

Soru 4) Beş katmanlı bir boruhattı yapısında katmanlar ve bunların tamamlanması için gereken süreler: **instruction fetch (30ns)**, **instruction decode (10ns)**, **operand fetch (30ns)**, **instruction execution (10ns)** ve **writeback (30ns)** olarak verilmektedir. Boruhattının katmanlarının **ortak bir saat işareti** ile yürütülmesi istenmektedir. Komut setindeki **komutların boruhattının tüm katmanlarını kullandığını** varsayarak:

- Boruhattı için kullanılabilir en yüksek saat frekansını hesaplayınız.
- 1000 adet komutun **boruhattı kullanılarak** yürütülmesinin ne kadar süreceğini hesaplayınız.
- 1000 adet komutun **boruhattı kullanılmadan** yürütülmesinin ne kadar süreceğini hesaplayınız.
- Sırasıyla K1, K2, K3 ve K4 komutlarının boruhattı kullanılarak yürütülmesi istenmektedir. Aşağıdaki tabloyu kullanarak, verilen zaman aralığında, boruhattı katmanlarında hangi komutların yürütülmekte olduğunu belirtiniz.

| Clock Period | Instruction Fetch | Instruction Decode | Operand Fetch | Instruction execution | Write Back |
|--------------|-------------------|--------------------|---------------|-----------------------|------------|
| T1 | K1 | | | | |
| T2 | K2 | K1 | | | |
| T3 | K3 | K2 | K1 | | |
| T4 | K4 | K3 | K2 | K1 | |
| T5 | | K4 | K3 | K2 | K1 |
| T6 | | | K4 | K3 | K2 |
| T7 | | | | K4 | K3 |
| T8 | | | | | K4 |

a) Boruhattında ortak saat işareti kullanmak istenirse en uzun katman süresi belirleyici olur \Rightarrow saat işareti periyodu 30 ns
 \Rightarrow saat işareti frekansı $\frac{1}{30 \cdot 10^{-9}}$ Hz = 33.33 MHz

b) 1000 komutun 5 katmanlı boruhattında yürütülebilmesi için 1004 alt işin yürütülmesi gerekir $\Rightarrow 1004 \times 30 \text{ ns} = 30120 = 30.12 \mu\text{s}$

c) Boruhattı kullanılmadığında 1 komutun yürütülmesi $30 + 10 + 30 + 10 + 30 = 110 \text{ ns}$ sürer
 1000 komutun boruhattı olmadan = $1000 \times 110 \text{ ns} = 110 \mu\text{s}$ yürütülme süresi

| | | | | | | |
|--------|-------------------------------|----------|----------|----------|----------|----------|
| İsim : | BLM2612 Bilgisayar Donanımı | S1 (30p) | S2 (20p) | S3 (27p) | S4 (23p) | Σ (100p) |
| No : | 2017/2 Vize 2 – 11 Mayıs 2018 | | | | | |
| İmza : | Süre: 90 dk | | | | | |

Soru 1) 32 bit adres yolu, 64 bit veri yolu genişliğine sahip bir mikroişlemci, 2MB kapasitesinde (tag alanları hariç), blok genişliği 32 byte olan cache yapısına sahiptir.

- Aşağıda verilen cache konfigürasyonları için adres alanlarının genişliklerini belirleyiniz.

| | tag | index | Word-Offset | Byte-offset |
|-----------------------|-----|-------|-------------|-------------|
| fully associative | 27 | 0 | 2 | 3 |
| direct mapped | 11 | 16 | 2 | 3 |
| 2-way set associative | 12 | 15 | 2 | 3 |
| 4-way set associative | 13 | 14 | 2 | 3 |

- Boş bir cache ile başlandığı varsayılarak, yukarıdaki direct mapped cache konfigürasyonu için sırasıyla aşağıdaki adreslere erişilmek istenirse oluşacak hit ve miss durumlarını belirtin.

| Referans Adres (Hex) | Adres binary formda, alanlarına ayrıştırılarak | H/M |
|----------------------|--|-----|
| ABF908A0 | 1010 1011 1111 1001 0000 1000 1010 0000 | M |
| ABF908A8 | 1010 1011 1111 1001 0000 1000 1010 1000 | H |
| ABF908A4 | 1010 1011 1111 1001 0000 1000 1010 0100 | H |
| ACF9BBAC | 1010 1100 1111 1001 1011 1011 1010 1100 | M |
| ACF9BBA4 | 1010 1100 1111 1001 1011 1011 1010 0100 | H |
| ACF9BBA0 | 1010 1100 1111 1001 1011 1011 1010 0000 | H |

MP'nin word genişliği = $\frac{64 \text{ bit}}{8 \text{ bit}} = 8 \text{ byte/word} \Rightarrow$ byte offset için $\log_2 8 = 3$ bit gerekli

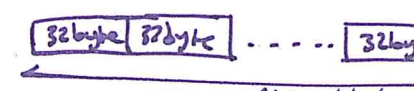
Cache bloğundaki word sayısı = $\frac{32 \text{ byte}}{8 \text{ byte}} = 4 \text{ word/block} \Rightarrow$ word offset için $\log_2 4 = 2$ bit gerekli

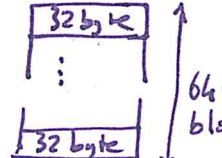
Fully associative cache'te index alanı yoktur, tag alanı = $32 - (3 + 2) = 27$ bit


Direct mapped cache için satır sayısı = $\log_2 \left(\frac{2 \text{ MB}}{32 \text{ byte}} \right) = \log_2 \left(\frac{2^{21} \text{ B}}{2^5 \text{ B}} \right) = 16 \text{ bit}$

2way set associative cache için satır sayısı = $\log_2 \left(\frac{2 \text{ MB}}{2 \times 32 \text{ byte}} \right) = \log_2 \left(\frac{2^{21} \text{ B}}{2^6 \text{ B}} \right) = 15 \text{ bit}$

4way set associative cache için satır sayısı = $\log_2 \left(\frac{2 \text{ MB}}{4 \times 32 \text{ byte}} \right) = \log_2 \left(\frac{2^{21} \text{ B}}{2^7 \text{ B}} \right) = 14 \text{ bit}$

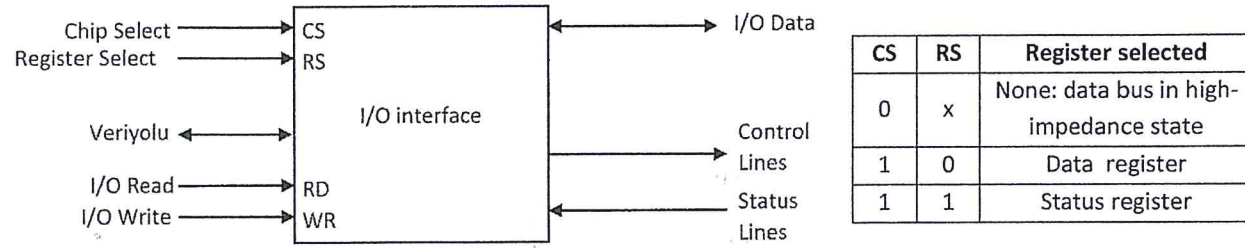
Fully associative cache: 

Direct mapped cache: 

2way set associative cache: 

4way set associative: 

Soru 2) Aşağıdaki şekilde verilen I/O cihazlarından 7 tanesinin, 8 bit adres yolu genişliğine sahip bir CPU ile bağlanması istenmektedir. I/O cihazlarının her birinin CS ucu farklı bir adres ucuna bağlanacaktır (A_7 yedinci cihazın CS ucuna, A_6 altıncı cihazın CS ucuna, ...). CPU'nun en düşük anlamlı adres ucu tüm I/O cihazlarının RS ucuna bağlanmıştır. Bu bağlantı sonucu tüm I/O cihazlarının yazmaçları için elde edilen adres değerlerini belirtiniz (toplamda 14 yazmaç için adres değeri).



| | | | | | | | | | | |
|-----|-------|-------|-------|-------|-------|-------|-------|-------|------|--------|
| | A_7 | A_6 | A_5 | A_4 | A_3 | A_2 | A_1 | A_0 | Data | Status |
| I01 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | x | 02H | 03H |
| I02 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | x | 04H | 05H |
| I03 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | x | 08H | 09H |
| I04 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | x | 10H | 11H |
| I05 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | x | 20H | 21H |
| I06 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | x | 40H | 41H |
| I07 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | x | 80H | 81H |

Soru 3) Komut tablosu, datapath yapısı, komut çözümleme devresi ile komut formatı verilen tek-çevrim hardwired CPU için

a) Aşağıda verilen komutlara karşılık komut çözümleme devresinin üreteceği binary değerleri tabloya yazınız. (5x3p=15p)

| Instruction | DA | AA | BA | MB | FS | MD | RW | MW | PL | JB | BC |
|----------------|-----|-----|-----|----|------|----|----|----|----|----|----|
| ADD R0, R7, R6 | 000 | 111 | 110 | 0 | 0010 | 0 | 1 | 0 | 0 | 0 | 0 |
| DEC R1, R0 | 001 | 000 | xxx | 0 | 0110 | 0 | 1 | 0 | 0 | 0 | 0 |
| NOT R3, R3 | 011 | 011 | xxx | 0 | 1011 | 0 | 1 | 0 | 0 | 0 | 1 |
| OR R4, R2, R1 | 100 | 010 | 001 | 0 | 1001 | 0 | 1 | 0 | 0 | 0 | 1 |
| SHR R4, R2 | 100 | xxx | 010 | 0 | 1101 | 0 | 1 | 0 | 0 | 0 | 1 |

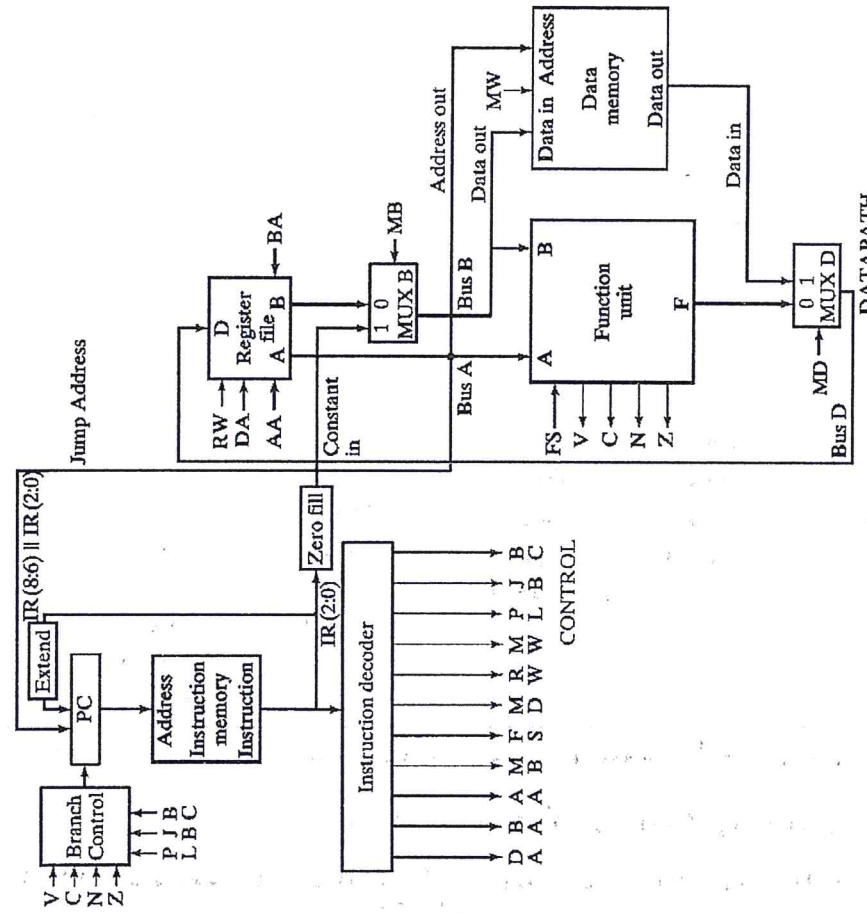
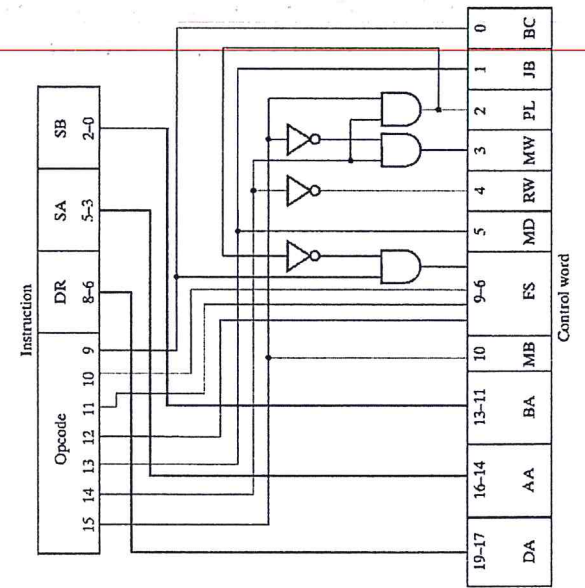
b) Aşağıda verilen komutların binary karşılıklarını bulunuz. (4x3p=12p)

| Instruction | Opcode | DR | SA | SB or Operand |
|----------------|----------|-----|-----|---------------|
| LD R0, R7 | 00100.00 | 000 | 111 | xxx |
| XOR R4, R3, R5 | 0001010 | 100 | 011 | 101 |
| SHL R0, R1 | 0001110 | 000 | xxx | 001 |
| ADI R3, R2, 3 | 1000010 | 011 | 010 | 011 |

Instruction Specifications for the Simple Computer

| Instruction | Mnemonic | Format | Description | Status Bits |
|--------------------|----------|----------------|---|-------------|
| Move A | 00000000 | MOVA RD, RA | $R[DR] \leftarrow R[SA]^*$ | N, Z |
| Increment | 00000001 | INC RD, RA | $R[DR] \leftarrow R[SA] + 1^*$ | N, Z |
| Add | 00000010 | ADD RD, RA, RB | $R[DR] \leftarrow R[SA] + R[SB]^*$ | N, Z |
| Subtract | 00000011 | SUB RD, RA, RB | $R[DR] \leftarrow R[SA] - R[SB]^*$ | N, Z |
| Decrement | 00000101 | DEC RD, RA | $R[DR] \leftarrow R[SA] - 1^*$ | N, Z |
| AND | 00010000 | AND RD, RA, RB | $R[DR] \leftarrow R[SA] \wedge R[SB]^*$ | N, Z |
| OR | 00010001 | OR RD, RA, RB | $R[DR] \leftarrow R[SA] \vee R[SB]^*$ | N, Z |
| Exclusive OR | 00010010 | XOR RD, RA, RB | $R[DR] \leftarrow R[SA] \oplus R[SB]^*$ | N, Z |
| NOT | 00010011 | NOT RD, RA | $R[DR] \leftarrow \neg R[SA]^*$ | N, Z |
| Move B | 00010100 | MOV B RD, RB | $R[DR] \leftarrow R[SB]^*$ | N, Z |
| Shift Right | 00010101 | SHR RD, RB | $R[DR] \leftarrow sr R[SB]^*$ | N, Z |
| Shift Left | 00010110 | SHL RD, RB | $R[DR] \leftarrow sl R[SB]^*$ | N, Z |
| Load | 10010000 | LDI RD, OP | $R[DR] \leftarrow z OP^*$ | N, Z |
| Add Immediate | 10000010 | ADI RD, RA, OP | $R[DR] \leftarrow R[SA] + z OP^*$ | N, Z |
| Load | 00100000 | LD RD, RA | $R[DR] \leftarrow M[SA]^*$ | N, Z |
| Store | 01000000 | ST RA, RB | $M[SA] \leftarrow R[SB]^*$ | N, Z |
| Branch on Zero | 11000000 | BRZ RA, AD | if $(R[SA] = 0) PC \leftarrow PC + se AD, N, Z$ | N, Z |
| Branch on Negative | 11000001 | BRN RA, AD | if $(R[SA] \neq 0) PC \leftarrow PC + 1$ | N, Z |
| Jump | 11100000 | JMP RA | if $(R[SA] < 0) PC \leftarrow PC + se AD, N, Z$ if $(R[SA] \geq 0) PC \leftarrow PC + 1$ | N, Z |

* For all of these instructions, $PC \leftarrow PC + 1$ is also executed to prepare for the next cycle.



| Opcode | Destination register (DR) | Source register A (SA) | Source register B (SB) |
|--------|---------------------------|------------------------|------------------------|
| Opcode | Destination register (DR) | Source register A (SA) | Source register B (SB) |
| Opcode | Destination register (DR) | Source register A (SA) | Source register B (SB) |