temp.reg. $t01 with item.**

**.....**

Sayılar varsayılan olarak taban 10'dur. Onlardan önce 0x gelirse, onaltılık olarak yorumlanır. Bu nedenle, 256 ve 0x100 aynı değeri ifade eder.

Dizeler "..." çift tırnak işareti içine alınır. Dizelerdeki özel karakterler C kuralını izler: yani yeni satır \ n, sekme \ t ve \ "

Bazı önemli SPIM (ve ayrıca MIPS) assembler direktifleri:

**.byte b1,...,bn #** n belirtilen değerleri hafızaya kaydeder.

**.data <address> #** veri segmenti adresini ayarlayın.  
SPIM, veri bölümünün başlangıcı olarak 0x10000000 kullanır. Veri etiketlerini adresleriyle doğru şekilde eşleşecek şekilde 0x10000000 olarak ayarlayın.

**.globl sym #** etiketi global olarak erişilebilir kılar.

**.space n #** Geçerli segmentte n bayt alan ayırır.

**.text <address> #** sonraki öğeler kullanıcı metin bölümüne yerleştirilir. Metin bölümündeki öğeler yalnızca kelimeler veya talimatlar olabilir.

**.word n #** Kelimelerin listelenen değerlerini hafızaya kaydeder.

PCSpim simülatör programı, aşağıda gösterildiği gibi açılır menü görünümüne sahiptir



**Open**

**Save Log File Exit**

**Clear Registers Reinitialize Reload…**

**Go Continue Single Step**

**Multiple Step Breakpoints Set Value**

**Display symbol table Settings**

**Next Previous Cascade Tile**

**Arrange Icons Messages Text Segment Data Segment Registers Console Clear Console Toolbar Status Bar**

**Help Topics About PCSpim**

**Help**

**Window**

**Simulator**

**File**

:

### Simülatör Nasıl Ayarlanır:

Bu deneyin ilk bölümünde, simüle etmek için SPIM kullanacaksınız.

* Çıplak bir MIPS makinesi,

 sözde kodlara izin vermeden ve

* Map eşlenmiş I / O seçeneği yok.
* herhangi bir tuzak özelliği yüklemeden.

Bu modda, derleyici programınızda sözde kodların (yani, li, mul, blt, ... vb. Ve ayrıca uzun ofset alanlarının) kullanılmasına izin vermeyecektir.

Kayıt içeriğini okuma rahatlığı için, ekran pencerelerinde onaltılık okumaları tercih edebilirsiniz.

İkinci bölümde, yüklü olan sözde kod ve tuzak özellikleri ile SPIM'i daha ayrıntılı modda kullanacaksınız.

Bu ayarların tümü, aşağı açılır menüden simülatör> ayarlar yoluyla erişilen ayarlar formunda ayarlanabilir.

**Bir MIPS montaj programının organizasyonu**

MIPS montaj programları, ".s" veya ".asm" uzantılı metin dosyalarıdır. SPIM, derleme kaynağını yazmak için yerleşik bir editör programına sahip değildir. En sevdiğiniz veya NOTEPAD.EXE gibi kullanılabilir bir metin düzenleyicisini kullanmanız gerekir. Windows 95/98 veya NT Not Defteri programı içinde bulunur:

StartProgramsAccessoriesNotepad

MIPS montaj programınızı yazın (**programın sonunda boş bir satır bırakmanız gerekir**) ve programınız için bir dosya adı belirterek kaydedin. Dosya adının uzantısının “.s” veya “.asm” olması gerektiğini unutmayın.

Önce tıklamanız gerekir **PCSpim for Windows**

PCSpim ana penceresini başlatmak için Ardından PCSpim’in Dosya Dosyası’nı kullanarak programınızı yükleyin > Aç. Açmak istediğiniz yolu ve montaj kaynağı dosyasını seçmek için açılan tarayıcıyı kullanın.

Dosyanızda herhangi bir sözdizimi veya yapı hatası varsa, SPIM size satır numarasını ve reddedilen satırın nedenini belirten bir mesaj verecektir. Programınızı sözdizimi hatalarından temizlemeniz ve SPIM'e yüklemeniz gerekir.

Montaj kaynağınızı yükledikten sonra çalıştırmaya veya izlemeye hazırsınız. Ana menünün Simülatörü> Git (F5 tuşu) veya Simülatörü> tek adımlı (F10 tuşu) kullanın. Başlangıç ​​adresi, derleyici tarafından ayarlar penceresinde kurduğunuz seçeneklere göre otomatik olarak tanımlanır.

Pencere içeriğini, ana menünün pencereleri> Kayıtlar'ı kullanarak izleyebilirsiniz. Kolay gözlemlenebilen bir sayfa organizasyonu için pencereyi deneyin> döşeme seçeneği.

