

BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

BLGM-324 BİLGİSAYAR MİMARİSİ

DENEY #1

PC-SPIM Girişi (MIPS R2000 Simulatörü)

Giriş:

PC-SPIM RISC mimarisi kullanan MIPS R2000/R3000 bilgisayarları için bir simülasyon programıdır. SPIM, MIPS assembly dili programları yürütülebilir. MIPS dosyasına dönüştürür, ve yürütülebilir. MIPS dosyalarını okuyup çalıştırabilir. SPIM, MIPS assembly programlarını yazmak, izlemek, ve hatadan arındırmak için kendi başına yeterli bir sistemdir.

Aynı zamanda bir MIPS simülasyonu olan SPIM www.mkp.com/cod2e.htm bu adresten ücretsiz olarak indirilebilir. SPIM yazılımını çoğu bilgisayar üreticisi desteklemektedir. SUN, VAX, IBM-PC, ve Windows SPIM'i kullanan bazı şirketlerdir.

PCSpim MIPS simülasyonu hata ayıklama konusunda çeşitli özellikler sağladığından deneylerimizde PCSpim kullanacağız. MIPS simülasyonu birçok bilgisayar ve işletim sisteminde de güncel olarak kullanılmaktadır. Ayrıca, MIPS simülasyonu yeni özellikleri güncelleyebilecek yapıdadır.

PC-SPIM, çevirici, işlemci, yükleyici, hafıza ve giriş-çıkış işlemlerinin birleştirilmesiyle oluşan bir simülasyon programıdır. Program Windows 32-bitlik işletim sistemine kolayca kurulabilir.

MIPS ASSEMBLER SÖZDİZİMİ

Yorumlar assembler dosyaları içinde '#' karakteri ile başlar.

Direktif ve Tanımlayıcılar sayı ile başlamayan alfanümerik karakterler dizisinde "_" (alt çizgi), ve "." (nokta) karakterleri kullanılarak oluşturulur. Örneğin:

```

.data
item: .word 1
.text
.globl main # must be global.
main: lw $t01,item # loads temp.reg. $t01 with item.
.....

```

Sayılar 10 luk tabanda varsayılır. 0x ile başlayanlar 16lık tabana çevrilir. Örneğin, 256 ve 0x100 aynı değere sahiptir. Dizgiler(strings) çift tırnak (" ") ile başlar ve sonlanır. Karakterler C dili notasyonu ile dizgiye gömülebilir. Örneğin, yeni satır: \n,tab: \t, ve quote: \" gibi.

Bazı önemli SPIM ve MIPS assembler direktifleri:

.byte b1,...,bn	#bellekten bayt depolar ve bunlara isim verir.
.data <address>	#veri bölütlerinin başlangıçadreslerini belirler.
.globl sym	#sınıfları global yapar.
.space n	#kullanılan bölütte n baytlık yer ayırır.
.text <address>	# metin bölütünün başlangıcını belirler. # metin bölütünde yalnızca kelimeler yada kodlar olabilir.
.word n	# listelenen sözcüklere bellekte yer açar.

PCSpim simülatör programının menüsü aşağıdaki gibidir.

File	Simulator	Window	Help
→ Open	→ Clear Registers	→ Next	→ Help Topics
→ Save Log File	→ Reinitialize	→ Previous	→ About PCSpim
→ Exit	→ Reload...	→ Cascade	
	→ Go	→ Tile	
	→ Continue	→ Arrange Icons	
	→ Single Step	→ Messages	
	→ Multiple Step	→ Text Segment	
	→ Breakpoints	→ Data Segment	
	→ Set Value	→ Registers	
	→ Display symbol table	→ Console	
	→ Settings	→ Clear Console	
		→ Toolbar	
		→ Status Bar	

Simülatörün Ayarlanması

Bu oturumun ilk bölümünde SPIM'i aşağıdaki seçeneklerle kullanmayı öğreneceksiniz.

- MIPS makinesi

- Pseudo-kodsuz
- I/O seçenekleri ile ilgili plan yok
- Herhangi bir yükleme seçeneği yok.

