Name, Lastname, Stu # :

KTU Faculty of Engineering, Dept. of Computer Engineering
COM/BİL 2000 Resit Exam

19.06.2019

1. Which of the following techniques should be used to design a highly complex microprocessor
Oldukça karmaşık bir mikroişlemci tasarlamak için aşağıdaki tasarım tekniklerinden hangisi kullanılmalıdır

a. Level triggered sequential circuit design techniques
Seviye tetiklemeli ardışıl devre tasarım teknikleri
b. Edge triggered sequential circuit design techniques
Kenar tetiklemeli ardışıl devre tasarım teknikleri
c. Sequence register and decoder
Sıralama kaydedici ve kod çözücü
d. One Flip-Flop per state
Durum başına bir flip-flop
e. none of the above
Yukarıdakilerden hiçbiri

2. Which of the following is not one of the required steps of instruction execution
AAşağıdakilerden hangisi emir çalıştırma için gerekli adımlardan biri değildir

a. Fetch instruction from the memory and store in IR
Emri bellekten okuyup IR'ye saklamak
b. Locate and fetch operands from memory
Operand'ları bellekten okumak
c. Exchange registers for better utilization of the resources
Kaynakları daha iyi kullanmak için kaydedicileri değiştirme
d. Execute the instruction and store the results
Emri çalıştırıp sonuçları saklama
e. Proceed (increment the PC) to the next instruction
Bir sonraki emre ilerleme (PC'yi artırma)

3. Which of the following addressing mode is required for relocatable code
Aşağıdaki adresleme modlarından hangisi yeniden yerleştirilebilir kod için gereklidir

a. Immediate addressing
İzleyen adresteki adresleme
b. Direct addressing
Doğrudan adresleme
c. Indirect addressing
Dolaylı adresleme
d. Relative addressing
Göreceli adresleme
e. Indexed addressing
İndisli adresleme

4. User and system modes of CPU operation is required for which of the following
MIB'in kullanıcı ve sistem modları aşağıdakilerden hangisi için gereklidir

a. All computers running a multitasking operating system
Çok görevlilik ikeli işletim sistemi çalıştıran tüm bilgisayarlar
b. To allow only some programs to modify parameters of operating system
Yalnız bazı programların işletim sistemi parametrelerini değiştirebilmesi için
c. For shared resources which require serialized access
Serileştirilmiş erişim gerektiren paylaşılan kaynaklar için
d. Unless a single level of process priority is sufficent
Tek bir süreç önceliği yeterli değilse
e. All of the above
Yukarıdakilerin tümü

5. Which of the following is not a reason to use interface circuits between I/O devices and CPU
Aşağıdakilerden hangisi MİB ve G/Ç cihazları arasında arayüz kullanma nedeni değildir

a. conversion of incompatible signal levels
uyumsuz işaret seviyelerinin dönüştürülmesi
b. synchronization to resolve data transfer rate differences
eşzamanlayarak veri aktarım hızı farklılıklarının çözümlenmesi
c. to reduce manufacturer dependent differences
üreticiyi kaynaklı farklılıkların azaltılması
d. data code and format difference between the CPU and I/O devices
MİB ve G/Ç cihazlarının veri kodu ve biçimi farklılıkları
e. to reduce the operating mode differences causing disturbance among peripherals
çevre cihazlarının arasında uyumsuzluk yaratan çalışma modu farklılıklarının azaltılması

6. Response time to interrupt request of an I/O device
G/Ç cihazının kesme isteğine tepki süresi

a. is constant and negligible compared to program controlled transfer
değişmezdir ve program denetimli aktarıma göre ihmal edilebilir
b. is not constant but negligible compared to program controlled transfer
değişkendir fakat program denetimli aktarıma göre ihmal edilebilir
c. is not deterministic and should be used with care in real time systems
belirtenebilir değildir ve gerçek-zamanlı sistemlerde dikkatle kullanılmalıdır
d. does not depend on the instruction being executed
o anda çalıştırılan emire bağımlı değildir
e. is negligible since there is no need to store the CPU registers
MİB kaydedicilerini saklama gereği olmadığı için ihmal edilebilir

7. Daisy Chain Priority Interrupt scheme
Papatya zinciri kesme öncelik yöntemi

a. has static priorities depending on the location of peripherals
çevre cihazının bulunduğu konuma bağlı değişmez önceliktedir
b. has dinamic priorities which can be modified by user
kullanıcı tarafından değiştirilebilen önceliklere sahiptir
c. has less delay than a parallel priority encoder
paralel öncelik çözümleyiciye göre daha düşük gecikmelidir
d. can not be cascaded to provide more interrupt inputs
daha fazla kesme girişi sağlamak için kaskat kullanılamaz
e. has more delay than polling each interrupt device
her bir cihazı tek tek yoklamamaktan daha fazla gecikmelidir

