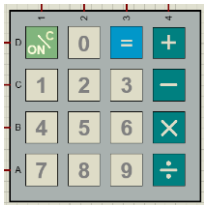
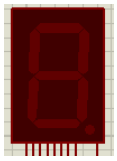
Lab 2 Mikroişlemciler Grup – EU-B & HOI-B

Lab gösterme tarihi 20 Kasım 2020 Cuma Lab saatine. (Yeşil renkli kısımlar işinizi kolaylaştırmak için verilen bilgilerdir. Bu bilgiler de dahil, bu dokumandaki tüm istenen koşullar, (özellikle mavi kısımlar, sağlanmalıdır.)

Soru: Handshaking sinyaleri ile tek yönlü veri gönderme ve alma ile ilgili basit bir hesap makinesi oluşturmanız beklenmektedir. 1 giriş (keypad) ve 2 çıkış (7 segment ile 16 segment) elemanlarını kullanmanız beklenmektedir. İki ayrı 8255 ile I/O kontrolü sağlayacaksınız. Birinci 8255'e keypad ve 7 segment bağlı olacak ve *handshake* ile veri alma-gönderme yapılacak (Mod1). İkinci 8255'e 16 segment bağlanacak ve PortA & PortB birarada output olarak bu göstergeyi kontrol edecek (Mod0). Birinci 8255 için PortA 200H adresinden başlayarak ardışık olarak çift adreste, ikinci 8255 için PortA 60H adresinden başlayarak ardışık olarak çift adreste işlem yapılacak.

Birinci 8255'e bağlı 7 segment ve keypad için; Mod1'de PortB veri çıkışı (7 segment bağlı), PortA veri girişi (keypad bağlı) olacak şekilde tasarımınızı tamamlayınız. PortC uçlarının *handshaking* için kullanılması gerektiğini hatırlayınız.

Keypad: 

7 segment (common cathode olacak): 

İkinci 8255'e bağlı 16 segment Mod0'da çalışacak ve PortC pinleri hiç kullanılmayacak.

16 segment (common anode olacak): 

Gerçeklenecek olan devre tek basamaklı 4 işlem yapan basit bir hesap makinesi devresidir ve aşağıdaki gibi çalışmaktadır:

1. Keypad den girilen 1. sayı değeri 7 segmentte gösterilmeli, (16 Segment henüz hiçbir şey göstermiyor)(Girdi alınan sayıya ilişkin *negatiflik*, *çok basamaklılık*, *arda arda 2 sayı girme* gibi kontrolleri yapmanıza gerek yok; ilk girilen değeri pozitif olarak alsın, yeterli.)
2. Sonra herhangi dört işlem tuşuna basıldığında (+ - x :), 7 segmentte "0" değeri gösterilmeli. (16 Segment henüz hiçbir şey göstermiyor)
3. 2. sayıya bastığımızda ise 2. sayıyı göstermeden, direkt işlemin sonucunu 7 segmentte göstermeli. Bu noktada *hatalı* ve *hatasız* işlemlere göre 16 segmentin davranışı kontrol edilmeli.

   Hatasız işlem: 3 + 5, 7 – 2, 6 : 2, 2 x 1 gibi sonucu pozitif olan, birbirine bölünebilen, 1 basamaklı çıkış veren işlemleri dikkate alın ve sonucunu 7 segmente yazdırın; 16 segmente ise sırasıyla 'O' , 'N' , 'A' , 'Y' harflerini aralarda kısa süreli *delay* vererek yazdırın (DELAY PROSEDÜRÜ).

   Hatalı işlem: 5 + 6, 2 – 7, 6 : 4, 2 x 8 … vb gibi sonucu negatif olan, tam bölünemeyen, 1 basamaktan fazla sonuç veren işlemleri ekranda, 7 segmenti tamamen söndürüp; 16 segmente sırasıyla 'H' , 'A' , 'T' , 'A' harflerini aralarda kısa süreli *delay* vererek yazdırın (DELAY PROSEDÜRÜ).

16 segmentte harfleri gösterirken bir miktar delay olacağı için, ister önce 7 segmentte, isterseniz önce 16 segmentte ilgili verileri yazdırabilirsiniz, paralel olamayacak, ardışıl gerçekleşecek; size kalmış.

Not1: Flagları kullanmayı ihmal etmeyin. Her olası *hatalı* işlem için kullanılması gerekli olan kontrolleri göz önüne alın. Taşma var mı? İşaretli bir durum söz konusu mu? Öte yandan kalanlı / kalansız DIVision işlemine bakınız (MOD -%- bir çözüm olabilir).