**2. Deneysel Çalışma**

**Bölüm 1**

Aşağıdaki program, R8'deki R8 kayıtlarındaki iki işaretsiz tam sayıyı çarparak R10'u kaydetmek için 32-bit ürünü yazar. Simülatör programınızın çalışmasını anlamak için, "exp1a.asm" adlı bir dosyaya yazın ve söz konusu talimat olmayan modda aşağıdaki MIPS montaj programını yürütün.

**.data 0x10000000**

**.text 0x00400000**

**.globl main main:**

**addi $8,$0,6 addi $9,$0,12**

**# multiplication of $8 \* $9 -> $10 add $2,$0,$8**

**add $10,$0,$0 mulloop:**

**beq $2,$0,mulexit # if zero exit addi $2,$2,-1**

**add $10,$10,$9 j mulloop**

**mulexit:**

**# multiplication loop is over,**

**# is the result in $10 correct? sll $0,$0,0**

**syscall**

* Keskin bir işaretten (#) sonra satırın sonuna yorum ekleyebilirsiniz.

 Aşağıdaki maddeleri uygulayarak tek adımlı çalıştırmayı başlatabilirsiniz.

- Önce PC'yi (prog.counter) programın başlangıç ​​adresine ayarlayın.

SPIM doğru ayarlanmışsa, başlangıç ​​adresi 0x00400000 şeklindedir.

Değeri ayarlamak için alt-s tuş takımını kullanın, v (veya menü simülatörü> değeri ayarlayın)

register-value atama iletişim kutusunu açın. PC'yi ve başlangıç ​​adresini onaltılık biçimde girin.

- Sonra, her basışta bir komut çalıştırmak için fn10 tuşunu kullanın. Progamın tamamını bir kerede yürütmek için fn5 tuşunu da kullanabilirsiniz. Tamam düğmesine basmadan önce başlangıç ​​adresini 0x00400000 olarak düzeltin.

* syscall işlemi durdurulduktan sonra yürütme, günlük dosyasını "exp1a.log" dosya adıyla kaydedin. Günlük dosyasını metin defterine sürükleyerek açın ve metin segmentini inceleyin. SPIM tarafından atanan onaltılık makine kodlarına göre aşağıdaki makine kodu tablosunu doldurun.

## Part-2

Bu bölümde, SPIM'i sözde kod izin verme modunda kullanacaksınız.

* Machine Çıplak makine ayarını silin ve sadece ayarlar iletişim kutusundaki (tuş dizilimi alt-s, s) sözde koda izin ver seçeneğini işaretleyin.

 Aşağıdaki metni "exp1b.asm" adlı bir dosyaya yazın.

**.data**

**.text**

**.globl main main:**

**li $8,0x3210**

**li $9,0x76543210**

**sge $11,$8,$9 mul $12,$11,$10**

**infloop:**

**bge $11,$0,infloop syscall**

* Dosyayı SPIM'e yükleyin ve her satırın ilgili makine kodlarını izleyin. Her komutun alanlarını daha iyi anlamak için aşağıdaki ikili makine kodu tablosunu doldurmak için günlük dosyasını kullanın.

Verilen metnin bir program olmadığını, izlenebilir olmadığını unutmayın. Sık kullanılan bazı MIPS sözde talimatlarının bir örneğini içerir.

**Raporlama**

Laboratuar süresi dolmadan laboratuar çalışmasını tamamladığınızda aşağıdaki rapor sayfasını doldurun ve yardımcınıza gönderin. Raporunuz, notlandırmanız için önemlidir.

Name: Student Number:

Submitted to (Asst.): Date:dd/mm/yy / /

1986

### EASTERN MEDITERRANEAN UNIVERSITY

**COMPUTER ENGINEERING DEPARTMENT**

**Güz 2019-20**

# CMPE 325 - Computer Architecture II DENEY 1 - Rapor Formu

# Birinci Bölüm: Talimatların gözlenen ikili makine kodları:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Instruction** | **opc** | **rs** | **rt** | **rd** | **sa** | **fn** |
| **addi $8,$0,6** |  |  |  |  |  |  |
| **addi $9,$0,12** |  |  |  |  |  |  |
| **add $2,$0,$8** |  |  |  |  |  |  |
| **add $10,$0,$0** |  |  |  |  |  |  |
| **beq $2,$0,mulexit** |  |  |  |  |  |  |
| **addi $2,$2,-1** |  |  |  |  |  |  |
| **add $10,$10,$9** |  |  |  |  |  |  |
| **j mulloop** |  |  |  |  |  |  |
| **sll $0,$0,0** |  |  |  |  |  |  |

Bölüm 2: Talimatların gözlenen ikili makine kodları:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Instruction** | **opc** | **rs** | **rt** | **rd** | **sa** | **fn** |
| **li $8,0x3210** | | | | | | |
|  |  |  |  |  |  |  |
| **li $9,0x76543210** | | | | | | |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
| **sge $11,$8,$9** | | | | | | |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
| **mul $12,$11,$10** | | | | | | |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
| **bge $11,$0,infloop** | | | | | | |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

**Derecelendirme:** Laboratuar Performansı:

Yard. Gözlemler: