Ad-Soyad : **İmza:**

No :

Email :

Final Sınavı (23.01.2008)

EENG5230- Bilgisayar Donanımı

S1. İşlevsel olarak bir bilgisayarı nasıl tanımlarsınız? Her bir işleve karşı gelen üniteler nelerdir? (10)

S2. Bir bilgisayarda her bir komutun yürütülmesi için 5 saat (clock) peryodu gerekmektedir. 100 komutluk bir programın yürütülmesi için 200 ns (nanosaniye) gerekiyorsa, bu bilgisayarın saat frekansı (clock frequency) nedir? (10)

S3. Belleğindeki her bir adreste 1 byte bulunan bir bilgisayara ait bir programın bir satırı yandaki gibidir. Buna göre:

Adres	Assembler	Makina kodu
200000	lda A4F051	12A4F051

(a) Bu bilgisayarda kaç tane komut olabilir? (4)

(b) Komut yazmacı (instruction register) nın uzunluğu kaç bittir? (3)

(c) Bu komutun bellekte kapladığı alan ne kadardır? (4)

(d) Bu bilgisayarın belleği kaç MByte tır? (3)

(e) Eğer bu bilgisayarda aşağıdaki formatta bir "direct-mapped cache" kulanılmışsa, A4F051 bellek adresi için Tag-id, Line-id ve Word-id değerlerini 16 tabanlı sayı formatında bulunuz.. (6)

Tag-id	Line-id	Word-id
7 bit	12 bit	5 bit

S4. Bir bilgisayar sistemi yandaki gibi bir kayan noktalı (floating point) sayı formatına sahiptir. 12.5 ve 1.25 sayıları verildiğine göre:

1 bit işaret	5 bit ötelenmiş üst	10 bit mantis

(a) Sayıların 16 bit kayan noktalı formatta karşılıklarını bulunuz. (8)

(b) (a) da bulduğunuz sonuçları kullanarak 12.5+1.25 işlemini gerçekleştiriniz. (7)

S5. **C*4** işleminin sonucunu Booth algoritması ile bulunuz (sayılar 2s complement sayı sistemine göredir)? (10)

S6.

```
00   in      // input number A
01   sta 30
02   in      // input number B
03   sta 31
04   sub 30
05   brp 08
06   lda 30
07   br  10
08   brz 10
09   lda 31
10   sta 40
11   hlt
```

Yandaki program VVM makina kodunda yazılmıştır. Buna göre aşağıdaki soruları cevaplayınız:

(a) A = 5, B = 3 olduğunda adres 40 ın içeriği ne olur? (5)

(b) A = 2, B = 6 olduğunda adres 40 ın içeriği ne olur? (5)

(c) A = 4, B = 4 olduğunda adres 40 ın içeriği ne olur? (5)

(d) Bu program ne yapar açıklayınız? (5)

S7. X ve Y, 2s complement sayı sistemine göre iki sayıdır. $X=(A)_{16}$ ve $Y=(4)_{16}$, olduğuna göre; X–Y işlemini yapınız. Sonuca göre taşma olup olmadığını yazınız. (15)

VVM instruction set (nn indicates an address location in the RAM), for example; machine code 510 corresponds to Assembly code LDA 10 (or lda 10).

Load Accumulator (5*nn*) [LDA *nn*] The content of RAM address *nn* is copied to the Accumulator Register, replacing the current content of the register. The content of RAM address *nn* remains unchanged. The Program Counter Register is incremented by one.
Store Accumulator (3*nn*) [STO *nn*] or **[STA *nn*]** The content of the Accumulator Register is copied to RAM address *nn*, replacing the current content of the address. The content of the Accumulator Register remains unchanged. The Program Counter Register is incremented by one.
Add (1*nn*) [ADD *nn*] The content of RAM address *nn* is added to the content of the Accumulator Register, replacing the current content of the register. The content of RAM address *nn* remains unchanged. The Program Counter Register is incremented by one.
Subtract (2*nn*) [SUB *nn*] The content of RAM address *nn* is subtracted from the content of the Accumulator Register, replacing the current content of the register. The content of RAM address *nn* remains unchanged. The Program Counter Register is incremented by one.
Input (901) [IN] or **[INP]** A value input by the user is stored in the Accumulator Register, replacing the current content of the register. Note that the two-digit operand does not represent an address in this instruction, but rather specifies the particulars of the I/O operation (see Output). The operand value can be omitted in the Assembly Language format. The Program Counter Register is incremented by one with this instruction.
Output (902) [OUT] or **[PRN]** The content of the Accumulator Register is output to the user. The current content of the register remains unchanged. Note that the two-digit operand does not represent an address in this instruction, but rather specifies the particulars of the I/O operation (see Input). The operand value can be omitted in the Assembly Language format. The Program Counter Register is incremented by one with this instruction.
Branch if Zero (7*nn*) [BRZ *nn*] This is a conditional branch instruction. If the value in the Accumulator Register is zero, then the current value of the Program Counter Register is replaced by the operand value *nn* (the result is that the next instruction to be executed will be taken from address *nn* rather than from the next sequential address). Otherwise (Accumulator >< 0), the Program Counter Register is incremented by one (thus the next instruction to be executed will be taken from the next sequential address).
Branch if Positive or Zero (8*nn*) [BRP *nn*] This is a conditional branch instruction. If the value in the Accumulator Register is nonnegative (i.e., >= 0), then the current value of the Program Counter Register is replaced by the operand value *nn* (the result is that the next instruction to be executed will be taken from address *nn* rather than from the next sequential address). Otherwise (Accumulator < 0), the Program Counter Register is incremented by one (thus the next instruction to be executed will be taken from the next sequential address).
Branch (6*nn*) [BR *nn*] or**[BRU *nn*]** or **[JMP *nn*]** This is an unconditional branch instruction. The current value of the Program Counter Register is replaced by the operand value *nn*. The result is that the next instruction to be executed will be taken from address *nn* rather than from the next sequential address. The value of the Program Counter Register is not incremented with this instruction.
No Operation (4*nn*) [NOP] or **[NUL]** This instruction does nothing other than increment the Program Counter Register by one. The operand value *nn* is ignored in this instruction and can be omitted in the Assembly Language format. (This instruction is unique to the VVM and is not part of the Little Man Model.)
Halt (0*nn*) [HLT] or **[COB]** Program execution is terminated. The operand value *nn* is ignored in this instruction and can be omitted in the Assembly Language format.