

Microprocessors Spring 2020 final test

1. Single byte signed address value of a single byte relative addressing mode `jl` (jump if less than) instruction of a 16-bit x86 processor at address `0x010A` is `F9`. Which of the following is the effective address of the instruction // `0x010A` adresindeki 16-bit x86 işlemcinin tek baytlık göreceli adresleme modu `jl` (küçükse atla) komutunun tek bayt işaretli adres değeri `F9`'dur. Aşağıdakilerden hangisi emrin etkin (efektif) adresidir

- A. `0x0101` B. `0x0105` C. `0x010C` D. `0x010E`

2. The content of the top of the memory stack is `0x5320`. The content of `SP` (Stack Pointer) is `0x3560`. A two- word `CALL` (call subroutine) instruction is at `0x1120` followed by the address field `0x6720` at location `0x1121`. Which of the following are the contents of `PC`, `SP` and the `TOS` (Top of Stack) after the `CALL` instruction is executed // Bellek yığınının en üst içeriği `0x5320`'dir. `SP` (Yığın İşaretçisi) içeriği `0x3560`'tır. İki kelimelik bir `CALL` (alt yordam çağır) komutu `0x1120` adresinde ve ardından `0x1121` adresindeki emrin adres alanında `0x6720` değeri vardır. Aşağıdakilerden hangisi `CALL` komutu yürütüldükten sonra `PC`, `SP` ve `TOS` (Top of Stack) içeriğidir

- A. `PC=0x6720`, `SP=0x355F`, `TOS=0x1122`
B. `PC=0x1122`, `SP=0x3560`, `TOS=0x5320`
C. `PC=0x1120`, `SP=0x3560`, `TOS=0x5320`
D. `PC=0x6720`, `SP=0x3559`, `TOS=0x1122`

3. The differences between a call subroutine instruction and program interrupt are // Bir altprogram çağrı emri ve kesme arasındaki farklar

- A. `call` is software and runs slower but interrupt is hardware and runs faster // altprogram çağrısı yazılımdır ve daha yavaş çalışır ancak kesme donanımdır ve daha hızlı çalışır
B. `call` without an argument pushes all registers to stack compared to interrupt which pushes just the return address // yalnızca dönüş adresini yığına iten kesmeye kıyasla değişkeni olmayan çağrı emri tüm kaydedicileri yığına iter
C. `call` without an argument pushes only the return address but interrupt needs to push all registers // değişkeni olmayan çağrı emri yalnızca dönüş adresini yığına iter, ancak kesme işleminin tüm kaydedicileri yığına itmesi gerekir
D. interrupt runs immediately after `IRQ` (Interrupt ReQuest) signal is received without waiting the completion of currently running instruction but subroutine call instruction needs to wait for the completion of currently running instruction // kesme, `IRQ` (Interrupt ReQuest) sinyali alındıktan hemen sonra çalışan komutun tamamlanmasını beklemeden hemen çalışır ancak altprogram çağrı emri o anda çalışan komutun tamamlanmasını beklemesi gerekir

4. What is the difference between synchronous and asynchronous serial transfer of information // Sayısal verinin eşzamanlı (senkron) ve eşzamansız (asenkron) seri aktarımı arasındaki fark nedir
A. Synchronous transmission has more overhead so it is less efficient // Senkron iletim daha fazla yüke sahiptir, bu nedenle daha az verimlidir.
B. Synchronous transmission is point to point whereas asynchronous is not // Senkron iletim noktadan noktaya, asenkron noktadan noktaya değildir
C. Asynchronous transmission has less overhead so it is more efficient // Asenkron iletim daha az ek yüke sahiptir, bu nedenle daha verimlidir
D. Asynchronous transmission has more overhead so it is less efficient // Asenkron iletim daha fazla ek yüke sahiptir, bu nedenle daha az verimlidir
5. What is the minimum number of bits that a frame must have in the bit-oriented protocol // Bit yönelimli protokolde bir çerçevede bulunması gereken en az bit sayısı nedir?
A. 16 B. 24 C. 32 D. 36
6. What is the basic advantage of using interrupt initiated data transfer over transfer under program control without an interrupt? // Kesme ile başlatılan veri aktarımı kullanmanın program denetimli aktarmaya göre temel avantajı nedir?
A. Interrupt initiated data transfer consumes more power since it needs to check the interrupt flip-flop after every instruction execution // Kesme ile başlatılan veri aktarımı, her komut yürütüldükten sonra kesme flip-flopunu kontrol etmesi gerektiğinden daha fazla güç tüketir.
B. Interrupt initiated data transfer responds faster compared to data transfer under program control // Kesme ile başlatılan veri aktarımı, program kontrolü altındaki veri aktarımına kıyasla daha hızlı yanıt verir.
C. Interrupt initiated data transfer responds slower compared to data transfer under program control // Kesme ile başlatılan veri aktarımı, program kontrolü altındaki veri aktarımına kıyasla daha yavaş yanıt verir.
D. Interrupt initiated data transfer frees CPU to perform other tasks while waiting for a slow I/O device to be ready to transfer data // Kesme ile başlatılan veri aktarımı, yavaş bir G/Ç aygıtının veri aktarmaya hazır olmasını beklerken CPU'yu serbest bıraktığı için diğer görevleri gerçekleştirmesini sağlar
7. Consider a computer without priority interrupt hardware. Any one of many sources can interrupt the computer and any interrupt results in storing the return address and branching to a common interrupt service routine. Priority can be established // Kesme öncelik donanımı olmayan bir bilgisayar düşünün. Birçok kaynaktan herhangi biri bilgisayar kesmeye uğratabilir ve herhangi bir kesme, dönüş adresinin depolanması ve ortak bir kesme hizmet altprogramına dallanma ile sonuçlanır. Aşağıdakilerden hangisi ile öncelikler oluşturulabilir
A. by checking all sources starting with the lowest priority // en düşük önceliğe sahip olan kaynaktan başlayarak tüm kaynakları kontrol ederek
B. by checking all sources starting with the highest priority // en yüksek önceliğe sahip olan kaynaktan başlayarak tüm kaynakları kontrol ederek.
C. by checking Enable Interrupt flip flop // Kesme Yetkilendir Flip-Flop'unu kontrol ederek
D. by checking the return address // geri dönüş adresini kontrol ederek

8. The three outputs x, y, z from an 8 x 3 Priority Encoder are used to provide an 8-bit vector address in the form 110xyz00. Which of the following is the eight address starting from the one with the highest priority // 8 x 3'lük Öncelik Kodlayıcısından gelen üç x, y, z çıkışlarını 110xyz00 biçimindeki 8 bit vektör adresi oluşturmak için kullanılmaktadır. Aşağıdakilerden hangisi en yüksek önceliğe sahiptir başlayarak sekiz vektör adresi değerleridir

- A. 0xDC, 0xD8, 0xD4, 0xD0, 0xCC, 0xC8, 0xC4, 0xC0
- B. 0xC0, 0xC4, 0xC8, 0xCC, 0xD0, 0xD4, 0xD8, 0xDC
- C. 0xD8, 0xD6, 0xD4, 0xD2, 0xD0, 0xC8, 0xC6, 0xC4
- D. 0xC0, 0xC2, 0xC4, 0xC6, 0xC8, 0xCA, 0xCC, 0xCE

9. Why are the read and write control lines in a DMA controller bidirectional // DMA denetleyicinin oku ve yaz hatları neden iki yönlüdür

- A. They are controlled by the CPU during a DMA operation. // DMA işlemi sırasında CPU tarafından denetlendiği için
- B. They are input lines during a DMA operation // DMA işlemi sırasında giriş hatlarıdır
- C. They are output lines before a DMA operation // DMA işlemi öncesinde çıkış hatlarıdır
- D. They are output lines during a DMA operation // DMA işlemi sırasında çıkış hatlarıdır

10. It is necessary to transfer 4096 byte sector from an Advanced Formatted (AF) 1066x Compact Flash disk to a memory disk buffer section starting from address 0x3000. The transfer is by means of the 16-bit Ultra DMA 7 (Ultra ATA/167). What are the Address Register and Word Count Register initial values that the CPU must transfer to the Ultra DMA controller // Gelişmiş Biçimlendirilmiş (AF) 1066x Kompakt Flash diskten 4096 bayt bir sektörü 0x3000 adresinden başlayarak bir bellek disk arabelleği bölümüne aktarmak gerekmektedir. Aktarım 16-bit Ultra DMA 7 (Ultra ATA / 167) ile yapılmaktadır. CPU'nun Ultra DMA denetleyicisine aktarması gereken Adres Kaydedici ve Kelime Sayısı Kaydedici başlangıç değerleri nelerdir?

- A. 0x3000, 0x1000 B. 0x3000, 0x0FFF
- C. 0x3000, 0x07FF D. 0x3000, 0x0800

ANSWERS

1 B, 2 A, 3 C, 4 D, 5 C, 6 D, 7 B, 8 A, 9 D, 10 C