Not2: *Algoritma akışı*: Öncelikle bir tuşa basıldığını anlamanız gerekli, ilgili değeri aldıktan sonra, handshake ile birinci 8255'e veriyi gönderip, oradan handshake ile 7 segmente bastırmalısınız. Sonra aynı işlemi, basılan yeni operator tuşu için (+, -, x, :) yaparak yine handshake ile gönderip 7 segmente 0 basmalısınız. Birinci 8255'te her *veri al-gönder* adımlarının handshake'e uygun olması gerekli. Bu noktaya kadar 16 segmentte herhangi bir karakter olmamalı. En son adımda, son defa sayısal tuş değerini (2. operand) handshake ile alıp, aritmetik işlemin ve kontrolünün ardından ilgili doğru sonucu 7 segmente handshake'e uyarak bastırınız (veya hatalı işlem varsa 7 segmenti söndürünüz). Son adımdaki işlem kontrolüne göre hatalı/hatasız işlemin etiketini ('O' , 'N' , 'A' , 'Y' veya 'H' , 'A' , 'T' , 'A') 16 segmentte DELAY prosedürü ile gösteriniz. (Dediğim gibi; 7 segment ve 16 segmentte veri gösterme sırası önemsiz. İsterseniz önce 16 segmente yazıp, sonra 7 segmenti güncelleyebilirsiniz.)

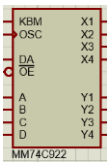
'O' DELAY 'N' DELAY 'A' DELAY 'Y'

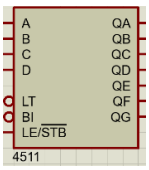
Kolaylık olması için örnek bir DELAY PROC →

```
DELAY PROC NEAR
PUSH CX
MOV CX, 05FFFh
COUNT:
LOOP COUNT
POP CX
RET
DELAY ENDP
```

Not3: Handshake kısmı derste anlatılan kısım gibi. Kolay, ara devreleri (NOT kapısı olan yapıdan bahsediyorum) oradan bakarak yapabilir; kodlayabilirsiniz.

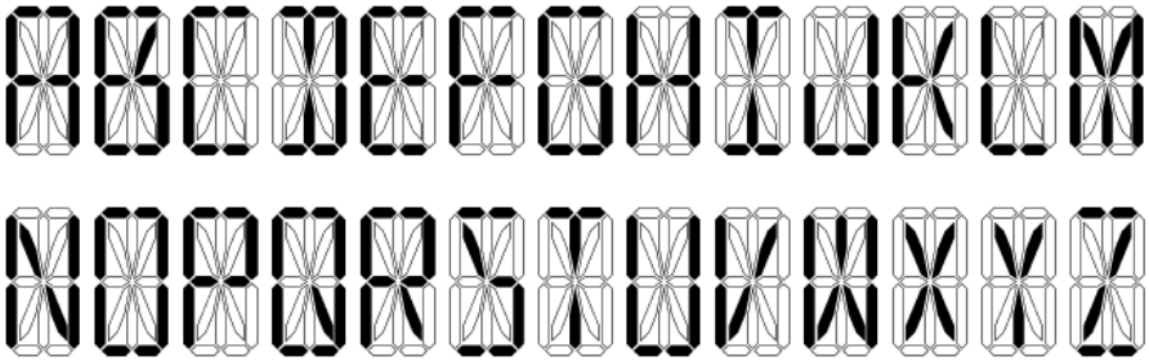
Not4: Key paddeki On/C tuşu ve '=' işareti önemsizdir, kullanmanıza gerek yok.

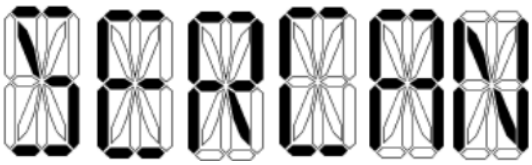
Not5: Keypad için 

yardımcı devresini, 7 segment için 

yardımcı devresini kullanmayı unutmayın; hem handshake için gerekli, hem de işlerinizi kolaylaştıracak. 16 segment Mod0'da çalışacağı ve ayrı bir 8255'e bağlı olduğu için ek eleman gerektirmiyor (yardımcı donanım KESİNLİKLE KULLANILMAYACAK; pinler direkt Port A & B ile sürülecek). İkinci 8255'in PortA ve PortB tüm pinlerini kullanmanız gerekecek (PortA ve PortB output; PortC hiç kullanılmayacak). Bu segmente veri gönderirken, göndereceğiniz veriyi, örneğin "T" harfini, 16 segmentin yanan ve yanmayan bacaklarına göre belirleyip, HEX değeri elde ettikten sonra PortA ve PortB'ye uygun **yarı değerleri** göndereceksiniz; yaptığınız bağlantı da burada önemli olacak. Segment pinleri PortA'nın / PortB'nin hangi pinlerine bağlı? (MSB – LSB sıralaması vb.)

16 segmente ilişkin harf tablosunu şöyle gösterebiliriz:



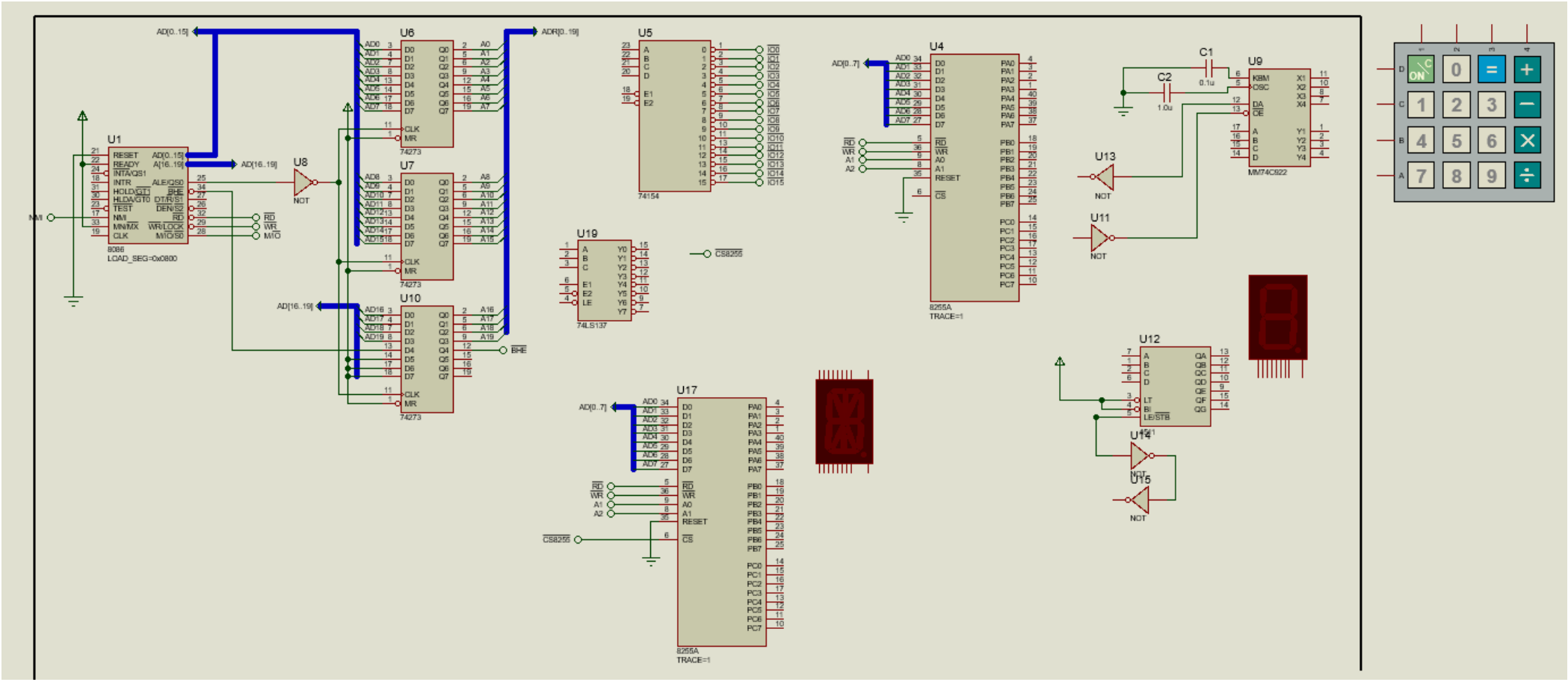
SERCAN = 

Segmentin *Common Anode* olmasını göz önüne almayı unutmayınız. Bacak numaraları için internetten kısa bir araştırma yapınız.

Not6: Devre 1 defa aritmetik işlemi yapınca 2. defa başka bir aritmetik işlemi yapmak için yeniden başlatılmak zorunda kalınsa da sorun yok; yeter ki 1 defa 2 sayıya aritmetik işlem uygulasın ve hata kontrolü yapsın. Simülasyonu durdurup, yeniden açsak da sıkıntı yok.

Not7: Proteus sürümleri arası sıkıntı olmaması adına, kullanılacak temel blokları aşağıda paylaşıyorum. Genel şablona ilişkin fikir oluşacaktır.

→ diğer sayfaya bakınız.



Notlandırma: Çalışan tasarım ve assembly kod & sorulan sorulara tam cevap → 100

Diğer durumlarda değerlendiren asistan her hatalı durum / yanlış tasarım / hantal ve yorum satırsız kod / yanlış cevap vb. için -25'er not düşecektir.