8. A parallel priority encoder with k inputs
k girişli bir paralel öncelik kodlayıcı

a. is a binary encoder with output corresponding to only one out of k inputs
k girişten yalnız birine karşı düşen kodu çıkaran bir ikili kodlayıcıdır
b. generates an output corresponding to the highest priority of all k inputs
en yüksek öncelikli girişe karşı düşen çıkışı üretir
c. can not be cascaded to provide more interrupt inputs
daha çok sayıda kesme girişi sağlamak için kaskatlanamaz (ard arda bağlanamaz)
d. does not prevent interrupt sources with less priority from being serviced
düşük öncelikli kesme kaynaklarına servis verilmesine engel olmaz
e. introduces more delay than a daisy chain priority resolver
papatya zinciri öncelik çözümleyiciden daha fazla gecikmeye neden olur

9. The read and write control lines of DMA controllers are
DMA denetleyicinin okuma ve yazma denetim uçları

a. unidirectional and can only be used as input
tek yönlüdür ve yalnız giriş olarak kullanılabilir
b. unidirectional and can only be used as output
tek yönlüdür ve yalnız çıkış olarak kullanılabilir
c. bidirectional and can be used as inputs under some conditions and as outputs otherwise
iki yönlüdür ve nazı koşullarda giriş diğer durumlarda çıkış olarak kullanılabilir
d. active (not in high-Z state) while the CPU is running
MİB çalışırken aktifdir (yüksek empedans değil)
e. not active (in high-Z) during DMA transfers
DMA aktarım sırasında aktif değildir (yüksek empedans)

10. What is the speed-up of a sector (512 bytes) transfer from a magnetic disk buffer to memory by DMA compared to 15 clock cycle per byte Program Controlled Transfer running on a 2 MHz CPU with 100 ns access time memory. // 100 ns erişim süreli bellek kullanan 2 MHz'de çalışan bir MİB'in bayt başına 15 saat dönemi gerektiren Program Denetimli aktarım yerine DMA aktarımı kullanarak manyetik disk tamponundan belleğe 512 byte'lık bir sektör aktarınmının hızlanması ne kadardır.

a. 5 b. 15 c. 25 d. 50 e. 75

Name, Lastname, Stu # :

Test	1	Toplam
1	3, 4	PÖÇ

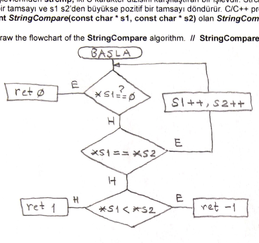
KTU Faculty of Engineering, Dept. of Computer Engineering
COM/BİL 2000 Resit Exam

19.06.2019

1. From **en.wikibooks.org/wiki/C_Programming/string.h/strcmp** : In POSIX and in the programming language C, ***strcmp*** is a function in the C standard library (declared in string.h) that compares two C strings. Function ***strcmp*** returns 0 when the strings are equal, a negative integer when s1 is less than s2, or a positive integer if s1 is greater than s2, according to the lexicographical order. Code ***StringCompare*** function with prototype **int *StringCompare* (const char * s1, const char * s2)** to be called from C/C++ programs.

en.wikibooks.org/wiki/C_Programming/string.h/strcmp 'den : POSIX ve C programlama dilinde, (string.h içinde tanımlı) standart kütüphane işlevlerinden **strcmp,** iki C karakter dizisini karşılaştıran bir işlevdir. Strcmp işlevi sözlük sıralamasına göre diziler eşitse 0, s1 s2'den küçükse negatif bir tamsayı ve s1 s2'den büyükse pozitif bir tamsayı döndürür. C/C++ programlarından çağrılabilecek ve işlev ön ürünü (function prototype) **int *StringCompare*(const char * s1, const char * s2)** olan ***StringCompare*** işlevini kodlayınız.

a. Draw the flowchart of the **StringCompare** algorithm. // **StringCompare** algoritmasının akış çizgesini çiziniz.



b. Code **StringCompare** algorithm in any high level language. // **StringCompare** algoritmasını herhangi bir yüksek seviyeli dil ile kodlayınız.

```
while (*S1++==*S2++)
  if (*S1++=='\Ø')
    return (0);
  if (*S1 < *S2)
    return (-1)
  else
    return (1)
```

c. Code **StringCompare** algorithm in Assembly language. // **StringCompare** algoritmasını Assembly dili ile kodlayınız.

```
Ara:        MOV IX, S1
            MOV IY, S2
            MOV A, [IX]
            MOV B, [IY]
            CMP A, #Ø
            BNE EşitDeğil
            MOV A, #Ø
            RET
EşitDeğil:  CMP A,B
            BNE KucukmuBuyukmu
            INC IX
            INC IY
            B Ara

KucukmuBuyukmu:  BN Küçük
                 MOV A, #1
                 RET
Küçük:           MOV A, #-1
                 RET
```