**BİLGİSAYAR MİMARİSİ**

**ÖDEV – 1**

**Teslim Tarihi : 9 Aralık 2020**

**SORU 1.**

Aşağıda aynı komut setini kullanan iki farklı işlemcili sistem (P1 ve P2) göz önüne alınıyor. Bu komut setinde 4 farklı komut grubu (A, B, C ve D) yer almaktadır. Her işlemci için bu komut gruplarının CPI değerleri ve işlemcilerin saat frekansları aşağıdaki tabloda verilmiştir. Bir program, 106 adet komut içermektedir ve bu komutların %15’i A grubu, %30’u B grubu, %35’i C grubu ve %20’si D grubu komutudur. Buna göre,

1. Bu programın her iki işlemci için ortalama CPI değerini hesaplayınız.

P1\_CPI\_Ort=(15\*1+30\*3+35\*5+20\*15)/100 = 5.8

P2\_CPI\_Ort=(15\*2+30\*5+35\*5+20\*18)/100 = 7.15

1. Bu program hangi işlemci üzerinde, ne kadar hızlı yürütülür? Hesaplayınız.

Yürütme Zamanı (T) = CPI \* KomutSayısı \* ClockTime

ClockTime (t) = 1/frekans ;

P1\_ClockTıme(t)=

1/ (2.2 \* 109 ) = “2.2 Gz – Hz e dönüştürülüyor)

P2\_ClockTıme(t)=

1/ (3.2 \* 109 ) “3.2 Gz – Hz e dönüştürülüyor)

P1\_T = (5.8 \* 106)/ (2.2 \* 109) = 0.00263636363636364

P2\_T = (7.15 \* 106) / (3.2 \* 109) = 0.002234375 (Daha Kısa Sürede)

C ) Bu programın yürütülmesi için gerekli saat çevrim miktarını bulunuz.

ti=Komut Sayısı \* CPI ;

P1\_ti= 10 000 000 \* 5.8 ;

P2\_ti= 10 000 000 \* 7.15 ;

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **İşlemci** | **Saat Frekansı** | **Komut Seti Komut Grupları** | | | |
| **A Grubu** | **B Grubu** | **C Grubu** | **D Grubu** |
| P1 | 2.2 GHz | 1 | 3 | 5 | 15 |
| P2 | 3.2 GHz | 2 | 5 | 5 | 18 |

**SORU 2.**

Bir bilgisayarda rutin bir işlem yükünün %60’ını CPU aktiviteleri ve %40’ını disk aktiviteleri oluşturmaktadır. Bu bilgisayarın sahibi size sisteminin yavaşlığından şikayet ederek, sistemini güncellemenizi istiyor. Yaptığınız bir araştırma sonucunda, mevcut disk işlemlerini 2.5 kat hızlandıracak ve maliyeti 8000 T.L olacak bir çözüm buluyorsunuz. Bu araştırma esnasında CPU işlemlerini 1.6 kat hızlandıracak ve maliyeti 5000 T.L. olacak başka bir çözümü de elde ediyorsunuz.

* En az maliyete en iyi performansı elde edebileceğiniz çözümü belirleyiniz. Bu çözüm ile tüm sistemin ne kadar hızlanacağını bulunuz.

Paranın tamamı maliyet kazanç çarpanına göre 5000 TL harcama ile

S1 = 1/(1 – Fractioncpu+ (Fractioncpu/Scpu))

S1= 1/((1-0.6)+(0.6/1,6)) = 1.290

* Maliyet önemli olmadığı durumda en hızlı sistemi elde edebilmek için hangi çözümü seçerdiniz? Bu çözüm ile tüm sistemin hızlanması ne olurdu? Bulunuz.

Maliyet önemli olmadığı durumda disk 2.5 kat , Cpuda 1.6 kat hızlandıırlır

S1 = 1/(1 – Fractiondisk+ (Fractiondisk/Sdisk))

S1= 1/((1-0.4)+(0.4/2,5)) = 1.315

S1 = 1/(1 – Fractioncpu+ (Fractioncpu/Scpu))

S1= 1/((1-0.6)+(0.6/1,6)) = 1.290

S=S1\*S2 = 1.315 \* 1.290 = 1.69635 Toplam Hızlanma

**SORU 3.**

Bellekte NxN boyutlu ve elemanları WORD uzunluğa sahip bir matris tanımlayınız (N değerini siz seçebilirsiniz). Matrisin her satırındaki en büyük elemanları bulan ve değerleri toplayıp sonucu belleğe kayıt eden ARM Assembly programını yazınız.

N= 3 Olarak yaptım.

​AREA main\_kod,​CODE,​READONLY

​​ENTRY

​​​​EXPORT \_\_main

​​\_\_main

​​;R0 SÜTUN DONGÜSÜ IÇIN

​​;R1 SATIR DONGÜSÜ IÇIN

​​​​;R4 MATRIS DIZISI BASLANGICI

​​;R5 MATRIS ELEMANI

​​​​;R9 EN KÜÇÜK DIZISI BASLANGICI

​​;R10 EN KÜÇÜK ELEMANI

​​​​LDR R4, =MATRIS ;DIZI baslangiç adresi R0 da

​​LDR R12, =ENKUCUK

​​​​MOV R0, #3 ;dizi sütun eleman sayisi -1 R1 de

​​​​MOV R5, #0 ;dizi sütun eleman sayisi -1 R1 de

​​;MOV R11, #0 ;dizi DÖNGÜ KONTROL ELEMANI

​​DONGUSUTUN

​​MOV R1, #3 ;dizi satir eleman sayisi. Her döngüdetekrar ilk deger atamasi var

​​;LDR R10, [R9], #4 ;eN KÜÇÜK sonuç dizisi

​​;MOV R9, #999;;eN KÜÇÜK sonuç dizisi

​​LDR R9, [R4]

DONGUSATIR

​​LDR R5, [R4], #4

​​CMP R9,R5

​​​​MOVGT R9,R5

​​SUBS R1, R1, #1 ; indis azaltilir.

​​BNE DONGUSATIR ;R1 0'dan farkliysa döngüdöner

​​LDR R11, [R12], #4

​​MOV R11,R9

​​SUBS R0, R0, #1 ; indis azaltilir.

​​BNE DONGUSUTUN ;R1 0'dan farkliysa döngüdöner

STOP

​​B STOP

​​MATRIS DCD 1, 3, 2, 6, 5, 4, 7, 8, 9

;MATRIS DCD 1, 2, 3, 4, 5, 6, 7, 8, 9

​​AREA bellek, DATA, READWRITE

ENKUCUK DCD 0,0,0

​​​​

​​END

**SORU 4.**

Bellekte tanımlı, eleman değerleri 0 ve 1’lerden oluşan bir BYTE dizisi olsun. Bu bir ikili (binary) sayıyı temsil etsin. Bu ikili sayının (dizinin) ikiye tümleyenini hesaplayan ARM assembly yordamını yazınız. Bu yordama iki adet byte dizisinin başlangıç adresi ve ilk dizinin (ikili sayı) eleman sayısı stack üzerinden parametre olarak geçirilecektir. Yordam ilk dizinin 2’ye tümleyenini hesaplayacak ve bu yeni ikili sayıyı ikinci BYTE dizisine yazacak ve bu dizi ile yordamdan geri döndürecektir.

Yordam yazıldıktan sonra bir ana program yazılarak test edilecektir.

/////Main.s/////

​​AREA main\_kod,​CODE,​READONLY

​​ENTRY

​​​​EXPORT \_\_main

​​IMPORT REVERSE

​​\_\_main

​​​​LDR R0, =byte1 ;DIZI baslangiç adresi R0 da

​​;LDR R5,[R0];ilk

​​MOV R1, #4

​​​​LDR R4, =byte2 ;DIZI baslangiç adresi R0 da

​​​​STMDB SP!,{R0,R1};

​​BL REVERSE

LDR R3, [SP, #4] ;geri dönüs degeri

STOP

​​B STOP

​​byte1 DCD 0,1,0,1

​​​AREA bellek, DATA, READWRITE

byte2 DCD 1,1,1,1

​​​​END

///////Toplam\_alt.s////////////////

​​AREA reverse\_kod,​CODE,​READONLY

​​ENTRY

​​​​EXPORT REVERSE

REVERSE

​​STMFD SP!,{R0,R1};

LOOP

​​LDR R2, [R0], #4

​​MVN R10,R2

​​​​LDR R11, [R4], #4

​​MOV R11,R10

​​​​SUBS R1, R1, #1 ; indis azaltilir.

​​BNE LOOP ;R1 0dan farkliysa döngü döner

​​END

**SORU 5.**

Aşağıda verilen ARM assembly programının her bir komutunu makine koduna çeviriniz. Makine kodunu on altılık tabanda yazınız.

LOOP

LDR r3, [r0], #4

**// 043090E4**

**CMP r3, #16**

**// 100053E3**

**STREQ r1, [r6, #8]**

**// 08108605**

**ANDLT r1, r1, r3, LSL #2**

**// 031101B0**

**ADDGT r1, r1, r3, LSR r4**

**// 331481C0**

**MUL r4, r5, r6**

**// 950604E0**

**SUBS r2, r2, #1**

**// 012052E2**

BNE LOOP