MIPS Assembly Programı Organizasyonu

MIPS Assembly programının uzantısı **“.s” yada “.asm”** olmalıdır. SPIM'in assembly kodu yazmak için editör programı yoktur ve dolayısıyla bilgisayarımızda var olan **NOTEPAD.EXE** gibi herhangi bir metin editörünü kullanabiliriz. Program yazma şu işlemlerden oluşur.

- 1- Editör programı başlatınız
- 2- yeni bir metin dosyası oluşturunuz.
- 3- MIPS assembly kodunu yazınız (programın en son satırını boş bırak)
- 4- programı kaydediniz (dosya adının sonu **“.s” yada “.asm”** olmalıdır)
- 5- PCspim programını açıp ve kaydettiğiniz dosyanızı yükleyeyiniz
 - a. eğer hata verirse, spim hatanın olduğu satırı ve sebebini gösterir.
- 6- assembly kodunu yükledikten sonra tüm komutları ya da tek bir komutu çalıştırmak için **simulator> go (f5-tuşu)** yada **simulator> single step (f10-tuşu)** tuşuna basınız.
- 7- Simulatör çalışıp durunca yazmaçların içeriğini **window> registers** ile görebilirsiniz.
 - a. sayfa düzenini kolaylaştırmak için **window> tile** kullanınız.

DENEY ÖDEVİ

a) BÖLÜM-1:

Dosya adı **"exp1a.asm"** olan bu program yazmaç **R8** ve **R9** daki iki işaretli sayıyı çarpar ve 32-bitlik R10 yazmacına yazdır.

```

.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

```

Gözlemler

Satırın sonuna (#) karakteri koyarak yorum yazabilirsiniz.

Yukarıdaki program aşağıdaki adımları izleyerek çalıştırabilirsiniz:

İlk olarak program sayacının adresini belirliyoruz

SPIM'in başlangıç adresi **0x00400000** olmalıdır.

Bu değerleri ayarlamak için: **alt-s** yada **menusimulator>set value** ve o değerın başlangıç adresini 16'lık tabanda yazıyoruz.

Daha sonra **fn10**tuşuna her bastığımızda bir satır çalıştırıyor.

fn5tuşuna bastığımız zaman programın tamamını çalıştırıyor. Ok tuşuna basmadan önce başlangıç adresini **0x00400000** olarak düzeltmelisin.

Program **syscall**'ı gördüğü zaman duruyor ve dosyamızı "**exp1a.log**"ismi ile kaydediyoruz.

Bir metin dosyasını açabilmek için dosyayı tutup daha önceden açılmış metin dosyası penceresine sürüklüyoruz ve açılan metin dosyasının içeriğini kontrol ediyoruz.

BÖLÜM-2:

Bu bölümde SPIM'in Pseudo-kod modunu kullanacağız.

Önce **Simulator>Settings, bare machine** kutusundaki tiki kaldırıp sadece **pseudo-kod** kutusunu işaretlediğinize emin olunuz.

Aşağıdaki kodu yeni bir metin dosyasının içine atıp ve "**exp1b.asm**" ismi ile kaydediyoruz.

```
.data
.text
.globl main
main:
    li    $8,0x3210
    li    $9,0x76543210
    sge   $11,$8,$9
    mul   $12,$11,$10
inloop:
    bge   $11,$0,inloop
    syscall
```

Bu dosyayı SPIM'e yüklüyoruz ve her bir satırdaki kodu takip ediyoruz.

RAPORLAMA:Deney bitmeden önce aşağıdaki deney raporunu doldurarak en kısa sürede deney asistanlarına teslim etmeniz gerekmektedir. Raporu vaktinde teslim etmeniz deney notunuz için çok önemlidir.

İsim: _____ Öğrenci No: _____

DOĞU AKDENİZ ÜNİVERSİTESİ BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ BLGM-324 BİLGİSAYAR MİMARİSİ

DENEY #1

Bölüm-1: Komutların SPIM ile gözlemlenen ikili makina kodlarını aşağıya doldurunuz.

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: Aşağıdaki sözde-komutların SPIM ile gözlemlenen ikili makina kodlarını aşağıya doldurunuz.

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						

Deney notu: