

YILDIZ TEKNİK ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

**80196 -16 BİTLİK MİKRODENETLEYİCİLİ
GELİŞTİRME SETİ TASARIMI VE FP 3000 FUZZY
İŞLEMCİYİ KULLANARAK ÇOK AMAÇLI
MİKRODENETLEYİCİLİ FUZZY LOJİK KONTROL VE
UYGULAMASI**

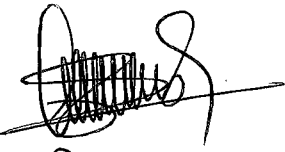
Elektronik. Yük. Müh. Ö. Faruk ÖZGÜVEN

**F.B.E. Elektronik ve Haberleşme Mühendisliği Anabilim Dalında
hazırlanan**

DOKTORA TEZİ

Tez Danışmanı

:Prof. Dr. Galip CANSEVER



Prof. Dr. Galip Cansever

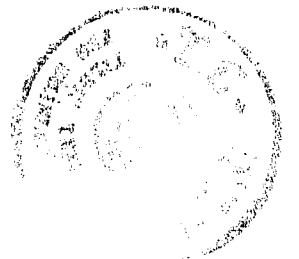


Prof. Dr. Uğur Çilingiroğlu



Prof. Dr. Sezgin Alsan

İSTANBUL, EKİM 1996



İÇİNDEKİLER:

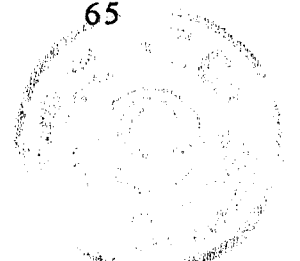
ÖZET

SUMMARY

1.GİRİŞ	1
1.1.Fuzzy lojik algoritması kullanan diğer deneysel çalışmalar	7
1.2.Fuzzy işlemci tasarımı için yapılan makale çalışmaları ve işlemcilerin özellikleri	16
2.FUZZY KÜME TEORİSİ	24
2.1.Fuzzy küme kavramı	24
2.2.Fuzzy kümelerde teorik işlemler	25
2.2.1.Fuzzy kümenin matematiksel gösterilimi	25
2.2.2.Birleşme özelliği	26
2.2.3.Keşisme özelliği	27
2.2.4.Evrik alma özelliği	27
2.2.5.Üs alma özelliği	28
2.2.6.Pozitif α sayısı ile çarpma özelliği	28
2.3.Fuzzy kümeler arasındaki mesafe	28
2.4.Klasik ve fuzzy bağıntılar	29
2.4.1.Klasik bağıntılar	29
2.4.1.1.Kartezyen çarpım	29
2.4.1.2.Klasik bağıntılarda işlemler	30
2.4.1.3.Klasik bağıntılarda kompozisyon	30
2.4.2.Fuzzy bağıntı	31
2.4.2.1.Kartezyen çarpım	31
2.4.2.2.Fuzzy bağıntılarda kompozisyon	32
2.4.2.3.Fuzzy çıkarım kompozisyonu	32
2.5.Üyelik fonksiyonları	33
2.5.1.Üyelik fonksiyonu kavramı ve çeşitleri	33
2.5.2.Normal ve iç bükey üyelik fonksiyonu tanımı	34
2.5.3.Bulanıklaştırma işlemi	35



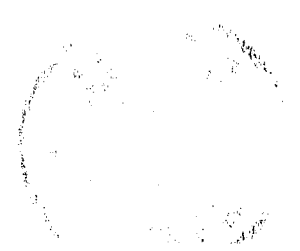
3.ÇIKARIM İŞLEMİ	36
3.1.Birinci tip karar verme	44
3.2.İkinci tip karar verme	45
3.3.Üçüncü tip karar verme	46
3.4.Dördüncü tip karar verme	47
3.5.Çıkarım işleminin bir uygulama üzerinde grafiksel ve cebirsel anlatımı	48
3.5.1.Grafiksel anlatım	48
3.5.2.Cebirsel anlatım	51
4.NETLEŞTİRME YÖNTEMLERİ	54
4.1.Maksimum üyelik yöntemi	54
4.2.Merkezi yöntem	54
4.3.Ağırlık ortalamasını bulma yöntemi	55
4.4.Maksimum üyelikleri ortalama yöntemi	56
4.5.Toplamların merkezini bulma yöntemi	56
4.6.En büyük alan merkezini bulma yöntemi	57
4.7.İlk veya son maksimum değeri bulma yöntemi	58
5.FUZZY LOJİK KONTROL	59
5.1.Temel Fuzzy Lojik Kontrol	59
5.2.Genel Fuzzy Lojik Kontrol	60
6. 8XC196KC/KD 16 BİT MİKRODENETLEYİCİ	62
6.1.MİB denetleme	62
6.2.Kaydedici bloğu	63
6.3.Aritmetik mantık birimi	64
6.4.Bellek denetleyicisi	64
6.5.Kesme isteği denetleyicisi	64
6.6.Standart Giriş/Çıkış portları	65
6.7.Seri Giriş/Çıkış portu	65
6.8.Yüksek hızlı Giriş/Çıkış ünitesi	65
6.8.1.Zamanlayıcı1 ve Zamanlayıcı2	65



6.8.2.Yüksek hızlı Giriş	66
6.8.3.Yüksek hızlı Çıkış	66
6.9.Analog dijital çevirici	66
6.10.Darbe genişlik modülatörü	67
6.11.Gözetlemeli sıfırlama sayıcısı	67
6.12.İç zamanlama	67
7. 8XC196KC/KD ADRESLEME TÜRLERİ	68
7.1.Dolaysız-Kaydedici adresleme	68
7.2.Dolaylı adresleme	69
7.3.Kendinden artmalı dolaylı adresleme	69
7.4.Anında adresleme	70
7.5.Kısa indekslemeli adresleme	70
7.6.Uzun indekslemeli adresleme	70
7.7.Sıfır kaydedici adresleme	71
7.8.Yığın gösteren kaydedici adresleme	71
8. 8XC196KC/KD BELLEK BÖLGELERİ	72
8.1.Dış bellek bölgesi	73
8.2.Port 3 ve Port 4	73
8.3.Program ve özel amaçlı bellek	73
8.4.İç veya dış bellek bölgesini seçme	73
8.5.Program belleği	74
8.6.Özel amaçlı bellek	74
8.6.1.Saklı tutulan bellek gözleri	75
8.6.2.Kesme isteği vektörü	75
8.6.3.Güvenlik anahtarı	75
8.6.4.Tüm devre ayarlama baytı	75
8.7.Kaydedici bloğu	75
8.7.1.Genel amaçlı kaydedici	76
8.7.2.Yığın gösteren kaydedici	77
8.7.3.Özel fonksiyon kaydedicileri	77

8.8.Yatay pencereleme	78
8.8.1.Ypencere seçme işlemi	78
8.8.2.Ypencere0	79
8.8.3.Ypencere1	79
8.8.4.Ypencere15	79
8.9.Düşey pencereleme	80
8.9.1.Dpencere seçme işlemi	82
9.SERİ GİRİŞ/ÇIKIŞ PORTU	82
9.1.Mod 0	83
9.2.Asenkron modlar	85
9.2.1. Mod 1	85
9.2.2. Mod 2	87
9.2.3. Mod 3	87
9.2.4. Mod 2 ve Mod 3 zamanlama	88
9.3.Seri portun programlanması	89
9.3.1.Eşitlik işlemine izin verme ve haberleşme modunu seçme	89
9.3.2.TXD ve RXD için şartlamalar	89
9.3.3.Seri port kesme isteğine izin verme	90
9.3.4.Karakter hızı ve saat darbesi kaynağını programlama	91
9.3.5.Seri portun durumunu tespit etme	93
10.MİKRODENETLEYİCİNİN DIŞ BELLEK ARABİRİMİ İLE BAĞLANMASI	95
10.1.Denetleyici Ayarlama Kaydedicisi	95
10.2.Yol genişliğinin ve belleğin ayarlanması	96
10.3.Yol denetleme modları	97
10.3.1.Standart yol denetleme modu	97
10.3.2.Yazma tutma modu	99
10.3.3.Geçerli adres tutma modu	101
10.3.4.Yazma tutma modu ile beraber geçerli adres modu	103
11.YÜKSEK HIZLI GİRİŞ/ÇIKIŞ ÜNİTESİ	104

11.1.Zamanlayıcılar	104
11.1.1.Zamanlayıcı1 işlevleri	104
11.1.2.Zamanlayıcı2 işlevleri	105
11.1.3. Zamanlayıcı2'yi programlama	105
11.1.4.Saat darbesi kaynağını seçme	106
11.1.5.İleri/Geri sayma	108
11.2.Yüksek Hızlı Giriş Modül(HSI)	108
11.2.1.Sonuç verisini okuma	109
12.2.2.HSI modülünü programlama	110
12.2.3.Kenar geçiş çeşitlerini tanımlama	110
12.3.4.HSI kesme isteklerine izin verme	111
12.3.5.HSI bacaklarına izin verme	113
12.FP 3000 FUZZY İŞLEMCİ	114
12.1.FP 3000 fuzzy işlemcinin bacak bağlantıları ve tanımları	115
12.2.Sistem donanımı	116
12.3.Fuzzy işlemcili sistemin bellek haritası	120
12.3.1.Giriş verilerinin yazılması	120
12.3.2.Çıkış verilerinin okunması	121
12.3.3.Çıkarım işlemine başlatma ve çıkarımın bittiğini tespit etme	122
12.3.4.Bilgi tabanı verileri	122
13.MİKRODENETLEYİCİLİ SİSTEMİN ANAKART DONANIMI VE DİĞER ÜNİTELER	124
13.1.Gösterge ve tuş takımı devresi	131
13.2.Dijital analog çevirici devresi	134
13.3.Motorlar ve manyetik algılayıcılar	137
14.PROGRAMLAR	138
14.1.Gösterge ve tuş takımı kullanan geliştirme seti programı	138
14.2.Programın seri porttan geliştirme setine gönderilmesi	142
14.3.Bilgi tabanı verilerinin fuzzy işlemciye gönderilmesi	145



14.4.Mikrodenetleyici ve fuzzy işlemciyi kullanarak Fuzzy Lojik Kontrol uygulama programı	148
14.5.Kesme isteği alt programı	148
15.ELDE EDİLEN EĞRİLERİN ANALİTİK YORUMU	154
15.1.Tasarlanan Fuzzy Lojik Denetleyicinin çeşitli ölçek değerleri için performansının eğriler ile değerlendirilmesi	170
15.2.Sistemin yüklenmesi ve yükten bırakılması durumu	177
16.SONUÇLAR	179
16.1.Çalışmanın maliyet hesabı ve diğer üniteler ile karşılaştırılması	182
KAYNAKLAR	185
EK1	197
EK2	208
EK3	216
EK4	218
EK5	224
ÖZGEÇMİŞ	



TEŞEKKÜR

Bu tezin hazırlanmasında ve çalışmalarım sırasında ilgisini, yardımını ve desteğini esirgemeyen, her türlü imkanın sağlanmasına neden olan, beni çalışmalara teşvik eden Sayın Hocam Prof. Dr. Galip CANSEVER' e ve ayrıca çalışmalarımda gerekli anlayışı gösterip bana yardımcı olan diğer arkadaşlara teşekkürlerimi sunarım.

İstanbul, 1996

Ö. Faruk Özgüven



ÖZET:

Geçen onyıl boyunca, fuzzy lojik kontrol fuzzy küme teorisi uygulamaları ile özellikle giriş-çıkış bağıntıların olmayışı yüzünden alışageldiğimiz yöntemler tarafından gereği gibi kontrol edilemeyen endüstriyel işlemlerde son derece etkili ve aktif araştırma alanlarından birisi olarak ortaya çıktı.

Fuzzy lojiğe dayanan fuzzy lojik kontrol alışageldiğimiz mantıksal sistemlerden daha fazla sözel anlatıma ve insan düşünme sistemine daha çok yakındır. Bu durum uzman bilgiye dayanan sözel kontrol stratejisinin otomatik kontrol stratejisine dönüştürmenin bir vasıtası olur.

Elektriksel sürücülerin yaptığı hız ve pozisyon kontrolü, son yıllarda fuzzy lojik yaklaşımı ile önerilmiş ve ayrıca geleneksel ayarlayıcıların performanslarının arttırılması çalışmaları yapılmıştır. Bazı kontrol sistemlerinde hız hatası ve türevi olduğu yerlerde bir genel şema çerçevesinde belirlenmekte ve bu durumda bulanıklaştırma, uygun üyelik fonksiyonları ile yapılmaktadır. Bu durumda kontrol işaretlerinin algılanıp sözel kurallar ile değerlendirilerek bulanıklaştırılan işaretler çıkarım işlemine gönderilmektedir. Fuzzy işlemciler fazla yazılıma ihtiyaç duyulmadan çıkarım işlemini donanım vasıtasıyla yaparlar. Ayrıca bu işlemciler mikrodenetleyiciler ile uyumlu çalışırlar.

Mikrodenetleyiciler içerisinde bir çok çevre birimlerini içerdikleri ve hızlı işlem yapabilme kapasitesine sahip oldukları için endüstride vazgeçilmez bir kontrol elemanı olmuşlardır.

Bu tezde önce 8XC196KC 16 bitlik mikrodenetleyici geliştirme seti ve sonra bu sisteme ilave edilmek üzere FP 3000 fuzzy işlemcili çıkarım kartı tasarlanmış ve gerçekleştirilmiştir.

Geliştirme setinde gerek tuş takımından yazılan gerekse bilgisayardan gönderilen makina dili programların çalıştırılması sağlanır. Bilgisayarın seri portundan fuzzy işlemcili çıkarım kartına ilgili bellek adreslerine yazılmak üzere kurallar ve üyelik fonksiyonlarının derlenmesinden oluşan bilgi tabanı verileri gönderilir. Sonra fuzzy işlemciye giriş değerleri verilerek çıkarım işlemi sonucundaki değer yine işlemci tarafından hesaplanır.

Ayrıca sisteme 12 bitlik dijital analog çevirici ve iki DC motor ilave edilmiştir. Son olarak bu sistem üzerinde fuzzy lojik uygulaması gerçekleştirilmiştir.



SUMMARY:

During the last decade the theory of fuzzy sets and fuzzy logic control as well as their applications have appeared as a quite affective and active research area on the industrial processes that cannot be properly controlled by using conventional methods.

Fuzzy logic control which is based on fuzzy logic is much mains natural language and way of reasoning than traditional systems. This is a way of converting of linguistic control strategy which is based on expert knowledge to the strategy of automatic control.

The speed and position control of electrical drivers using fuzzy logic approach was recently proposed, mostly to improve the performances of conventional regulators. Such controllers are based on a general scheme where the speed error and its derivative are evaluated and fuzzified exploiting suitable membership functions. The fuzzified signals are sent to the inference block where control signals are deduced evaluating linguistic rules. Fuzzy processors do inference by means of hardware without necessitating much software. Further these processors are work in compatible with microcontrollers.

Since they include many peripheral units and they have capacity of fast performance, the microcontrollers become giving up control element in industry.

In this thesis, first a 16 bit microcontroller evaluation board of 8XC196KC/KD and then in addition to this system a fuzzy inference board is designed and realized.

In the evaluation board, it is achieved to run machine language programs that are written by through keyboard or sent by computer. Data of knowledge base that is composed of rules and membership functions are sent through to serial port of the computer to be written the relevant memory addresses to the fuzzy inference board. Later, by giving input values to processor value after the inference is computed by the processor again.

A part from above, two DC motors and a 12 bit digital analog converter is added to the system. Least, fuzzy logic control is realized on this system.

1.GİRİŞ:

Bu tezde bir fuzzy lojik denetleyici sisteminin gerçel zamanda çalıştırılıp, denetlenen sistem üzerinde daha etkin bir performans sağlanması, sistem kararlılığının geçici durumunun ve aynı zamanda sönüm olayının en aza indirilmesi problemine çözüm getirilmesi hedeflenmiştir.

Gerçek zamanda 80196 mikrodenetleyicili ve FP 3000 fuzzy işlemcili bir sistem ile donanım ve yazılım tasarlanarak daha önce bu konu ile yapmış olduğumuz simulasyon çalışmalarında bulunan sonuçların gerçel zaman ortamında elde edilmesi yoluna gidilmiştir. Ayrıca bu sistemi gerçekleştirirken maliyetin en aza indirgenmeside diğer bir hedef olmuştur. Bu hedefin gerçekleşmesi için üniversite ortamında sistemin tamamını her şeyi ile göz önüne alarak yazılımın ve donanımın gerçekleştirilmesi sağlanmıştır.

Problemlere çözüm getirebilmek için, bu tez çalışmasında; iki adet D.C. motorun farklı hızlarda dönmesi ile meydana getirdikleri aksaklıkları 80196 'lı ve fuzzy işlemcili bir sistem üzerinde yazılan bir makine dili program yardımı ile hızı sabit olan motorun hızına çok kısa bir süre içerisinde ulaşılmaya çalışılmış ve bu iki hız birbirine eşitlenerek kilitlenmiştir.

Fuzzy lojik denetleyici için bulanıklaştırma, çıkarım ve netleştirme işlemleri yalnız yazılım kullanarak bir program aracılığı ile gerçekleştirilebilir. Bu durum yazılımın büyük olmasına, gerçel zaman kontrolunda zaman kaybına ve bilgisayar ile kontrol gerçekleştiği için başlı başına bazı taşınabilir cihazların kontrolunu yapmak mümkün olmayabilir. Modüler bir yapıya sahip olmaz. Ayrıca kontrol edilecek cihazın durumuna göre bilgisayarın içerisine çeşitli (zamanlayıcı, DAC, ADC, giriş-çıkış portu gibi) kartların ilave edilmesi gerekir. Bu durum maliyetin daha fazla büyümesine neden olur. Fuzzy lojik denetleme işleminde belirtilen bu durumların ortadan kalkması için

mikrodenetleyici ve fuzzy işlemci ile beraber makine dili yazılım kullanarak sağlanabilir. Böylece mikrodenetleyici fuzzy işlemler için gerekli olan hesaplamaları sadece fuzzy işlemcili sistemin donanımını kullanarak gerçekleştirir. Fuzzy işlemcinin kullanılması sayesinde yazılım küçüldüğü gibi ve makine dilinin de kullanılması gerçel zaman işlemlerinde en önemli bir parametre olan hesaplama hızının artmasına hesaplama süresinin azalmasına neden olur. Sonuçta bu çalışma ile bir bilgisayara bağlı olmaksızın her yerde çalışabilen ve hızı yüksek olan bir fuzzy lojik denetleyici tasarlanmıştır. Ayrıca çok girişli ve çok çıkışlı, fazla kural sayısı içeren ve hızlı fuzzy lojik denetleyicisi en az maliyet ile elde edilmiştir.

Bu denetleyicinin uygulama alanı olarak şimdiye kadar tek motorlu sistem yerine iki motorlu sistem düşünülmüş ve ayrıca ilgili motorlara yükleme yapılarak çıkış eğrileri gerçek zamanda bilgisayar ekranında incelemiştir. Ayrıca modüler bir cihaz ve daha az malzeme ile fuzzy lojik algoritması kullanılarak diğer sistemlerin rahatlıkla denetlenmesi ve böyle bir sistemin bu denetleme işlemine nasıl cevap verdiği gerçek zamanda incelenmesi mümkün duruma getirilmiştir.

Yukarıda belirtilen problemlere getirilen çözüm için yapılan bu çalışmalar kontrol alanında bu sistemin kendisine özel ilk defa yapılan orijinal çalışmalardır.

Bu bilgiler ışığında sistem tasarlanırken sistemi oluşturan elemanların özellikle kontrol alanında kullanım özellikleri incelenmiştir. Bu nedenden dolayı bu tezde her elemanın rahatlıkla kullanılabilmesi için o elemana ilişkin özellikler detaylı olarak verilmiştir. Mikrodenetleyici ile ilgili bilgiler geniş bir şekilde tutulmuştur. Ayrıca mikrodenetleyici programlanırken çok önem taşıyan ve kullanılan özel amaçlı kaydediciler ,üniteler detaylı bir şekilde anlatılmıştır. Bu tez mikrodenetleyici ile çalışma yapanlar için bir referans olabilecek niteliktedir. Gerek mikrodenetleyicinin ve

gerekse fuzzy işlemcinin kullanılan fonksiyonlarının çoğu üzerinde durulmuştur.

Sonuçta oluşturulan bu sistem sadece bilgisayar ekranında kalacak bir simulasyon çalışmasından daha yararlı olabilecek endüstriyel uygulamalar için kullanılabilecek duruma getirilmiştir. Sistem yeni gelecek önerilere ve geliştirilmeye açık durumdadır. Ayrıca çalışmanın kolay ve pahalı yönü olan donanım hazır alınmadan, hem donanım ve hem de yazılım üzerinde temelden çalışılarak tasarım yapılmış ve kontrol edilecek sisteme uygulanmış, sistemin bu kontrol olayına nasıl cevap verdiği bilgisayar ekranında gerçek zamanda çizilen eğriler incelenerek tespit edilmiştir.

Şimdiye kadar motor kontrolunda yapılan uygulamalar yalnız bir motorun devri ölçülerek bu devir istenilen sabit bir devir hızına getirilmeye çalışılmıştır. Fakat bu tezde, daha önceki çalışmalarda sabit hız olarak alınan devir, hızı isteğe göre potansiyometre ile değiştirilen ikinci bir motorun devri referans alınmıştır. Böylece aynı anda iki motorun devir hızı ölçülmüş ve fuzzy lojik mantığı kullanarak iki motorun hızları kilitlenmiştir. Bu duruma frekans olarak bakıldığında, farklı iki frekansın birbirine kilitlenmesi işlemi fuzzy lojik denetleyici kullanarak sağlanmıştır. Eğer hızı kontrol edilen motorun yerine gerilim kontrollü osilatör (VCO) kullanılırsa elde edilen uygulama fuzzy lojik denetleyici kullanan faz kilitlemeli çevrime benzemektedir.

Bu sisteme daha sonra üçüncü bir motor daha ilave edilmiştir. Bu motor hızı kontrol edilen motora bağlanarak jeneratör olarak kullanılmıştır. Bu jeneratörün çıkış uçlarına 2.2Ω 'luk direnç yük olarak bağlanmıştır. Kontrol edilen motorun devri diğer motorun devrine ulaştığında bu direnç ile yükleme yapılarak sonuçta fuzzy lojik denetleyicinin motoru tekrar ne kadar sürede yüklenmeden önceki devrine getirdiği gözlenmiştir. Yine benzer bir şekilde yüklü motor

istenilen devire geldikten sonra jeneratör yüklenmeden bırakıldığı zaman yine motorun kendini istenilen ikinci motorun devrine ne kadar bir zamanda getirdiği eğriler üzerinde incelemiştir.

Uygulama için 12 V'luk D.C. motorlar seçilmiştir. Bu tercihin yapılmasının nedeni her yere taşınabilir ve sürücü devrelerine ihtiyaç duyulmadan bir devir hızı kontrol uygulamasını yapmak içindir. Bilineceği üzere servo motorların sürücü devreleri girişine 0 ile 10 V arası gerilim verilerek devir değiştirilmekte ve gerekli devir hızı bilgileri shaft encoder ünitesinden alınmaktadır. Bu tezde kontrol için elde edilen gerilimler kolaylıkla bu sürücü devresine fazla değişiklik yapmadan uygulanabilir. Bu durumda sistemden daha iyi cevap alınabilir.

Denetleme işleminde en önemli işlem olan motorların devrini ölçme mikrodenetleyicinin zamanlayıcı1 modülüne bağlı olan HSI (yüksek hızlı Giriş-Çıkış) bacakları kullanılmıştır. Bu sayede bir veya birden fazla giriş değerlerinin periyodunu ölçmek mümkündür. Sistem iki veya daha fazla motorun devrini ölçme yeteneğine sahiptir.

Sistemde motorların dönüş hızlarını alıp değerlendirebilmek için her motor için ayrı olarak magnetik algılayıcı ve bu algılayıcıların önünden geçen metal pervane kullanılmıştır. Bir devir ölçülmesi için pervanenin iki kanadının magnetik algılayıcının önünden geçmesi gerekir. Bu durumdan sonra ancak kontrol işlemine geçilmektedir. Eğer daha kısa bir sürede örneğin shaft encoder ile devir hakkında bilgi alınırsa sistemin kontrol hızı daha da artacaktır.

Sistem üzerinde bulunan boş slotları sayesinde başka kontrol olaylarında kullanılmaya uygundur. Ayrıca istenilirse daha değişik fuzzy işlemciler sisteme uyarlanıp çalıştırılabilir.

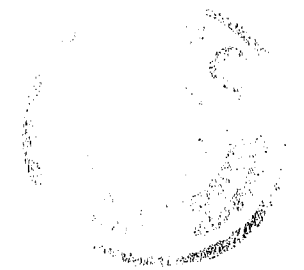


Fuzzy küme teorisi ile ilgili ilk bağıntılar L.A. Zadeh tarafından bir makalede yayınlandı, (Zadeh, 1965). Zadeh bu makaleyi yayınlarken insanların düşünme sisteminin konu ile ilgili iki ana hususun farkına vardı. Bunlardan birincisi insanların tamamıyla anlaşılmamış belirsiz kavramları çözömleme yeteneğinin ve insan düşüncesinin ana öğesinin sayısal ifadeler olmadığının farkına varması ve diğeri ise insanların kontrol ettikleri sistemler ile o zamana kadar geleneksel yöntemler ile kontrol edilen sistemler arasındaki ilişkinin uygun olmadığını görmesi olarak söylenebilir.

1970 ve 1980 tarihlerinde araştırmacılar teorisinin matematiksel yönleri ile ilgili çalışmalar yaptılar. Temel teorileri geliştirme sırasında bazı araştırmacılar aktif çalışmalar yaptılar, (Zadeh, 1973; Kaufman, 1975). Temel fuzzy küme teorisinin matematiksel olarak geliştirilmesi günümüzde devam etmektedir.

Fuzzy küme teorisi ile teorik hesaplamalarda sürekli olarak kullandığımız matematik ve olasılık teorisi arasında farklar vardır. İki teori farklı alanları ve ifadeleri gösterirler. Olasılık teorisi veri ölçömlerine dayanır ve tecrübe ile test edilirler. Olasılıklar bir olayın olup olmayacağını ölçer. Fuzzy küme teorisi ise bir olayın ne dereceye kadar var olduğunu ve bir olayın olma ağırlığını ölçer. Yani " Yüzde 30 olasılıkla hava serin olacak " önermesi serin hava olasılığını dile getirir. Fakat " Sabah hava yüzde 30 serin geliyor " önermesi hava bir dereceye kadar serin, aynı zamanda değişen derecelerde ılık ve sıcak demektir.

Daha önce yukarıda sözü geçen araştırmacılar konunun uygulamasından çok teorisiyle ilgilendiler. Bu durum fuzzy küme teorisi ve uygulaması arasında bir boşluk oluşmasına neden oldu. Mamdani 1974 yılında matematiksel olarak ifade edilmesi çok zor olan

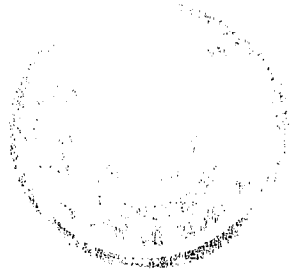


sistemin kontrolunda uzmanın yerini fuzzy lojik kontrolun alabileceğini gösterdi, (Mamdani, 1974). Daha sonra Mamdani konu ile ilgili ilk endüstriyel uygulamayı bir buhar makinası için gerçekleştirdi, (Mamdani and Assilian, 1975). Bu çalışma fuzzy lojik temelli bu tür bir uzman sistemle türbin hızının ve performansının çok başarılı bir şekilde kontrol edilebileceğini göstermiştir. Bu sonuçlar araştırmacıları fuzzy kümeler teorisinin uygulama yönü ile ilgilenmeye teşvik etti.

Kontrol işlemi için uygulama sırasında tasarım yapılırken herşeyden önce sistemin matematiksel modeline ihtiyaç duyulur. Ancak pratikte bu modelleme her zaman mümkün olmayabilir. Bazı durumlarda ise doğru model kurulsa bile bunun uygulamada kullanılması karmaşık problemlere yol açabilir. Bu gibi sorunlarla karşılaşıldığı zaman genellikle kontrol olayını gerçekleştiren uzman kişinin bilgi ve deneyimlerinden yararlanma yoluna gidilir. Uzman kişi sözel değişkenler olarak tanımlanan; uygun, çok uygun değil, yüksek, biraz yüksek, fazla, çok fazla gibi günlük yaşantımızda sıkça kullandığımız kelimeler doğrultusunda esnek bir kontrol mekanizmasını geliştirir. İşte fuzzy küme teorisinin ve fuzzy lojiğin uygulaması olan fuzzy lojik kontrol bu tür mantıksal ilişkiler üzerine kurulmuştur. Ayrıca fuzzy küme teorisinin günümüzde içerdiği diğer uygulama alanları ve tarihsel gelişimi aşağıdaki gibi sıralanabilir.

1. Kontrol işlemi: (Mamdani, 1976; King and Mamdani, 1977; Rutherford and Bloore, 1976; Braae and Rutherford, 1979; Tong, 1977; Kickert, 1976; Sugeno, 1985; Pedrycz, 1989; Huang and Tomozuka, 1990; Liaw and Wang, 1991; Yamakawa, 1993)

2. Karar verme: Karar verme için ilk uygulamalar 1970 yılında başlatıldı, (Bellman and Zadeh, 1970). Daha sonra karar verme geliştirilmeye başlandı, (Yager, 1977; Tong ve Benissone, 1980; Watson et. al.)



3. Görüntü tanıma: (Pal and Majumder 1986, Kandel, 1982, Bezdek, 1981)
- 4.Yapay zeka: (Zadeh, 1978a, 1978b, 1984; Tong and Shapiro, 1985; Whalen and Schott, 1985; Bandler and Khout, 1985; Leung and Lam, 1988)
- 4 .Sınıflama: (Xie, 1984; Miyamoto and Nakayama,1986)

1.1.Fuzzy Lojik Algoritması Kullanan Diğer Deneysel Çalışmalar:

Bir çok makale taraması yapılmıştır. Özellikle bu tezde yapılan çalışma ile diğer çalışmaları karşılaştırabilmek için deneysel olarak gerçel zamanda yapılan uygulamaları içeren makaleler alınarak tablo oluşturulmuştur. Ayrıca çalışmada kullanılan donanım ve malzeme listesi olarak ilgili makalede belirtilenler bu tabloya aktarılmıştır.

Tablo 1.1. Deneysel çalışmalar

Makale Yazarı ve Tarih	Yapılan Çalışma	Uygulamada kullanılan donanım ve malzemeler	İlave açıklamalar
Wu J.C. et. Al. 1996	Fuzzy Kontrol tasarımı için Kayma modu yaklaşımı	Bilgisayarda yazılım kullanılarak simülasyon ve deneysel çalışma yapılmıştır.	Uygulama olarak dönen bir disk üzerinde evirici sarkacın açısı denetlenmiştir.
El-Metwally K.A. et. al. 1996	Enerji iletim güç sistemlerinde kararlılık işlemi	Üzerinde Intel 8051FA 8 Bit mikrodenetleyici, 32KB ROM, 16 KB statik RAM,Klavye ve LCD display bulunan MCC51 geliştirme seti, Intel486 mikroişlemcili bilgisayar, 12 bit ADC ve DAC	Denetleme algoritması makine dili yazılım ile yapılmıştır. Bilgisayar ile haberleşmek için RS232 haberleşme standartı kullanılmış.

Makale Yazarı ve Tarih	Yapılan Çalışma	Uygulamada kullanılan donanım ve malzemeler	İlave açıklamalar
Hesburg T. et. al. 1994	Kullanıcı olmaksızın bir arabanın yoldaki hareketinin denetlenmesi	Texas TMS320C30 DSP	DSP girişine, yol çizgisine göre arabanın durumu belirleyen magnetometreden ve sapma oranı sensöründen bilgi gelmekte , DSP çıkışı direksiyon hidrolik sistemi denetlemektedir.
Shenoi S. et al. 1995	Öğrenme yapısına sahip fuzzy denetleyici Sıcaklık ve ayrıca yükseklik kontrolü üzerinde çalışma yapılmıştır	Intel MCS-51 (8031) mikrodeneleyici, Bilgisayar	Bilgisayar ile haberleşme için RS232 standardı kullanılmıştır. Mikrodeneleyici osilatör frekansı 11.06 MHz 'dir. Elde edilen verilerin saklanması için Statik RAM pil ile desteklenmiştir.
Teeter J. T. et. al. 1996	Fuzzy sürtüşme kompanzasyon yaklaşımı kullanılarak DC motor kontrol sisteminin randımanının artırılması	486 mikroişlemcili Bilgisayar, Grafik çizimler için LabVIEW programı, ADC ve DAC (Bit sayısı belirtilmemiş.)	DC motor hız kontrol işlemine yeni bir yöntem uygulanmıştır.
So W.C. et. al. 1996	DC/DC konvertör denetleme işlemi	Üzerinde Texas TMS 320C50 DSP bulunan geliştirme seti, AD668 12 bit DAC, AD1671 12 bit ADC ve bilgisayar	Denetleme işlemi DSP yazılımı ile gerçekleştirilmiştir. Simülasyon ve deneysel çalışma yapılmıştır.

Makale Yazarı ve Tarih	Yapılan Çalışma	Uygulamada kullanılan donanım ve malzemeler	İlave açıklamalar
Lai M. F. et al. 1996	İndüksiyon Motor Hız Kontrolunda faz kilitlemeli çevrimin ve Fuzzy denetleyicinin beraber kullanılması	486 mikroşlemcili Bilgisayar, 12 bit ADC, 12 bit sayıcı, tutma ikilisi	Fuzzy denetleyici algoritması C++ programlama dilinde gerçekleştirilmiştir.
Kung Y. S. et al. 1994	Motor hız kontrolunda Fuzzy denetleyicinin geliştirilmesi için Lineer Model izleyen denetleyici tasarımı	80386/387 Mikroşlemcili PC bilgisayar, ADC ve DAC (Bit sayısı belirtilmemiş.)	—
Silva C.W. et. al. 1994	Fuzzy lojik denetleme sisteminde veri tabanını ayırma tekniğini kullanarak DC servo motorun kontrolu	486 Mikroşlemcili Bilgisayar, ICB930 arabirim kartı, DMC-400 digital denetleyici	—
Beom R. H. et. al. 1995	Fuzzy Lojik ve geliştirilmiş öğrenme işlemini kullanan robotun yol üzerindeki hareketinin denetlenmesi	Bilgisayar, Ultrasonik sensörler, 8031 mikrodnetleyicisi	Sensörler ise uzaklık ve yön tayini yapmak için , Mikrodnetleyici sensörlerden gelen bilgiyi bilgisayara seri yoldan göndermek için kullanılmıştır.



Makale Yazarı ve Tarih	Yapılan Çalışma	Uygulamada kullanılan donanım ve malzemeler	İlave açıklamalar
Corbet T. et. al. 1996	Endüstriyel hidrolik Robot Kontrolü	Bilgisayar, Unimate MK-II hidrolik robot ve DAC, ADC (Bit sayısı belirtilmemiş.)	—
Huang T.C. et. al. 1996	Çok katmanlı Fuzzy lojik denetleyici kullanarak indüksiyon motorunun yüksek randımanlı hız ve pozisyon kontrolü	Bilgisayar ve Mikrodenetleyici. (Mikrodenetleyicinin adı belirtilmemiş.)	İndüksiyon motorundan alınan bilgilerin bilgisayara seri yoldan gönderilmesi için mikrodenetleyici kullanılmış.
Wu Z. Q. et. al. 1990	Sıvı yakıtlı tavlama fırının denetlenmesi	Apple II bilgisayar	Bilgisayar içerisinde Mikroişlemci kartı, Monitör, Disk sürücü, DAC, ADC ve ayrıca yazıcı, bulunmaktadır.
Maeda M. et. al. 1991	Robotun belirli bir yol üzerindeki hareketinin kontrolü	Robot, CCD kamera, TV monitör, Bilgisayar	Programlar BASIC ve makine dilinde yazılmıştır.
Hara K. et. al. 1991	Perçin yerleştirme işlemi	NEC PC-9801VM2 Bilgisayar, HZ NM-6620S robot kolu	Robot ile bilgisayar haberleşmesinde RS232 standardı kullanılmış.

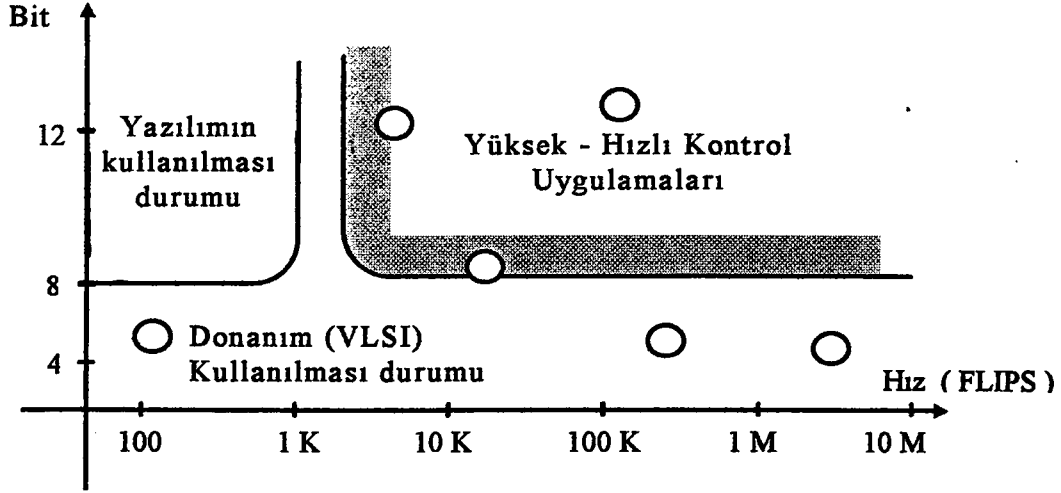
Makale Yazarı ve Tarih	Yapılan Çalışma	Uygulamada kullanılan donanım ve malzemeler	İlave açıklamalar
Liaw C. M. et. al. 1995	Serbest iki dereceli fuzzy denetleyici ile İndüksiyon motor hız kontrolü	80386/387 mikroişlemcili bilgisayar	—
Sousa G.C.D. et. al. 1995	İndirekt Vektör kontrollu İndüksiyon motorunun hız kontrolunda fuzzy lojik tabanlı gelişmiş denetleyici tasarımı	Texas TMS320C25 DSP kontrol kartı	Temel vektör kontrol çevrim örnekleme zamanı 200 μ S 'dir.
Lu Y. S. et. al. 1994	Tek eksenli magnetik süspansiyon sistemi için Fuzzy kayma mod denetleyicisi	DASH 16F Bilgisayar	Bilgisayar içerisinde 12 bit ardışıl yaklaşımlı, 8 μ S çevirim zamanlı. ADC, 12 Bit DAC mevcuttur. Uygulamada magnetik devre vasıtasıyla çeşitli ağırlıktaki metal kütleler aşağıda belirli bir mesafede tutulmuştur.
Yen Y.J. et. al. 1993	Sabit disk üzerindeki izlerin takip edilmesi	Zentek 3100 sabit diski, Texas TMS320C30 DSP kartı	Sabit disk 120 Mbayt kapasitesindedir. DSP kartı üzerinde 4096x32 bit ROM, 2048x32 bit RAM bulunmaktadır. Saat darbesi frekansı 33.3 MHz 'dir.

Makale Yazarı ve Tarih	Yapılan Çalışma	Uygulamada kullanılan donanım ve malzemeler	İlave açıklamalar
Hilloowala R.M. et. al. 1996	Rüzgar gücü ile çalışan elektrik üreten sistemin denetlenmesi	Bilgisayar, içerisinde DAC ve ADC bulunan DT2821 kartı, rüzgar tribünü, İndüksiyon jeneratörü, köprü tipi doğrultucu, PWM inverter, transformator ve yük	—
Guillem P. 1996	DC motor hız kontrolü	SGS-Thomson ST6 (ST6265-8 bit) mikrodnetleyici, FuzzyTECH yazılım programı, 400 W gücünde DC motor, Analog takometre	Mikrodnetleyicide PWM ,zamanlayıcı, ADC ve portlar mevcuttur. Kuralları, sözel değişkenleri ve üyelik fonksiyonlarını belirleyip derlemek için FuzzyTECH programı kullanılmış, Motor devri ST6 tarafından okunup 9600 Baud hızında bilgisayara gönderilmiştir. Motorun yüklü ve yüksüz durumları için devir eğrileri çizdirilmiştir.
Cheng F.F. et. al. 1993	A.C. motor servo sistemin hız kontrolü	Intel 80196 mikrodnetleyicisi, Sürücü devre için M57918LS, inverter için FM50 DY-10 MOSFETS, Stator akımı için LT100p Hall effect sensörü, 2 Hp'lik İndüksiyon motoru	80196 Mikrodnetleyicide 8 kanal 10 bit ADC, PWM çıkışı, , yüksek hızlı Giriş/Çıkış portları bulunduğdan bu denetleme işlemi için çok uygun olduğu belirtilmiştir.

Makale Yazarı ve Tarih	Yapılan Çalışma	Uygulamada kullanılan donanım ve malzemeler	İlave açıklamalar
Stenz R. et. al. 1995	Kimya endüstrisinde damıtma işleminin denetlenmesinde	Bilgisayar, ADC ve DAC (Bit sayısı belirtilmemiş), Shell Fuzzy Tech yazılım programı	Shell Fuzzy Tech programında sözel değişkenler kullanıcı tarafından belirlenir ve üyelik fonksiyonları grafiksel olarak tanımlanır. Bu programda üyelik fonksiyonlarını, fuzzy lojik işlemcilerini ve netleştirme işlemlerini kullanarak gerekli verilerin elde edilmesi için hazır bir C programı mevcuttur.
Surman H. et. al. 1996	Nikel-Kadmiyum pillerinin sarj işlemi için Fuzzy denetleyici kullanılması	SGS-Thomson ST6 mikrodenetleyicisi	Yüksek güce sahip NiCd pili 6A akım ile yaklaşık 10 Dakikada sarj edilmiştir.
Reay D.S. et. al. 1995	Fuzzy adaptif sistem kullanılarak anahtarlama relüktans motorun hızının denetlenmesi	Texas TMS320C25 DSP	Bu çalışmada motorun düşük ve orta hızlarında DSP ile denetleme işleminin iyi sonuç verdiği belirtilmiştir. Fakat yüksek hızlarda daha iyi sonuç alabilmek için yüksek hızlı mikrodenetleyici kullanılması önerilmiştir.

Makale Yazarı ve Tarih	Yapılan Çalışma	Uygulamada kullanılan donanım ve malzemeler	İlave açıklamalar
Suyitno A. et. al. 1993	Bir servomotorun kontrolu için Fuzzy lojik kullanılarak değişken yapılı güçlendirilmiş denetleyici	Texas TMS320C14E DSP, 8086 mikroişlemcili bilgisayar, ADC, DAC, 7.5 kW, 200 V, 600 Hz, 4p ve 1800 d / dak İndüksiyon motoru, yük olarak 15kW'lık DC makine	Algoritma için bilgisayarda yazılan program seri yoldan DSP kartına gönderilmiş ve DSP üzerinde çalıştırılmıştır. Örneklem zamanı 100 μ S 'dir.

Tablodaki uygulamalarda denetleme işlemi mikrodenetleyici, dijital sinyal işlemcisi veya bilgisayar kullanılarak gerçekleştirilmiştir. Fuzzy lojik denetleme işlemi için donanım olarak fuzzy işlemcisi veya fuzzy çıkarım kartı kullanılmamıştır. Denetleme algoritması yazılım ile sağlanmıştır. Yazılım ile yapılan fuzzy lojik denetleme işlemi donanım kullanılarak yapılan denetleme işlemine göre yavaş kalmaktadır (Kazuo et. al. 1993). Yüksek hızlı kontrol uygulamasında çıkarım hızının fazla olması gerekmektedir. Bu durum Şekil.1 'de görülmektedir. Yazılım kullanılması durumunda hassasiyet 8 bit veya daha yukarısında çıkarım hızı en fazla 1 KFLIPS olmaktadır. Fakat donanım kullanıldığında ve hassasiyet 8 bit veya daha yukarısı seçildiğinde çıkarım hızı artmakta aynı zamanda yüksek hızlı kontrol işlemleri bölgesine gelinmektedir. Bu bölgede çıkarım hızı 10 KFLIPS ve 10 MFLIPS yukarıdır. Küçük daireler ile gösterilen noktalar çeşitli fuzzy işlemcilerin çıkarım hızını ve hassasiyetlerini gösterir. Tezde kullanılan fuzzy işlemci 12 bit hassasiyetindedir.



Şekil 1

Yukarıdaki tabloda motor hız uygulamalarında yalnız bir motor alınmış denetleme yapılmıştır. Gücü fazla olan motorlar tercih edilmiştir. Fakat bu tezde kullanılan motorların gücü oldukça küçüktür. Böyle olmasının nedeni laboratuvar imkanlarının kısıtlı olmasından dolayıdır. Sonuçta yine hız kontrolü yapılmıştır. Ayrıca tek motorun hızı yerine ayrı iki motor hızı değerlendirilmiş ve denetleme işlemi buna göre yapılmıştır.

Bu sistemde diğer fuzzy denetleme çalışmalarının deneysel yapılabilmesi mümkündür. Yapılacak olan çalışmalarda sistem matematiksel olarak modellendikten sonra bilgisayarda simulasyon çalışması yapılabilir. Bu simulasyon çalışmadan sonra bu sistem üzerinde deneysel olarak uygulamak mümkündür. Sistemde yeni kartlara ihtiyaç duyulduğunda sisteme ilave edebilmesi mümkündür. Çalışmada gerçel zamanda elde edilen verilerin bilgisayara seri yoldan gönderilip grafiksel olarak eğrilerin çizdirilip incelenebilir.

1.2.Fuzzy İşlemci Tasarımı İçin Yapılan makale Çalışmaları ve İşlemcilerin Özellikleri:

Fuzzy lojik kontrol ile ilgili ilk uygulamalar bilgisayar üzerinde paket yazılım programaları kullanılarak gerçekleştirildi. Fakat bu durum kontrol olayının istenildiği kadar hızlı olmamasına ve yazılımın uzunluğunun artmasına neden oluyordu. Daha sonra fuzzy lojik kontrol işlemini donanım kullanarak gerçekleştiren fuzzy işlemciler geliştirilmeye başlandı. Böylece kontrol işlemi için yazılan yazılımların uzunlukları daha da azaldı. Bu işlemciler kontrol kurallarını saklayacak ve işleyecek biçimde tasarlanmışlardır. 1985 yılında Masaki Togai ve Hiroyaki Watanabe, AT&T Bell laboratuvarında ilk sayısal fuzzy işlemciyi gerçekleştirdiler. Bu işlemci 12.5 μ s 'de 16 basit kuralı işleme kapasitesine sahipti. Daha sonra Togai, InfraLogic Inc., fuzzy hesap hızlandırıcısı donanımıyla saniyede iki milyon kuralı işleyen işlemciler üretmiştir. Mikroişlemci firmalarının çoğu fuzzy işlemci üzerinde araştırma projeleri başlatmıştır. Sonra Yamakawa bu konu üzerindeki çalışmaları geliştirmiştir,(Yamakawa,1988). Ayrıca OMRON, TEXAS ve SGS-THOMSON firmaları yeni fuzzy işlemciler üretmiştir.

Aşağıdaki tabloda ilgili makalede çalışması yapılan fuzzy işlemcilerin teknik özellikleri bu makalede verilen bilgiler kapasitesinde belirlenmiştir.

Tablo 1.2. Fuzzy işlemci tasarımı için yapılan makale çalışmaları

Makale Yazarı ve tarihi	Fuzzy İşlemcinin makalede belirtilen teknik özellik ve kapasitesi
Huertas J. L. et. al. 1996	Giriş: 2 Çıkış: 1 Hızı: 2 MFLIPS Türü: Analog

Makale Yazarı ve tarihi	Fuzzy İşlemcinin makalede belirtilen teknik özellik ve kapasitesi
Eichfeld H. et. al. 1996.	İşlemci adı: SAE81C991 Evrensel Kümedeki Giriş Giriş: 4096 Üyelik fonk.: sayısı: 15360 Çıkış: 1024 Evrensel Kümedeki Çıkış Çalışma modu: 10 Üyelik fonk. sayısı: 1024 Kural sayısı: 131 072 Üyelik fonk. şekil sayısı: Osilatör frekansı: 25 MHz Keyfi, sınırsız Hızı: 10 Milyon Kural /sn Netleştirme yöntemi: 4 Hassasiyet: 12 Bit Bacak sayısı: 68 pin İç ROM: 144x12b(MASK) Türü: Dijital İç RAM: 4 Kx16 Bit Firma: Siemens AG
Eichfeld H. et. al. 1995.	İşlemcinin adı: SAE81C99 Hassasiyet: 8 Bit Giriş: 256 İç ROM: 64 Kbit Çıkış: 64 İç RAM: 256x6 Bit Çalışma modu: 4 Türü: Dijital Kural sayısı: 16 384 Firma: Siemens AG Osilatör frekansı: 20 MHz Hızı: 10 Milyon Kural /sn
Guo, S., et. al. 1996.	Giriş: 3 Çıkış: 1 Kural sayısı: 13 Hızı: 6 M FLIPS Türü: Analog

Makale Yazarı ve tarihi	Fuzzy İşlemcinin makalede belirtilen teknik özellik ve kapasitesi
SGS - Thomson Information Sheet 1996	Giriş: 3 Çıkış: 3 Kural sayısı: 16 Evrensel kümede için üyelik fonk. sayısı: 15 Giriş için üyelik fonk. şekil sayısı: 2 (Üçgen ve yamuk) Çıkış için üyelik fonk. grafiği: Tek nokta Bacak sayısı: 28 Türü: Analog Firma: SGS - Thomson
Ascia G. et. al. 1996	Giriş: 8 Çıkış: 8 Kural sayısı: 64 Hızı: 468 KFLIPS Osilatör frekansı: 60 MHz Türü: Analog
Costa A. et. al. 1996	Giriş: 5 Çıkış: 1 Kural sayısı: 75 Osilatör frekansı: 75 MHz Evrensel kümedeki üyelik fonk. sayısı: 5 Türü: Dijital

Makale Yazarı ve tarihi	Fuzzy İşlemcinin makalede belirtilen teknik özellik ve kapasitesi
Russo G. et. al. 1996	Giriş: 6 Çıkış: 2 Kural sayısı: 15 Osilatör frekansı: 40 MHz Hassasiyet: 5 bit Türü: Dijital
Huertas J. L. et. al. 1996	Giriş: 2 Çıkış: 1 Hızı: 2 MFLIPS Türü: Analog
Fattaruso J. W. et.al. 1994	Giriş: 8 Çıkış: 4 Kural sayısı: 32 Türü: Analog - Dijital
Manaresi N. et. al. 1996	Giriş: 2 Çıkış: 2 Kural sayısı: 25 Türü: Analog n giriş, m çıkış ve r kural bulunan genel bir yapı üzerinde çalışılmıştır.

Makale Yazarı ve tarihi	Fuzzy İşlemcinin makalede belirtilen teknik özellik ve kapasitesi
NeuraLogix, Information sheet 1992	İşlemci adı: NLX230 Giriş: 8 Çıkış: 8 Kural sayısı: 64 Hızı: 30 Milyon Kural/sn Türü: Analog Firma: American Neura Logic Inc.
Lee S. et al. 1994 (Makalenin giriş bölümünde bu işlemcinin özellikleri kısaca anlatılmıştır.)	İşlemci adı: FP5000 Hızı: 500 MFLIPS Türü: Dijital Firma: OMRON
SGS-Thomson Advanced Data , 1994	İşlemci adı: W.A.R.P. 1.1 Giriş: 4 Çıkış: 1 Giriş hassasiyeti: 7 bit Çıkış hassasiyeti: 10 bit Kural sayısı: 256 Osilatör frekansı: 40 MHz Bacak sayısı: 64 Türü: Dijital Firma: SGS-Thomson

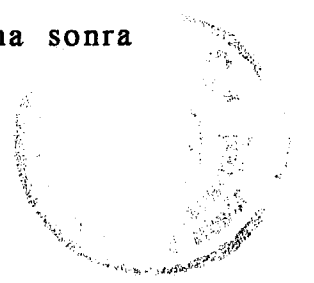
Makale Yazarı ve tarihi	Fuzzy İşlemcinin makalede belirtilen teknik özellik ve kapasitesi
Hollstein, T. et. al. 1996	İşlemci adı:XC4006 Giriş: 128 Çıkış: 4 Kural sayısı: 256 Evrensel kümede giriş için üyelik fonk. sayısı: 256 Hızı: 1.25 Milyon kural/sn Türü: Analog, Dijital
Ascia, et. al. 1996.	Giriş : 4 Çıkış: 1 Evrensel kümede kullanılabilecek sözel değişkenler: 64 Üyelik Fonksiyonu sayısı: 8 Osilatör Frekansı: 50 MHz Hızı: 50 Milyon Kural/sn Türü: Analog
Nakamura K. et. al. 1993	Giriş: 2 Çıkış: 1 Hassasiyet: Giriş 12 bit , Çıkış 16 Bit Üyelik Fonksiyonu Sayısı: 30 000 Hızı: 200 KFLIPS Kural sayısı: 15 000 'den fazla Osilatör Frekansı: 20 MHz Bacak Sayısı: 80 Türü: Dijital Firma: Mutsubishi Electric Corporation

Tezde kullanılan FP-3000, fuzzy işlemciler arasında teknik özellikler bakımından üst sıralarda yer almaktadır. Fakat Siemens firmasının SAE81C991 ve FP3000 'nün bir sonraki sürümü olan FP-5000 en başta gelen işlemcilerdir. Tabloda bulunan makalelerde sadece fuzzy işlemcinin tasarımı için çalışılmış mikrodenetleyicili bir uygulama yapılmamıştır.

Fuzzy işlemciler mikrodenetleyiciler ile beraber kullanıldığı zaman fuzzy lojik kontrol işleminin daha kolaylıkla gerçekleştirilmesi ve gerçekleştirme maliyetinin düşmesine neden olmuştur. Fuzzy işlemci ile beraber çalışacak 16 bitlik mikrodenetleyici 8XC196KC/KD seçilmiştir. İşlemci 12 bitlik olduğu ve işlemleri daha kolaylıkla hızlı yapabilmek için böyle bir tercih yapılmıştır. Ayrıca bu denetleyicinin içerisinde zamanlayıcı, dijital analog çevirici gibi çeşitli üniteler bulunmasından dolayı endüstriyel uygulamalarda ihtiyaçlara cevap verebilecek kapasitededir.. Fuzzy lojik ile kontrol edilen sistemlerde giriş bilgilerini okuyabilmek için algılayıcılara ihtiyaç vardır. Kullanılan denetleyici algılayıcılardan gelen bilgiyi almaya elverişli yapıdadır.

Bu çalışmada önce fuzzy küme teorisi anlatılmıştır. Temel işlemler hakkında bilgi verilir. Ayrıca fuzzy lojik kontrol olayı için ana konularından biri olan çıkarım teknikleri üzerinde durulur ve bir uygulama üzerinde çıkarım işlemi grafiksel ve cebirsel olarak anlatılır. Çıkarım işlemi sonucundaki ifadeden bir sonuç değeri elde edebilmek için netleştirme yöntemleri verilir. Bu bölümde son olarak fuzzy lojik kontrol sisteminin tasarımı için gerekli bilgiler anlatılır.

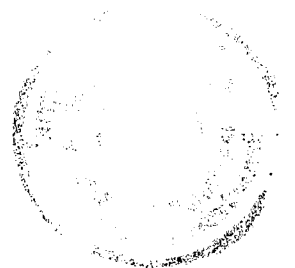
Sonraki bölümde mikrodenetleyici tanıtılır. Denetleyicinin adresleme türleri, özel bellek adresleri, kaydedicileri ve pencereleme teknikleri incelenir. Ayrıca adres ve veri yolunu kullanma şekilleri şemaları ile birlikte açıklanır. Daha sonra mikrodenetleyicinin donanım şeması tasarlanır ve gerçekleştirilir. Bu devre üzerinde program yazabilmek için gösterge ve tuş takımı ilave edilmiştir. Daha sonra



makina dili programların kodlarını sistemin yazılır silinir belleğe yazabilmek monitor programı yazılmış ve program akış diyagramı verilmiştir. Seri port şartlanırken kullanılan kaydedicilerin fonksiyonları incelenir. Bilgisayarda yazılan bir programın seri porttan denetleyiciye gönderilebilmesi için mikrodnetleyicide bulunan seri port ünitesi hakkında bilgi verilir. Bilgisayarda yazılan makine dili programın mikrodnetleyicili sisteme gönderen programın akış diyagramı anlatılır. İleride fuzzy lojik kontrol uygulamasında kullanılacak olan zamanlayıcılar ve zamanlayıcılara ilişkin kaydediciler incelenir.

Devreye ilave edilecek fuzzy işlemci kartının en önemli elemanı olan fuzzy işlemcinin özellikleri incelenir. İşlemcinin bacaklarının işlevleri ve özel adres bölgeleri tanıtılır. Mikrodnetleyiciye bağlanmak üzere fuzzy işlemcili ünitenin yapısı anlatılır. Bilgisayarda kontrol edilecek sisteme ait kurallar ve üyelik fonksiyolarının derlenmesinde oluşan bilgi tabanı verilerinin mikrodnetleyicili sisteme gönderilmesi ile ilgili programın akış diyagramı açıklanır.

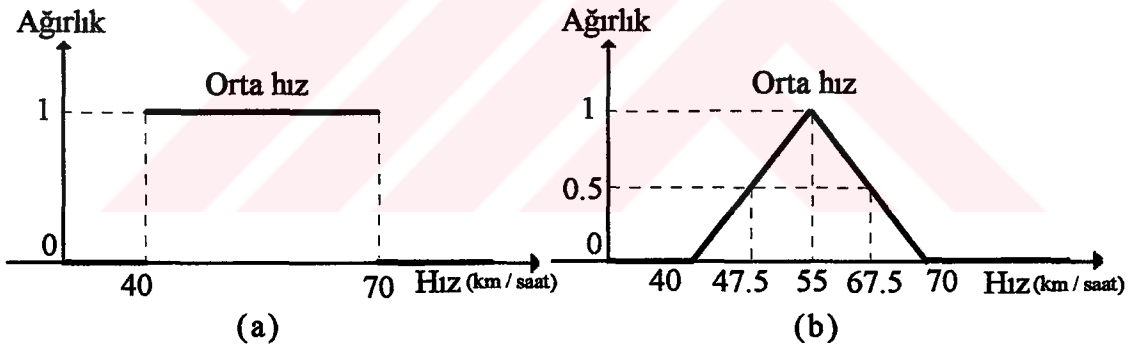
Son olarak dijital analog çevirici kullanılarak iki motorlu bir sistem üzerinde fuzzy lojik uygulama programının akış diyagramı anlatılır.



2. FUZZY KÜME TEORİSİ

2.1.Fuzzy küme kavramı

Bilindiği gibi klasik (keskin) mantıkta bir olayın olabilirlik derecesi 1 (var) ve 0 (yok) olmak üzere iki çeşittir. Dolayısıyla klasik mantıkta olayın olabilirlik derecesi 0 ile 1 arasında herhangi bir değer alamaz. Olay olarak hız düşünülecek olursa, 40 km/saat ve 70 km/saat arası orta hız olarak kabul edilirse bu durumda şekil 2.1.a 'da gösterildiği gibi klasik mantıkta 40 km/saat ve 70 km/saat dahil olmak üzere bu iki değer arasındaki her hız değeri orta hız olarak kabul edilir. Yani bu değerlerin olabilirliği 1 olur. Sınır hız değerleri arasında yorum yapmak mümkün değildir. Bu durumu daha olumlu hale getirmek için Fuzzy Mantık kullanılır.



Şekil 2.1 Klasik ve Fuzzy mantıkta hız grafiği

Fuzzy mantıkta sınır ve ara değerler için yalnız bir durum yoktur. Olabilirlik derecesi $[0,1]$ arasında çeşitli değerler alabilir. Çok seviyeli mantığın bir çeşidi olarak görülebilir. Şekil 2.b 'de gösterildiği gibi 40-70 km/saat arası yine orta hız olarak kabul edilirse bu iki değer arasındaki hızların olabilirlik dereceleri değişik değerlere sahiptir. Bu durumda 40 km/saat ve 70 km/saat hızlarının olabilirlik derecesi 0 olur. Yani orta hız sayılmazlar. 55 km/saat hızının olabilirlik derecesi 1 'dir. Yani bu değer tam orta hız sayılır. Dolayısıyla bir

değerden diğer bir değere daha yumuşak keskin olmayan geçiş sağlanmış olur. Ayrıca 47.5 km/saat ve 67.5 km/saat hızlarının olabilirlik derecesi 0.5 'tir. Bu hız değerleri noktalarına geçiş noktası adı verilir. Fuzzy mantıkta Şekil 2.1.b 'de gösterilen eğriye üyelik fonksiyonu denir. Hız ekseninde başka hız grupları için üyelik fonksiyonları gösterilebilir. (çok yavaş ,yavaş, orta, hızlı, çok hızlı gibi.) Hız eksenindeki tüm hız değerlerinin bulunduğu küme hız için evrensel küme olarak adlandırılır. Her hızın üyelik fonksiyonunda aldığı olabilirlik derecesi, üyelik ağırlığı olarak isimlendirilir. Ayrıca orta hız üyelik fonksiyonu evrensel kümenin her elemanı ve bu elemana karşılık gelen üyelik ağırlığı ile matematiksel olarak gösterilebilir. Yine orta hız üyelik fonksiyonu hız evrensel kümesinin bir fuzzy alt kümesidir.

2.2. Fuzzy kümelerde teorik işlemler

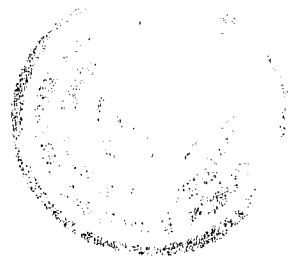
Fuzzy kümelerin teorik işlemler oldukça fazladır. Fakat bu çalışmada sadece belirli bir kısım teorik işlemler anlatılacaktır, (Ross, 1995).

2.2.1. Fuzzy kümenin matematiksel gösterilimi

Bir X evrensel kümesindeki bir A fuzzy küme $[0,1]$ aralığında değer alan μ_A üyelik fonksiyonu tarafından karakterize edilir. Bu yüzden X 'deki A fuzzy küme bir x ve x 'in şekil 2.2'de gösterilen üyelik fonksiyonu tarafından belirlenen ağırlığı ile beraber tarif edilir. Yani;

$$A = \{(x, \mu_A(x)) | x \in X\} \quad (2.1)$$

olur.



2.2.3. Kesişme özelliği

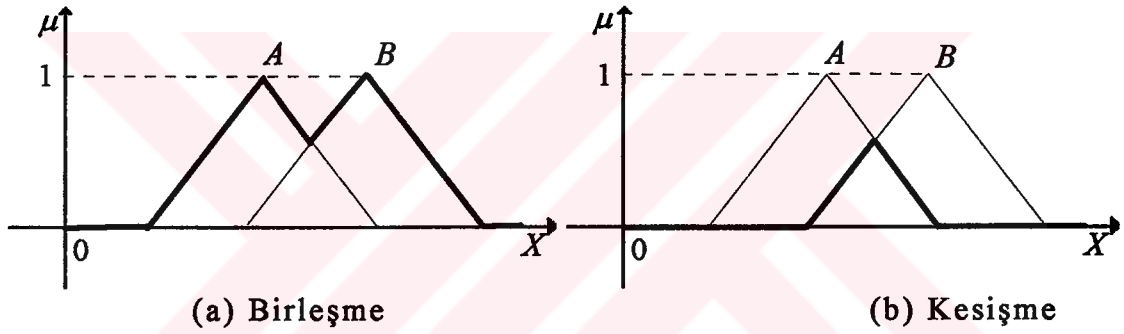
$A \cap B$ kesişimin üyelik fonksiyonu $\mu_{A \cap B}(x)$ üyelik fonksiyonu, $x \in X$ 'in her değeri için ;

$$\mu_{A \cap B}(x) = \mu_A(x) \wedge \mu_B(x) \quad (2.6)$$

veya;

$$\mu_{A \cap B}(x) = \min \{ \mu_A(x), \mu_B(x) \} \quad (2.7)$$

şeklinde gösterilebilir. Kesişme özelliği şekil 2.3.b 'deki gibi grafiksel olarak gösterilebilir.



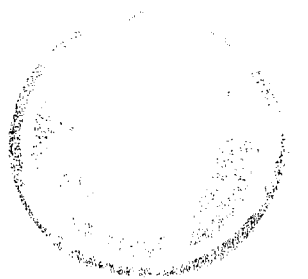
Şekil 2.3. Birleşme ve kesişme özelliği

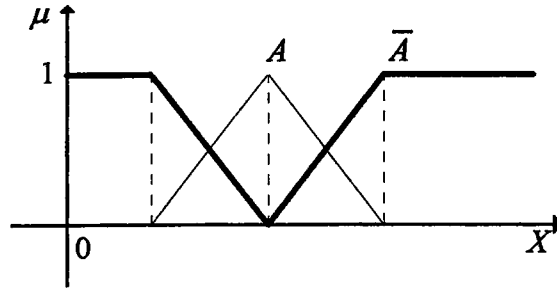
2.2.4. Evrik alma özelliği

Bulanık A kümesinin evriğini $x \in X$ 'in her değeri için $\mu_A(x)$ üyelik fonksiyonu türünden aşağıdaki gibi gösterilir.

$$\mu_{\bar{A}}(x) = 1 - \mu_A(x) \quad (2.8)$$

Evrik alma özelliği şekil 2.4 'deki gibi grafiksel olarak gösterilebilir.





Şekil 2.4. Evrik alma özelliği

2.2.5. Üs alma özelliği

Fuzzy A kümesinin $x \in X$ 'in her değeri için α . dereceden üssü;

$$A^\alpha = \sum_i \frac{[\mu_A(x_i)]^\alpha}{x_i} \quad (2.9)$$

şeklinde ifade edilir.

2.2.6. Pozitif α sayısı ile çarpma özelliği

Fuzzy A kümesinin $x \in X$ 'in her değeri için pozitif α sayısı ile çarpma;

$$\alpha.A = \sum_i \frac{\alpha.\mu_A(x_i)}{x_i} \quad (2.10)$$

şeklinde gösterilir.

2.3. Fuzzy Kümeler Arasındaki Mesafe

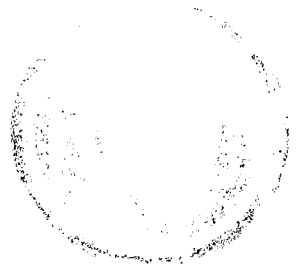
İki fuzzy küme arasındaki mesafe iki fuzzy kümenin eşit veya farklı olup olmadığını belirten bir işlemdir.

2. A ve B kümeleri arasında normalize edilmiş lineer mesafe;

$$d_{nl}(A,B) = \left(\frac{1}{n}\right) \sum_{i=1}^n |\mu_A(x_i) - \mu_B(x_i)| \quad (2.11)$$

şeklinde tanımlanır.

2. A ve B kümeleri arasında normalize edilmiş ikinci dereceden mesafe;



$$d_m(A,B) = \left(\frac{1}{n}\right) \left[\sum_{i=1}^n \left| \mu_A(x_i) - \mu_B(x_i) \right|^2 \right]^{0.5} \quad (2.12)$$

şeklinde gösterilir.

2.4. Klasik ve Fuzzy bağıntılar

2.4.1. Klasik bağıntılar

2.4.1.1. Kartezyen çarpım

X ve Y belirli elemana sahip iki evrensel küme ise bunların kartezyen çarpımı;

$$X \times Y = \{(x,y) \mid x \in X, y \in Y\} \quad (2.13)$$

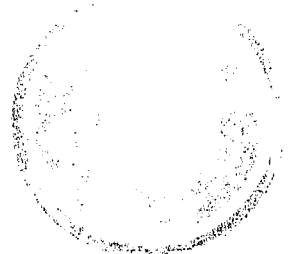
şeklindeir. Kartezyen çarpım X kümesindeki her elemana karşılık Y kümesindeki her eleman ile oluşturulan ikilileri kapsar. Göz önüne alınan herhangi bir eleman $X \times Y$ kartezyen çarpımındaki kümeye ait ise ağırlığı 1, aksi takdirde ağırlığı 0 olarak gösterilir. Yani;

$$\chi_{X \times Y}(x,y) = \begin{cases} 1, (x,y) \in X \times Y \\ 0, (x,y) \notin X \times Y \end{cases} \quad (2.14)$$

şeklinde ifade edilir. $X \times Y$ kartezyen çarpımı küme bağıntı matrisi olarak gösterilebilir. $X = \{1, 2, 3\}$ ve $Y = \{a, b, c\}$ şeklinde kümeler ise $X \times Y$ kartezyen çarpımında oluşan bağıntı kümesi ve bağıntı matrisi aşağıdaki gibi olur.

$$X \times Y = \{(1,a), (1,b), (1,c), (2,a), (2,b), (2,c), (3,a), (3,b), (3,c)\} \quad (2.15)$$

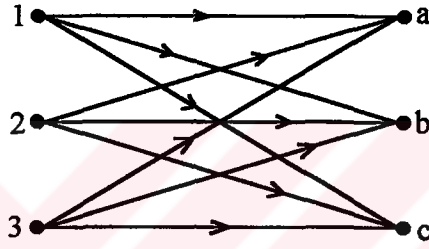
$$R = \begin{matrix} & \begin{matrix} a & b & c \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \end{matrix} & \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \end{matrix} \quad (2.16)$$



Bağıntı matrisi şekil 2.4. 'deki gibi ağ yapısı şeklinde gösterilebilir. Eğer sadece $X \times Y$ kartezyen çarpımında oluşan kümenin alt kümesi olan bir bağıntıyı sadece üyelik ağırlıkları ile ifade edersek;

$$\chi_{X \times Y}(x, y) = \begin{cases} 1, (x, y) \in R \\ 0, (x, y) \notin R \end{cases} \quad (2.17)$$

şeklinde gösterilir ve bağıntı $X \times Y$ kartezyen çarpımında oluşan kümenin alt kümesi olduğu için $R \subset X \times Y$ ifadesi yazılabilir.



Şekil 2.4 X ve Y kümeleri için bağıntı ağı

2.4.1.2. Klasik bağıntılarda işlemler

$X \times Y$ kartezyen çarpımın alt kümeleri olan R ve S keskin bağıntıları için aşağıdaki teorik işlemler verilebilir.

$$\begin{aligned} \text{Birleşme: } R \cup S &\rightarrow \chi_{R \cup S}(x, y): \chi_{R \cup S}(x, y) = \max[\chi_R(x, y), \chi_S(x, y)] \\ \text{Kesişim: } R \cap S &\rightarrow \chi_{R \cap S}(x, y): \chi_{R \cap S}(x, y) = \min[\chi_R(x, y), \chi_S(x, y)] \\ \text{Evrik alma: } R &\rightarrow \chi_{\bar{R}}(x, y): \chi_{\bar{R}}(x, y) = 1 - \chi_R(x, y) \end{aligned} \quad (2.18)$$

$$\text{Kapsama: } R \subset S \rightarrow \chi_R(x, y): \chi_R(x, y) \leq \chi_S(x, y)$$

2.4.1.3. Klasik bağıntılarda kompozisyon

$X \times Y$ kartezyen çarpımın bir alt kümesi R bağıntısı ve $Y \times Z$ kartezyen çarpımın bir alt kümesi S bağıntısı olsun. R bağıntısında S bağıntısından farklı olarak X evrensel kümesi ve S bağıntısında R bağıntısından farklı olarak Z evrensel kümesi vardır. Eğer bu bağıntılardan yararlanarak $X \times Z$ kartezyen çarpımındaki T bağıntısının

hesaplanması istenilirse kompozisyon işlemi kullanılır. Kompozisyon max-min ve max-çarpma kompozisyonu olarak iki değişik şekilde gösterilir.

$$T = R \circ S \quad (2.19)$$

Burada o kompozisyon işlemcisini gösterir. Kompozisyon işlemi sonucunda oluşan T bağıntısının ağırlıkları aşağıdaki ifadeler ile hesaplanır.

Max-min kompozisyonu ;

$$\chi_T(x, z) = \bigvee_{y \in Y} \chi_R(x, y) \wedge \chi_S(y, z) \quad (2.20)$$

veya ;

$$\chi_T(x, z) = \max_{y \in Y} \left\{ \min \left[\chi_R(x, y), \chi_S(y, z) \right] \right\} \quad (2.21)$$

Max-Çarpım kompozisyonu ;

$$\chi_T(x, z) = \bigvee_{y \in Y} \chi_R(x, y) \bullet \chi_S(y, z) \quad (2.22)$$

veya ;

$$\chi_T(x, z) = \max_{y \in Y} \left\{ \chi_R(x, y) \bullet \chi_S(y, z) \right\} \quad (2.23)$$

Burada $\bullet$ normal çarpma işlemi göstermektedir.

2.4.2. Fuzzy bağıntı

Bu bağıntılar ve işlemler klasik bağıntılara benzerdir. Bağıntılarda ağırlıklar $[0,1]$ arasında değişir ve $\mu_R(x, y)$ üyelik fonksiyonu ile ifade edilirler.

2.4.2.1. Kartezyen çarpım

A fuzzy kümesi X evrensel kümesinin, B fuzzy kümesi Y evrensel kümesinin alt kümeleri ise $A \times B$ ile gösterilen A ile B 'nin kartezyen çarpımı R bağıntısı olarak ifade edilir .

$$A \times B = R \subset X \times Y \quad (2.24)$$

ve R bulanık bağıntısının üyelik fonksiyonu;

$$\mu_R(x, y) = \mu_{A \times B}(x, y) = \min\{\mu_A(x), \mu_B(y)\} \quad (2.25)$$

şeklinde gösterilir.

Fuzzy bağıntılar arasındaki özellikler klasik bağıntılardaki gibidir.

2.4.2.2. Fuzzy bağıntılarda kompozisyon

R , $X \times Y$ kartezyen çarpımının bir fuzzy bağıntısı ve S , $Y \times Z$ kartezyen çarpımının bir fuzzy bağıntısı ise $X \times Z$ kartezyen çarpımının T fuzzy bağıntısı klasik bağıntılarda anlatıldığı gibi gösterilebilir .

$$T = R \circ S \quad (2.26)$$

Max-min kompozisyonu ;

$$\mu_T(x, z) = \bigvee_{y \in Y} \mu_R(x, y) \wedge \mu_S(y, z) \quad (2.27)$$

veya;

$$\mu_T(x, z) = \max_{y \in Y} \left\{ \min[\mu_R(x, y), \mu_S(y, z)] \right\} \quad (2.28)$$

Max-Çarpım kompozisyonu ;

$$\mu_T(x, z) = \bigvee_{y \in Y} \mu_R(x, y) \cdot \mu_S(y, z) \quad (2.29)$$

veya;

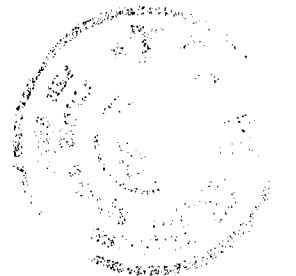
$$\mu_T(x, z) = \max_{y \in Y} \left\{ \mu_R(x, y) \cdot \mu_S(y, z) \right\} \quad (2.30)$$

2.4.2.3. Fuzzy çıkarım kompozisyonu

R , X 'den Y 'ye Fuzzy bağıntı, x , X evrensel kümenin fuzzy alt kümesi ve y , Y evrensel kümenin fuzzy alt kümesi ise ;

$$y = x \circ R \quad (2.31)$$

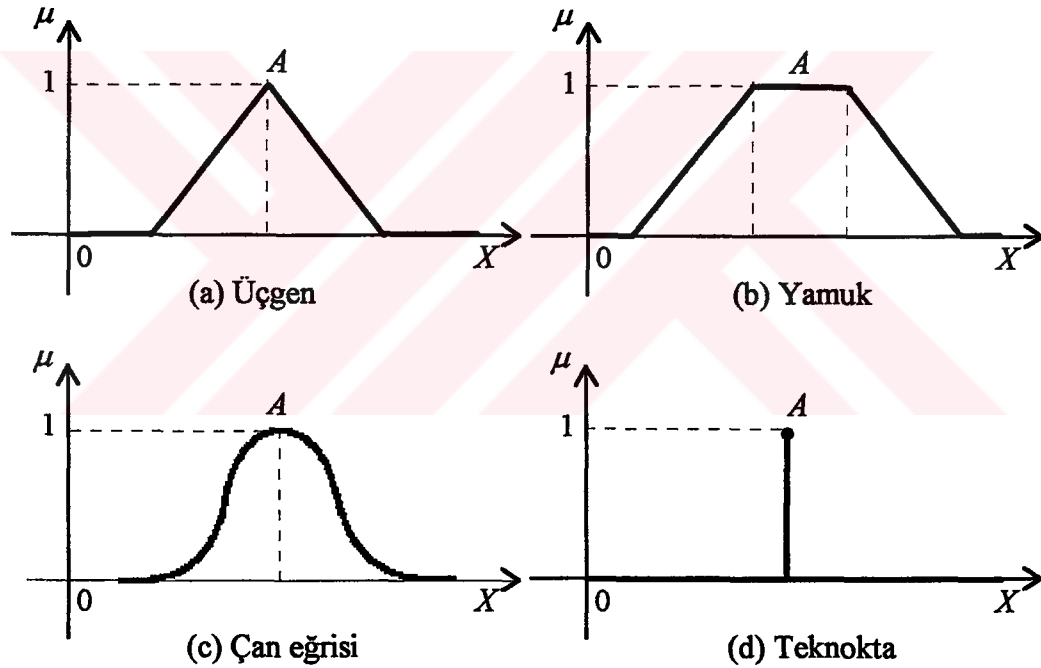
ifadesi çıkarım kompozisyonunu gösterir. Burada $x \circ R$, x ve R 'nin max-min kompozisyonudur.



2.5. Üyelik fonksiyonları

2.5.1. Üyelik fonksiyonu kavramı ve çeşitleri

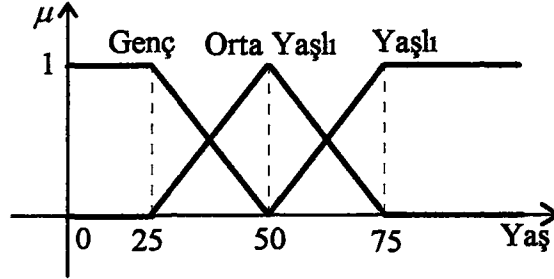
Evrensel kümenin alt kümesi olan fuzzy kümenin üyelerine karşılık gelen ağırlıklar eğri şeklinde gösterilebilir. Bu eğriler üyelik fonksiyonu olarak adlandırılır. Evrensel kümede sadece elemanlara karşılık ağırlıklar alınırsa üyelik fonksiyonu söz konusu olur. Şayet küme eleman ve bu elemana karşılık gelen ağırlık ile beraber gösterilirse bu takdirde bu küme bir fuzzy alt kümeyi teşkil eder.



Şekil 2.5 Değişik üyelik fonksiyonları

Fuzzy alt kümelerin üyelik fonksiyonlarının kullanımı belirsiz ve kapalı kavramların işleme konulmasında yardımcı olur. Üyelik fonksiyonları kontrol edilen sistemin özelliklerine göre üçgen, yamuk ve can eğrisi şeklinde olabilir.(Şekil 2.5.a, b, c) Eğer fuzzy küme yalnız bir tek elemanın ağırlığına sahip ise bu kümenin üyelik fonksiyonu teknokta olan bir şekli gösterir.(Şekil 2.5.d)

Evrensel kümede birden fazla fuzzy küme mevcut ise bu durumda her kümeye ait bir üyelik fonksiyonu belirlenebilir,(Şekil 2.6).



Şekil 2.6. Evrensel kümede üyelik fonksiyonlarının belirlenmesi

Burada 'yaş' sözel bir değişkeni ifade eder. Evrensel kümeyi Y şeklinde gösterirsek bu kümeyi aşağıdaki gibi yazabiliriz.

$$Y(\text{yaş}) = \{ \text{genç, orta yaşlı, yaşlı} \} \quad (2.32)$$

Burada yaş değişkeni $[0, 100]$ arasında değişir. "genç", "orta yaşlı" ve "yaşlı" sözel ifadeler fuzzy kümelerin isimleridir. y tanım olarak yaş sözel değişkeni ise fuzzy alt kümeleri matematiksel olarak aşağıdaki gibi gösterilebilir.

$$\text{genç} = \int_0^{25} 1/y + \int_{25}^{50} [(50-y)/25]/y$$

$$\text{orta yaşlı} = \int_{25}^{50} [(y-25)/25]/y + \int_{50}^{75} [(75-y)/25]/y \quad (2.33)$$

$$\text{yaşlı} = \int_{50}^{75} [(y-50)/25]/y + \int_{75}^{100} 1/y$$

2.5.2. Normal ve iç bükey üyelik fonksiyonu tanımı

Bir üyelik fonksiyonunda en büyük ağırlık 1 ise bu üyelik fonksiyonuna normal üyelik fonksiyonu denir.

$$\max_{x \in X} \mu_A(x) = 1 \quad (\text{normal üyelik fonksiyonu}) \quad (2.34)$$

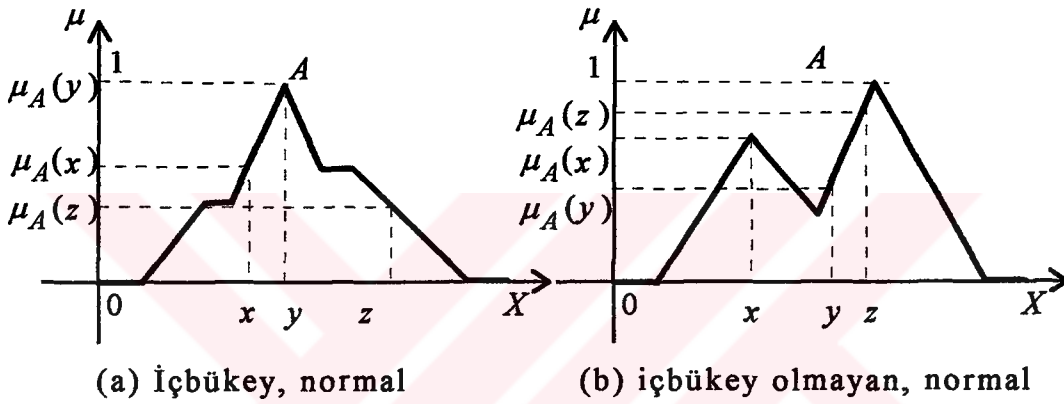
Eğer üyelik fonksiyonunda aşağıdaki tanımların biri geçerli ise bu üyelik fonksiyonu içbükey olur.

$$\mu_A(\lambda x_1 + (1-\lambda)x_2) \geq \min[\mu_A(x_1), \mu_A(x_2)], x_1, x_2 \in U, \lambda \in [0,1] \quad (2.35)$$

veya

$$\mu_A(y) \geq \min[\mu_A(x), \mu_A(z)] \quad x < y < z \quad (2.36)$$

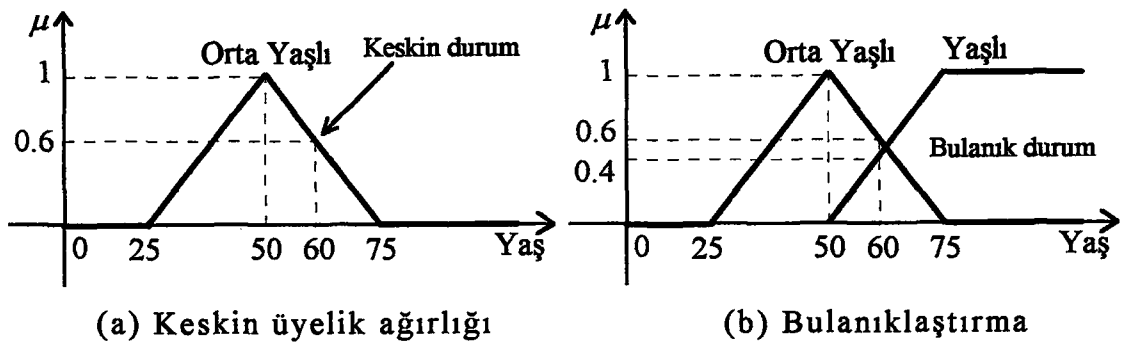
Bu durum şekil 2.7. 'de grafiksel olarak gösterilmektedir.



Şekil 2.7. İç bükey ve normal üyelik fonksiyonları

2.5.3. Bulanıklaştırma işlemi

Şekil 2.8.a 'da gösterilen üyelik fonksiyonunda yaş=65 için üyelik ağırlığı hesaplanırsa 0.6 olarak yalnız bir değer bulunur. Bu değer keskin değerdir. Çünkü bu şekilde yalnız "Orta Yaşlı" isimli bir



Şekil 2.8. Keskin durum ve Bulanıklaştırma

üyelik fonksiyonu vardır. Şayet Şekil 2.8.b. için aynı işlemler yapıldığında 0.6 ve 0.4 gibi iki üyelik ağırlık değeri bulunur. Çünkü bu şekil " Orta Yaşlı" ve "Yaşlı " isimli iki üyelik fonksiyonuna sahiptir. Dolayısıyla yapılan işlem bulanıklaştırmadır.

3. ÇIKARIM İŞLEMİ

Çıkarım işlemi Fuzzy Lojik Kontrol için en önemli kavramlardan birisidir. Algılayıcılardan alınan giriş bilgileri çıkarım işleminden geçirildikten sonra kontrol edilecek sisteme verilecek olan çıkış hesaplamak için kullanılır.

Fuzzy Lojik Kontrol (FLK) çıkarım sistemi genel olarak uzman sistemden daha kolaydır. Çünkü FLK önceki sonuçlardan bağımsız sadece sonraki olaylarla ilgilidir. FLK'nın kural tabanı uzman bilgidен yararlanılarak elde edilir. Kural tabanı Çok Girişli Çok Çıkışlı (ÇGÇÇ) sistem yapısına sahiptir,(Lee, 1990b) .

$$R = \{R_{\text{ÇGÇÇ}}^1, R_{\text{ÇGÇÇ}}^2, \dots, R_{\text{ÇGÇÇ}}^n\} \quad (3.1)$$

Burada $R_{\text{ÇGÇÇ}}^i$: if $(x = A_i \text{ and } \dots \text{ and } y = B_i)$ then $(z_1 = C_i, \dots, z_q = D_i)$ kuralını ifade eder. $Ux \dots xV$ kartezyen çarpımdaki $A_i x \dots x B_i$ bir fuzzy küme $R_{\text{ÇGÇÇ}}^i$ 'ın giriş kısmındaki ifadesini oluşturur. Sonuç q tane bağımsız kontrol olayının bileşimidir. i . $R_{\text{ÇGÇÇ}}^i$ bağıntısı bir bulanık anlam olarak ifade edilebilir.

$$R_{\text{ÇGÇÇ}}^i : (A_i x \dots x B_i) \rightarrow (z_1 + \dots + z_q) \quad (3.2)$$

R bağıntısı aşağıdaki gibi bileşim işlemi olarak da ifade edilebilir.

$$\begin{aligned} R &= \left\{ \bigcup_{i=1}^n R_{\text{ÇGÇÇ}}^i \right\} \\ &= \left\{ \bigcup_{i=1}^n [(A_i x \dots x B_i) \rightarrow (z_1 + \dots + z_q)] \right\} \end{aligned}$$

$$\begin{aligned}
&= \left\{ \bigcup_{i=1}^n [(A_i x \dots x B_i) \rightarrow z_1], \bigcup_{i=1}^n [(A_i x \dots x B_i) \rightarrow z_2], \dots, \bigcup_{i=1}^n [(A_i x \dots x B_i) \rightarrow z_q] \right\} \\
&= \left\{ \bigcup_{k=1}^q \bigcup_{i=1}^n [(A_i x \dots x B_i) \rightarrow z_k] \right\} \tag{3.3}
\end{aligned}$$

$$R = \{RB_{\text{ÇGTÇ}}^1, RB_{\text{ÇGTÇ}}^2, \dots, RB_{\text{ÇGTÇ}}^q\}$$

Sonuçta FLK 'nın R bağıntısı Çok Girişli Tek Çıkışlı (ÇGTÇ) bağıntı ($RB_{\text{ÇGTÇ}}^i$) alt kural tabanının bir kümesi biçimine getirilmiş olur. Her bir $RB_{\text{ÇGTÇ}}^i$ alt kural tabanı bir kontrol çıkışı ve işlem giriş değişkenleri çarpımını içeren n tane fuzzy kontrol kuralından oluşur. ÇGTÇ 'lı fuzzy sistemin kural yapısı ÇGTÇ 'lı bir sistem yapılarının toplamı olarak ifade edilebilir .

$$R = \{RB_{\text{ÇGTÇ}}^1, RB_{\text{ÇGTÇ}}^2, \dots, RB_{\text{ÇGTÇ}}^q\} \tag{3.4}$$

Burada $R_{\text{ÇGTÇ}}^k$: if $(x = A_i \text{ and } \dots \text{ and } y = B_i)$ then $(z_k = D_i)$ $i = 1, 2, \dots, n$ ifadesini gösterir. ÇGTÇ 'lı fuzzy kontrol kurallarını iki girişli / tek çıkışlı fuzzy sistemde aşağıdaki gibi tanımlayalım.

$$\begin{aligned}
&\text{giriş: } x = A' \text{ and } y = B' \\
&\quad R_1: \text{ if } x = A_1 \text{ and } y = B_1 \text{ then } z = C_1 \\
&\text{also} \quad R_2: \text{ if } x = A_2 \text{ and } y = B_2 \text{ then } z = C_2 \\
&\quad \dots \tag{3.5} \\
&\quad \dots \\
&\text{also} \quad R_n: \text{ if } x = A_n \text{ and } y = B_n \text{ then } z = C_n
\end{aligned}$$

$$z = C'$$

x ve y giriş, z çıkış değişkenidir. A_i, B_i ve C_i sırasıyla U, V ve W evrensel kümesinde x, y ve z sözel değişkenlerin sözel değerleridir.

Fuzzy lojik kontrol kuralı "if $(x = A_i \text{ and } y = B_i)$ then $(z = C_i)$ " R fuzzy bağıntı yerine geçer ve aşağıdaki gibi tanımlanır.

$$\mu_{R_i}^{\Delta} = \mu_{(A_i \text{ and } B_i \rightarrow C_i)}(u, v, w) = [\mu_{A_i}(u) \text{ and } \mu_{B_i}(v)] \rightarrow \mu_{C_i}(w) \quad (3.6)$$

Burada " $A_i \text{ and } B_i$ " $U \times V$ 'deki $A_i \times B_i$ fuzzy kümedir.

$R_i^{\Delta} = (A_i \text{ and } B_i) \rightarrow C_i$, $U \times V \times W$ 'daki bir fuzzy bağıntıdır ve " $\rightarrow$ " sembolü bir fuzzy bağıntı kümesini ifade eder. C' sonucu " and " ve fuzzy bağıntıyı içeren çıkarım kurallarından max-min kompozisyon işlemi kullanılarak hesaplanır.

FLK çıkarım sisteminin bazı elverişli özellikleri vardır. Kuralların hepsi max-min ve " also " bağlayıcısı ile bir araya gelirler. Böylece fuzzy kontrol kümesinin hepsinden elde edilecek bir kontrol sonucu tek kontrol kuralından çıkan bir ağırlığa eşittir. Ayrıca aynı ağırlık sonucu sonradan anlatılacağı gibi max-çarpma kompozisyon işleminden de elde edilir. Aşağıdaki işlemlerin sonucu klasik lojik işlemlerinde geçerli değildir.

$$\text{Teorem 1.: } (A', B') \circ \bigcup_{i=1}^n R_i = \bigcup_{i=1}^n (A', B') \circ R_i \quad (3.7)$$

İspat :

$$C' = (A', B') \circ \bigcup_{i=1}^n R_i = (A', B') \circ \bigcup_{i=1}^n (A_i \text{ and } B_i \rightarrow C_i) \quad (3.8)$$

Her $w \in W$ için C' fuzzy kümesinin $\mu_{C'}$ üyelik fonksiyonu belirlenir.

$$\begin{aligned} \mu_{C'}(w) &= (\mu_{A'}(u), \mu_{B'}(v)) \circ \max_{u, v, w} (\mu_{R_1}(u, v, w), \mu_{R_2}(u, v, w), \dots, \mu_{R_n}(u, v, w)) \\ &= \max_{u, v} \left\{ \min \left[(\mu_{A'}(u), \mu_{B'}(v)), \max_{u, v, w} (\mu_{R_1}(u, v, w), \mu_{R_2}(u, v, w), \dots, \mu_{R_n}(u, v, w)) \right] \right\} \\ &= \max_{u, v} \left\{ \max_{u, v, w} \left\{ \min [(\mu_{A'}(u), \mu_{B'}(v)), \mu_{R_1}(u, v, w)] \right\}, \dots \right\} \end{aligned}$$

$$\min\left\{\left(\mu_{A'}(u), \mu_{B'}(v)\right), \mu_{R_n}(u, v, w)\right\} \quad (3.9)$$

$$= \max_{u, v, w} \left\{ \left[\left(\mu_{A'}(u), \mu_{B'}(v)\right) \circ \mu_{R_1}(u, v, w) \right], \dots, \left[\left(\mu_{A'}(u), \mu_{B'}(v)\right) \circ \mu_{R_n}(u, v, w) \right] \right\}$$

Bu durumda;

$$C' = \left[(A', B') \circ R_1 \right] \cup \left[(A', B') \circ R_2 \right] \cup \dots \cup \left[(A', B') \circ R_n \right]$$

$$= \bigcup_{i=1}^n (A', B') \circ R_i \quad (3.10)$$

$$= \bigcup_{i=1}^n (A', B') \circ (A_i \text{ and } B_i \rightarrow C_i)$$

$$\overset{\Delta}{=} \bigcup_{i=1}^n C'_i$$

Teorem 2: R_c , R_p , R_{bp} , ve R_{dp} bağlayıcı ifadelerini bulmak için aşağı -
daki işlemler yazılabilir.

eğer $\mu_{A_i \times B_i} = \mu_{A_i} \wedge \mu_{B_i}$ ise;

$$(A', B') \circ (A_i \text{ and } B_i \rightarrow C_i) = [A' \circ (A_i \rightarrow C_i)] \cap [B' \circ (B_i \rightarrow C_i)] \quad (3.11)$$

eğer $\mu_{A_i \times B_i} = \mu_{A_i} \cdot \mu_{B_i}$ ise;

$$(A', B') \circ (A_i \text{ and } B_i \rightarrow C_i) = [A' \circ (A_i \rightarrow C_i)] [B' \circ (B_i \rightarrow C_i)]$$

İspat:

$$C'_i = (A', B') \circ (A_i \text{ and } B_i \rightarrow C_i)$$

$$\mu_{C'_i} = (\mu_{A'}, \mu_{B'}) \circ (\mu_{A_i \times B_i} \rightarrow \mu_{C_i})$$

$$= (\mu_{A'}, \mu_{B'}) \circ \left[\min(\mu_{A_i}, \mu_{B_i}) \rightarrow \mu_{C_i} \right] \quad (3.12)$$

$$= (\mu_{A'}, \mu_{B'}) \circ \min \left[(\mu_{A_i} \rightarrow \mu_{C_i}), (\mu_{B_i} \rightarrow \mu_{C_i}) \right]$$

$$= \max_{u, v} \left\{ \min \left[(\mu_{A'} \rightarrow \mu_{B'}), \min \left[(\mu_{A_i} \rightarrow \mu_{C_i}), (\mu_{B_i} \rightarrow \mu_{C_i}) \right] \right] \right\}$$

$$\begin{aligned}
&= \max_{u,v} \left\{ \min \left[\min \left[\mu_{A'}, \left(\mu_{A_i} \rightarrow \mu_{C_i} \right) \right], \min \left[\mu_{B'}, \left(\mu_{B_i} \rightarrow \mu_{C_i} \right) \right] \right] \right\} \\
&= \min \left\{ \left[\mu_{A'} \circ \left(\mu_{A_i} \rightarrow \mu_{C_i} \right) \right], \left[\mu_{B'} \circ \left(\mu_{B_i} \rightarrow \mu_{C_i} \right) \right] \right\}
\end{aligned}$$

Böylece;

$$C'_i = \left[A' \circ (A_i \rightarrow C_i) \right] \cap \left[B' \circ (B_i \rightarrow C_i) \right] \quad (3.13)$$

ifadesi elde edilmiş olur.

Aşağıdaki iki özel teorem FLK ' nın anlaşılmasında önemli iki durumu ifade eder.

Teorem 3: Eğer $A' = u_0$, $B' = v_0$ olan teknokta girişleri ise R_c kuralı "min" işlemini ve R_p kuralıda çarpma işlemini gösterir ve aşağıdaki gibi ifade edilirler.

$$\begin{aligned}
1) \quad R_c : \alpha_i^{\wedge} \wedge \mu_{C_i}(w) & \quad 2) \quad R_c : \alpha_i^{\wedge} \wedge \mu_{C_i}(w) \\
R_p : \alpha_i^{\wedge} \cdot \mu_{C_i}(w) & \quad R_p : \alpha_i^{\wedge} \cdot \mu_{C_i}(w)
\end{aligned} \quad (3.14)$$

Burada $\alpha_i^{\wedge} = \mu_{A_i}(u_0) \wedge \mu_{B_i}(v_0)$ ve $\alpha_i^{\wedge} = \mu_{A_i}(u_0) \cdot \mu_{B_i}(v_0)$ ifadelerine eşittir.

İspat:

$$\begin{aligned}
1) \quad C'_i &= \left[A' \circ (A_i \rightarrow C_i) \right] \cap \left[B' \circ (B_i \rightarrow C_i) \right] \\
\mu_{C'_i} &= \min \left\{ \left[u_0 \circ \left(\mu_{A_i}(u) \rightarrow \mu_{C_i}(w) \right) \right], \left[v_0 \circ \left(\mu_{B_i}(v) \rightarrow \mu_{C_i}(w) \right) \right] \right\} \\
&= \min \left\{ \left[\left(\mu_{A_i}(u_0) \rightarrow \mu_{C_i}(w) \right) \right], \left[\left(\mu_{B_i}(v_0) \rightarrow \mu_{C_i}(w) \right) \right] \right\}
\end{aligned} \quad (3.15)$$

$$\begin{aligned}
2) \quad C'_i &= \left[A' \circ (A_i \rightarrow C_i) \right] \cdot \left[B' \circ (B_i \rightarrow C_i) \right] \\
\mu_{C'_i} &= \left[u_0 \circ \left(\mu_{A_i}(u) \rightarrow \mu_{C_i}(w) \right) \right] \cdot \left[v_0 \circ \left(\mu_{B_i}(v) \rightarrow \mu_{C_i}(w) \right) \right] \\
&= \left[\left(\mu_{A_i}(u_0) \rightarrow \mu_{C_i}(w) \right) \right] \cdot \left[\left(\mu_{B_i}(v_0) \rightarrow \mu_{C_i}(w) \right) \right]
\end{aligned} \quad (3.16)$$

Aşağıda anlatılacağı gibi son teorem hesaplama işlemlerini kolaylaştırdığı gibi FLK 'de fuzzy çıkarım işleminin grafik yoldan yapılmasında sağlar. max-çarpma işlemine tekrar bakılacak olursa;

$$\text{Teorem 1': } (A', B') \cdot \bigcup_{i=1}^n R_i = \bigcup_{i=1}^n (A', B') \cdot R_i \quad (3.17)$$

Teorem 2': R_c, R_p, R_{bp} , ve R_{dp} bağlayıcı ifadelerini bulmak için aşağıdaki işlemler yazılabilir.

Eğer $\mu_{A_i \times B_i} = \mu_{A_i} \wedge \mu_{B_i}$ ise;

$$(A', B') \cdot (A_i \text{ and } B_i \rightarrow C_i) = [A' \cdot (A_i \rightarrow C_i)] \cap [B' \cdot (B_i \rightarrow C_i)] \quad (3.18)$$

Eğer $\mu_{A_i \times B_i} = \mu_{A_i} \cdot \mu_{B_i}$ ise;

$$(A', B') \cdot (A_i \text{ and } B_i \rightarrow C_i) = [A' \cdot (A_i \rightarrow C_i)] \cdot [B' \cdot (B_i \rightarrow C_i)] \quad (3.19)$$

Teorem 3': Eğer $A' = u_0, B' = v_0$ olan teknokta girişleri ise R_c kuralı "min" işlemini ve R_p kuralıda çarpma işlemini gösterir ve aşağıdaki gibi ifade edilirler.

$$\begin{aligned} 1) R_c : \alpha_i^{\wedge} \wedge \mu_{C_i}(w) & \quad 2) R_c : \alpha_i^{\wedge} \wedge \mu_{C_i}(w) \\ R_p : \alpha_i^{\wedge} \cdot \mu_{C_i}(w) & \quad R_p : \alpha_i^{\wedge} \cdot \mu_{C_i}(w) \end{aligned} \quad (3.20)$$

Burada $\alpha_i^{\wedge} = \mu_{A_i}(u_0) \wedge \mu_{B_i}(v_0)$ ve $\alpha_i = \mu_{A_i}(u_0) \cdot \mu_{B_i}(v_0)$ ifadelerine eşittir..

Böylece en son durum aşağıdaki gibi olur.

$$R_c : \mu_{C'} = \bigcup_{i=1}^n \alpha_i^{\wedge} \wedge \mu_{C_i} \quad (3.21)$$

$$R_p : \mu_{C'} = \bigcup_{i=1}^n \alpha_i \cdot \mu_{C_i}$$

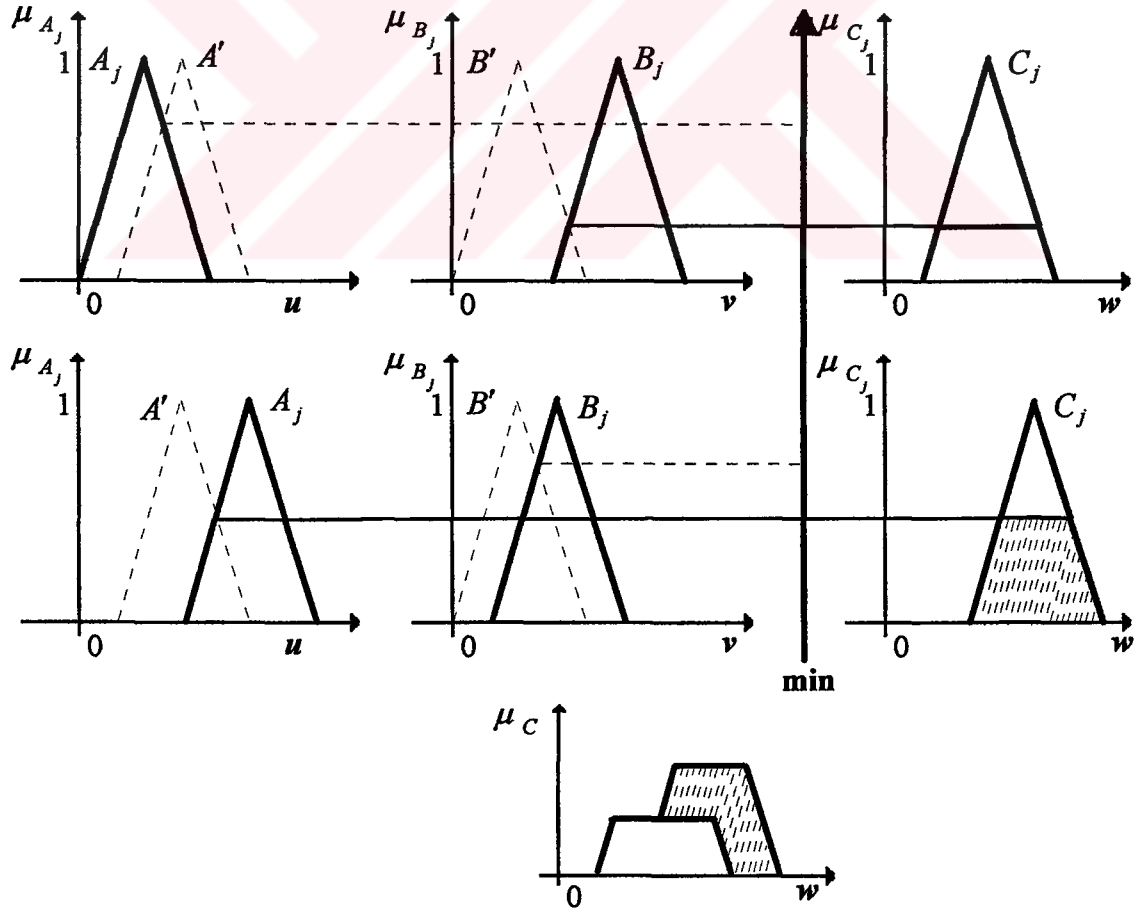
Burada α_i ağırlık faktörü i . kuralda fuzzy kontrol işleminin bir ölçüsüdür. Söz konusu ağırlık faktörü iki yöntem belirlenir. Birinci yöntem FLK uygulamalarında geniş ölçüde kullanılan kartezyen çarpımda bulunan "min" işlemi ve ikinci yöntem kartezyen çarpımdaki

cebirsel çarpma işlemidir. Giriş işlemlerindeki kolaylığı sağlamak bakımından birinci yöntem ikinci yöntemden daha etkindir. Bu yüzden uygulamalarda birinci yöntem daha çok kullanılmaktadır.

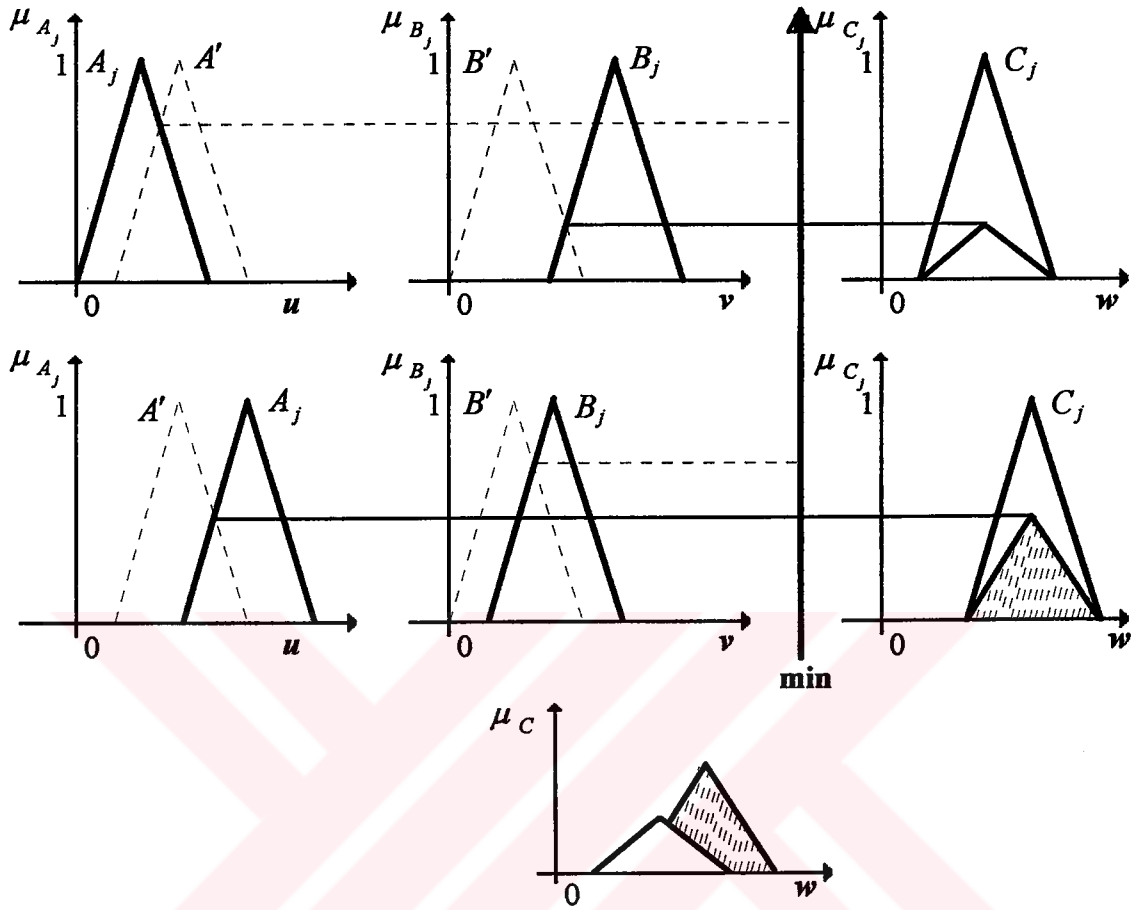
Basit olarak iki fuzzy kontrol kuralının aşağıdaki gibi kabul edildiğini kabul edersek ;

$$\begin{aligned} R_1: & \text{ if } x = A_1 \text{ and } y = B_1 \text{ then } z = C_1 \\ R_2: & \text{ if } x = A_2 \text{ and } y = B_2 \text{ then } z = C_2 \end{aligned} \quad (3.22)$$

Şekil 3.1. R_c ve $\alpha_i^{\wedge}$ şartları altında Teorem 2 'nin grafik yorumunu gösterir. Şekil 3.2. R_p ve $\alpha_i^{\wedge}$ şartları altında Teorem 2 'nin grafik yorumunu gösterir.



Şekil 3.1. R_c ve $\alpha_i^{\wedge}$ şartları için Teorem 2 'nin grafik gösterilimi.



Şekil 3.2. R_p ve α' şartları için Teorem 2 'nin grafik gösterilimi

Girişler algılayıcılar tarafından ölçülür ve keskin değere sahiptirler. Giriş değerinin ölçekleme vasıtasıyla fuzzy kümelerdeki giriş bilgisine çevrilmesi daha uygun olur. Aynı zamanda bu değerler fuzzy teknoktaya karşılık gelirler. Birinci ve ikinci kuralların α_1 ve α_2 ağırlık faktörleri aşağıdaki gibi açıklanır.

$$\alpha_1 = \mu_{A_1}(x_0) \wedge \mu_{B_1}(y_0) \quad (3.23)$$

$$\alpha_2 = \mu_{A_2}(x_0) \wedge \mu_{B_2}(y_0)$$

Burada $\mu_{A_1}(x_0)$ ve $\mu_{B_1}(y_0)$ kural tabanındaki bilgi ile kaynak bilgisi arasında aynı derecede etkiye sahiptir. Bu bağıntılar dört çeşit fuzzy karar vermede önem taşırlar.

3.2. İkinci tip fuzzy karar verme; Fuzzy anlam ifadesi olarak "çarpma" işlemi

Bu tür karar vermede i . kural kontrol kararına yön gösterir.

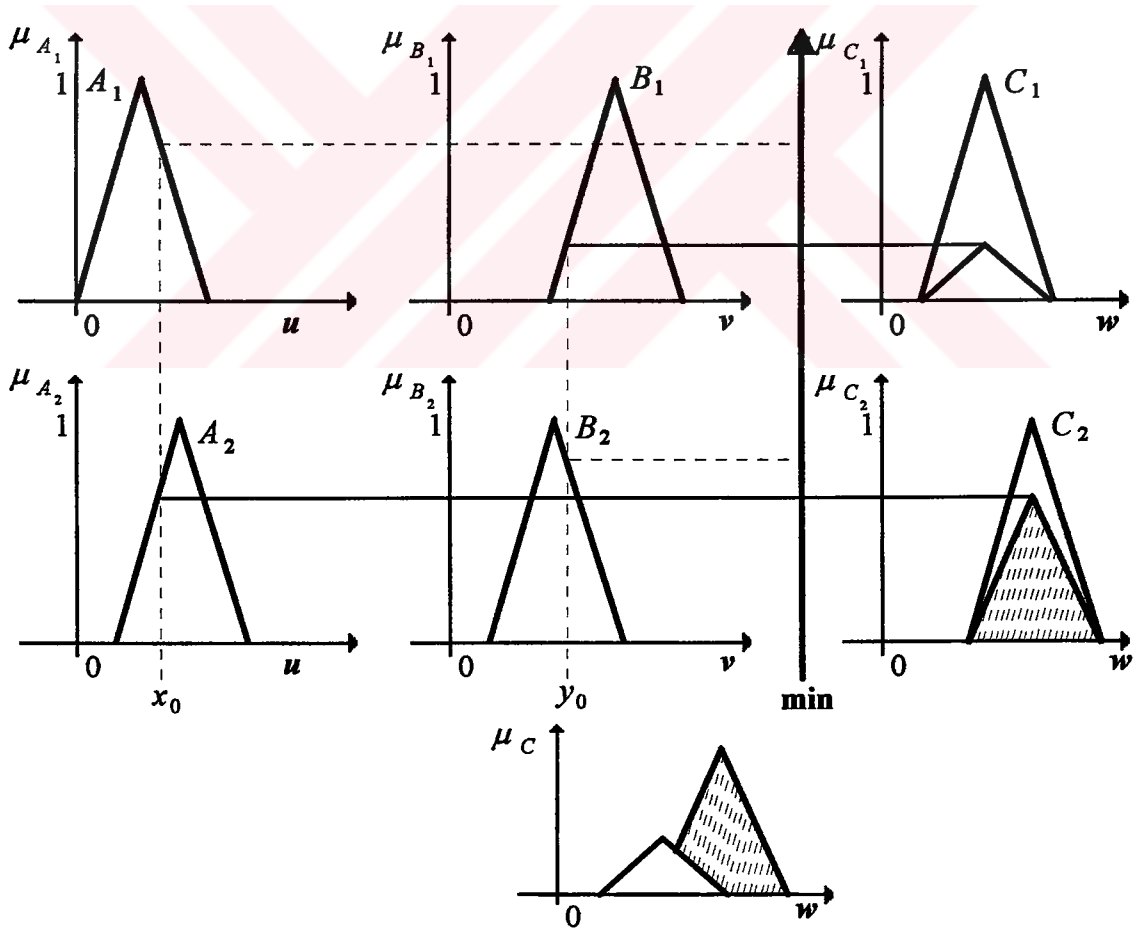
$$\mu_{C_i}(w) = \alpha_i \cdot \mu_{C_i}(w) \quad (3.26)$$

Çıkarımı yapılan μ_C üyelik fonksiyonu aşağıdaki gibi hesaplanır.

$$\mu_C(w) = \mu_{C_1} \vee \mu_{C_2} \quad (3.27)$$

$$\mu_C(w) = [\alpha_1 \cdot \mu_{C_1}(w)] \vee [\alpha_2 \cdot \mu_{C_2}(w)]$$

Şekil 3.4. fuzzy karar verme işleminde R_p yöntemi yoluyla teorem 3'ün bir grafiksel yorumunu gösterir.



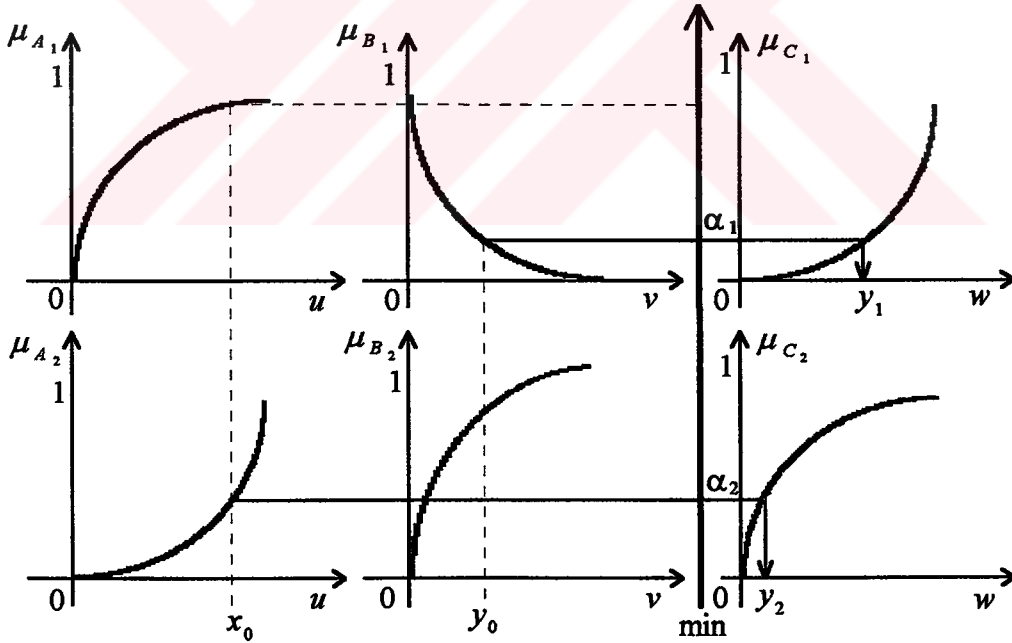
Şekil 3.4. Fuzzy karar verme 2 işleminin grafiksel gösterilimi

3.3. Üçüncü tip fuzzy karar verme ; Monotonik üyelik fonksiyonu olarak sözel terimli metod

A_i , B_i ve C_i fuzzy kümelerin üyelik fonksiyonu monotonik olduğu birinci tip karar verme ile aynı yolu kullanan basit bir yöntemdir. Bununla beraber işlemlerde A_i , B_i monotonik olmayabilir, fakat C_i monotonik olması gerekir.

Bu yöntemde birinci kuraldan elde edilen sonuç $\alpha_1 = C_1(y_1)$ ifadesindeki α_1 'e ikinci kuraldan elde edilen sonuç $\alpha_2 = C_2(y_2)$ ifadesindeki α_2 'ye eşittir. Keskin kontrol sonuç değeri aşağıdaki gibi ağırlıkların kombinasyonu olarak alınır.(Şekil 4.5)

$$z_0 = \frac{\alpha_1 y_1 + \alpha_2 y_2}{\alpha_1 + \alpha_2} \quad (3.28)$$



Şekil 3.5. Fuzzy karar verme 3 işleminin grafiksel gösterilimi

3.4. Dördüncü tip fuzzy karar verme; Kuralın sonucu sözel giriş değişkenlerinin bir fonksiyonu

Dördüncü tip fuzzy karar verme sonuç durumu eşitlik fonksiyonu ile değiştirilmiş bir şeklidir. Karar vermenin bu metodunda i . kontrol kuralı aşağıdaki yapıdadır.

$$R_i: \text{if } (x = A_i \text{ and } \dots \text{and } y = B_i) \text{ then } z = f_i(x, \dots, y) \quad (3.29)$$

Burada sırasıyla $x, \dots, y$ giriş değişkenlerini ve z kontrol çıkış değerini gösteren sözel değişkenlerdir. $A_i, \dots, B_i, U, \dots, V$ evrensel kümesindeki $x, \dots, y$ sözel değişkenlerinin sözel değerleridir. f_i ve $i = 1, 2, \dots, n$ giriş alt uzayında tanımlanan $x, \dots, y$ giriş değişkenlerinin bir fonksiyonudur.

Basit olarak çıkış değerini hesaplamak için aşağıdaki iki kuralı ele alalım ;

$$R_1: \text{if } x = A_1 \text{ and } y = B_1 \text{ then } z = f_1(x, y) \quad (3.30)$$

$$R_2: \text{if } x = A_2 \text{ and } y = B_2 \text{ then } z = f_2(x, y)$$

Birinci kuraldan kontrol olayının çıkarılan sonuç değeri $\alpha_1 f_1(x_0, y_0)$ ve ikinci kuraldan kontrol olayının çıkarılan sonuç değeri $\alpha_2 f_2(x_0, y_0)$ 'dır. Keskin çıkış kontrol sonuç değeri aşağıdaki ifade ile hesaplanır.

$$z_0 = \frac{\alpha_1 f_1(x_0, y_0) + \alpha_2 f_2(x_0, y_0)}{\alpha_1 + \alpha_2} \quad (3.31)$$

Bu yöntem Tagaki ve Sugeno tarafından geliştirilmiş ve bir model arabayı kullanıcı olmaksızın uygun bir şekilde graja park etme işlemine kullanmışlardır, (Sugeno and Murakami, 1985).



3.5.Çıkarım işleminin bir uygulama üzerinde grafiksel ve cebirsel anlatımı

3.5.1. Grafiksel anlatım

FLD 'nin giriş değişkenleri hata (e), hatanın değişimi (ce) ve çıkış değişkeni (u) alınmıştır (Şekil3.6). Evrensel kümeler e , ce ve u sözel değişkenleri temsil eder. Bu duruma göre aşağıdaki kurallar isteğe göre belirlenmiş olsun, (Li and Lau, 1989).

$$\begin{aligned}
 &\text{if } e = \text{SF and } ce = \text{KP then } u = \text{KN} \\
 &\text{if } e = \text{SF and } ce = \text{SF then } u = \text{SF} \\
 &\text{if } e = \text{KN and } ce = \text{KN then } u = \text{KP} \\
 &\text{if } e = \text{KN and } ce = \text{SF then } u = \text{BP}
 \end{aligned} \tag{3.32}$$



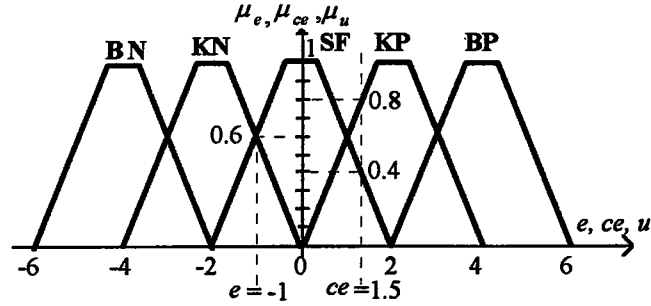
Şekil 3.6. FLK 'nin giriş ve çıkış değişkenleri

Burada e , ce ve u sözel değişkenli üç evrensel küme her biri fuzzy alt küme olan sözel etiketleri içerir.

- Büyük Pozitif (BP)
- Küçük Pozitif (KP)
- Sıfır (SF)
- Küçük Negatif (KN)
- Büyük Negatif (BN)

Bu sözel etiketlere ilişkin üyelik fonksiyonları şekil 3.7 'de gösterilmiştir. Bu şekilde üyelik fonksiyonları e , ce ve u sözel değişkenlerinin hepsi için aynı seçilmiştir. e , ce ve u sözel

değişkenleri için kontrol edilecek sistemin durumuna göre birbirinden farklı üyelik fonksiyonları ve başka sözel etiketler seçilebilir.

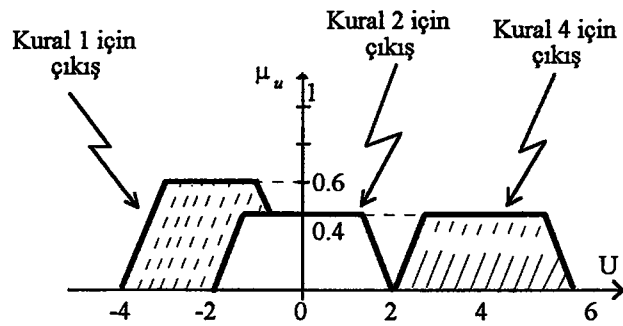
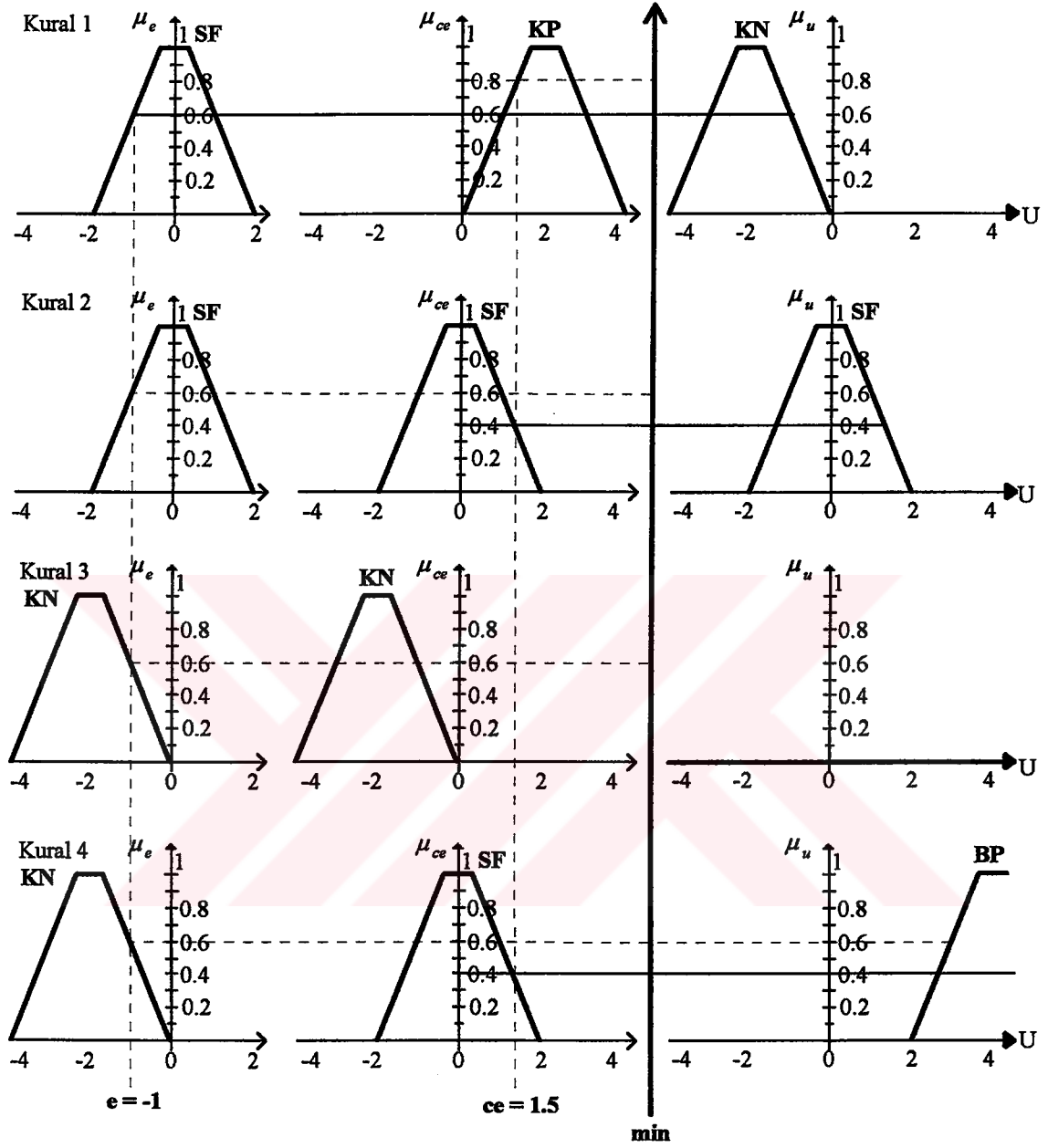


Şekil 3.7. e , ce ve u için üyelik fonksiyonları.

Tablo 3.1. Kuralların gösterilmesi

$\begin{smallmatrix} ce \\ e \end{smallmatrix}$	BN	KN	SF	KP	BP
BN	X	X	X	X	X
KN	X	KP	BP	X	X
SF	X	X	SF	KN	X
KP	X	X	X	X	X
BP	X	X	X	X	X

Başta verilen kurallar tablo 3.1 'de gösterilmiştir. Bu tabloda e için BN, KP ve BP, ce için BN, BP ve u için BN sözel etiketli fuzzy alt kümeler kullanılmamıştır. Eğer sistemi daha iyi kontrol edebilmek için yeni kurallar ilave edilmesi istenilirse duruma göre bu sözel etiketler kullanılabilir.



Şekil 3.8. Bulanıklaştırma işlemi ve çıkış grafiğinin bulunması

Şekil 3.7 'de görüldüğü gibi $e = -1$ ve $ce = 1.5$ ise e için KN ve SF sözel etiketli ce için SF ve KP sözel etiketli fuzzy alt kümelerde bir ağırlığa sahip olurlar. Bu durumda e ve ce evrensel kümelerindeki fuzzy alt kümeleri arasında oluşturulacak kartezyen çarpımlar aşağıdaki gibi olur.

1. $e = \text{KN}$ $ce = \text{SF}$
 2. $e = \text{KN}$ $ce = \text{KP}$
 3. $e = \text{SF}$ $ce = \text{KP}$
 4. $e = \text{SF}$ $ce = \text{SF}$
- (3.33)

Fakat yukarıdaki 1, 3, 4 nolu durumlar daha önce yazılan kurallarda mevcut durumdadır. 1 nolu durum kural 4 'e, 3 nolu durum kural 1 'e ve 4 nolu durum kural 2 'ye karşılık gelir. Bu kurallara ilişkin bulanıklaştırma işlemi ve sonuçta çıkışın bulunması şekil 3.8 'de grafiksel olarak gösterilmiştir.

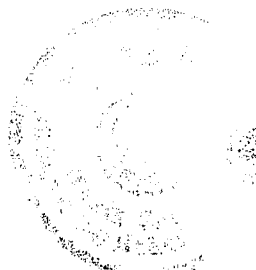
Yukarıda elde edilen çıkış grafiğine bölüm 4 'de anlatılan netleştirme yöntemlerinden biri uygulanarak kontrol edilecek sisteme verilecek olan keskin sonuç hesaplanmış olur.

3.5.2.Cebirsel anlatım

Şekil 3.7 'de gösterildiği gibi e , ce ve u ;

$$\begin{aligned}
 e &= \{\text{BP}, \text{KP}, \text{SF}, \text{KN}, \text{BN}\} \\
 ce &= \{\text{BP}, \text{KP}, \text{SF}, \text{KN}, \text{BN}\} \\
 u &= \{\text{BP}, \text{KP}, \text{SF}, \text{KN}, \text{BN}\}
 \end{aligned}$$
(3.34)

fuzzy alt kümeleri kapsarlar. Daha önce anlatılan kurallara ait fuzzy bağıntılar aşağıdaki gibi yazılabilir.



$$\begin{aligned}
R_1 &= [\mu_{SF}(e) \wedge \mu_{KP}(ce)] \circ \mu_{KN}(u) \\
R_2 &= [\mu_{SF}(e) \wedge \mu_{SF}(ce)] \circ \mu_{SF}(u) \\
R_3 &= [\mu_{KN}(e) \wedge \mu_{KN}(ce)] \circ \mu_{KP}(u) \\
R_4 &= [\mu_{KN}(e) \wedge \mu_{SF}(ce)] \circ \mu_{BP}(u)
\end{aligned} \tag{3.35}$$

Fuzzy bağıntı tanımındaki max-min kompozisyonu kullanılarak R_1, R_2, R_3 ve R_4 bağıntıları aşağıdaki duruma getirilir.

$$\begin{aligned}
R_1(e, ce, u) &= \max \left\{ \min [\min [\mu_{SF}(e), \mu_{KP}(ce)], \mu_{KN}(u)] \right\} \\
R_2(e, ce, u) &= \max \left\{ \min [\min [\mu_{SF}(e), \mu_{SF}(ce)], \mu_{SF}(u)] \right\} \\
R_3(e, ce, u) &= \max \left\{ \min [\min [\mu_{KN}(e), \mu_{KN}(ce)], \mu_{KP}(u)] \right\} \\
R_4(e, ce, u) &= \max \left\{ \min [\min [\mu_{KN}(e), \mu_{SF}(ce)], \mu_{BP}(u)] \right\}
\end{aligned} \tag{3.36}$$

$e = -1$ ve $ce = 1.5$ için şekil 3.7 'den faydalanarak her fuzzy alt kümede bu değerler ilişkin aşağıdaki ağırlıklar bulunur.

$$\begin{aligned}
\mu_{BN}(e)|_{e=-1} &= 0 & \mu_{BN}(ce)|_{ce=1.5} &= 0 \\
\mu_{KN}(e)|_{e=-1} &= 0.6 & \mu_{KN}(ce)|_{ce=1.5} &= 0 \\
\mu_{SF}(e)|_{e=-1} &= 0.6 & \mu_{SF}(ce)|_{ce=1.5} &= 0.4 \\
\mu_{KP}(e)|_{e=-1} &= 0 & \mu_{KP}(ce)|_{ce=1.5} &= 0.8 \\
\mu_{BP}(e)|_{e=-1} &= 0 & \mu_{BP}(ce)|_{ce=1.5} &= 0
\end{aligned} \tag{3.37}$$

Bu ağırlıklara göre R_1, R_2, R_3 ve R_4 fuzzy bağıntıları aşağıdaki duruma gelirler.

$$\begin{aligned}
R_1(e, ce, u) \Big|_{e=-1, ce=1.5} &= \max \left\{ \min \left[\min(0.6, 0.8), \mu_{KN}(u) \right] \right\} \\
R_2(e, ce, u) \Big|_{e=-1, ce=1.5} &= \max \left\{ \min \left[\min(0.6, 0.4), \mu_{SF}(u) \right] \right\} \\
R_3(e, ce, u) \Big|_{e=-1, ce=1.5} &= \max \left\{ \min \left[\min(0, 0), \mu_{KP}(u) \right] \right\} \\
R_4(e, ce, u) \Big|_{e=-1, ce=1.5} &= \max \left\{ \min \left[\min(0.6, 0.4), \mu_{BP}(u) \right] \right\}
\end{aligned} \tag{3.38}$$

parantez içindeki "min" işlemi yapılırsa ifadeler;

$$\begin{aligned}
R_1(e, ce, u) \Big|_{e=-1, ce=1.5} &= \max \left\{ \min[0.6, \mu_{KN}(u)] \right\} = \min[0.6, \mu_{KN}(u)] \\
R_2(e, ce, u) \Big|_{e=-1, ce=1.5} &= \max \left\{ \min[0.4, \mu_{SF}(u)] \right\} = \min[0.4, \mu_{SF}(u)] \\
R_3(e, ce, u) \Big|_{e=-1, ce=1.5} &= \max \left\{ \min[0, \mu_{KP}(u)] \right\} = 0 \\
R_4(e, ce, u) \Big|_{e=-1, ce=1.5} &= \max \left\{ \min[0.4, \mu_{BP}(u)] \right\} = \min[0.4, \mu_{BP}(u)]
\end{aligned} \tag{3.39}$$

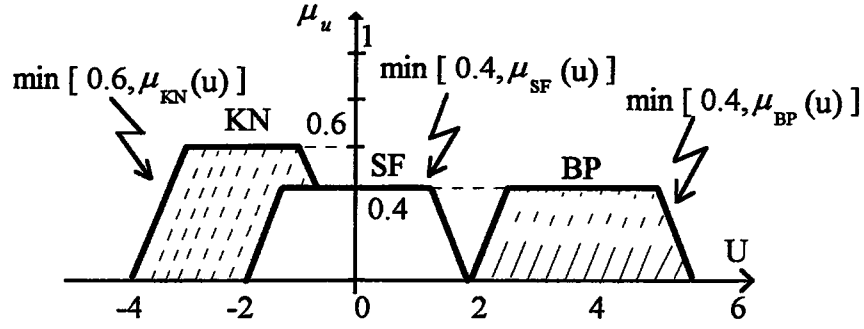
şeklinde yazılır. Bütün bu bağıntılar bileşme özelliği kullanılarak bir bağıntı durumuna dönüştürülür ve sonuç çıkış ifadesi bulunur.

$$\begin{aligned}
R &= R_1 \vee R_2 \vee R_3 \vee R_4 \\
R(e, ce, u) &= \max_{e, ce, u} (R_1, R_2, R_3, R_4)
\end{aligned} \tag{3.40}$$

ise R_1, R_2, R_3 ve R_4 sonuç fuzzy bağıntıları kullanılarak;

$$\begin{aligned}
R(e, ce, u) \Big|_{e=-1, ce=1.5} &= \max \left\{ \min[0.6, \mu_{KN}(u)], \min[0.4, \mu_{SF}(u)], 0, \min[0.4, \mu_{BP}(u)] \right\} \\
R(e, ce, u) \Big|_{e=-1, ce=1.5} &= \max \left\{ \min[0.6, \mu_{KN}(u)], \min[0.4, \mu_{SF}(u)], \min[0.4, \mu_{BP}(u)] \right\}
\end{aligned} \tag{3.41}$$

şekline gelir ve bu sonuç şekil 3.9 'da grafiksel olarak gösterilmiştir.



Şekil 3.9 Çıkış ifadesi

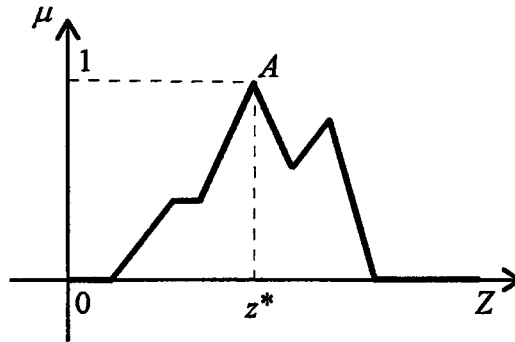
4. NETLEŞTİRME YÖNTEMLERİ

Yedi ayrı netleştirme yöntemi vardır. Bunlar aşağıda grafiksel ve cebirsel olarak anlatılmıştır, (Ross, 1995).

4.1. Maksimum-üyelik yöntemi

Yükseklik metodu olarak da bilinir. Bu yöntemde üyelik fonksiyonunda en büyük ağırlık keskin sonuç olarak alınır (Şekil 4.1). Aşağıdaki gibi ifade edilir.

$$\mu_c(z^*) \geq \mu_c(z) \quad \forall z \in Z \quad (4.1)$$



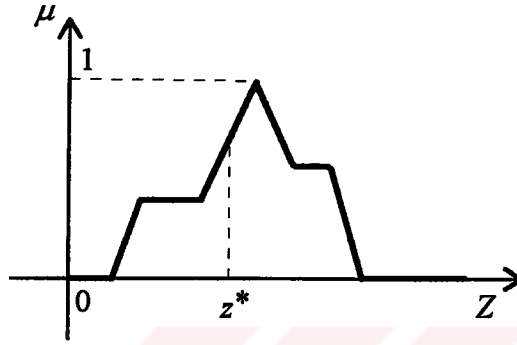
Şekil 4.1. Maks-Üyelik yöntemi

4.2. Merkezi yöntem

Bu yöntem ağırlık merkezini bulma veya alan merkezini bulma olarak adlandırılır. En çok kullanılan netleştirme yöntemlerin birisidir. Bu yöntem aşağıdaki gibi ifade edilir.

$$z^* = \frac{\int \mu_c(z).z.dz}{\int \mu_c(z).dz} \quad (4.2)$$

Burada $\int$ cebirsel integrali gösterir. Bu yöntem şekil 4.2. 'de gösterilmiştir.



Şekil 4.2. Merkezi yöntem

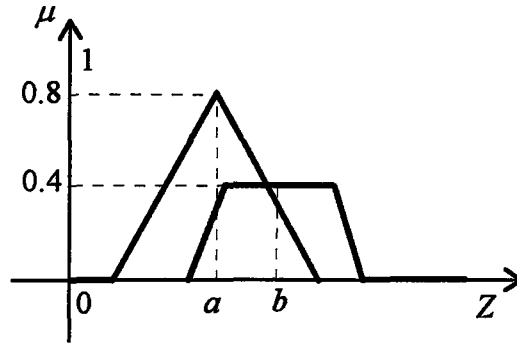
4.3. Ağırlık ortalamasını bulma yöntemi

Bu yöntem sadece çıkış grafiğinin simetrik olması durumunda uygulanır. Aşağıdaki gibi gösterilir.

$$z^* = \frac{\sum \mu_c(\bar{z}).\bar{z}}{\sum \mu_c(\bar{z})} \quad (4.3)$$

$\sum$ cebirsel toplam sembolünü gösterir. Şekil 4.3 'de gösterildiği gibi çıkıştaki üyelik fonksiyonlarından en büyük ağırlıklar alınır ve aşağıdaki işlem yapılarak netleşmiş sonuç bulunur.

$$z^* = \frac{a(0.8) + b(0.4)}{0.8 + 0.4} \quad (4.4)$$

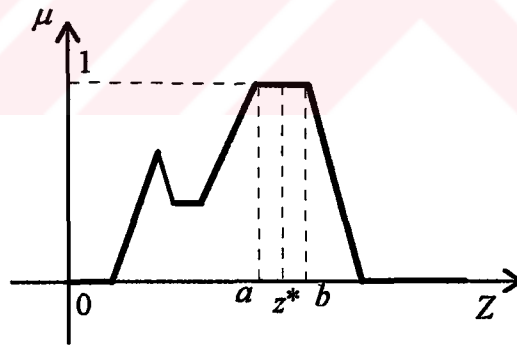


Şekil 4.3. Ağırlık ortalamasını bulma yöntemi

4.4. Maksimum üyelikleri ortalama yöntemi

Bu yöntem maksimum üyelik noktalarını ortalama yöntemi olarakda bilinir. Maksimum-üyelik yöntemine benzerdir. Farklı olarak maksimum noktaların evrensel kümedeki değerlerinin ortalaması alınır. Şekil 4.4 için netleşmiş sonuç aşağıdaki gibi bulunur.

$$z^* = \frac{a+b}{2} \quad (4.5)$$



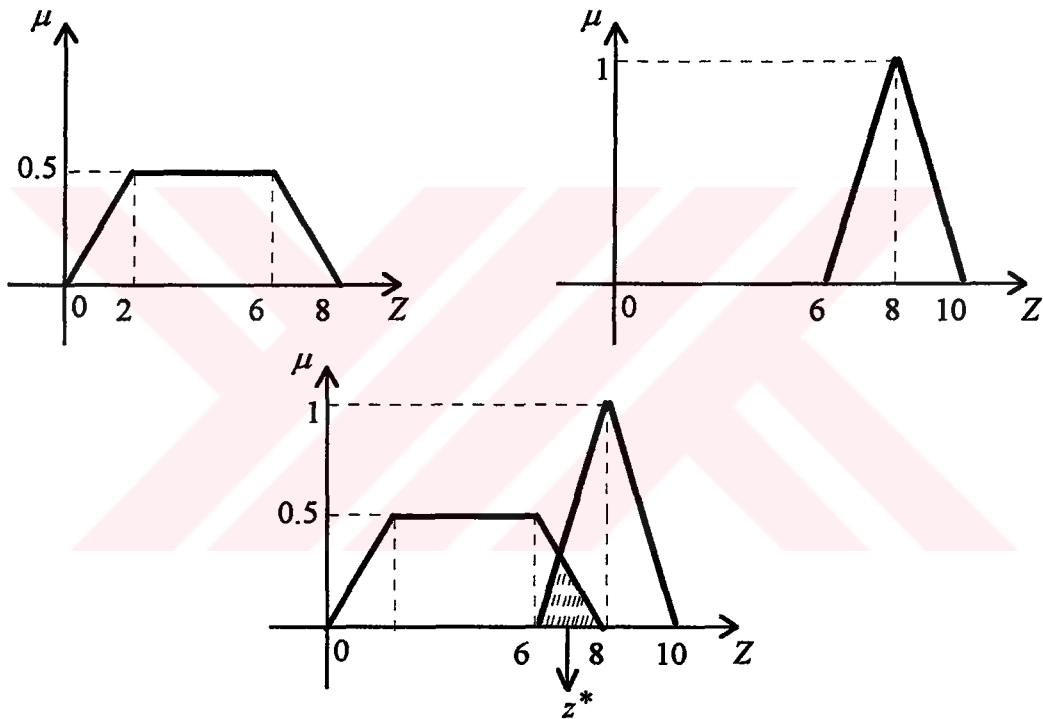
Şekil 4.4. Maksimum üyelikleri ortalama yöntemi

4.5. Toplamların merkezini bulma yöntemi

Diğer hesabı uzun olan netleştirme yöntemlerinden daha çabuk sonucun bulunmasını sağlar. Şekil 4.5. 'de gösterildiği gibi çıkıştaki fuzzy küme C_1 ve C_2 üyelik fonksiyonlarının bileşimi şeklindedir. Netleşmiş sonuç aşağıdaki gibi hesaplanır.

$$z^* = \frac{\int_z \sum_{k=1}^n \mu_{c_k}(z) \cdot z \cdot dz}{\int_z \sum_{k=1}^n \mu_{c_k}(z) \cdot dz} \quad (4.6)$$

Bu yöntem ağırlıkları ortalama yöntemine benzer. Yalnız bu yöntemde sonuçtaki üyelik fonksiyonlarının merkezdeki ağırlıkları yerine bu üyelik fonksiyonlarının alanları alınarak hesap yapılır.



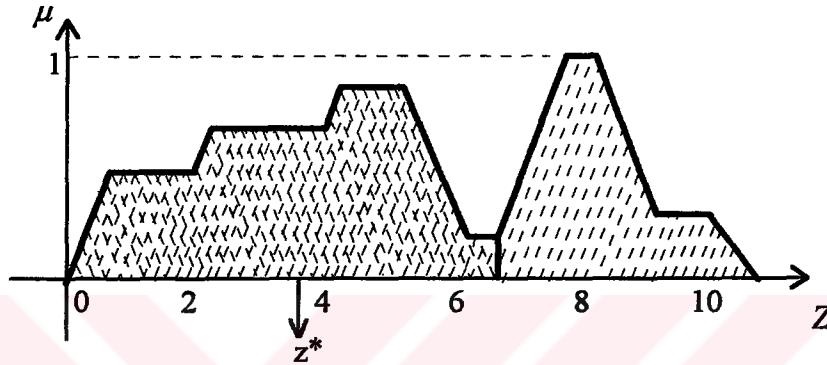
Şekil 4.5. Toplamların merkezini bulma yöntemi

4.6. En büyük alan merkezini bulma yöntemi

Eğer çıkışta birden fazla içbükey bölge var ise bu içbükey bölgelerden en büyük alana sahip olan alınır. Netleşmiş z^* çıkış değeri hesaplamak için bu bölgeye daha önce anlatılan merkezi bulma yöntemi uygulanır. Bu yöntem grafik olarak şekil 4.6 'da gösterilir ve cebirsel olarak aşağıdaki gibi hesaplanır.

$$z^* = \frac{\int \mu_{c_m}(z).z.dz}{\int \mu_{c_m}(z).dz} \quad (4.7)$$

Burada C_m , C_k 'daki en büyük içbükey alanı sembolize eder.



Şekil 4.6.En büyük alan merkezini bulma yöntemi

4.7. İlk veya son maksimum değeri bulma yöntemi

Bu yöntemde çeşitli üyelik fonksiyonlarının bileşimi olan C_k 'nın çıkış evrensel kümesinde ağırlığı büyük olan en yakın değer veya ağırlığı büyük olan en son değer alınır.Aşağıdaki gibi ifade edilir.

İlk maksimum değerini alma yöntemi;

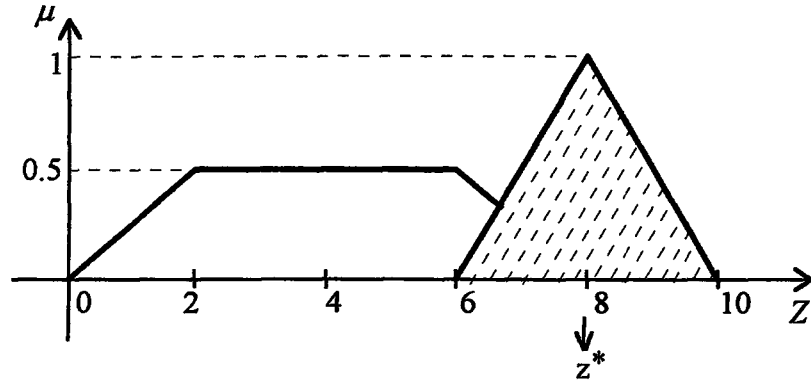
$$z^* = \inf_{z \in Z} \{z \in Z \mid \mu_{C_k} = hgt(C_k)\} \quad (4.8)$$

şeklinde ve son maksimum değerini alma yöntemi;

$$z^* = \sup_{z \in Z} \{z \in Z \mid \mu_{C_k} = hgt(C_k)\} \quad (4.9)$$

şeklinde gösterilir.

Burada infimum (inf) evrensel kümede ağırlığı en büyük olan ilk değeri ve subremum (sub) ağırlığı en büyük olan en son değeri gösterir.



Şekil 4.7. İlk veya son maksimum değeri bulma yöntemi

5. FUZZY LOJİK KONTROL

5.1. Temel Fuzzy Lojik Kontrol

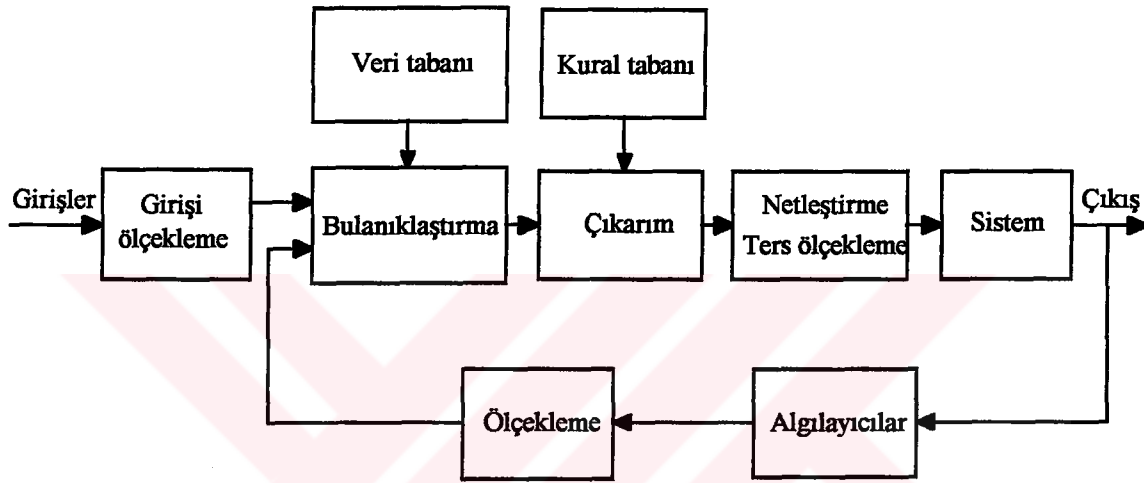
Temel Fuzzy Lojik Kontrol sisteminin blok diyagramı şekil 5.1 'de gösterilmiştir.

Şekil 5.1 'deki veri tabanı modülü giriş ve çıkışın fuzzy kısımlarının tümü hakkındaki verileri, kural tabanlı sistemde tanımlanan giriş değişkenlerine karşılık gelen üyelik fonksiyonlarını ve sistemin kontrolü için kontrol olaylarını veya çıkış değişkenlerini içerir.

Temel fuzzy lojik kontrol sisteminin tasarım basamakları aşağıdaki gibidir.

1. Giriş ve çıkış değişkenleri belirlenir.
2. Her değişkene karşılık gelen evrensel kümelerde fuzzy alt kümelerin bölümleri ve bu fuzzy alt kümelere karşılık gelen sonradan kurallarda kullanılacak sözel etiketler belirlenir.
3. Her fuzzy alt kümenin üyelik fonksiyonu belirlenir.
4. Sistemin kontrol edecek kurallar belirlenir.
5. Giriş ve çıkış normalize etmek için ölçekleme faktörü belirlenir.

6. FLK girişleri bulanıklaştırılır.
7. Fuzzy çıkarımı kullanarak girişlerin değerleri için her kuralın çıkış değeri bulunur ve elde edilen bu çıkışlar birleştirilir.
9. Sonuçtaki çıkışa netleştirme yöntemlerinden biri uygulanarak keskin çıkış elde edilir.



Şekil 5.1 Temel Fuzzy Lojik Kontrol sisteminin blok diyagramı.

5.2. Genel Fuzzy Lojik Kontrol

Temel tasarım maddeleri aşağıdaki gibidir, (Lee, 1990a).

1. Bulanıklaştırma metodları ve bulanıklaştırma işlemcisinin veya bulanıklaştırıcının ifadesi.
2. Veri tabanı;
 - a. Evrensel kümenin ayrıştırılması / normalize edilmesi.
 - b. Giriş ve çıkış uzayının bulanık bölümlere ayrılması .
 - c. Bölümlerin tamamlanması.
 - d. Temel bir bulanık kümenin üyelik fonksiyonunun seçimi.
3. Kural tabanı
 - a. İşlem giriş ve çıkış değişkenlerinin seçimi.
 - b. Fuzzy lojik kontrol kurallarının elde ediliş kaynağı.

- c.Fuzzy lojik kontrol kurallarının yazılması.
- d.Fuzzy lojik kontrol kurallarının içeriği, etkileşimi ve tamamlanması.
- 4. Sayısal Karar verme:
 - a.Bulanıklığın belirlenmesi.
 - b.Bağlayıcı AND ifadesinin yorumu.
 - c.Bağlayıcı OR ifadesinin yorumu.
 - d.Çıkarım mekanizması.
- 5.Netleştirme yöntemleri ve bir netleştirme işlemcisinin belirlenmesi,
(Netleştirici)



6. 8XC196KC/KD 16 BİT MİKRODENETLEYİCİ

8XC196KC ve 8XC196KD çok hızlı hesaplama ve giriş çıkış yapmak için tasarlanmış 16-bit CHMOS mikrodeneleyicidir, (Intel,1990,1991,1992). Aynı zamanda komutları MCS 96 ailesi ile uyumludur, (Intel, 1998a,1988b).

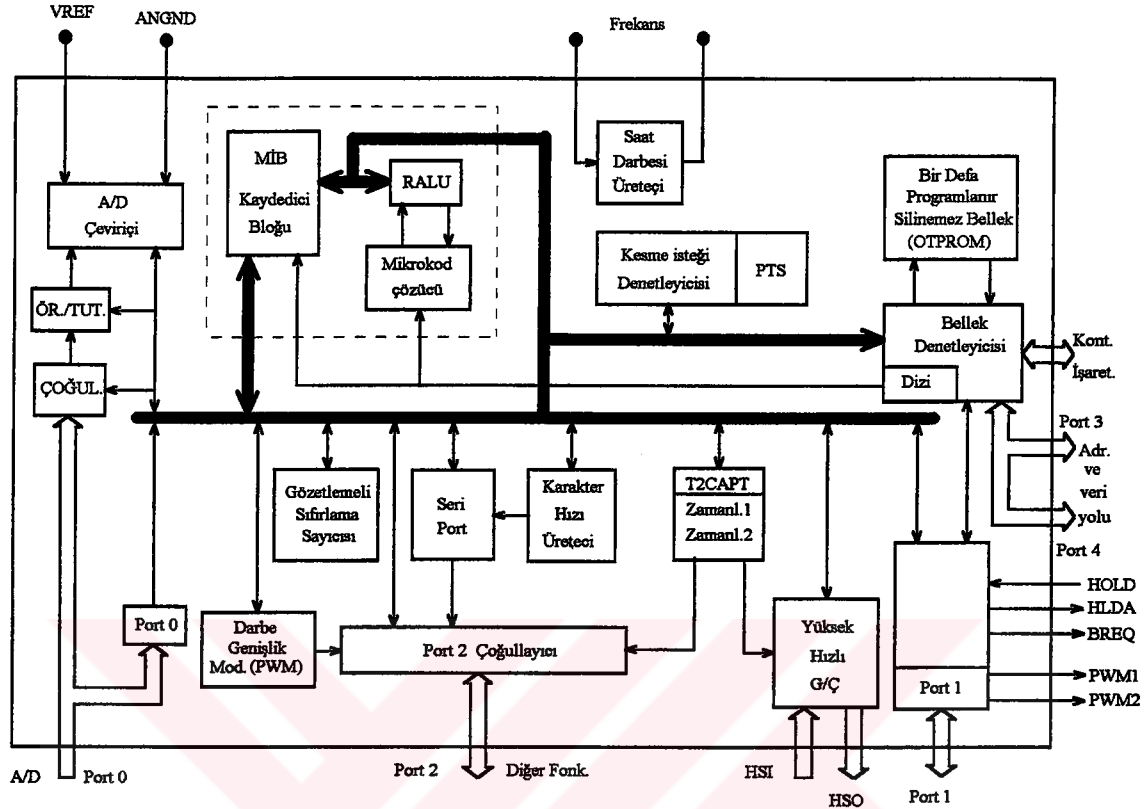
MCS 96 ailesi kapalı çevrim kontrolü, dijital sinyal işleme, modem, motor kontrol sistemi, yazıcılar, sıvı yakıtlı motor sistemi, fotokopi makinaları, klima kontrol sistemi, disk sürücüler ve tıbbi cihazları uygulamalarında kullanılır.

8XC196KC ve 8XC196KD mikrodeneleyicisinin iç yapısının blok diyagramı şekil 6.1 'de gösterilmiştir. Her çevre birimi 16 bitlik yapıya sahiptir. Kesme isteği ve bellek denetleyicisi 16 bitlik yol ile Merkezi İşlem Birimine (MİB) bağlanırlar. 8 bit MİB yolu, komutları bellek denetleyicisinden Aritmetik mantık birimi kaydedicisi içerisindeki komut kaydedicisine transfer eder. 8XC196KC/KD mikrodeneleyicisinin ana yapısında merkezi işlem birimi, bellek ve kesme isteği denetleyicisi vardır. Ayrıca merkezi işlem biriminde ise aritmetik mantık birimi/kaydedicisi (RALU) ve kaydedici bloğu vardır.

6.1. MİB denetleme:

Görevi 256 byte alt kaydedici bloğunda veya pencereleme ile dolaysız olarak erişilebilen üst kaydedici bloğunda bayt, kelime veya çift kelime kullanan işlemleri yerine getirmek için RALU 'ya komut vermek olan MİB, mikrokod çözücüsü tarafından kontrol edilir. MİB komutları bellek denetleyicisindeki dört-byte bellek dizisinden RALU komut kaydedicisine yükler. Mikrokod çözücüsü komutların anlamını çözer ve istenilen komutların sıra ile sonuçlarının yerine getirilmesini sağlar.





Şekil 6.1. 8XC196KC/KD mikrodnetleyicisinin iç yapısının blok diyagramı

6.2. Kaydedici bloğu:

Kaydedici bloğu üst ve alt blok olmak üzere ikiye ayrılır. Alt kaydedici bloğu özel fonksiyonlu kaydedicilerin 24 baytını ve genel amaçlı kaydedicilerin 232 baytını, üst kaydedici bloğu sadece genel amaçlı yazılır-silinir bellek (RAM) kaydedicilerini içerir. (Üst kaydedici bloğu 8XC196KC 'de 256 bayt ve 8XC196KD 'de 768 bayt). Genel amaçlı kaydedicilere bayt, kelime veya çift kelime olarak ulaşmak mümkündür. RALU üst ve alt kaydedici bloğuna farklı biçimde erişir. Alt kaydedici bloğuna direkt adresleme ile erişir (Bölüm 7). Üst kaydedici bloğuna sadece düşey pencerelemeye izin verildiğinde dolaysız-kaydedici adresleme türü ile ulaşır (Bölüm 8).

6.3. Aritmetik mantık birimi (RALU)

Aritmetik Mantık Birimi kaydedicisi 17 bitlik Aritmetik Mantık Birimini, Program Durum Kelimesini, Ana Program Sayıcısını, Denetleme Kaydedicisini, Mikrokod yürütücüsünü, Sabitler Kaydedicisini, 3 bit Seçici Kaydedicisini, Çevrim Sayıcısını ve üç tane geçici kaydediciyi içerir. Seçici kaydedici 3 bit ve diğer kaydediciler 16 veya 17 bit genişliğindedir.

Aritmetik Mantık Birimi kaydedicisi Sabitler Kaydedicisinde saklanan sabit sayılarla ilgili hesaplamalar yapmak için kullanılır. Aynı zamanda bu kaydedici bayt ve kelime olarak evrik alma, arttırma, azaltma, kaydediciyi öteleme, toplama, çıkarma, çarpma ve bölme işlemlerini yapar.

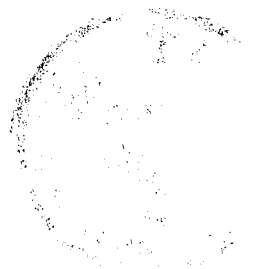
6.4. Bellek denetleyicisi

Bellek denetleyicisi, kaydedici bloğu dahil olmamak şartıyla Aritmetik Mantık Biriminin bellek ile haberleşmesini sağlar. Bellek denetleyicisi adres ve veri kaydedicisini, dört bayt bellek dizisini, Uydu Program Sayıcısını ve yol denetleyicisini içerir.

Yol denetleyicisi iç RAM, OTPROM (bir defa programlanabilir sadece okunur bellek) yolunu ve dış adres ve veri yolunu sürer.

6.5. Kesme isteği denetleyicisi

Programlanabilir kesme isteği denetleyicisi kullanıcının yazılımı tarafından ayarlanan öncelikli kesme yapısına sahiptir. Bu kesme istekleri ilgili kullanıcı tarafından yazılan alt programlarda yerine getirilir. 8XC196KC/KD Paralel İşlem Yardımcısını (PTS) içermektedir. PTS veri transferini, Analog dijital (A/D) çevrime başlama, Yüksek Hızlı Giriş modülünü okuma ve Yüksek Hızlı Çıkış modülüne sonuçları yükleme kesme isteklerini ayarlar.



6.6. Standart Giriş/Çıkış (G/Ç) portları

8XC196KC/KD beş tane 8 bit G/Ç portuna sahiptir. Bir kısmı sadece giriş, sadece çıkış, bir kısmı çift yönlü ve bir kısmında çok amaçlı işlemler için kullanılabilir. Port 0 giriş portu ve aynı zamanda A/D çevrim için analog giriş işleminin yapılmasını sağlar. Port 1 yarı çift yönlü porttur. Port 1 'in uçları Darbe Genişlik Modülasyonunun iki çıkışı ve yol denetleme sinyali ile paylaşmalı olarak çalışır. Port 2 yarı çift yönlü, giriş ve çıkış olmak üzere üç çeşit port uçlarını içerir. Port 2 giriş ve çıkış için aynı uçlarını ortaklaşa olarak kullanır. Port 3 ve 4 açık-drain çift yönlü port uçları ve adres/veri yolu olmak üzere paylaşmalı bir yapıya sahiptir.

6.7. Seri G/Ç portu

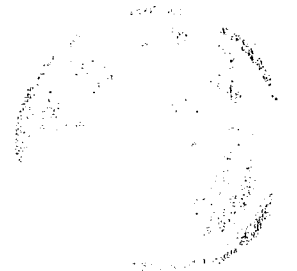
Seri G/Ç portu genel senkron olmayan alıcı, vericiyi (UART) içeren asenkron ve senkron olan bir porttur. UART bir senkron (Mod 0) ve üç asenkron moda (Mod 1, 2 ve 3) sahiptir. Senkron olmayan modda gönderme ve alma işlemini aynı zamanda gerçekleştirebilir.

6.8. Yüksek Hızlı Giriş / Çıkış ünitesi (HSIO)

HSIO ünitesi dört özel modülü içerir. Bunlar zamanlayıcı 1, Zamanlayıcı 2, Yüksek Hızlı Giriş ve Yüksek Hızlı Çıkış modülleridir. Bu modüller çeşitli şekillerde kullanılabilen sayıcı/zamanlayıcı, G/Ç sistemi şeklindedir.

6.8.1. Zamanlayıcı 1 ve Zamanlayıcı 2

Zamanlayıcı 1 her 8 durum zamanında bir artan, serbest çalışan bir zamanlayıcıdır. Yüksek Hızlı Giriş modülü için periyot ölçme ve Yüksek Hızlı Çıkış modülü için isteğe bağlı olarak zaman ölçümü ile ilgili çeşitli ölçmeler yapılabilir.



Zamanlayıcı 2 çıkan veya düşen kenarın durumuna göre sayma işlemini gerçekleştirebilir. Yüksek Hızlı Çıkış modülü için periyot ölçme, yukarı / aşağı sayıcı veya ilave bir sayıcı olarak kullanılabilir.

6.8.2. Yüksek Hızlı Giriş (HSI)

HSI modülü sekiz durum zamanı hassiyetinde dış sonuç zamanlarını kaydedebilir. HSI modülü dört bağımsız şekilde şartlanabilir girişleri denetleyebilir ve sonuç olduğu zaman zamanlayıcı 1 'deki değeri tutabilir. Sonuç çıkan kenar, düşen kenar, yükselen veya düşen kenar ve her sekiz periyotta çıkan kenar olmak üzere dört çeşittir. HSI modülü sekiz tane sonucu (zamanlayıcı 1 değerlerini) kaydedebilir.

6.8.3. Yüksek Hızlı Çıkış (HSO)

HSO modülü zamanlayıcı 1 veya zamanlayıcı 2 'de özel olarak bölünen zaman aralıkları sonuçlarını ölçmek için tetikleme yapabilir. Bu programlanabilir sonuçlar A/D çevirmeye başlamayı, zamanlayıcı 2 'yi sıfırlamayı (reset), dört tane zamanlayıcısını oluşturmayı ve 6 tane HSO çıkışlarından birini veya fazlasını lojik 0 veya lojik 1 yapmayı içerir. HSO ünitesi Öz Adreslenebilir Bellek (CAM) bloğunda belirlenen zamanları ve bir sonuca kadar olan değerleri depolama özelliğine sahiptir. Her komut zamanlayıcı 1 veya zamanlayıcı 2 'nin referans zaman değerine geldiğinde ve kesintinin oluşmasına yani olayın referans zamanı eşitliğine göre olayın zamanını belirler.

6.9. Analog Dijital (A/D) Çevirici

Analog Dijital Çevirici analog giriş gerilimini sayısal eşdeğerine çevirir. Hassasiyet 8 veya 10 bittir. Örnekleme veya çevirim zamanı ayarlanabilir. Kendiliğinden A/D çevirimi sağlama ve sonucu saklama PTS 'nin A/D modülü tarafından kolaylaştırılır. A/D çeviricinin ana

bölümleri örnekleme ve tutma devresi, 8 kanal çoğullayıcı ve 8 veya 10 bit Ardışıl yaklaşımlı Analog Dijital çeviriciden oluşur.

6.10. Darbe Genişlik Modülatörü (PWM)

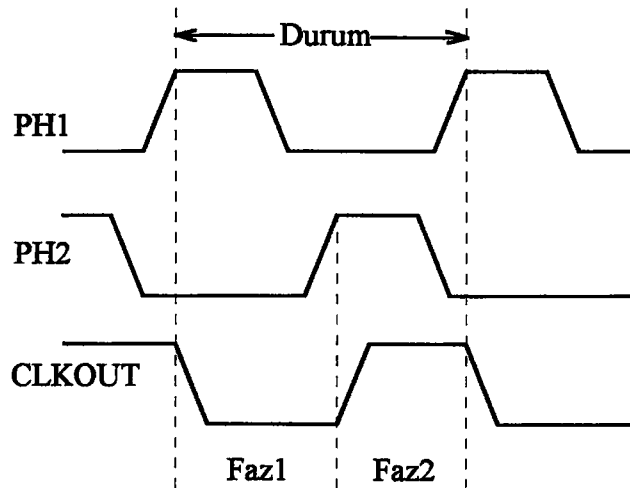
8XC196KC/KD üç PWM modülüne sahiptir. Her birinin çıkış dalga şekli, programlanmış olarak her 256 veya 512 durum zamanında oluşan değişken amaçlı periyot darbesidir. Elektrik motorlarını etkili bir şekilde denetlemek için PWM dalga şekli kullanılır. PWM dalga işareti süzgeç devresinden geçirilirse periyodun değişmesi ile 256 kademeli doğru gerilim seviyesi elde edilir.

6.11. Gözetlemeli sıfırlama sayıcısı

Bu sayıcı yazılım istenildiği gibi çalışmaz ise mikrodenetleyicinin bir müddet sonra sıfırlanmasını sağlar.

6.12. İç zamanlama

Saat darbesi üretici XTAL1 'deki sinyalin frekansını ikiye böler ve PH1 ve PH2 gibi iki iç zamanlama sinyalini üretir. Bu sinyaller lojik 1 oldukları zaman aktif olurlar. PH1 ve PH2 'nin yükselen kenarı ile iç saat darbesi üreticinin çıkışı olan CLKOUT sinyali üretilir.



Şekil 6.2. İç saat darbe fazları

CLKOUT 'a IOC3 kaydedicisindeki bit 1 ile izin verilmeyebilir. Bu bit lojik 1 olursa CLKOUT sinyali önlenir. Bu şekilde sistemin gürültüden etkilenmesi azaltılabilir.

7. 8XC196 KC/KD ADRESLEME TÜRLERİ:

8XC196KC/KD 'nin altı tane adresleme çeşidi vardır. Bunlar;

- Dolaysız-Kaydedici adresleme
- Dolaylı adresleme
- Kendinden artmalı dolaylı adresleme
- Anında adresleme
- Kısa indekslemeli adresleme
- Uzun indekslemeli adresleme

Ayrıca Sıfır Kaydedici Adresleme ve Yığın Gösteren Kaydedici adresleme türleride vardır. Sıfır Kaydedici adresleme uzun indekslemeli adresleme ile ZERO_REG kaydedicisinin bir araya gelmesiyle oluşur. Aynı zamanda bellek bölgesinde herhangi bir bellek gözüne dolaysız olarak ulaşmayı sağlar. Yığın Gösteren Kaydedici adresleme türü yığının tepesindeki veriye ulaşmak için dolaylı adresleme ile Yığın Gösteren Kaydedicinin bir araya gelmesi ile oluşur. Ayrıca yığın içerisindeki bir veriye ulaşmak için kısa indekslemeli adresleme ile yine Yığın Gösteren Kaydedicinin bir araya gelmesi durumunda da oluşur.

7.1. Dolaysız-Kaydedici adresleme

Dolaysız-Kaydedici adresleme alt kaydedici bloğundaki 256 bayt uzunluğundaki bir bölgede bir kaydediciye dolaysız olarak ulaşmak için kullanılır. Bu adresleme türü pencereleme ile üst kaydedici bloğuna dolaysız olarak ulaşabilir. Bu adreslemede kullanılacak

kaydediciler programın başında 8 veya 16 bit olarak belirlenirler. Bu adreslemede sadece kaydedicilerle ilgili kaydediciler arası işlem yapılır. Örneğin;

```
ADD  AX, BX, CX    ; AX ← BX + CX
MUL  AX, BX        ; AX ← AX * BX
INCB BL            ; CL ← CL + 1
```

Bu örnekte ve bundan sonraki örneklerde kullanılan AX, BX ve CX kaydedicileri 16 bit uzunluğundadır. CL kaydedicisi CX kaydedicisinin hafif ağırlıklı baytı ve 8 bit uzunluğundadır.

7.2. Dolaylı adresleme

Bu adresleme türünde kullanılan 16 bitlik kaydedicideki değer bir adresi gösterir. İlgili işlem bu adresin gösterdiği bellek gözü ile yapılır. Dolaylı adreslemede adresi gösteren kaydedici 16 bit olması gerekmektedir. Örneğin;

```
LD    AX, [AX]      ; AX ← MEM_WORD (AX)
ADDB AL, BL, [CX]   ; AL ← BL + MEM_BYTE (CX)
POP   [AX]          ; MEM_WORD (AX) ← MEM_WORD (SP)
                        ; SP ← SP + 2
```

7.3. Kendinden artmalı dolaylı adresleme

Dolaylı adresleme türüne benzer. Fakat adresi gösteren 16 bitlik kaydedici işlem sonucunda artar. Eğer kaydedicinin gösterdiği adresteki veri ile 8 bitlik işlem yapılırsa bu kaydedici bir, 16 bitlik işlem yapılırsa aynı kaydedici iki artar. Örneğin;

```
LD    AX, [BX] +    ; AX ← MEM_WORD (BX)
                        ; BX ← BX + 2
```



```

ADDB AL, BL, [CX] +      ; AL ← BL+MEM_BYTE (CX)
                          ; CX ← CX +1
PUSH [AX] +              ;MEM_WORD(SP) ←MEM_WORD (AX)
                          ; AX ← AX + 2

```

7.4.Anında adresleme

Bu adreslemede bilginin alındığı kaynak sabit bir değeri ifade eder. Komutun içerisinde işlem yapılan sayı değeri görünür. Bu değer işlemin durumuna göre 8 bit veya 16 bit olabilir. Örneğin;

```

ADD  AX, #340H           ; AX ← AX + 340H
PUSH #1234H              ; SP ← SP - 2
                          ; MEM_WORD (SP) ← 1234H
DIVB AX, #10H            ; AL ← AX / 10H
                          ; AH ← AX MOD 10H

```

7.5.Kısa indekslemeli adresleme

16 bitlik bir kaydedicideki değer ile 8 bitlik sabit bir değer toplanır ve oluşan yeni değer bir adresi gösterir. Sonuçta bu adresteki veri ile ilgili işlem yapılır. 8 bitlik sabit değer komut içerisinde görülür ve işaretli bir sayıdır. Dolayısıyla temel alınan değer 127 bayt sonrası ve 128 bayt öncesi için adresleme yapılabilir.Örneğin;

```

LD    AX, 12 [BX]        ; AX ← MEM_WORD (BX +12)
MULB  AX, BL, 3 [CX]     ; AX ← BL * MEM_BYTE (CX + 3)

```

7.6. Uzun indekslemeli adresleme

Kısa indekslemeli adresleme türüne benzerdir. Herhangi bir komutta adres oluşturulurken 16 bitlik kaydedicideki değer ile 16 bitlik sabit bir değer toplanır. Bu durumdan dolayı 16 bitlik

kaydedicide temel alınan değerin 65535 bayt yukarısı
adreslenebilir.Örneğin;

AND	AX, BX, TABLE [CX]	;AX←BX AND MEM_WORD(TABLE+CX)
ST	AX, TABLE [BX]	; MEM_WORD (TABLE + BX) ← AX
ADDB	AL, BL, LOOKUP [CX]	;AL←BL +MEM_BYTE(LOOKUP+ CX)

7.7. Sıfır kaydedici adresleme

Kaydedici bloğunun ilk iki baytı Sıfır kaydedicisini (ZERO_REG) gösterir. Bu ilk bayt 8XC196 KC/KD donanımı vasıtasıyla daima sıfırlanır. Programda değeri sıfır ile ilgili olan hesaplamalarda ve karşılaştırmalarda kolaylık sağlar. Sıfır kaydedicisi uzun indekslemeli adreslemede 16 bitlik temel adresi gösteren kaydedici olarak alınabilir. Bu kaydedicinin sürekli sıfır olması sebebiyle bellekte herhangi bir gözü adresleyebilir. Örneğin;

```
ADD    AX, 1234H [ZERO_REG]    ;AX←AX +MEM_WORD (1234H)
POP     5678H[ZERO_REG]        ;MEM_WORD(5678H)←MEM_WORD(SP)
                                ; SP ← SP + 2
```

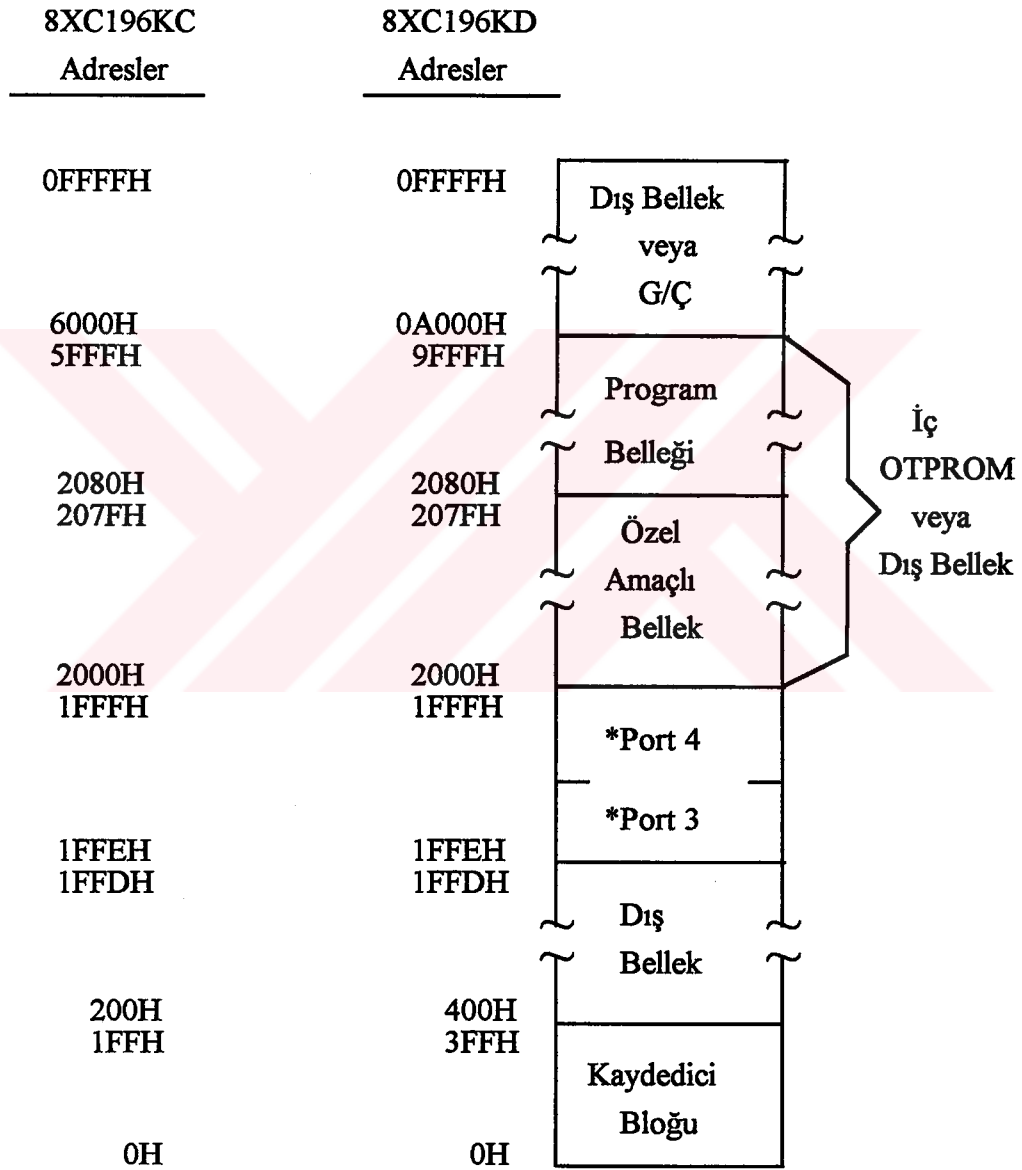
7.8. Yığın Gösteren Kaydedici adresleme

Alt kaydedici bloğunda 18H ve 19H adreslerindeki baytlar Yığın Gösteren Kaydedici için ayrılmıştır. Yığın Gösteren Kaydedici yığın adresinin en üstünü gösteren 16 bitlik bir adres olarak kullanılır. Bu adreslemeye örnek;

PUSH [SP]	; Yığının tepesindeki bayt bir defa daha yığının tepesine kopyalanıyor.
LD AX, 2[SP]	; Yığının tepesinden iki fazla olan adresteki veri AX 'e yükleniyor

8. 8XC196KC/KD BELLEK BÖLGELERİ

Bu bölümde 8XC196KC ve 8XC196KD için adreslenebilir bellek bölgeleri tanımlanmaktadır. Her iki denetleyici 64 KBayt bellek bölgesine sahiptir. Bu bölgelerin büyük bir bölümü program ve.



* Port 3 ve Port 4 beraber kullanılır.

Şekil 8.1. 8XC196KC ve 8XC196KD için bellek bölgeleri

veri bölgeleri olarak kullanılır. Her bellek gözü bir bayttır. Şekil 8.1 gerekli bellek bölgelerini göstermektedir. 8XC196KC ve 8XC196KD 'nin bellek bölgeleri birbirinden farklıdır

8.1. Dış bellek bölgesi

Kaydedici bloğunun yukarı adres bölgeleri dış bellek veya giriş/çıkış olarak ayrılır (Şekil 8.1). 8XC196KC/KD çeşitli dış bellek ünitelerine bağlanabilir. Bu durum bölüm 10 'da anlatılmaktadır.

8.2. Port 3 ve Port 4

16 bit uzunluğundaki bir kelimenin alt baytı port 3'e ve üst baytı port 4'e dahil olmak üzere bu portlardan okuma ve yazma kelime olarak yapılır. Bu portlar giriş/çıkış veya sistemin adres/veri yolu olarak, aynı zamanda Dış Erişim (EA#) bacağına göre her ikisi beraber olarak kullanılabilir. Eğer EA# bacağı lojik 0 ise bu portlar Adres/Veri yolu olarak kullanılır.

8.3. Program ve özel amaçlı bellek

Bu bölge bellek bölgesinde 8XC196KC için 16 KBaytlık ve 8XC196KD için 32 Kbaytlık yer tutar (Şekil 8.1). Bu bölgeler iç OTPROM veya dış bellek olarak ifade edilir.

8.4. İç veya dış bellek bölgesini seçme

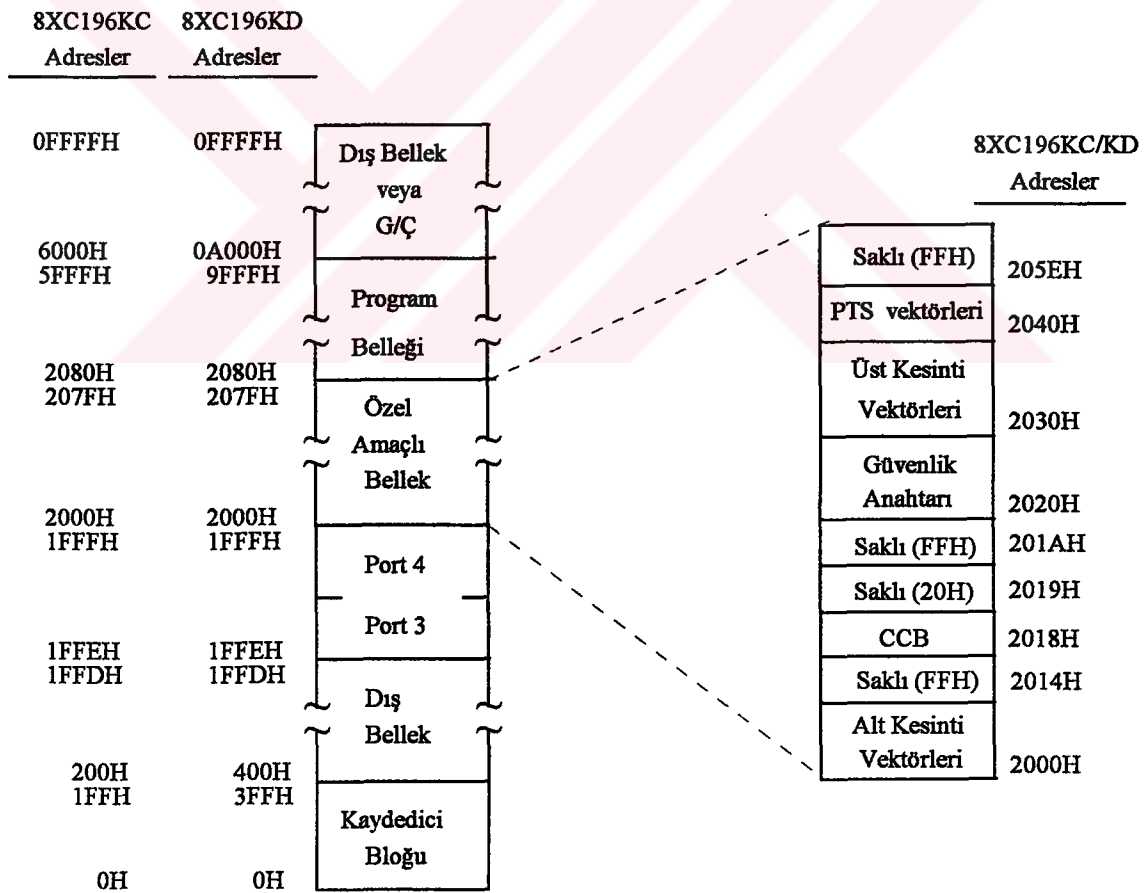
Program veya özel bellek bölgelerine ulaşmak iç OTPROM veya dış bellek olmak üzere iki değişik şekilde olur. Dış veya iç belleği kullanmak EA# bacağının değeri ile belirlenir. Bu değer denetleyici tarafından sıfırlama sinyalinin çıkan kenarı ile görülür ve sistem bir daha sıfırlanıncaya kadar tutulur. Eğer EA# lojik 0 ise bu adres aralığında dış belleğe, eğer EA# lojik 1 ise iç OTPROM 'a ulaşılır.

8.5. Program Belleği

Program belleği 8XC196KC 'de 2080H-5FFFH ve 8XC196 KD'de 2080H-9FFFH adresleri arasındaki bölgededir. Denetleyici sıfırlanınca 2080H bellek gözündeki komutu alır ve yürürlüğe koyar.

8.6. Özel amaçlı bellek

Özel amaçlı bellek 2000H-207FH adresleri arasındaki bölgedir (Şekil 8.2). Bu bölge saklı tutulan bellek gözlerini, standart kesintiler ve Paralel İşlem Yardımcısı (PTS) için vektörleri, güvenlik anahtarını ve Tümdevreyi Ayarlama Baytını (CCB) içerir.



Şekil 8.2. 8XC196KC/KD Özel amaçlı bellek haritası

8.6.1. Saklı tutulan bellek gözleri

Bu tür bellek gözleri geliştirilecek yeni ürünler veya test etme işlemi için saklanır. Kullanıcının yapacağı program, saklı tutulan bellek gözlerini okuma ve bu bellek gözlerine yazma işlemini yapmaması gerekir.

8.6.2. Kesme İsteği vektörleri

Üst ve alt kesme isteği vektörleri kesme isteği servis alt programlarının adreslerini içerir. Paralel İşlem Yardımcısı (PTS) vektörleri PTS denetleme bloklarının adreslerini içerir.

8.6.3. Güvenlik anahtarı

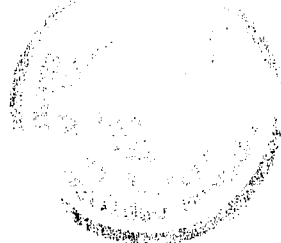
Güvenlik anahtarı OTPROM 'u programı yazanların dışındaki kimsenin programı okumasını önlemek için kullanılır. Eğer Tümdevre Ayarlama Kaydedicisindeki (CCR) güvenlik kilitleme bitleri (CCR.6 ve CCR.7) sıfırlanırsa, 16 bayt olarak programlanabilen güvenlik kodlarının her biri dış bellekte karşılık gelen baytlar ile karşılaştırılır. Eğer dış veri ile güvenlik anahtarı birbirinin aynısı değil ise denetleyici istenilen program moduna giriş yapmaz.

8.6.4. Tümdevre Ayarlama Baytı (CCR)

Bu bayt sıfırlama işleminden önce bellekten alınan bir bayttır. Bu işlemde sonra sıfırlama işlemi gerçekleşir. CCR iç belleğin okunmasını önlemeyi, iç READY modunu, yol denetleme sinyallerini, yol genişliğini seçme işlemini ve düşük güç harcama modunu denetler.

8.7. Kaydedici bloğu

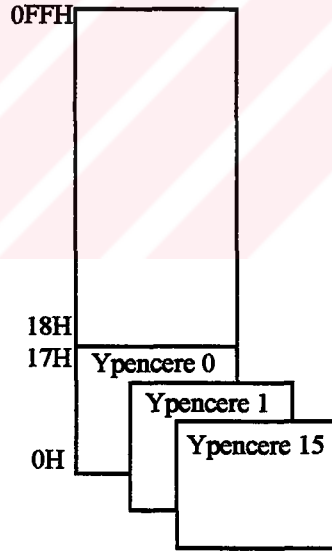
Kaydedici bloğu üst ve alt kaydedici bloğu olmak üzere ikiye ayrılır (Şekil 8.3) Üst kaydedici bloğu genel amaçlı kaydedici RAM 'ı içerir. Alt kaydedici bloğu genel amaçlı kaydedici RAM 'ı, Yığın



önceden bilinmesi gerekir. Sonuçlar SFR 'nin içeriğini değiştirebilir ve bazı SFR 'ler okunduktan sonra sıfırlanabilir.

8.8. Yatay pencereleme (Ypencere)

8XC196KC/KD alt kaydedici bloğunun alt 24 baytının içerisinde değişik üç tane 24 bayt bellek bloğu olan (Ypencere 0, 1 ve 15) yatay pencereleme düzenini kullanır (Şekil .8.4). Her Ypencere SFR 'nin değişik durumlarına ulaşp okumayı ve yazmayı sağlar. Bir kısım kaydedicilere bir bayt olarak ve diğerlerine ise kelime olarak ulaşılır. Pencere Seçme Kaydedicisi (WSR) gibi bazı kaydedicilere yatay pencerelerin hepsinde ulaşmak mümkün olur. Diğer kaydedicilere yatay pencerenin birinde yazmak mümkün olduğu gibi diğer yatay pencerede okumak mümkündür.



Şekil 8.4. Yatay pencereleme (Ypencere)

8.8.1. Ypencere seçme işlemi

Pencere Seçme Kaydedicisi (WSR, 14H) yatay ve dişey pencelere ulaşmayı sağlar. Bir yatay pencereyi seçmek için istenen pencerenin numarası WSR.0-WSR. 3 bitlerine yazılır. Bu durumda

WSR.4-WSR.6 bitleri lojik sıfır olmalıdır. Tablo 8.1 'de her bir pencerenin nasıl seçileceği gösterilmiştir.

Tablo 8.1. Yatay pencere seçimi

Ypencere	WSR içeriği
0	X000 0000B= 00H
1	X000 0001B= 01H
15	X000 1111B= 0FH

8.8.2. Ypencere 0

Ypencere 0 şartlanma yapılmadan önceki normal durumdaki penceredir. Bu pencere okumada ondokuz kaydediciye ve yazmada yirmibir kaydediciye ulaşmayı sağlar (Şekil 8.5). Bazı kaydediciler (Örn: INT_MASK 1) Ypencere 0 'da okunabilir ve aynı pencerede yazılabilir türdendir. Diğer kaydediciler ise Ypencere 0'da okunabilir fakat aynı pencereye yazma durumunda başka bir kaydedici karşıya çıkar (Örn: IOS1). Bu çeşit kaydediciler Ypencere 0'da okunabilir ve aynı tür kaydedicilere Ypencere 15 'de bilgi yazılabilir.

8.8.3. Ypencere 1

Ypencere 1 okuma ve yazma işleminde 12 kaydediciye ulaşmayı sağlar. Aynı zamanda bu penceredeki bazı kaydedicilere Ypencere 0 ve Ypencere 15 'de erişmek mümkündür (Şekil 8.5, Şekil 8.6).

8.8.4. Ypencere 15

Ypencere 15 0CH-10H arasındaki adresler hariç Ypencere 0'daki benzer kaydedicilere ulaşmayı sağlar (Şekil 8.6). 0CH ve 0DH adresleri arasındaki baytlar Ypencere 0 'da 16 bit zamanlayıcı 2 kaydedicisi fakat Ypencere15'de 16 bit T2CAPTURE kaydedicisini gösterir..



Ypencere 0 (Oku)		Ypencere 0 (Yaz)		Ypencere 1 (Oku/Yaz)	
17H	IOS2		PWM0_CONTROL	17H	PWM2_CONTROL
16H	IOS1		IOC1	16H	PWM1_CONTROL
15H	IOS0		IOC0	15H	Saklı
14H	WSR		WSR	14H	WSR
13H	INT_MASK1		INT_MASK1	13H	INT_MASK1
12H	INT_PEND1		INT_PEND1	12H	INT_PEND1
11H	SP_STAT		SP_CON	11H	Saklı
10H	IOPORT2		IOPORT2	10H	Saklı
0FH	IOPORT1		IOPORT1	0FH	Saklı
0EH	IOPORT0		BAUD_RATE	0EH	Saklı
0DH	TIMER2(ÜST B.)		TIMER2(ÜST B.)	0DH	Saklı
0CH	TIMER2(ALT B.)		TIMER2(ALT B.)	0CH	IOC3
0BH	TIMER1(ÜST B.)		IOC2	0BH	Saklı
0AH	TIMER1(ALT B.)		WATCHDOG	0AH	Saklı
09H	INT_PEND		INT_PEND	09H	INT_PEND
08H	INT_MASK		INT_MASK	08H	INT_MASK
07H	SBUF(RX)		SBUF(TX)	07H	PTSSRV(ÜST B.)
06H	HSI_STATUS		HSO_COMMAND	06H	PTSSRV(ALT B.)
05H	HSI_TIME(ÜST B.)		HSO_TIME (ÜST B.)	05H	PTSSEL(ÜST B.)
04H	HSI_TIME(ALT B.)		HSO_TIME (ALT B.)	04H	PTSSEL(ALT B.)
03H	AD_RESULT(ÜST B.)		HSI_MODE	03H	AD_TIME
02H	AD_RESULT(ALT B.)		AD_COMMAND	02H	Saklı
01H	ZERO_REG(ÜST B.)		ZERO_REG(ÜST B.)	01H	ZERO_REG(ÜST B.)
00H	ZERO_REG(ALT B.)		ZERO_REG(ALT B.)	00H	ZERO_REG(ALT B.)

Şekil 8.5. Ypencere 0, 1

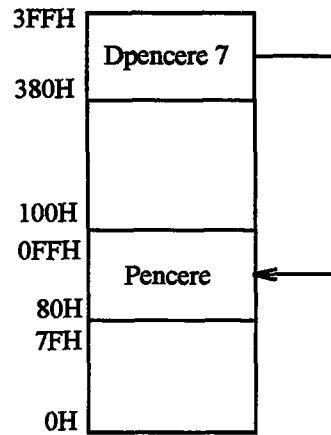
0EH ve 10H adresleri arasındaki baytlar Ypencere 0 'da IOPORT0, IOPORT1 ve IOPORT2 kaydedicilerdir. Fakat aynı baytlar Ypencere 15 'de saklı tutulur. Ypencere 15 'de yazma işlemi yapılan kaydediciler, Ypencere 0 'da okunabilirler.

8.9. Düşey pencereleme (Dpencere)

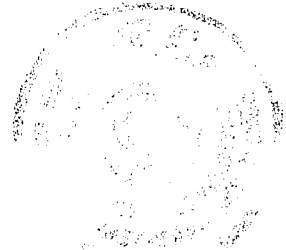
Düşey pencereleme alt kaydedici dosyasını 32, 64, 128 baytlık bloklar halinde üst kaydedici bloğuna haritalandırma işlemidir.

	Ypencere 15 (Oku)	Ypencere 15 (Yaz)
17H	PWM0_CONTROL	IOS2
16H	IOC1	IOS1
15H	IOC0	IOS0
14H	WSR	WSR
13H	INT_MASK1	INT_MASK1
12H	INT_PEND1	INT_PEND1
11H	SP_CON	SP_STAT
10H	Saklı	Saklı
0FH	Saklı	Saklı
0EH	Saklı	Saklı
0DH	T2CAPTURE(ÜST B.)	T2CAPTURE(ÜST B.)
0CH	T2CAPTURE(ALT B.)	T2CAPTURE(ALT B.)
0BH	IOC2	TIMER1(ÜST B.)
0AH	WATCHDOG	TIMER1(ALT B.)
09H	INT_PEND	INT_PEND
08H	INT_MASK	INT_MASK
07H	SBUF(TX)	SBUF(RX)
06H	HSO_COMMAND	HSI_STATUS
05H	HSO_TIME(ÜST B.)	HSI_TIME (ÜST B.)
04H	HSO_TIME (ALT B.)	HSI_TIME(ALT B.)
03H	HSI_MODE	AD_RESULT(ÜST B.)
02H	AD_COMMAND	AD_RESULT(ALT B.)
01H	ZERO_REG(ÜST B.)	ZERO_REG(ÜST B.)
00H	ZERO_REG(ALT B.)	ZERO_REG(ALT B.)

Şekil 8-6. Ypencere 15



Şekil 8.7. Yatay pencereleme



8XC196KC 16 çeşit 32 bayt, 8 çeşit 64 bayt ve 4 çeşit 128 bayt düşey pencereleme yapılabilir. 8XC196KD ise 8XC196KC 'nin sahip olduğu düşey pencerelemenin iki katına sahiptir. Şekil 8.7. düşey pencerelemeye ait bir örneği gösterir

8.9.1. Dpencere seçme işlemi

Pencere seçme kaydedicisi (WSR, 14H) yatay pencerelere veya düşey pencelere ulaşmayı sağlar. WSR.4, WSR.5 veya WSR.6 bitlerinden herhangi birisinin lojik 1 olması şartıyla sırayla 128 , 64 ve 32 bayt Dpenceleri seçilir(Şekil 8.7). Bu durumda WSR 'nin alt bitlerine (WSR.3, WSR.2, WSR.1ve WSR.0) Dpencerenin numarası yazılır.

9. SERİ GİRİŞ/ÇIKIŞ PORTU

8XC196KC/KD 'de seri Giriş/Çıkış portu genel asenkron alıcı ve vericiyi (UART) içeren asenkron/senkron bir porttur. UART 'ın alma ve gönderme işlemi yapabilen senkron modu (Mode 0) ve üç asenkron modu (Mode 1, 2 ve 3) vardır. UART 'ın şartlandırılması ve kullanılması için bazı kaydediciler vardır. Bu kaydediciler daha sonra detaylı olarak anlatılacaktır.

SP_CON kaydedicisi haberleşme modunu seçmeyi, alma işlemine izin verip vermemeyi, Çift Eşitlik bitini denetlemeyi ve dokuz bit veri göndermeyi sağlar. SP_STAT kaydedicisi Alma İşlemi Kesme İsteği bayrağını (RI) ve Gönderme İşlemi Kesme İsteği bayrağını (TI) ve diğer durum bitlerini içerir. SP_STAT kaydedicisi okunduktan sonra TXE biti hariç diğer bütün bayraklar sıfırlanır. Bu yüzden SP_STAT kaydedicisinin içeriği bir görüntü kaydedicisine kopyalanır. Daha sonra görüntü kaydedicisinde JBC ve JBS bit-test komutları yürütülür.

Alınan veri SBUF(RX) kaydedicisine yazılır. Aynı zamanda gönderilen veride aynı adreste bulunan SBUF(TX) kaydedicisine

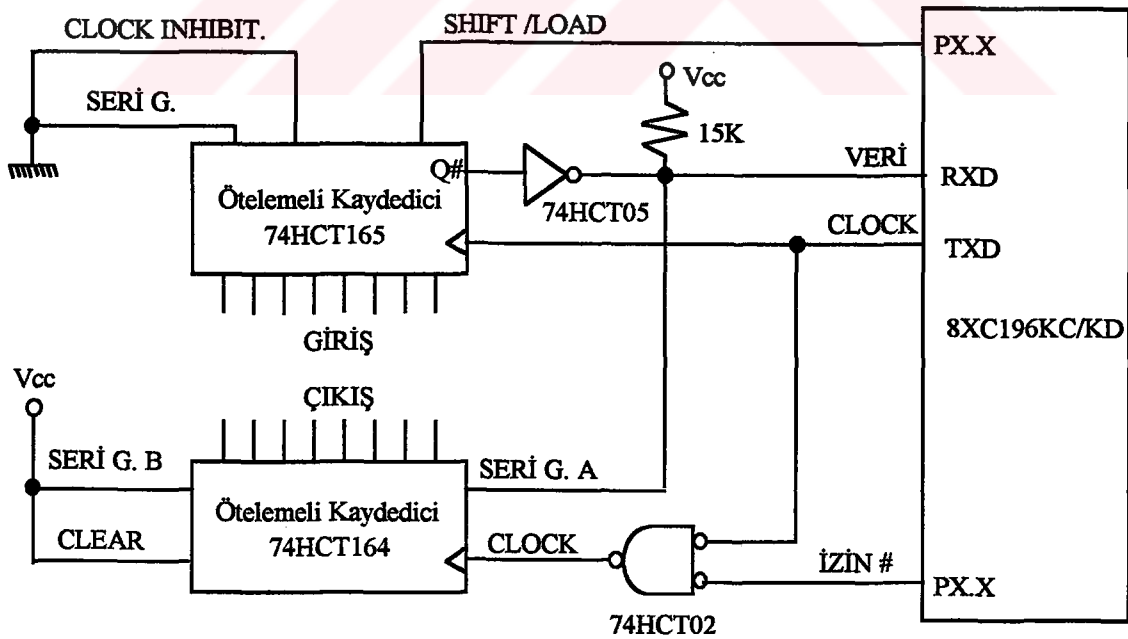
yazılır. Bu iki kaydedici Ypencere 0 'da bulunurlar. Gönderici ve alıcı devamlı olarak göndermeyi ve alınan birinci bayt okunmadan sonraki ikinci baytın alınmasını sağlayabilir.

Seri port kesme isteklerine, Kesme İsteği Önleme Kaydedicisi (INT_MASK veya INT_MASK1) vasıtasıyla izin vermek veya izin vermemek mümkün olur. Beklemekte olan kesme istekleri, Kesme İstekleri Bekletme Kaydedicisi (INT_PEND veya INT_PEND1) tarafından gösterilir.

Bağımsız Karakter Hızı Üretici seri portun alma veya göndermede saniyedeki karakter hızını denetler. XTAL1 veya T2CLK'tan her birisi saat darbe işaretini sağlar. BAUD_RATE kaydedicisi saniyedeki karakter hızını ve saat darbesi kaynağını seçer.

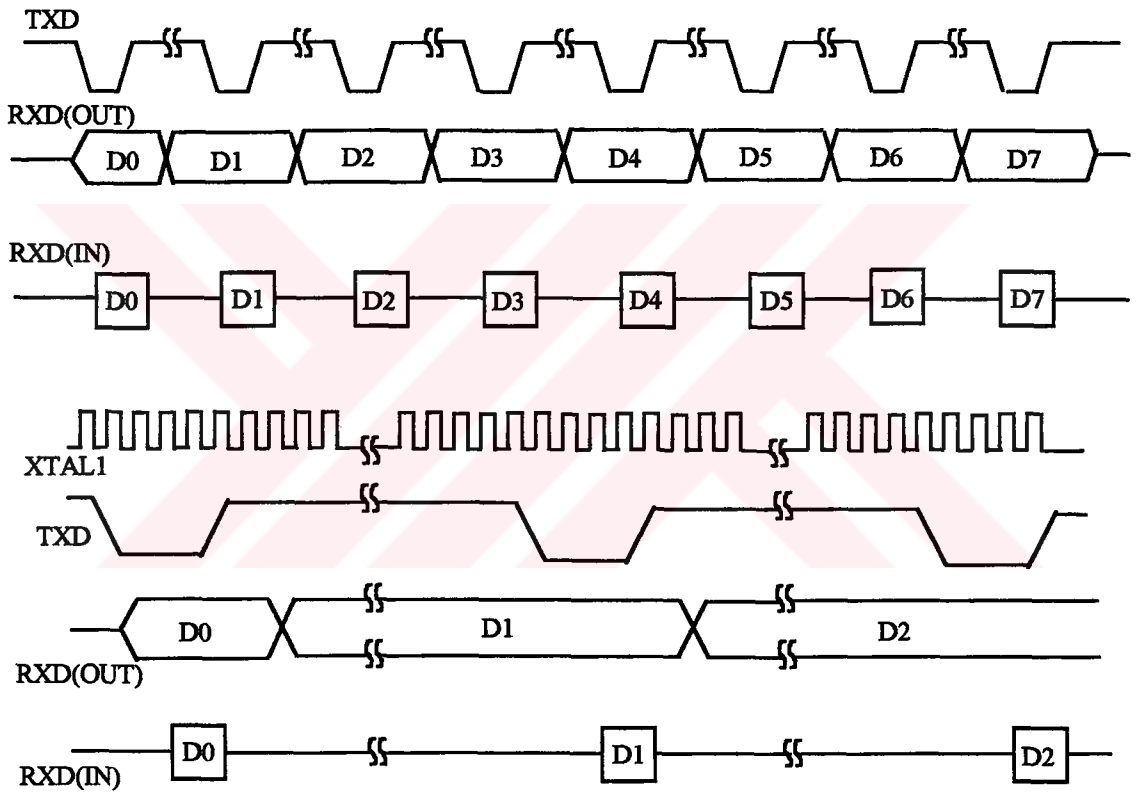
9.1. Mod 0

Senkron mod olan Mod 0 'ın çok yaygın kullanımı, ötelemeli kaydediciler kullanarak 8XC196 KC/KD 'nin Giriş / Çıkış özelliğinin



Şekil 9.1. Mode 0 için ötelemeli kaydedici devresi

geliştirilmesidir (Şekil 9.1). RXD ucu gönderme ve alma işlemlerinin her ikisini sağlarken TXD sekiz tane saat darbesi göndermeye şartlandırılır. Verinin önce en hafif biti gönderilir. Şekil 9.2. Mod 0'da gönderme ve alma işlemi gerçekleşirken oluşan zamanlama işaretlerini gösterir.



Şekil 9. 2. Mod 0 için zamanlama işaretleri

Mod 0'da RXD 'ye gönderme işlemi sırasında izin verilmemeli ve alma işlemi sırasında izin verilmelidir. RXD 'ye izin verme veya izin vermeme seri portun şartlamasını yapan kaydediciler tarafından sağlanır. RXD 'ye izin verilmediği zaman SBUT(TX)'e veri yazıldığında gönderme işlemi başlar. Eğer RXD 'ye izin verilirse RXD 'nin yükselen kenarı veya alma işlemi kesintisi (RI) bayrağının sıfırlanması ile alma işlemi başlar.

RXD 'ye izin verilmemesiyle devam eden alma işlemi durur ve sonraki alma işlemi yasaklanır. İstenmeyen veya yarım bilgi alma durumunu ortadan kaldırmak için RI bayrağı sıfırlanmadan önce RXD 'ye izin verilmemelidir.

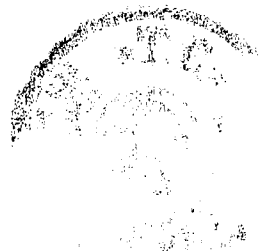
9.2. Asenkron modlar

Mod 1,2 ve 3 çift yönlü seri gönderme/alma işlemini gerçekleştiren modlardır. Bu modların aynı anda bilgi gönderebilmesi ve bilgi alabilmesi mümkündür. Mod 1 standart 8 bitlik veri için normal seri haberleşme için kullanılan asenkron moddur. Mod 2 ve 3 ise mikro işlemciler arası 9 bit haberleşme için kullanılır. Mod 2 'de sadece 9.bit lojik 1 olursa seri port kesme isteği üretir. Mod 3 'de seri port daima verinin alınması veya gönderilmesinin tamamlanması durumunda kesme isteği üretir.

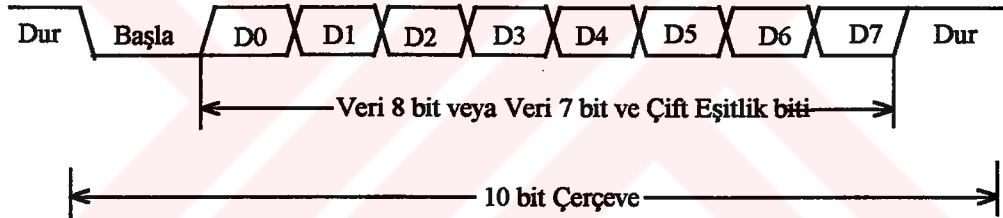
Seri port Mod 1, 2 veya 3 için ayarlandığı zaman SBUF(TX) 'e bilginin yazılması seri portun veri gönderme işlemine başlamasına neden olur. SBUF(TX)'e yeni bir veri yazılırsa Durma Biti gönderilinceye kadar yeni veri gönderilmez. RXD 'ye izin verilirse RXD giriş işaretinin düşen kenarı seri portun veri alma işlemine başlamasına neden olur. RXD 'ye izin verilmemesi durumunda devam eden alma işlemi durur ve sonraki alma işlemleri yasaklanır.

9.2.1. Mod 1

Mode 1 standart asenkron haberleşme modudur. Bu modda kullanılan Veri Çerçevesi 10 bitten oluşur (Şekil 9.3). Bunlar bir başlama biti (0), sekiz veri biti (önce hafif ağırlıklı bit - LSB) ve bir Dur biti(1) 'dir. Eğer Eşitlik Bitine izin verilirse, Çift Eşitlik biti 8. veri biti yerine gönderilir ve alma işleminde eşitliği denetlemek için kullanılır.



Alma işlemi sırasında RI bayrağı son veri bitinin alınmasından sonra lojik 1 olur. Alma kesme isteği işareti oluşturulmadan önce RI bayrağı lojik 1 durumundadır. Gönderme işlemi sırasında 8. veri biti gönderildikten sonra lojik 1 olur. TI bayrağı lojik 1 olduğu zaman gönderme kesme isteği işaretinin oluşması mümkün olur. Gönderme ve alma işlemleri ayrı öteleme için kullanılan saat darbeleri tarafından denetlenir. Gönderme işlemi için kullanılan öteleme saat darbeleri Karakter Hız Üretici şartlandığında başlar. Alma işleminde kullanılan saat darbesi düşen kenar tetiklemesi alındığı zaman sıfırlanır. Bu yüzden gönderme işlemi için kullanılan saat darbesi, alma işlemindeki saat darbesi ile eşit frekansta olmalarına rağmen bu iki saat darbesi uyumlu olmayabilir.



Şekil 9.4. Mod 1 için seri port veri çerçevesi

Gönderme İşlemi Kesme İsteği (TI) ve Alma İşlemi Kesme İsteği (RI) bayrakları işlemlerin tamamlandığını göstermek için lojik 1 olurlar. Alma işlemi sırasında RI bayrağı durma bitinden hemen önce lojik 1 olur. Alma İşlemi Kesme İsteği sinyali, RI bayrağı lojik 1 olduğu zaman oluşur. Gönderme işlemi sırasında TI bayrağı durma bitinin başlamasıyla lojik 1 olur. Gönderme İşlemi Kesme İsteği sinyali TI bayrağı lojik 1 olursa üretilir. Sonraki bayt durma biti gönderilinceye kadar gönderilmez.



9.2.2. Mod 2:

Mod 2, 9 bit asenkron moddur. Bu mod çok işlemcili sistemlerin haberleşmeleri için Mod 3 ile ortaklaşa kullanılır. Şekil 9.5. bu modda kullanılan veri çerçevesini gösterir. Bu mod başlama bitini (0), dokuz veri bitini (önce en hafif ağırlıklı bit - LSB) ve bir durma bitini (1) içerir. Gönderme işlemi sırasında 9. biti göndermek için, SBUF(TX) 'e bilgi yazılmadan önce SP_CON kaydedicisinde 9. bit görevini yapan TB8 biti kullanılır. Mikrodenetleyici donanımı her gönderme işleminden sonra TB8 bitini sıfırlar. Bu nedenle SBUF(TX) 'e gönderilecek veri yazılmadan önce 9. bit yerine geçen TB8 biti belirlenmelidir. Alma işlemi sırasında RI bayrağı lojik 1 olur ve sadece TB8 biti lojik 1 ise kesme isteği üretilir. Bu modda Eşitlik İşlemine izin verilmez.



Şekil 9.5. Mod 2 için seri port veri çerçevesi

9.2.3. Mod 3

Mod 3, asenkron 9 bitlik moddur. Bu mod için veri çerçevesi Mod 2 ile aynı yapıdadır. Mod 3 'ün Mod 2 'den farkı, eşitlik işlemine izin verilmesidir. Bu nedenle 9. bit çift eşitlik biti olur. Eşitlik işlemine izin verilmediği zaman 0-7 arasındaki veri bitleri verici Destek Kaydedicisine ve 9. veri biti SP_CON kaydedicisindeki bit 4 'e (TB8) yazılır. Mod 3 'de alma işlemi 9. bitin durumu ne olursa olsun bir kesme isteğine neden olur. Eğer Eşitlik İşlemine izin verilmez ise alma işleminde 9. veri biti SP_STAT kaydedicisinin bit 7 'sinden (RB8)

okunabilir. Eşitlik işlemine izin verilmez ise RB8 , Alma İşlemi Eşitlik Hata bayrağı (RPE) olur.

Tablo 9.1. Seri portu şartlamak için gerekli kaydediciler

Kayd. sembolü	Kaydedici adı	Tanımlama
BAUD_RATE	Karakter Hızı Ayarı	Bu kaydedici seri portun karakter hızını ve saat darbesi kaynağını seçer. Kaydedici adresine ilk bayt, hafif ağırlıklı bayt olmak üzere iki defa yazma işlemi yapılır. En yüksek ağırlıklı bit saat darbesi kaynağını seçer. En hafif 15 bit saniyedeki karakter hızı olarak belirlenen işaretli tamsayı değeri BAUD_VALUE 'yi gösterir.
IOC1	I/O kontrol kaydedicisi	Bu kaydedicinin 5. biti lojik 1 olması P2.0 bacağına TXD fonksiyonu olmasına izin verir.
SBUF (RX)	Seri port alma kaydedicisi	Bu kaydedici seri porttan alınan veriyi içerir.
SBUF (TX)	Seri port gönderme kaydedicisi	Kaydedici göndermek için hazır olan veriyi gösterir. Mod 1,2 ve 3 'de SBUF(TX) 'e veri yazılırsa ve alma işlemine izin verilmemiş (SP_CON.3 = 0) ise gönderme işlemi başlar.
SP_CON	Seri port kontrol kaydedicisi	Bu kaydedici haberleşme modunu seçer ve alma işlemine, çift eşitlik işlemine ve 9 bit veri göndermeye izin verebilir. Her gönderme işleminden sonra TB8 sıfırlanır.
SP_STAT	Seri port durum kaydedicisi	Bu kaydedici seri port durum bitlerini içerir. Alma kaydedicisi Üzerine Yazma Hatası (OE), Gönderme Kaydedicisi Boş (TXE), Çerçeveleme Hatası (FE), Gönderme İşlemi Kesme isteği (TI), Alma İşlemi Kesme İsteği (RI) ve Alma İşlemi Eşitlik Hatası (RPE) veya Alma işlemindeki 8. bit (RB8) için durum bitlerine sahiptir. SBUF(TX) 'e yazma işlemi yapılırsa TXE biti sıfırlanır.

9.2.4. Mod 2 ve Mod 3 zamanlama

Mod 2 ve Mod 3 'deki çalışma Mod 1 'deki çalışmaya benzer. Aralarındaki fark gönderme ve alma işleminde veri 9 bit ve bilgi çerçevesinin 11 bit uzunluğunda olmasıdır. Alma işlemi sırasında RI bayrağı durma bitinin sonunda lojik 1 olur. Alma işlemi kesme isteği sinyali RI bayrağı lojik 1 olduğu zaman üretilir. Gönderme işlemi sırasında TI bayrağı durma bitinin başlamasıyla lojik 1 olur. Gönderme



işlemi kesme isteği sinyali TI bayrağı lojik 1 olduğu zaman üretilir. 9. bit eşitlik işlemi veya çok işlemcili haberleşmeler için kullanılabilir.

9.3. Seri portun programlanması

Tablo 9.1. seri port fonksiyonlarını ve işlemlerini ayarlayan kaydedicileri gösterir.

9.3.1. Eşitlik işlemine izin verme ve haberleşme modunu seçme

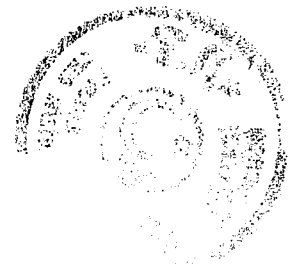
SP_CON kaydedicisinin bit 0 ve bit 1 'i seri haberleşme modunu seçer.(Tablo 9.2, Şekil 9.6). Yeni mod seçilirse seri giriş/çıkış portu sıfırlanır ve devam etmekte olan alma veya gönderme işleminin durmasına neden olur. Bit 2 'nin lojik 1 olması Eşitlik işlemine izin verir. Bit 3 'ün lojik 1 olması P2.1 bacağıının RXD fonksiyonu olmasına izin verir. Bit 4, Mod 2 veya Mod 3 'de gönderme işleminde 9. veri biti veya eğer eşitlik işlemine izin verilmiş ise Çift Eşitlik biti değeridir. Bit 4 (TB8) her göndermeden sonra sıfırlanır

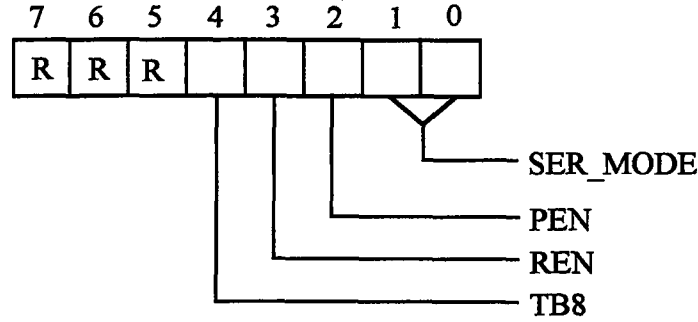
Tablo 9.2. Mod seçme

Bit 1	Bit 0	Seçilen mod
0	0	Mod 0
0	1	Mod 1
1	0	Mod 2
1	1	Mod 3

9.3.2. TXD ve RXD için şartlamalar

Tablo 9.3. seri port ile biraradaki diğer işlevli bacakları listeler. IOC1 kaydedicisinin 5. biti lojik 1 ise TXD 'ye izin verilir eğer aynı bit lojik 0 ise TXD 'ye izin verilmez. SP_CON kaydedicisinin 3. biti lojik 1 ise RXD 'ye izin verilir eğer bu bit lojik 0 ise RXD 'ye izin verilmez.





Şekil 9.6. SP_CON kaydedicisi

Tablo 9.3. Seri port sinyal bacakları

Fonk.	Ek fonk.	Seçilme dur.	Tip	Tanımlama
RXD	P2.1/ PALE	SP_CON=1	I/O	Seri veri alma. Mod 1,2 ve 3 'de RXD karşı üniteden gönderilen verinin giriş bacağı. Mod 0 'da giriş ve çıkış.
T2CLK	P2.3	BAUD_ RATE MSB=0	I	Zamanlayıcı 2 saat darbe girişi ve Karakter Hızı Üretici için iki saat darbesi kaynağından birisi.
TXD	P2.0/ PVER#	IOC1.5 = 1	O	Seri veri gönderme. Mode 1, 2 ve 3 'de TXD seri port çıkış verisini karşıya gönderen bacak. Mod 0 'da seri saat darbesi çıkışıdır.

9.3.3. Seri port Kesme isteğine izin verme

Seri port gönderme işlemi kesme isteğini ve alma işlemi kesme isteğinin her ikisini veya yalnız seri port kesme isteğini üretmek için ayarlanabilir. Eğer seri port kesme isteği önlenir, alma ve gönderme işlemi kesme isteklerine izin verilirse RI ve TI bayrakları ayrı ayrı alma ve gönderme kesme istekleri üretir. Eğer 8096BH ile uyumluluk söz konusu değil ise bu ayarlama tercih edilir.

8096BH ile uyumluluk istenilirse alma ve gönderme işlemi kesme isteklerine izin verilmez ve seri port kesme isteğine izin verilir. RI ve TI bayraklarının her ikisinde seri port kesme isteklerini üretebilir. TI bayrağı kesme isteği üretirse SP_STAT kaydedicisinde 5. bit lojik 1



olur. RI bayrağı kesme isteğı üretirse SP_STAT kaydedicisindeki 6. bit lojik 1 olur.

Tablo 9.4. Seri port kesme isteğı kaydedicileri

Kayd. semb.	Kaydedici adı	hex adr	Tanımlama
INT_MASK	Kesme isteğini önleme	08H	Bu kaydedicinin 6. bitinin lojik 1 olması 8096BH ile uyumlu olan seri port kesme isteğine (INT 06H - 200CH) izin verir. Aynı bitin lojik 0 olması kesme isteğini önler.
INT_MASK1	Kesme isteğini önleme1	13H	Bu kaydedicinin 0. bitinin lojik 1 olması gönderme işlemi kesme isteğine (INT 08H - 2030H) izin verir. 0. bitin lojik 0 olması kesme isteğini önler. Bu kaydedicinin 1. bitinin lojik 1 olması alma işlemi kesme isteğine (INT 09H-2032H) izin verir. 1. bitin lojik 0 olması kesme isteğini önler.
INT_PEND	Kesme isteğini bekletme	09H	Bu kaydedicinin 0. biti lojik 1 olduğu zaman bekletilen seri port kesme isteğini (INT 06H) gösterir. Kesme isteğı vektörü 2030H olduğu zaman bu bit lojik 0 olur.
INT_PEND1	Kesme isteğini bekletme 1	12H	Bu kaydedicinin 0. biti lojik 1 olduğu zaman bekletilen gönderme işlemi kesme isteğini (INT 08H) gösterir. Kesme isteğı vektörü 2030H olduğu zaman bu bit lojik 0 olur. Bu kaydedicinin 1. biti lojik 1 olduğu zaman bekletilen alma işlemi kesme isteğini (INT 09H) gösterir. Kesme isteğı vektörü 2032H olduğu zaman bu bit lojik 0 olur.

Özel kesme isteklerine izin vermek için INT_MASK1 kaydedicisindeki TI_MASK ve RI_MASK bitleri lojik 1 olmalıdır. 8096BH ile uyumlu kesme isteğine izin vermek için INT_MASK kaydedicisindeki SER_MASK bitinin lojik 1 olması gerekir.

9.3.4. Karakter hızı ve saat darbesi kaynağını programlama

BAUD_RATE kaydedicisi Karakter hızı üreteci için saat darbesi girişini seçer ve seri Giriş/Çıkış modlarının hepsi için karakter hızını tanımlar. Kelime uzunluğundaki bu kaydediciye ard arda gelen iki



bayt yüklenmelidir. Bu kaydedici adresine karakter hızını belirleyen kelimenin önce hafif ağırlıklı baytı yüklenir. Kaydedicinin en fazla ağırlıklı biti (Bit 15) Karakter Hız Üreteci için iki uygun saat darbesi kaynağından birisini seçer. Eğer 15. bit lojik 1 ise XTAL1 giriş sinyali (f_{osc}) veya bu bit lojik 0 ise T2CLK bacağındaki dış giriş sinyali Karakter Hız Üreteci saat darbesi kaynağı olarak seçilir. En yüksek T2CLK giriş frekansı $f_{osc} / 4$ 'tür.

Bu kaydedicinin alt 15 biti karakter hızı olarak hesaplanan işaretli tamsayı olan BAUD_VALUE 'yi gösterir. Bu değer en fazla 7FFFH (32767) olabilir. Verilen bir saniyedeki karakter hızı BAUD_VALUE değerini hesaplamak için aşağıdaki ifadeler kullanılır.

Senkron mod 0 için;

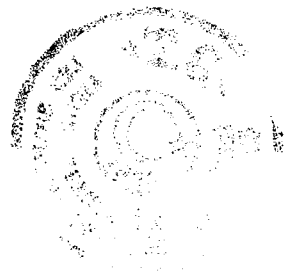
$$\text{BAUD_VALUE} = \frac{f_{osc}}{\text{Karakter hızı} \times 2} - 1 \text{ veya } \frac{\text{T2CLK}}{\text{Karakter hızı}} \quad (9.1)$$

Asenkron mod 1, 2, ve 3 için;

$$\text{BAUD_VALUE} = \frac{f_{osc}}{\text{Karakter hızı} \times 16} - 1 \text{ veya } \frac{\text{T2CLK}}{\text{Karakter hızı} \times 8} \quad (9.2)$$

Tablo 9.5. 16 MHz 'de XTAL1 kullanıldığı zaman BAUD_RATE değerleri

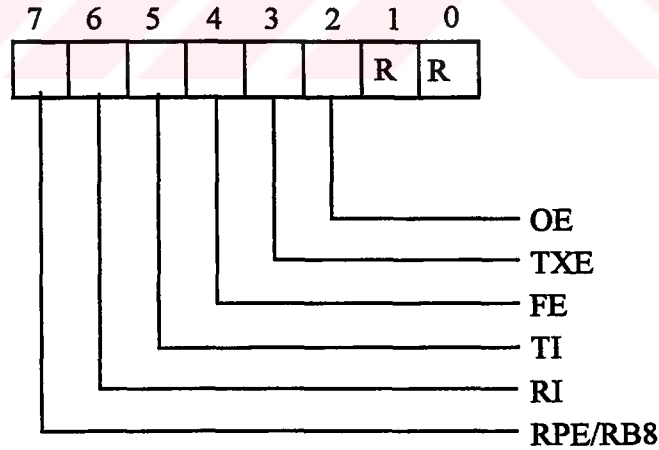
Karakter hızı	BAUD RATE		% Hata	
	Mod 0	Mod 1,2,3	Mod 0	Mod 1,2,3
9600	8340H	8067H	0.04	0.16
4800	8682H	80CFH	0.02	0.16
2400	8D04H	81A0H	0.01	0.08
1200	8A9AH	8340H	0	0.04
300	E82BH	8D04H	0	0.01



Tablo 9.5. Karakter hız üreteci için 16 MHz'lik XTAL1 saat darbesi giriş seçildiği zaman çok kullanılan karakter hızları için BAUD_RATE değerlerini gösterir. BAUD_VALUE denklemi tam kesin denklem olmadığından sonuçlar yaklaşık olarak hesaplanmıştır. Tablo 9.5 aynı zamanda bu yaklaşık değerler ile gerçek değerler arasındaki hatayı yüzde olarak gösterir. Çoğu zaman karakter hızlarındaki %5 'lik fark ihmal edilebilir.

9.3.5. Seri portun durumunu tespit etme

Seri portun durumu SP_STAT kaydedicisinde görülür (Şekil 9.6). SP_STAT kaydedicisi okunduğu zaman TXE biti hariç diğer bitler sıfırlanır. Bu yüzden SP_STAT kaydedicisinin içeriği görüntü kaydedicisi gibi kullanılan başka bir kaydediciye kopyalanır. Sonra JBC ve JBS gibi bit test komutları görüntü kaydedicisi üzerinde yürütülür.



Şekil 9.6. SP_STAT kaydedicisi

8XC196 KC/KD 'deki alıcı, geçerli durma bitini denetler. Eğer durma biti kendine ayrılan zaman içerisinde bulunamamış ise SP_STAT kaydedicisindeki Çerçeve Hata (FE) biti lojik 1 olur. Durma biti tespit edildiği zaman alma işlemini yapan ötelemeli kaydedicideki veriyi



SBUF(TX) kaydedicisine yükler ve alma işlemi kesinti (RI) bayrağı lojik 1 olur. Eğer bu olay SBUF(RX) kaydedicisindeki önceki baytın okunmasından önce meydana gelmesi durumunda Üzerine Yazma Hata (OE) biti lojik 1 olur. SBUF(RX) daima en son alınan baytı içerir.

Alma İşlemi kesme İsteği (RI) bayrağı yeni veri baytı alınmasında lojik 1 olur. Gönderme İşlemi Kesme İsteği (TI) bayrağı bir veri baytının gönderilmesi bittiğinde lojik 1 olur. Bu bayraklar kesme isteklerini tetikler ve Kesme İsteği Bekletme kaydedicisinde karşılık gelen bitleri lojik 1 yaparlar. Bilgi alma /gönderme işlemi sonucu sırasında SP_STAT 'daki RI/TI biti ve karşılık gelen Kesme İsteği bekletme biti lojik 1 olur. SP_STAT kaydedicisindeki RI/TI bitine programda herhangi bir şey yazılması Kesme İsteği Bekletme bitleri etkilemez ve bir kesme isteğine neden olmaz. Benzer şekilde SP_STAT kaydedicisinin okunmasıyla RI ve TI bitleri sıfırlanır. Fakat Kesme İsteği Bekletme Kaydedicisinde bu bitlere karşılık gelen bitler sıfırlanmaz.

Eğer SBUF (TX) ve destekleyicisi boş ve iki bayt almaya hazır ise lojik 1 olur. TXE biti SBUF(TX) 'e bir bayt yazılır yazılmaz sıfırlanır. Eğer yalnız TI lojik 1 ise bir bayt yazılabilir. Eğer TXE henüz lojik 1 ise gönderme işlemi tamamlandığında TI lojik 1 olacaktır.

Alma İşlemi Eşitlik Hata (RPE) bayrağı veya Alınan Bit 8 (RB8) bayrağı Eşitlik İşlemine izin verip vermeme için kullanılır. Eğer İşitlik İşlemine izin verilir ve Eşitlik Hatası tespit edilirse RPE biti lojik 1 olur. Eğer Eşitlik İşlemine izin verilmez ise RB8 biti Mod 2 ve 3 'de alınan 9. veri biti olur.

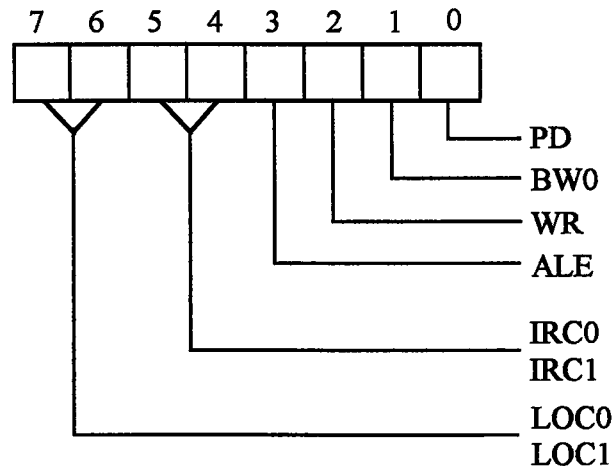


10. MİKRODENETLEYİCİNİN DIŞ BELLEK İLE BAĞLANMASI

8XC196 KC/KD çeşitli dış belleklerle sabit 8 bit yol genişliğinde veya dinamik 8 bit/16 bit yol genişliğinde bağlanabilir. Ayrıca hızı yavaş olan dış bellekler için iç READY denetlemeyi, yol üzerinde yer alan dış birimlere izin veren yol tutma işlemini ve çeşitli yol denetleme modlarını sağlayabilir. Bunların en önemli özelliği dış bellek birimleri ile bağlanıldığı zaman büyük bir oranda uyumluluk sağlamasıdır.

10.1. Denetleyici Ayarlama Kaydedicisi (CCR)

Denetleyici Ayarlama Kaydedicisi (CCR) yol genişliğini, yazma tutma sinyalini üretmeyi, adres geçerli sinyalini üretmeyi ve READY bacağı lojik 0 'da kaldığı zamanda araya konulan bekleme durumu süresini denetler. CCR kaydedicisi önce bellekten bir bayt alır ve sonra sistem sıfırlanır. Sistem sıfırlandığında yol denetleyici 2018H bellek gözünden Denetleyiciyi Ayarlama Baytını (CCB) alır ve denetleyiciyi ayarlama kaydedicisine (CCR) yükler. CCB eğer EA# bacağı lojik 1 ise iç OTPROM 'dan ve EA# bacağı lojik 0 ise dış bellekten alınır.



Şekil 10.1. Denetleyiciyi Ayarlama Kaydedicisi

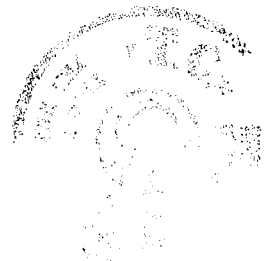


Denetleyiciyi ayarlama kaydedicisine sadece ard arda gelen sıfırlama işaretleri sırasında yüklenme olur. Bir defa yükleme yapıldıktan sonra CCR sonraki sistem sıfırlanmasına kadar değişmez. CCB kullanıcı programı derlendiği zaman bir defa programlanır ve işlemler sırasında tekrar tanımlanmaz.

CCR ilk şartlamalar yapılmadan kendi değerinde olması CCB 'yi alma sırasında yol genişliğini ve zamanlamayı denetlemek için READY ve BUSWITH sinyallerine izin verir. Eğer CCB hızı yavaş olan dış bellekte saklanırsa READY sinyali lojik 0 'a çekilir. Bu CCB 'yi alma sırasında üç bekletme durum zamanının geçmesine neden olur. (CCR:4 ve CCR.5 bekletme durum zamanının sayısını denetler, kendi ilk değerinde olması üç durum zamanı ve en yüksek bekletme zamandır.) Eğer CCB 16 bitlik dış belleğe yazılı durumda ise BUSWITH sinyali lojik 1 , CCB 8 bitlik dış belleğe yazılı durumda ise BUSWITH sinyali lojik 0 olması gerekir. CCB, CCR 'ye yüklendikten sonra yol genişliği CCR.1 ve BUSWITH sinyali tarafından 8 bit veya dinamik 16 bit olarak denetlenebilir şekilde ayarlanır.

10.2. Yol genişliğinin ve belleğin ayarlanması

8XC196KC/KD dış yolu 8 bit veya 16 bit Adres/Veri çoğullamalı olarak çalışabilir (Şekil 10.2). CCR.1 'in değeri BUSWITH sinyali tarafından denetlenebilen yol genişliğini 8 bit yol genişliği veya 16 bit/ 8bit yol genişliği olarak ayarlar. Eğer CCR.1 sıfırsa yol denetleyicisi 8 bit yol moduna ayarlanmış olur. CCR.1 lojik 1 olursa, BUSWITH sinyali yol genişliğini kontrol eder. BUSWITH bacağı lojik 1 olduğu zaman yol genişliği 16 bit ve BUSWITH lojik 0 olduğu zaman yol genişliği 8 bit olur.



10.3. Yol denetleme modları

Dış okuma ve yazma periyotlarında hangi yol denetleme sinyallerinin üretileceğini CCR.2 ve CCR.3 bitleri belirler.

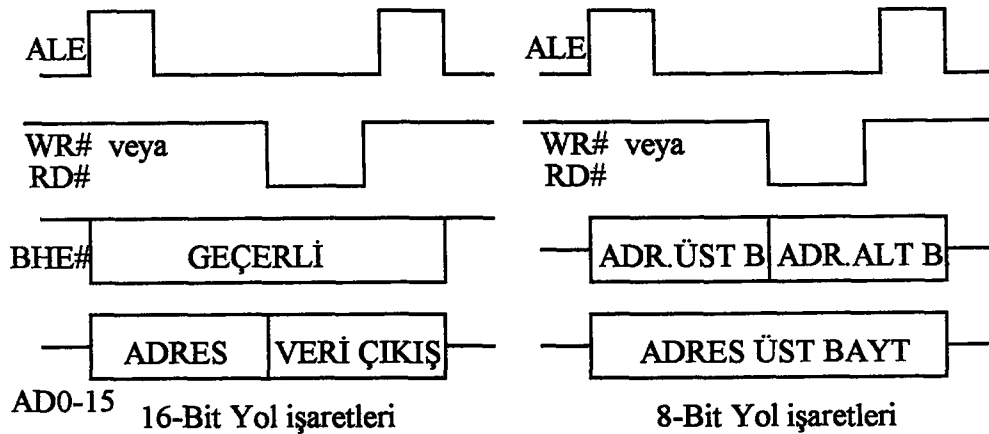
Tablo 10.1. Yol denetleme modları

Yol denetleme modu	Yol denetleme Sinyalleri	CCR.2	CCR.3
Yazma tutma modu için geçerli adres	ADV#, RD#, WRL#, WRH#	0	0
Tutma geçerli adres modu	ADV#, RD#, WR#, BHE#	0	1
Yazma tutma modu	ALE, RD#, WRL#, WRH#	1	0
Standart yol denetleme modu	ALE, RD#, WR#, BHE#	1	1

Tablo 10.1. dört yol denetleme modlarını listeler ve CCR.3, CCR.2'nin herbirinin lojik 1 olması durumlarını gösterir.

10.3.1. Standart yol denetleme modu

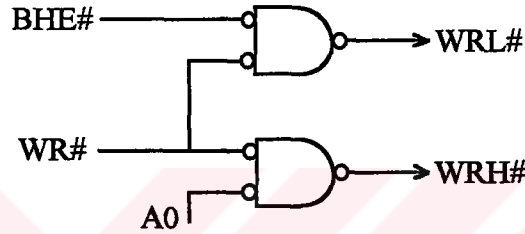
Standart yol denetleme modunda 8XC196 KC/KD ALE, WR#, RD# ve BHE# standart yol denetleme işaretlerini üretir (Şekil 10.2.). Mikrodenetleyici adresi dışarı verdiğinde ALE işaretini üretir ve ALE dışarıdaki işareti tutmak için kullanılabilir. BHE# lojik 0 olduğu zaman veri yolunun üst baytı tarafından gösterilen bellek gözünü seçer.



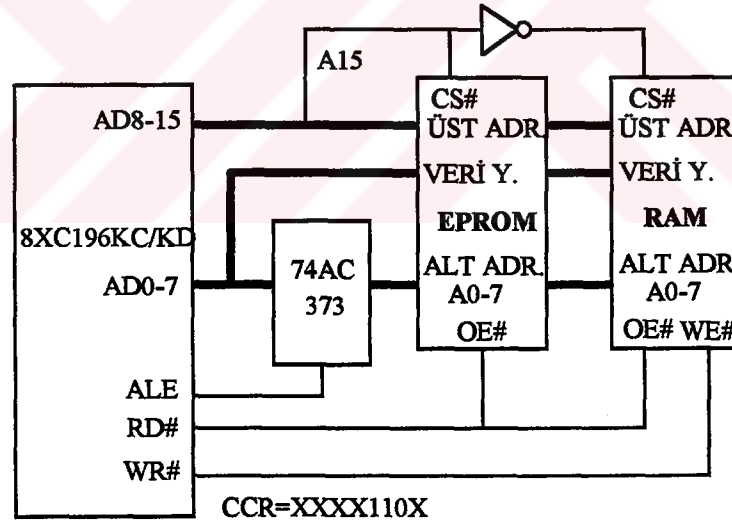
Şekil 10.2. Standart yol denetleme işaretleri

RD# işareti dış belleği her okuma ve WR# işareti belleğe her yazma işleminde oluşur.

Sistem 16 bitlik yol için ayarlandığı zaman, her defasında bir bayt yazmak üzere alt ve üst bayt sinyalleri ayrı ayrı üretilir. Şekil 10.3 WRL# ve WRH# sinyallerini üretmek üzere A_0 ve BHE# işaretlerinin bileşimi olan bir devreyi gösterir. Okuma işareti için ayrı iki işarete ihtiyaç duyulmaz. 16 bitlik yoldan bir bayt okunacağı zaman veri yoluna üst ve alt baytın her ikisinde yüklenir.



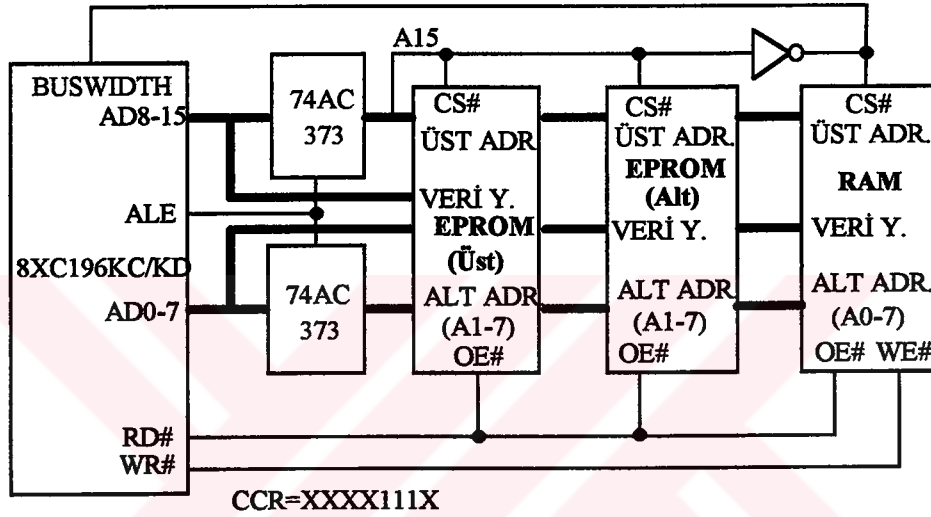
Şekil 10.3. WRL# ve WRH# işaretlerinin elde edilişi



Şekil 10.4. EPROM ve RAM 'ın beraber kullanıldığı 8 bitlik sistem

Şekil 10.4 EPROM 'un ve RAM 'ın beraber kullanıldığı 8 bitlik bir sistemi gösterir. EPROM alt yarı bellekte ve RAM üst yarı bellekte yer almaktadır. Bu sistem en fazla ağırlıklı adres biti (A_{15}) Tümdevre seçme işareti ve ALE adres tutma işareti olarak kullanılır.

Şekil 10.5 dinamik yol genişliği özelliğini kullanan bir sistemi gösterir. Komut kodları iki EPROM 'dan alınır ve veri bir bayt genişliğinde RAM 'da saklanır. RAM üst bellek bölgesine yerleştirilmiştir. RAM, A_{15} lojik 1 olduğu zaman ve aynı zamanda 8 bit yol genişliği modunda BUSWIDTH sinyali lojik 0 olduğu zaman seçilir.



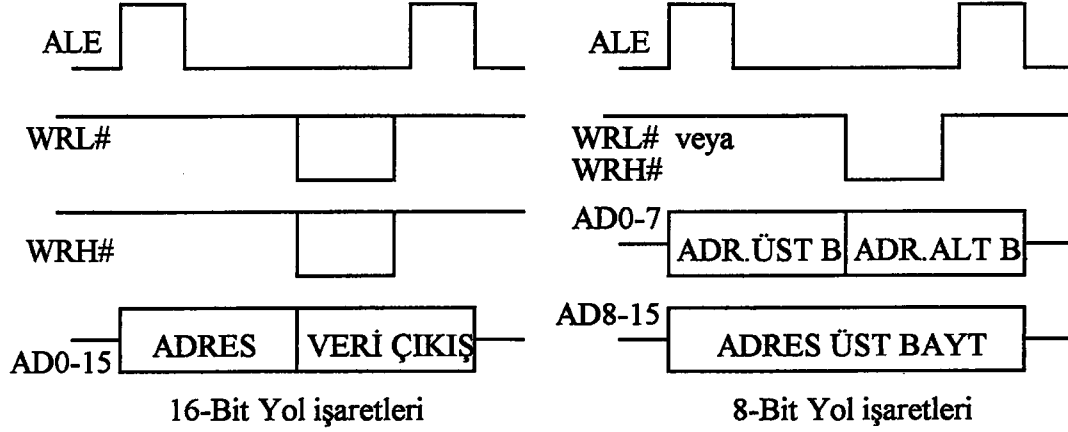
Şekil 10.5. 16 bitlik Dinamik yol genişliğindeki sistem

10.3.2. Yazma tutma modu

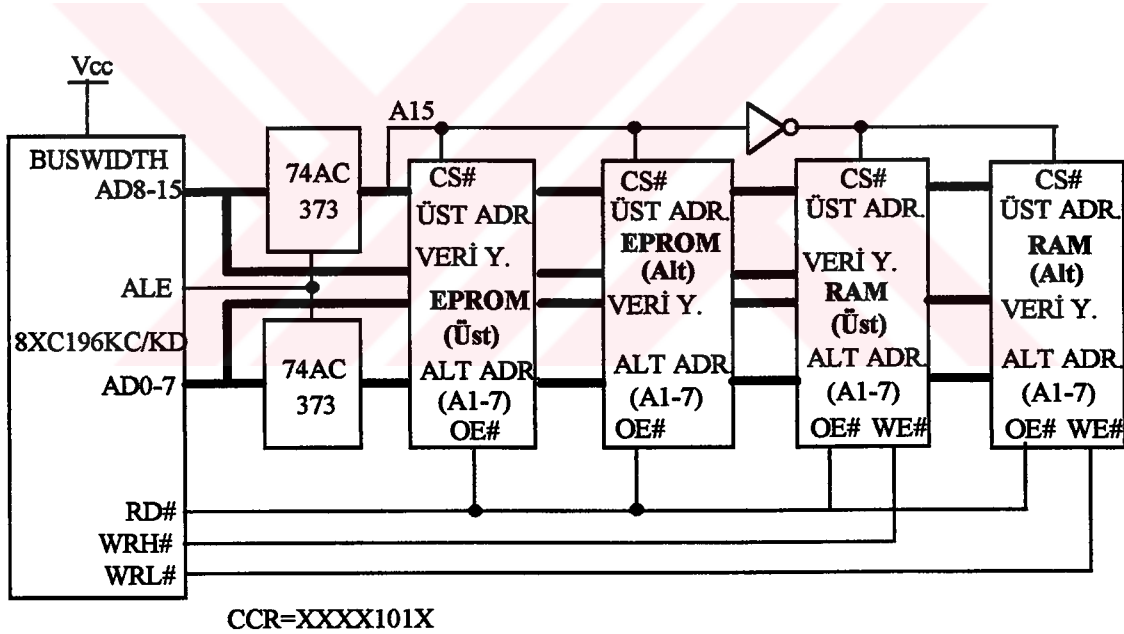
Yazma tutma modu üst ve alt baytları 16 bitlik yol modunda dış 16 bit RAM 'a yazan dış seçme devresine olan ihtiyacı ortadan kaldırmak için kullanılır. Yazma tutma modu kullanıldığında zaman 8XC196KC/KD WR# ve BHE# işaretleri sayesinde kendisi içeriden WRL# ve WRH# işaretlerini üretir. WRL# alt baytlar çift adreslere yazılırken ve WRH# üst baytlar tek adreslere yazılırken oluşturulur. 8 bitlik yol modunda, WRH# ve WRL# çift ve tek adreslerin her ikisi için ayrı ayrı üretilir. Yazma tutma modundaki zamanlama işaretleri Şekil 10.6. ' de gösterilmiştir.

Şekil 10.7. iki EPROM ve iki RAM kullanan 16 bitlik bir sistemi gösterir. Bu örnek yazma tutma modunu kullanmak için tasarlanmıştır.



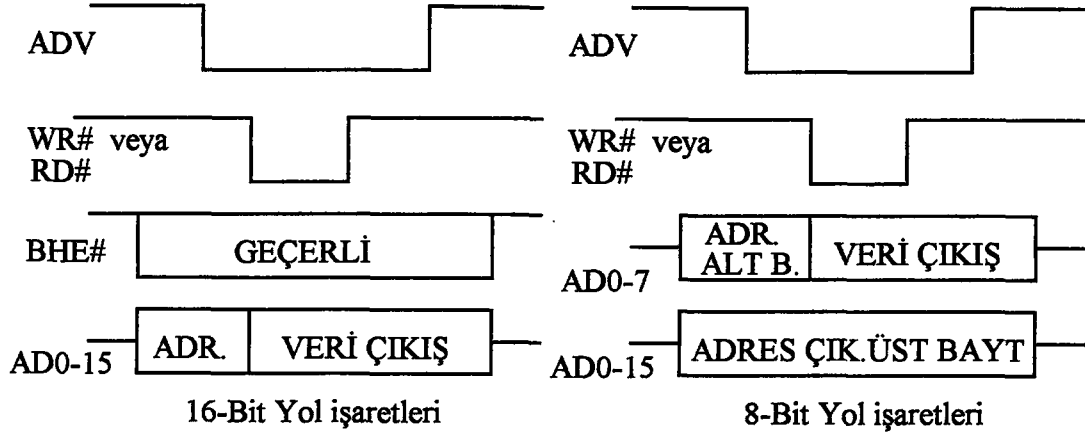


Şekil 10.6.Yazma tutma modundaki zamanlama işaretleri



Şekil 10.7 RAM 'e tek bayt yazan 16 bitlik sistem

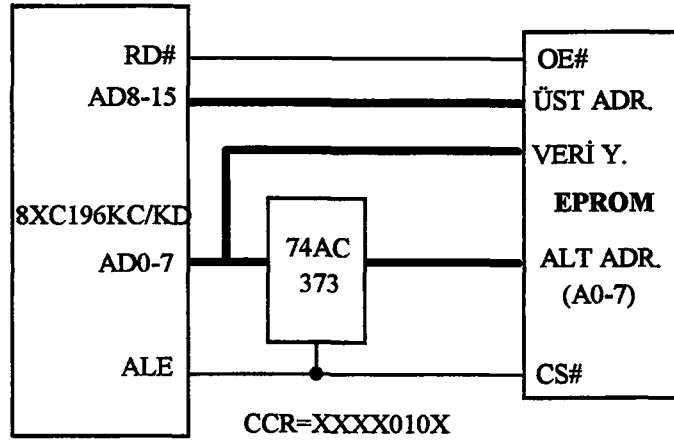
ALE sinyali adresi tutma işine yarar ve A_{15} EPROM ve RAM için tümdevre seçme sinyalidir. Kelime yazılırken WRL#, alt baytı yazma sırasında ve WRH# üst baytı yazma sırasında üretilir. Bellekler adres yolunun A_0 bitini kullanmazlar. WRL# ve WRH# alt bayt ($A_0=0$) veya üst bayt ($A_0=1$) durumları oluşurken elde edilir.



Şekil 10.8. Geçerli adres tutma modu zamanlama işaretleri

10.3.3. Geçerli adres tutma modu

Bu mod seçildiği zaman 8XC196 KC/KD adres tutmaya izin verme (ALE) işareti vasıtasıyla geçerli adres (ADV#) işareti üretilir. ADV# dış adres geçerli olduktan sonra oluşur (Şekil 10.8). Bu işaret geçerli adresi tutmak ve aynı anda dış belleğe izin vermek için kullanılabilir.

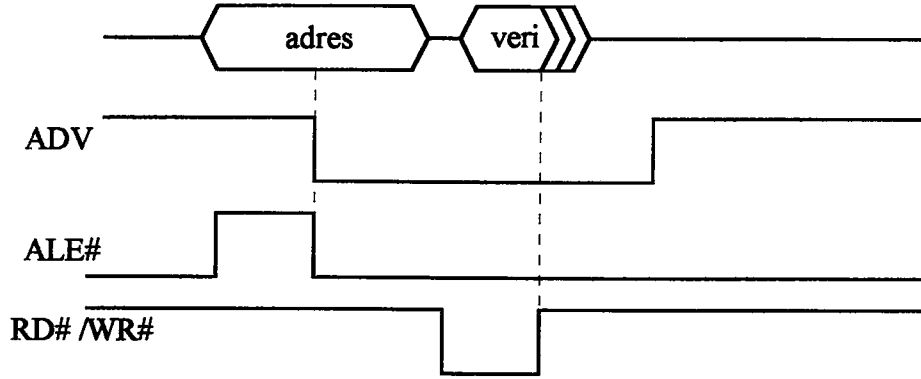


Şekil 10.9. EPROM kullanan 8 bitlik sistem

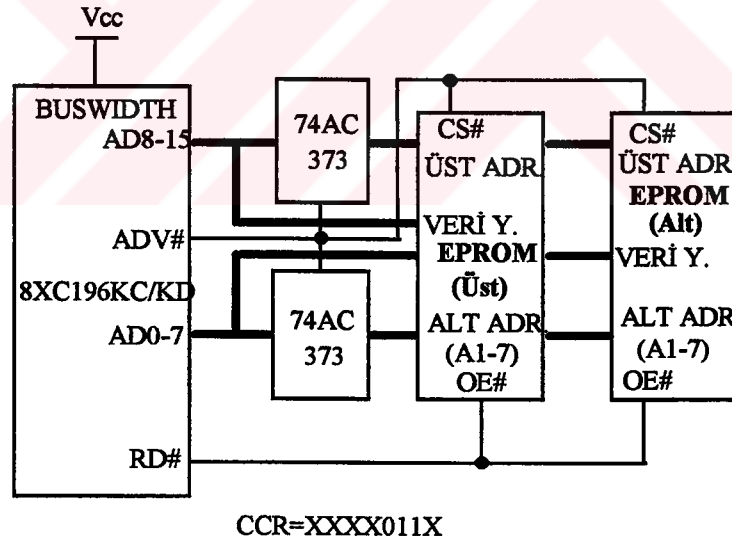
ALE ve ADV# arasındaki fark ALE adres oluştuğunda hemen konum değiştirir ve lojik 1 olur. Fakat ADV# adresin oluşup bitme süresinin



ortasında konum değiştirir ve lojik 0 olur (Şekil 10.10). Ayrıca ADV# işareti lojik 1 'de iken dış belleğe izin verilmez. Şekil 10.9 ve 10.11 geçerli adres tutma modu kullanan temel devreleri gösterir.



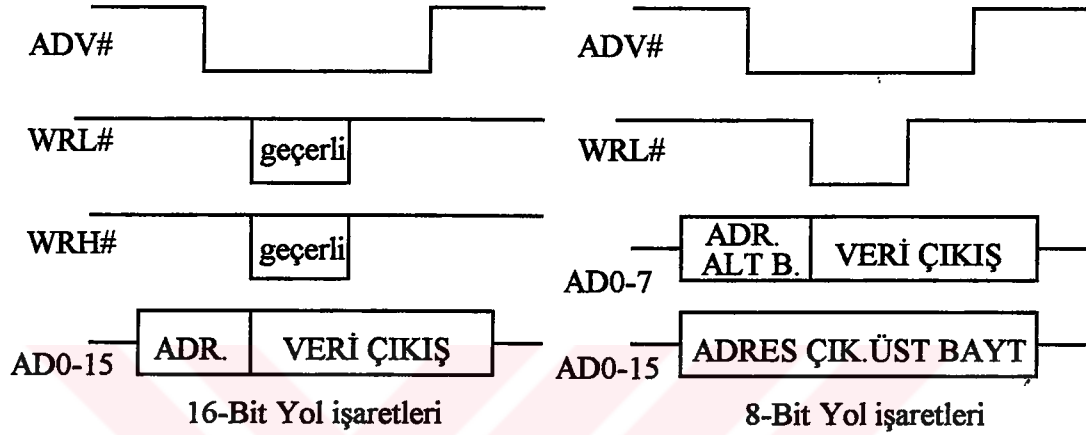
Şekil 10.10. ALE ve ADV# yol işaretlerinin zamanlamalarının karşılaştırılması



Şekil 10.11. İki EPROM kullanan 16 bitlik sistem

Şekil 10.9. tek EPROM kullanan temel 8 bitlik bir sistemi gösterir. Geçerli adres tutma modu için ayarlanmıştır. Bu sistem ADV# işaretini EPROM tümdevresini seçme işareti ve adres tutma işaretinin her ikisini sağlamak için kullanılır.

Şekil 10.11. iki EPROM 'lu 16 bitlik bir sistemi gösterir. Bu sistem EPROM tümdevresini seçme işareti ve adres tutma işaretinin her ikisini oluşturmak için ADV# işaretini kullanır.

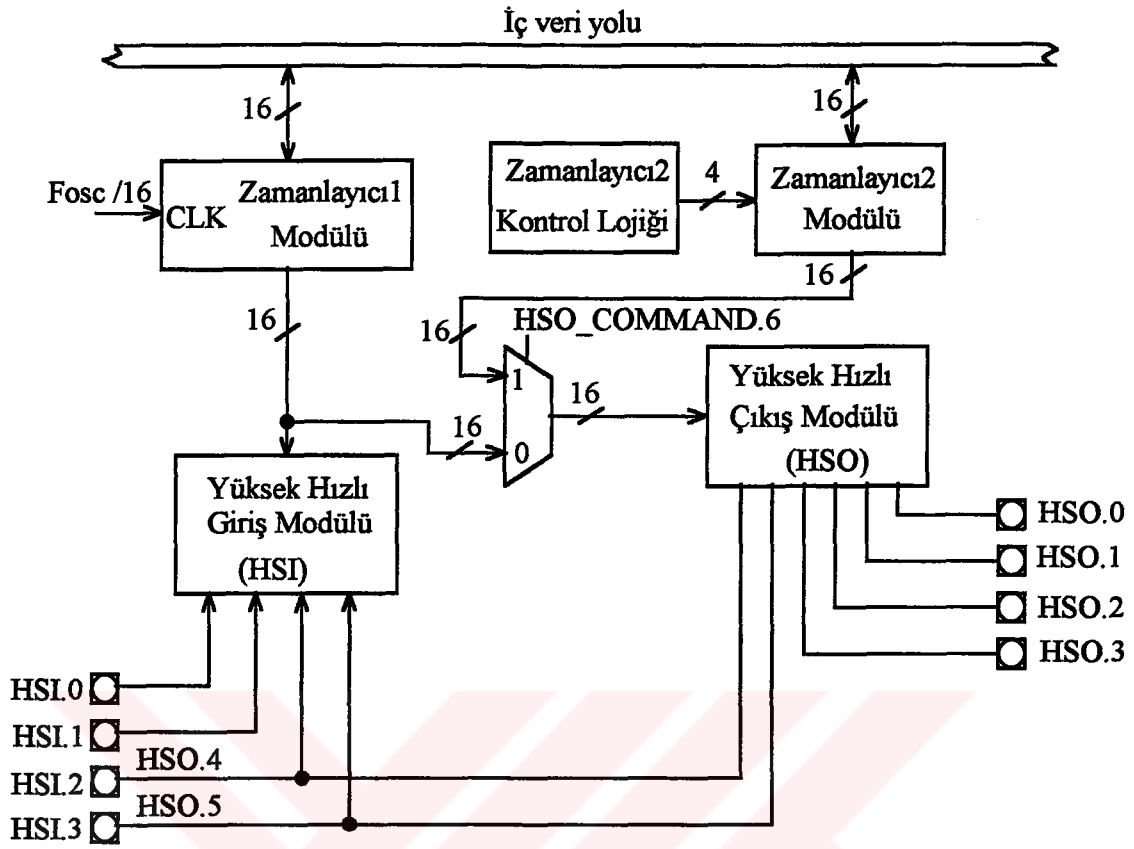


Şekil 10.12. Yazma tutma modu ile beraber geçerli adres modu zamanlama işaretleri.

10.3.4. Yazma tutma modu ile beraber geçerli adres modu

Bu mod seçildiği zaman 8XC196KC/KD ADV#, WRL# ve WRH# yol denetleme işaretlerini üretirler. Bu mod 16 bitlik dış RAM kullanan temel bir sistem için kullanılır. Şekil 10.12. bu moddaki zamanlama işaretlerini gösterir. Bu şekilde görünmeyen RD# işareti WRL#, WRH# ve WR# 'işareti gibidir. Şekil 10.13 'deki örnek sistem yazma tutma modu ile beraber geçerli adres modunu kullanır.





Şekil 11.1. HSIO blok diyagramı

11.1.2. Zamanlayıcı 2 işlevleri

Zamanlayıcı 2 ileri/geri sayıcı, HSO modülü için zaman aralığı ve 16 bitlik programlanabilir sayıcı olarak kullanılabilir. Zamanlayıcı 2'ye dışarıdan veya içeriden saat darbesi verilebilir. T2CLK veya HSI.1 bacaklarının herhangi birisi dışarıdan saat darbesi girişi olarak seçilebilir. Zamanlayıcı 2 hızlı veya normal sayma modunda kullanılabilir (hızlı $f_{osc} / 4$, normal $f_{osc} / 32$). Zamanlayıcı 2'ye yüklenecek sayı Ypencere 0 'da TIMER2 'ye yazılır. Yanlış yazılacak değer sıfırdan farklı olmalıdır.

11.1.3. Zamanlayıcı 2 'yi programlama

Tablo 11.1. zamanlayıcı 2 modülünü programlamaya yarayan kaydedicilerin listesini gösterir.



Tablo 11.1.Zamanlayıcı 2 modülünü programlayan kaydediciler

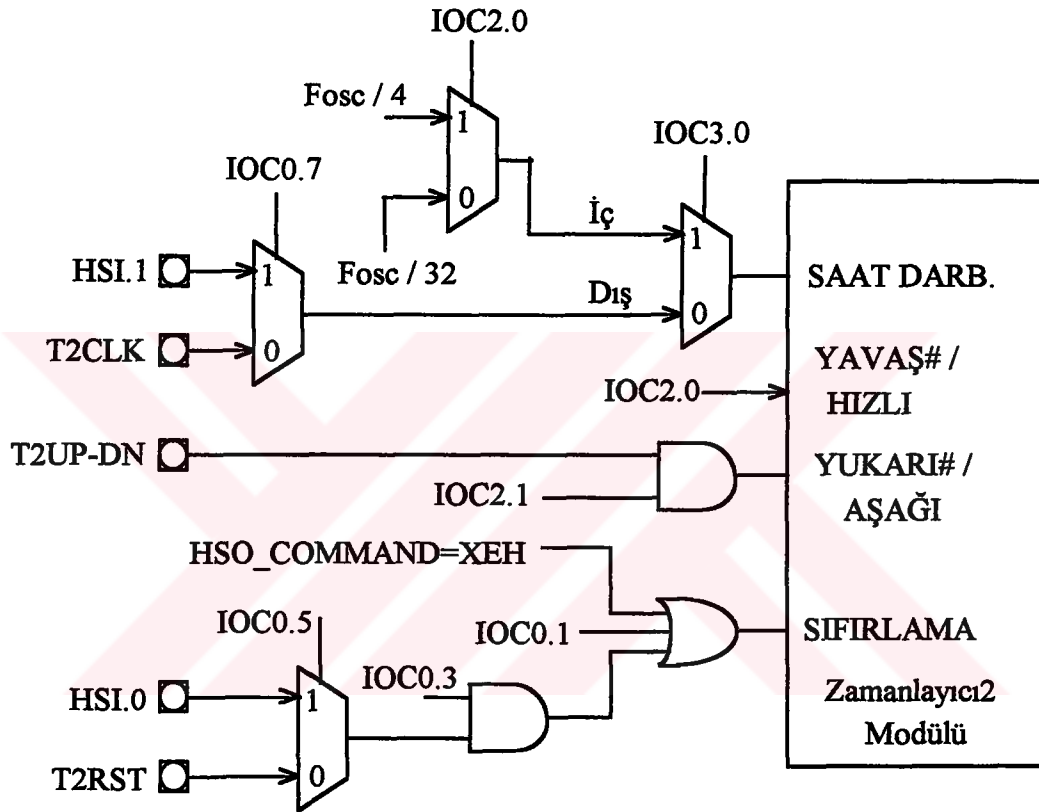
Kayd. sembolü	Kaydedici adı	Tanımlama
TIMER2	Zamanlayıcı2	Zamanlayıcı 2'nin değerini içerir.
T2CAPTURE	Zamanlayıcı2'yi tutma	P2.7 bacağına uygulanan sinyalin çıkan kenarında zamanlayıcı 2 değerinin T2CAPTURE kaydedicisinde tutulmasına neden olur ve Zamanlayıcı 2 'yi Tutma Kesme isteği oluşur (INT 11H).
INT_MASK INT_MASK1	Kesme isteği önleme Kesme isteği önleme1	Zamanlayıcı 2 'nin kesme isteklerine izin verme veya önleme
IOC0	Giriş / Çıkış kontrol Kaydedicisi 0	Zamanlayıcı 2 için sıfırlama ve saat darbesi kaynağını seçer.
IOC1	Giriş / Çıkış kontrol Kaydedicisi 1	Zamanlayıcı taşma kesme isteği (INT 07) için kesme isteği kaynağını seçer.
IOC2	Giriş / Çıkış kontrol Kaydedicisi 1	Hızlı sayma modunu, ileri/geri sayıcı fonksiyonunu ve Zamanlayıcı 2 taşma kesme isteği (INT 12H) için taşma sınırını seçer.
IOC3	Giriş / Çıkış kontrol Kaydedicisi 3	Zamanlayıcı 2 için dış veya iç saat darbesi kaynağını seçer.

11.1.4. Saat darbesi Kaynağını seçme

Zamanlayıcı 2 düşen ve çıkan kenar tetikleme her ikisinde sayma işlemi yapabilir. Saat darbesi içeriden veya dışarıdan verilebilir. Bu işlem IOC3 kaydedicisindeki T2_ENA (IOC3.0) bitinin durumuna göre olur. IOC3.0 lojik 1 ise saat darbesi içeriden, lojik 0 ise saat darbesi dışarıdan verilir (Şekil 11.2).

Dış saat darbesi kaynağı seçildiği zaman IOC0 kaydedicisindeki T2CLK_SRC biti (IOC0.7) iki dış saat darbesi kaynağını kullanmayı denetler. Dış saat darbesi IOC0.7 biti lojik 1 ise HSI.1, eğer bu bit

lojik 0 ise T2CLK (P2.3) bacağından verilir. Zamanlayıcı 2 'ye dışarıdan saat darbesi verildiği zaman IOC2 kaydedicisindeki (IOC2.0) FAST_T2_ENA biti ile saat darbesi frekansı bire veya sekize bölünebilir. Eğer FAST_T2_ENA biti lojik 1 ise en yüksek giriş frekansı oluşur. Yani giriş saat darbesi bölünmeden zamanlayıcıya



Şekil 11.2. Zamanlayıcı 2 'yi denetleme

ulaşır. Aynı bit lojik 0 ise sayıcı her sekiz saat darbesinde bir sayar. saat darbesi frekansı sekize bölünmüş olur. Zamanlayıcı 2 'ye içeriden saat darbesi verilirse FAST_T2_ENA biti giriş saat darbesi frekansını seçer. FAST_T2_ENA lojik 1 ise saat darbesi frekansı $f_{osc} / 4$ ' e eşit ve sayıcı her saat darbesinde bir artar (hızlı sayma). FAST_T2_ENA lojik 0 ise saat darbesi frekansı $f_{osc} / 32$ 'e eşit ve zamanlayıcı 2 her sekiz saat darbesinde bir artar.

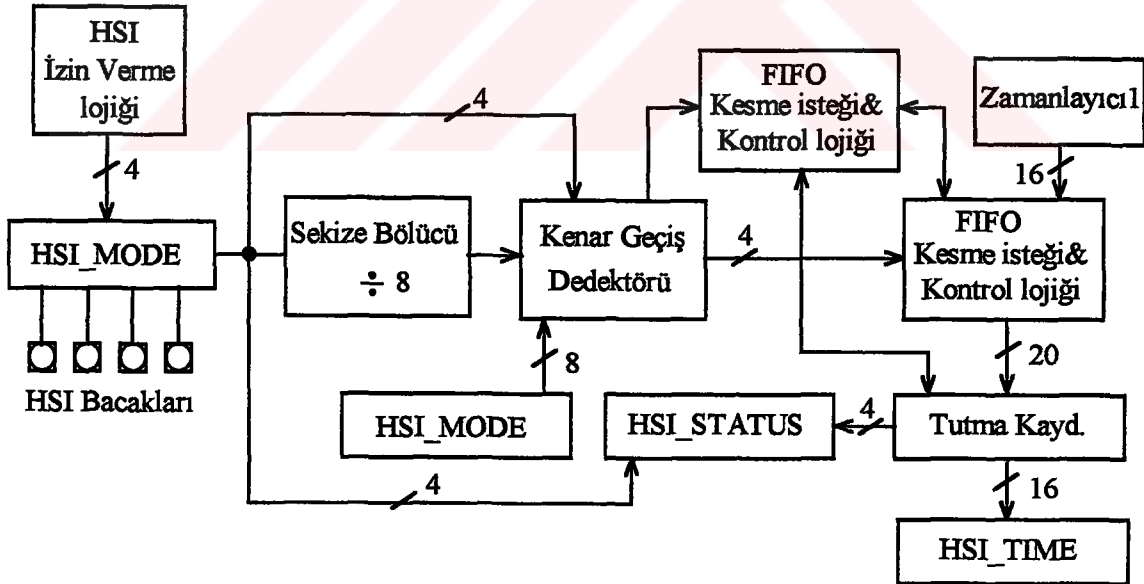


11.1.5. İleri/geri sayma

İleri/geri sayma zamanlayıcı 2 'de mevcuttur. Bu işlemi IOC2 kaydedicisindeki T2UD_ENA biti (IOC2.1) ile denetleyicinin T2UP-DN bacağı denetler (Şekil 11.2). IOC2.1 biti lojik 0 ise zamanlayıcı 2 daima ileri doğru sayar. Bu durumda T2UP-DN bacağının konumu önemli değildir. Eğer IOC2.1 biti lojik 1 ise sayma T2UP-DN bacağının durumuna göre değişir. T2UP-DN lojik 0 ise ileri aksi takdirde geri sayma işlemi gerçekleştirilir. Zamanlayıcı 2 'yi sıfırlama ve dış veya iç saat darbesinin seçilme işlemlerinin gerçekleştirilmesinde ilgili bacakların ve bitlerin nasıl kullanılacağı şekil 11.2 'de gösterilmiştir.

11.2. Yüksek Hızlı Giriş Modülü (HSI)

HSI modülünün dört tane dış giriş bacağı (HSI.0-HSI.3) vardır. Önceden tanımlanan bacakların biri veya daha fazlasında sonuç



Şekil 10.3. HSI blok diyagramı

oluşursa her giriş için durum bitleri ile beraber zamanlayıcı 1 'deki sayma değerleri kaydedilir. HSI modülü 7x20 bitlik FIFO (First Input First Output) yedi sonuç bilgisini ve tutma kaydedicisindeki veriyi

saklar (Şekil 10.3). Veri FIFO 'dan tutma kaydedicisine yüklenir ve bu veri tutma kaydedicisine yüklendikten sonra okunabilir.

HSI her çıkan kenar geçişlerinde, düşen kenar geçişlerinde, kenar geçişlerinin her ikisinde ve sekiz darbe sonra çıkan kenar geçişinde zamanlayıcı 1 'deki değeri tutabilir. Böylece HSI modülü girişlerin durumunu (darbe genişliğini, periyodu, faz farkını) ölçmek için kullanılabilir. Her sekiz darbede zamanlayıcı değeri tutma kaydedicisine kaydedilerek çok hızlı işaretler ölçülebilir ve sayılabilir.

11.2.1. Sonuç verisini okuma

Sonuçtaki veri tutma kaydedicisine yüklendikten sonra okunabilir. Sekiz darbe (CLKOUT) geçtiği zaman FIFO 'da mevcut olan zamanlayıcı 1 değeri tutma kaydedicisine yüklenir. Bu durumda IOS1 kaydedicisindeki HSI_READY biti (IOS1.7) lojik 1 olur. Aynı zamanda kesme işlemine izin verilmiş ise HSI 'da bilgi okunabilir kesme isteği (INT 02) oluşabilir.

Sonuç içeren tutma kaydedicisi okunmadan önce HSI_STATUS kaydedicisi okunur. HSI_STATUS kaydedicisinin çift bitleri HSI.0 - HSI.3 bacakları için sonuç durum bitlerini gösterir. Eğer bu bitlerden herhangi birisi veya daha fazlası lojik 1 ise bu bitlere karşılık gelen HSI bacaklarında çıkan veya düşen kenar sonucu oluştuğunu gösterir. HSI_STATUS kaydedicisindeki tek bitler ise HSI.0 - HSI.3 bacaklarındaki o anda devam eden durumu gösterir.

Tutma kaydedicisi zamanlayıcı 1 sayma değerini HSI_TIME kaydedicisine yükler. Böylece sayma sonucu HSI_TIME kaydedicisinden alınabilir. HSI_TIME kaydedicisini okuma, FIFO 'da sonraki sonuç verisinin tutma kaydedicisine yüklenmesine neden olur.



11.2.2. HSI modülünü programlama

Tablo 11.2. HSI modülünü programlamaya yarayan kaydedicilerin listesini gösterir.

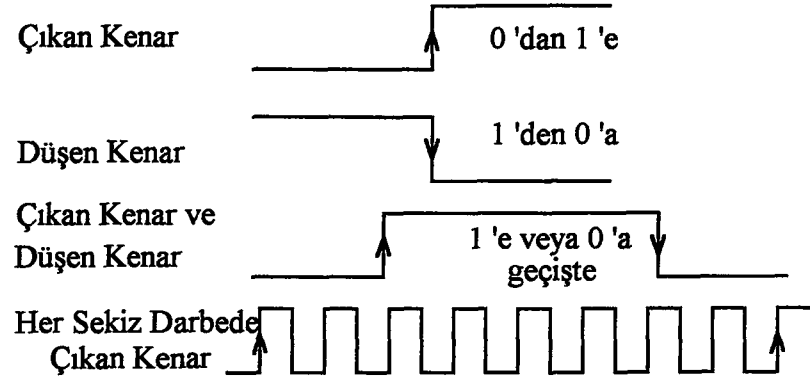
Tablo 11.3. HSI modülünü programlayan kaydediciler

Kayd. sembolü	Kaydedici adı	Tanımlama
HSI_MODE	HSI mod	Zamanlayıcıda sonuca neden olacak HSI bacaklarının kenar geçiş şeklini ayarlar.
HSI_STATUS	HSI 'daki durum	HSI bacaklarından hangisinin sonuca neden olduğunu ve HSI bacaklarının sürekli değişen konumlarını gösterir.
HSI_TIME	HSI zaman değeri	Sonuçtaki TIMER1'in 16 bitlik değerini içerir.
INT_MASK INT_MASK1	Kesme isteğini önleme Kesme isteğini önleme1	HSI kesme isteklerine izin verir veya önler.
IOC0	Giriş / Çıkış kontrol Kaydedicisi 0	Her bir HSI bacağına izin verir veya izin vermez. HSI.0 ve HSI.1 bacakları için diğer fonksiyonları seçer.
IOC1	Giriş / Çıkış kontrol Kaydedicisi 1	HSI 'da bilgi okunabilir kesme isteği için kesme isteği oluşturacak kaynağı seçer (INT 02H).
IOS1	Giriş / Çıkış durum kaydedicisi	HSI FIFO 'nun durumunu gösterir.

12.2.3. Kenar geçiş çeşitlerini tanımlama

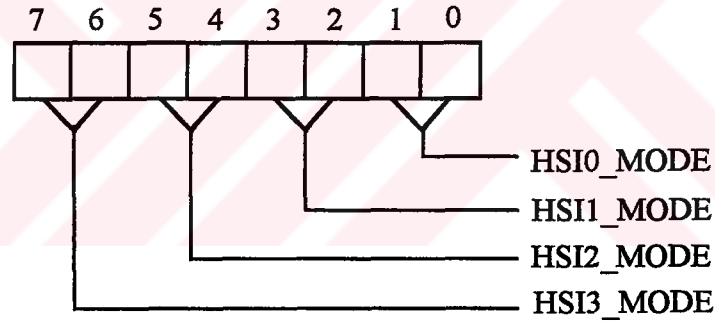
HSI_MODE kaydedicisinin denetleme özellikleri, her HSI bacağı için FIFO 'da sonuç bilgisini tutmak için dört ayrı kenar geçişini tanımlar.





Şekil 11.1. HSI 'da sonuç oluşturan kenar geçişleri

Bu kenar geçişleri çıkan kenar, düşen kenar, çıkan kenar ve düşen kenarın her ikisinde veya her sekiz darbeden sonra çıkan kenar geçişleri olarak ifade edilir (Şekil 11.1).



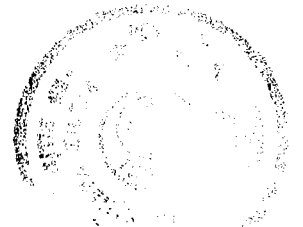
Şekil 11.2. HSI_MODE kaydedicisi

HSI bacaklarına karşılık gelen girişlerin kenar geçişlerinin nasıl ayarlanacağı şekil 11.2 ve tablo 11.4 'de gösterilmektedir.

11.2.4. HSI kesme isteklerine izin verme

HSI modülü aşağıdaki her sonuçta kesme isteği üretebilir.

- Dördüncü sonuç FIFO 'ya transfer edilirse,
- Altıncı sonuç FIFO 'ya transfer edilirse,
- Bir sonuç FIFO 'dan Tutma kaydedicisine yüklenirse



Tablo 11.4. Kenar geiş modlarını seçme işlemi

HSI_MODE	Tanımlama
00	HSI_FIFO 'ya her sekiz darbeden sonra çıkan kenar, sayma sonucunu transfer eder.
01	Aynı durum her çıkan kenarda oluşur.
10	Aynı durum her düşen kenarda oluşur.
11	Aynı durum her düşen ve çıkan kenarda oluşur.

INT_MASK ve INT_MASK1 kaydedicilerin bitleri ilgili kesme isteğine izin vermek veya kesme isteğini önlemek için kullanılır. Aşağıdaki tablo ilgili kesme isteğinin nedenini, kesme isteği listesini, kesme isteği nosunu ve öncelik sıralarını belirtir.

Tablo 11.5. HSI'daki kesme isteği isimleri ve nedenleri

Kesme isteği Kaynağı	Üretilme nedeni	Kesme isteği adı ve nosu	Önce lik
HSI_FIFO dolu	INT_MASK1.6 = 1 ve altıncı sonuç FIFO 'ya yüklenirse.	HSI_FIFO dolu (INT 14)	14
	IOC1.7 = 1, INT_MASK.2 = 1, INT_MASK1.6 = 0 ve altıncı sonuç FIFO 'ya yüklenirse	HSI 'da bilgi okunabilir (INT 02)	2
HSI_FIFO 4	INT_MASK1.2=1 ve dördüncü sonuç FIFO 'ya yüklenirse	HSI_FIFO (INT 10)	10
HSI.0	INT_MASK1.2=1, HSI.0 bacağındaki sinyalin çıkan kenarında ve bu sinyal son bir durum zamanında lojik 1 'de kalırsa. Bu kesme isteği bağımsız dış kesme isteğidir.	HSI.0(INT 04)	4
HSI'da bilgi okunabilir	IOC1.7 = 0, INT_MASK.2 = 1 ve sonuç FIFO 'dan tutma kaydedicisine yüklenirse. Bu kesme isteği işlemler için HSI'da en son bir sonuçun oluştuğunu gösterir.	HSI 'da bilgi okunabilir (INT 02)	2

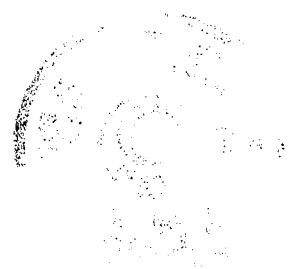
11.2.5. HSI bacaklarına izin verme

HSI bacaklarına karşılık gelen IOC0 kaydedicisindeki bitler lojik 0 veya lojik 1 yapılarak ilgili uçlara HSI giriş fonksiyonu olarak izin verilir veya verilmeyebilir.

IOC0 'daki ilgili bitlerin lojik 1 olması ile HSI geçişlerine izin verilmiş olur. Eğer bu bitler sıfırlanırsa HSI giriş fonksiyonlarına izin verilmez. Fakat HSI_STATUS kaydedicisindeki tek nolu bitlerden bu girişlerin konumu okunabilir ve HSI bacakları diğer ilave fonksiyonları yerine getirir. Bu ilave fonksiyonlar aşağıdaki tabloda gösterilmiştir.

Tablo 11.6. HSI bacaklarının ilave fonksiyonları

HSI bacağıнын adı	İlave fonksiyonları	Seçilme işlemi
HSI.0	HSI.0 dış kesme isteği (INT 04)	INT_MASK1.4 = 1 ise bu kesme isteğine izin verilir. (INT 04)
	Zamanlayıcı 2'yi sıfırlama kaynağı	IOC.5 = 1 ise HSI.0 zamanlayıcı 2 'yi sıfırlama bacağı olur. Çıkan kenar ile zamanlayıcı 2 sıfırlanır.
HSI.1	Zamanlayıcı 2 için saat darbesi kaynağı	IOC0.7 = 0 ve IOC3.0 = 0 ise HSI.1 zamanlayıcı 2 için saat darbesi kaynağı olur.
HSI.2	HSO.4	IOC1.4=1 ise bu bacak HSO.4 çıkış işlemini yerine getirir. HSI.2 ve HSO.4 aynı anda giriş ve çıkış işlemlerini yerine getirebilir.
HSI.3	HSO.5	IOC1.6 =1 ise bu bacak HSO.5 çıkış işlemini yerine getirir. HSI.3 ve HSO.5 aynı anda giriş ve çıkış işlemlerini yerine getirebilir.



12.FP 3000 FUZZY İŞLEMCI

Fuzzy işlemci, donanım kullanarak çok hızlı bulanıklaştırma, çıkarım ve netleştirme yapabilen OMRON firmasının üretmiş olduğu bir tümdevredir,(Omron, 1991a). FP 3000, 24 MHz saat frekansında 5 girişi 2 çıkışı olan 20 kuralı, ağırlık merkezini bulma yöntemini kullanarak sonucu bulma zamanı 650µs'dir. Bu işlemci mikro işlemciler veya mikrodenetleyiciler ile uyumlu çalışabilmektedir,(Omron,1991b). İşlemci işlemleri tamsayılarla yaptığı için mikrodenetleyiciler için yazılan makina dili programlarda kolaylık sağlar. 12 bit hassasiyetinde işlem yapabilme kapasitesine sahiptir. Fuzzy işlemcinin yapmış olduğu bulanıklaştırma, çıkarım ve netleştirme işlemini yazılımı kullanarak bir bilgisayar yardımı ile yapmak mümkün olabilir. Fakat bu durum hız gereken işlemlerde zaman kaybettireceği için uygun olmayan yönleri vardır. Ayrıca mikrodenetleyici veya mikro işlemcilerde yazılımı kullanarak bu işlemlerin yapılması daha zahmetli olmaktadır.

İşlemciye sekiz tane giriş değeri vermek ve sonuçta netleşmiş olarak dört çıkış değeri almak mümkündür. Bu durum yazılacak olan kuralların sekiz girişinin ve dört çıkışının olmasını sağlar. Ayrıca herbir giriş veya çıkış için her etiket bir bulanık alt kümeyi göstermesi şartıyla yedi tane sözel etiket belirlenebilir. İşlemcide fuzzy alt kümeler için kullanılacak üyelik fonksiyonları üçgen ($\wedge$), yamuk(∇) ve teknokta($\perp$) şeklinde olabilir. Fuzzy işlemciye kurallar ve üyelik fonksiyonu ile ilgili verileri, giriş ve çıkış ile ilgili verileri, işlemcinin meşgul olma durumunu gösteren verileri içeren 8Kbaytlık RAM bağlanabilir.

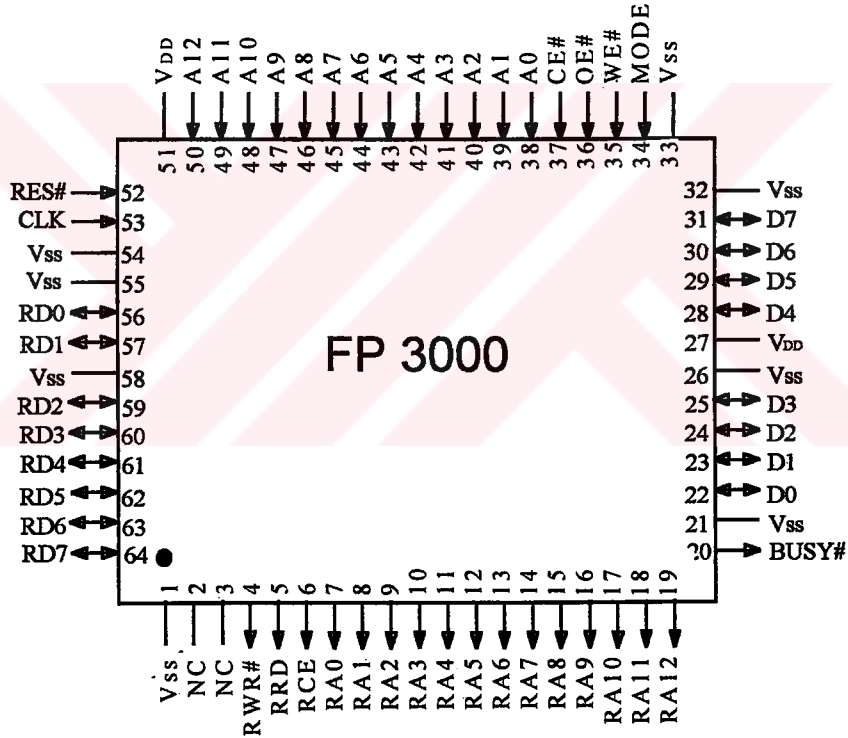
Bu işlemçi, çıkarım yönteminde max-min kompozisyonunu kullanır ve netleştirme işleminde ise maksimum ağırlığı alma veya ağırlık merkezini bulma yöntemlerinden birini kullanır.



FP 3000 'nin MODE bacağı ile denetlenebilen tek mod ve dış mod olarak adlandırılan iki ayrı modu vardır. Tek modda beş giriş ve iki çıkış kısmı olan en fazla 29 kuralı işleyebilir. Dış modda ise sekiz giriş ve dört çıkış kısmı olan en fazla 128 kuralı işleyebilme kapasitesine sahiptir.

FP 3000 besleme gerilimi +5 Volt kullanır ve dış kılıf şekli yüzey montajlıdır.

12.1.FP 3000 Fuzzy işlemcinin bacak bağlantıları ve tanımları



Şekil 12.1. FP 3000 fuzzy işlemcisinin bacak bağlantıları

A0 - A12 (Giriş) :13 bitlik adres yolu.Mikrodenetleyicinin adres yoluna bağlanırlar.

RA0-RA12(Çıkış): 13 bitlik adres yolu. Fuzzy işlemcinin çıkarım yapması için gerekli bilgilerin saklandığı RAM 'ın adres yoluna bağlanırlar.

D0-D7 (Giriş-Çıkış): 8 bitlik Veri yolu. Mikrodenetleyicinin veri yoluna bağlanırlar.

RD0-RD7 (Giriş-Çıkış): 8 bitlik Veri yolu. RAM 'ın veri yoluna bağlanırlar.

OE# (Giriş): Okunma işlemi. Dış ünitenin işlemciyi okuması için işaret giriş bacağı. Mikrodenetleyicinin RD# bacağına bağlanır.

RRD# (Çıkış): Dış üniteyi okuması için gerekli işareti üretir. Fuzzy işlemciye bağlanan RAM'in okuma girişine bağlanır.

WE# (Giriş): Dış ünitenin işlemciye yazma işlemi yapabilmesi için işaret giriş bacağı. Mikrodenetleyicinin WRL# bacağına bağlanır.

RWR# (Çıkış): İşlemcinin RAM 'e yazma işlemi yapabilmesi için gerekli işareti üretir. RAM'in WR# bacağına bağlanır.

CE# (Giriş): İşlemciyi aktif duruma getirmek için seçme işlemi.

RCE# (Çıkış): RAM 'ı aktif duruma getirmek için seçme işlemi.

CLK (Giriş): 24 MHz saat darbesi giriş bacağı.

RES# (Giriş): İşlemciyi ilk şartlarına getirmek için sıfırlama bacağı.

BUSY (Çıkış): Çıkarım işleminin bitip bitmediğini tespit etme. Çıkarım tamamlandığında mikrodenetleyiciye kesme isteği işaretini gönderir.

MODE (Giriş): Tek mod veya dış modu seçme. Lojik 0 ise tek mod, lojik 1 ise dış mod seçilir.

V_{DD}: + 5 Volt

V_{SS}: GND

12.2. Sistemin donanımı

Mikrodenetleyicinin adres yolundaki A0 adres bacağı gerek diğer ünitelerde gerekse fuzzy işlemcili sistemde kullanılmamıştır. Bu durumda 16 bitlik veri yolundan dış belleğe 16 bitlik veri alt ve üst baytlar olmak üzere ayrı ayrı iki bayt şeklinde yazılır. Çift adrese 16 bitlik veri yazılırken önce alt bayt yazılır bir sonraki adrese üst bayt yazılmaktadır. Çünkü her adres gözü bir baytı gösterir. Alt bayt

yazılırken mikrodenetleyici WRL#, üst bayt yazılırken WRH# işareti üretir. WRL# işareti çift adresler için WRH# işareti tek adresler için oluşturulur. Bu yüzden Fuzzy işlemci 16 bitlik veri yolunun sadece alt sekiz baytını okuyabilir. Çünkü işlemcinin veri yolu 8 bittir. Mikrodenetleyicinin WRL# bacağı işlemcinin WR# bacağına bağlanır. Bu şekilde sadece çift adreslerde işlem yapılır. Dolayısıyla fuzzy işlemci bellekte 16 KBaytlık yer tutar. Fakat bu bölgenin tek adreslerle temsil edilen yarı bölgesi kullanılmaz. İşlemciye gerekli verileri yazarken veya işlemciden gerekli verileri okurken bu duruma dikkat etmek gerekir.

Fuzzy işlemciyi seçmek için kullanılan CE# işareti adres seçici devreden gelir. Tasarlanan donanıma göre bu işaret 8000H-BFFFH adresleri arasında lojik 0 olur ve işlemciyi aktif duruma getirir. Bu bölgenin sadece çift adresleri kullanılır.

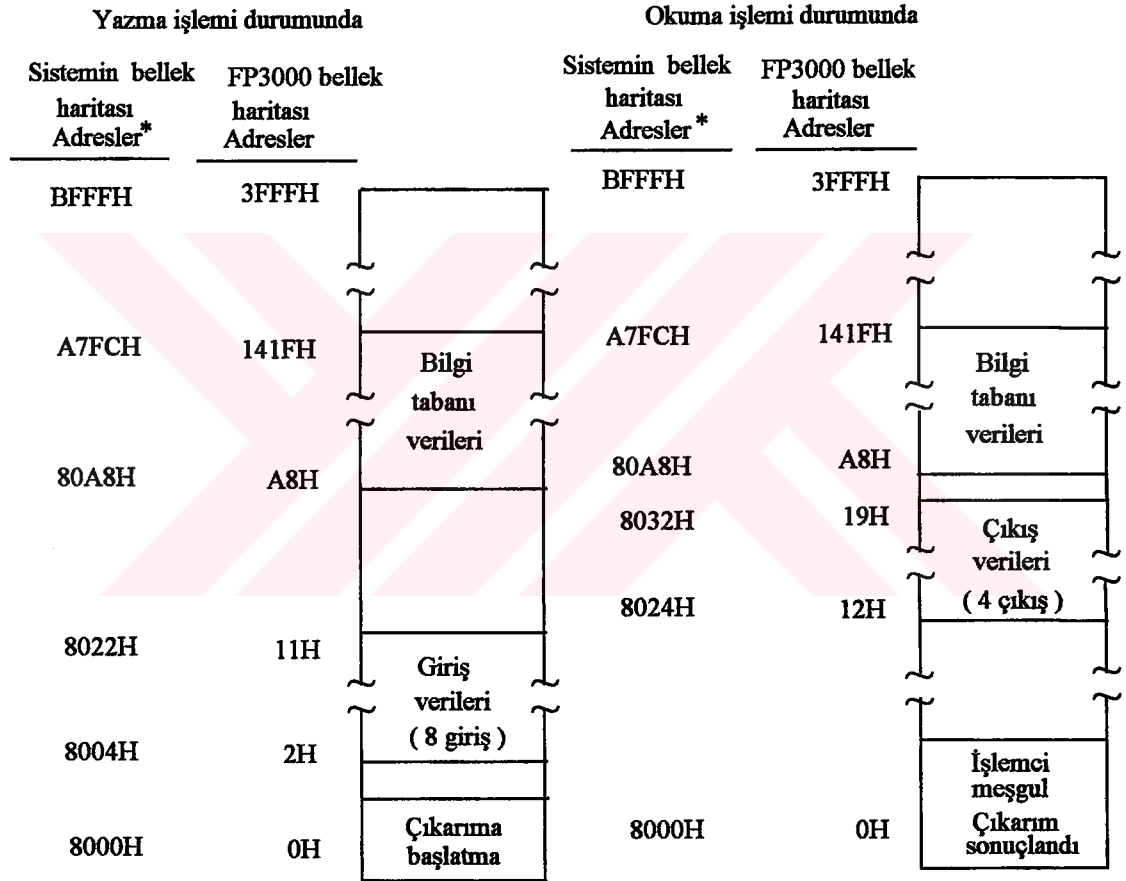
FP 3000 'nin veri yolu girişi 74HCT245 tümdevresi ile kuvvetlendirilir (Şekil 12.2). Eğer bu tümdevre kullanılmaz ve sadece CE# kullanılarak fuzzy işlemci seçilmeye çalışılırsa veri yolu mikrodenetleyicinin veri yoluna yük olarak geleceği için sistemin çalışmamasına neden olur. Bu tümdevre giriş ve çıkış yapılabilecek şekilde çift yönlüdür. Bu kuvvetlendiriciye izin verme (G#) işlemi aynı zamanda fuzzy işlemciye bağlı olan CE# işareti ile sağlanır. Veri yolunun işlemciden denetleyiciye veya denetleyiciden işlemciye doğru olacağı kuvvetlendiricinin DIR bacağı ile belirlenir. DIR bacağına mikrodenetleyicinin RD# bacağı bağlanır. Eğer DIR ve G# lojik 0 ise veri yolu fuzzy işlemciden mikrodenetleyiciye, şayet G# lojik 0 ve DIR lojik 1 ise veri yolu mikrodenetleyiciden fuzzy işlemciye doğrudur.

Devrede Mod bacağı lojik 1 durumundadır. Böylece sistem dış moda ayarlanmış olup en fazla 128 kuralı işleyebilir. Ayrıca devre üzerinde her iki modu seçebilmek için kısa devre elemanı ilave edilmiştir. Bu çalışmada sürekli dış mod kullanılmıştır.



Saat derbesi üreticinin ana kısmı 74HCT04 tümdevresidir. Frekans 24 MHz 'e bir kristal vasıtasıyla ayarlanmıştır. Osilator çıkışındaki evirici saat derbesi işaretini kuvvetlendirerek işlemciye verir.

FP 3000 'nin BUSY# işareti sistemin genişleme yuvalarına bağlıdır. İstenilirse işlemci çıkartım işlemini tamamladığında mikrodnetleyiciye kesme isteği gönderilebilir. Bu sistemde çıkartımın



* Yalnız çift adresler kullanılır.

Şekil 12.3. FP 3000 ve sistemin bellek haritası

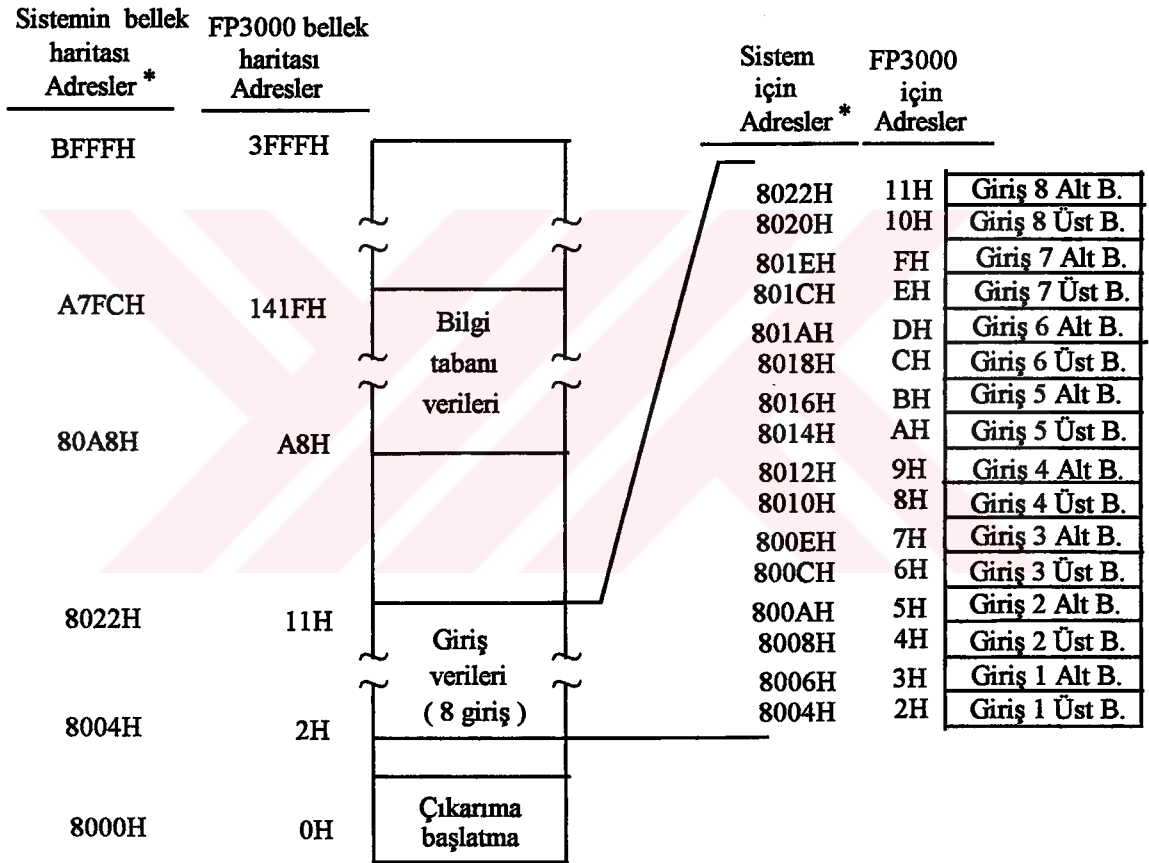
tamamlandığını anlama işlemi yazılım ile denetlenir. Kesme isteği işlemi kullanılmaz. Ayrıca işlemcinin bacaklarının mikrodnetleyiciye bağlanma durumu Şekil 12.2 'de gösterilmektedir.

12.3. Fuzzy İşlemcili Sistemin Bellek Haritası

Bellek bölgesi giriş, çıkış verilerini ayrıca kuralların ve üyelik fonksiyonun bir araya getirilmesi ile oluşturulan bilgi tabanı verilerini içerir. Ayrıca bu bölgede FP3000 'ni çıkarıma başlatmak ve çıkarım işleminin tamamlandığını anlamak için bir bayt bulunur(Şekil 12.3.).

12.3.1 Giriş verilerinin yazılması

Yazma işlemi durumunda



* Yalnız çift adresler kullanılır.

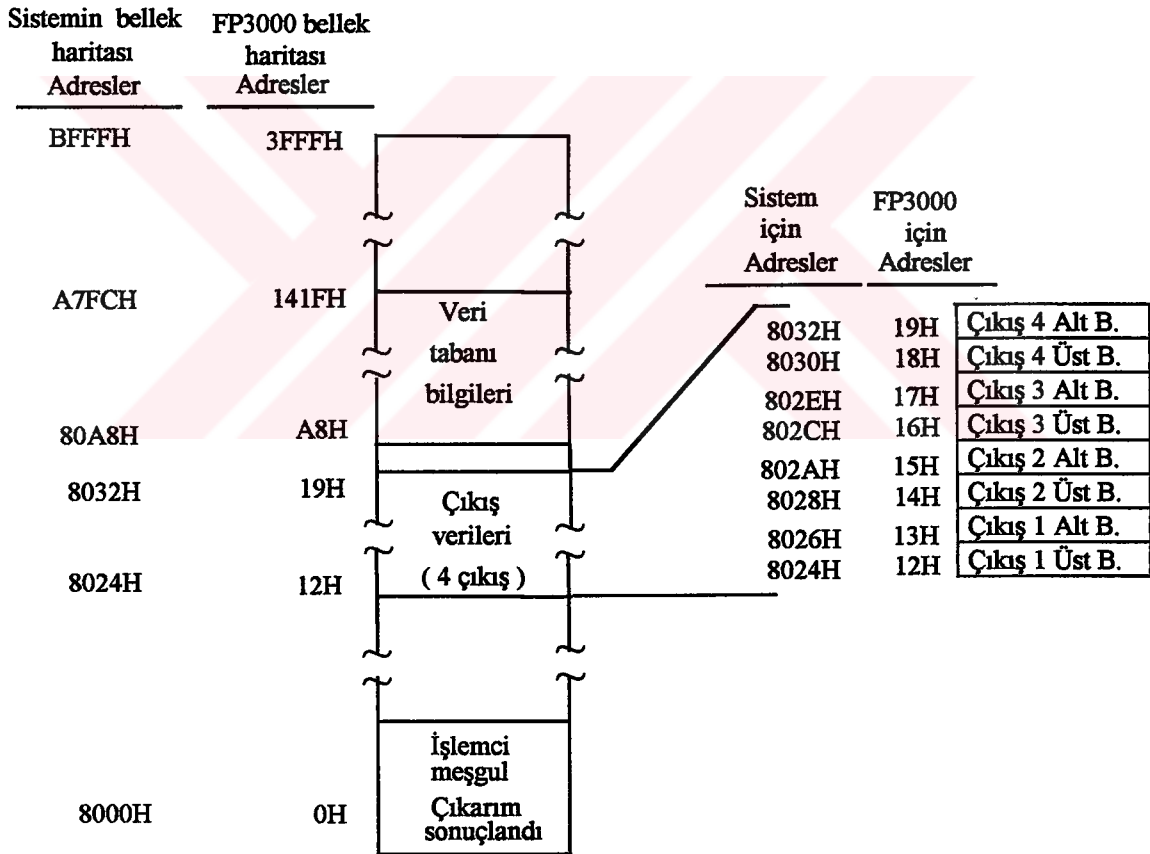
Şekil 12.4. Giriş için bellek düzeni

Şekil 12.4.'den görüleceği gibi giriş verileri 8004H-8022H adresleri arasındaki bellek gözlerine yazılır. Her giriş 12 bit olduğu için bellek iki baytlık yer kaplar. Giriş verileri yazılırken ilk çift adrese

önce giriş verisinin üst baytı yazılır .Bu baytın üst dört biti kullanılmayacağı için sıfırlanması gerekir. Bir sonraki adrese giriş verisinin alt baytı yazılır. Daha önce anlatıldığı gibi donanımın özelliğinden dolayı sistemde tek adresler kullanılmaz. FP 3000 'ne yazılacak kuralın sekiz tane girişi olduğu için girişlerin bellekte kullandığı göz sayısı en fazla 16 bayttır. Eğer kullanılmayan tek olan adreslerde sayılırsa göz sayısı 32 bayt olur.

12.3.2.Çıkış verilerinin okunması

Okuma işlemi durumunda



Şekil.12.5. Çıkış için bellek düzeni

Dört ayrı çıkış verisi 8024H- 8032H adresleri arasındaki çift adreslerden okunur (Şekil.12.5.).Her bir çıkış verisi 12 bit uzunluğundadır.İlk çift adreste (8024H) ilk çıkışın üst baytı vardır. Bu

adresten sonraki çift adreste aynı çıkışın alt baytı mevcuttur. Bir kuralda en fazla 4 çıkış olduğu için çıkış verileri bellekte en fazla 8 baytlık yer tutar. Eğer kullanılmayan tek adreslerde ilave edilirse çıkış için ayrılan bellek göz sayısı 16 baytlık olur. Bellekten alınan bir çıkış verisinin üst baytının üst 4 biti sıfır olması gerekir. Eğer bu 4 bit sıfır değilse elde edilen çıkış sonucu kullanılmaz. Yani o andaki giriş verilerine karşılık, kurallardan ve üyelik fonksiyonundan çıkarım ile bulunacak sonuç verisi mevcut değildir.

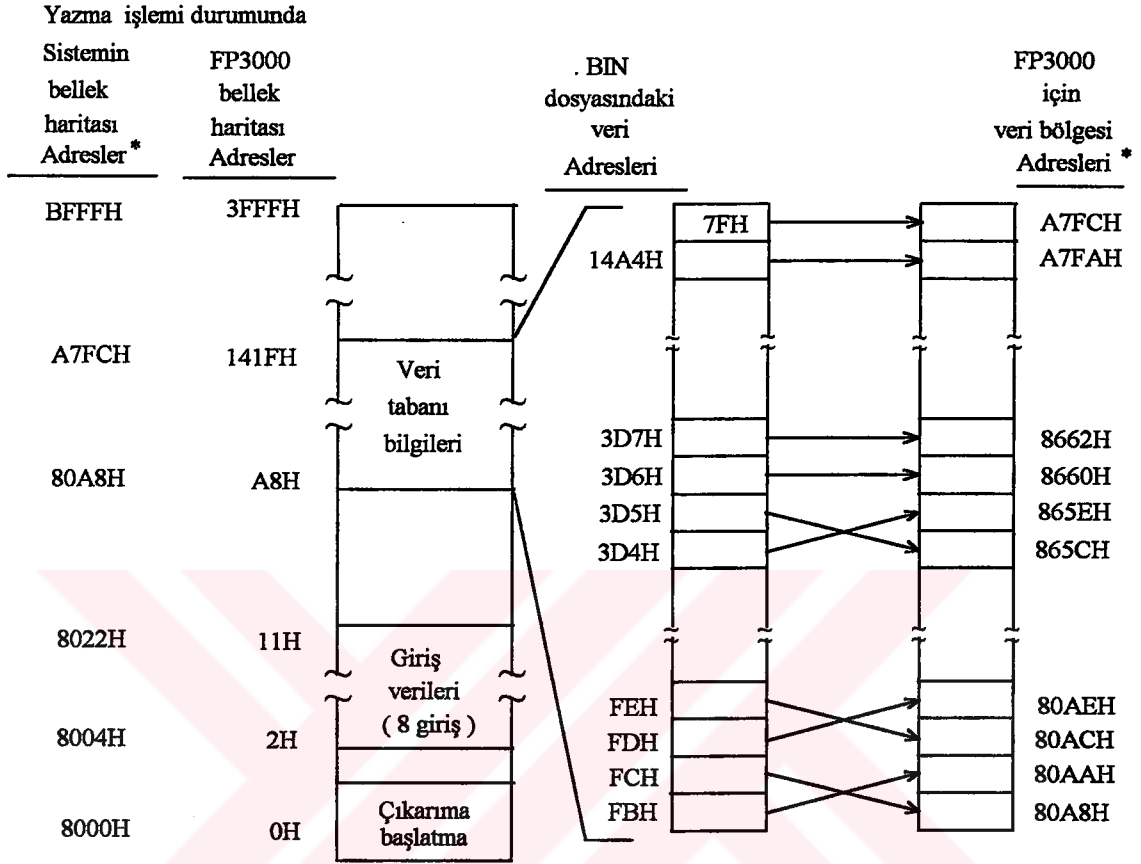
12.3.3. Çıkarım işlemine başlatma ve çıkarımın bittiğini tespit etme

0000H adresindeki bellek gözü bu işlemleri sağlamak için kullanılır. Bellek gözünün 4.biti bu iki işlemi gerçekleştirmeye yarar. Yazma durumunda işlemci çıkarım işlemine başlayabilir. Okuma durumunda ise çıkarım işleminin tamamlandığı anlaşılabılır. Gerekli giriş verileri belleğe yazıldıktan sonra 0000H bellek gözünün 4. biti lojik 1 yapılır. Bu durumda çıkarım işlemi başlar. Sonra aynı bellek gözünün aynı biti okunur. Eğer bu bit lojik 1 ise çıkarım işlemi bitmemiş ve işlemci meşgul durumdadır. Lojik 0 olursa çıkarım işleminin tamamlandığı anlaşılr. Çıkarım sonucunda ilgili adreslerden çıkış verileri alınır.

12.3.4. Bilgi tabanı verileri

Bu bilgiler 80A8H-A7FCH adresleri arasında çift adreslere yazılır. Bilgisayarda FB-30AT fuzzy çıkarım kartı için kullanılan FS-30AT Fuzzy Çıkarım yazılımında kurallar ve üyelik fonksiyonları yazılır. Daha sonra seçilen netleştirme yöntemine göre bu kurallar üyelik fonksiyonları ile bir araya getirilerek derleme yapılır. Derleme işlemi sonucunda bilgi tabanı olarak kullanılacak gerekli veriler elde edilir. Bu bilgilerin saklandığı dosyanın uzantısı *. BIN şeklindedir.





* Yalnız çift adresler kullanılır.

Şekil 12.6. Veri tabanı için bellek düzeni

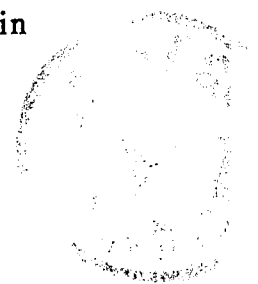
Daha sonra bu dosyadan fuzzy işlemcinin çıkarım işlemini yapabilmesi için gerekli bilgiler alınır.*. BIN adlı dosyadaki FBH-3D5H arasındaki her bayt bir sonraki adres gözündeki bayt ile yeri değiştirilir. Daha sonra FBH-14A4H adresleri arasındaki dosya bilgileri fuzzy işlemcide karşılık gelen 80A8H-A7FBH arasındaki çift adres gözlerine seri porttan gönderilir. En sonunda işlemcinin A7FCH adres gözüne sonu belirtmek için 7FH verisi yazılır (Şekil 12.6.).Gönderilen veriler Fuzzy işlemcili sistemdeki belleğe kaydedilir. İşlemciye bağlı bellek tümdevresi pil ile beslenecek olursa bilgi tabanı için gönderilen veriler sürekli olarak mikrodenetleyicili sistemde saklanabilir.

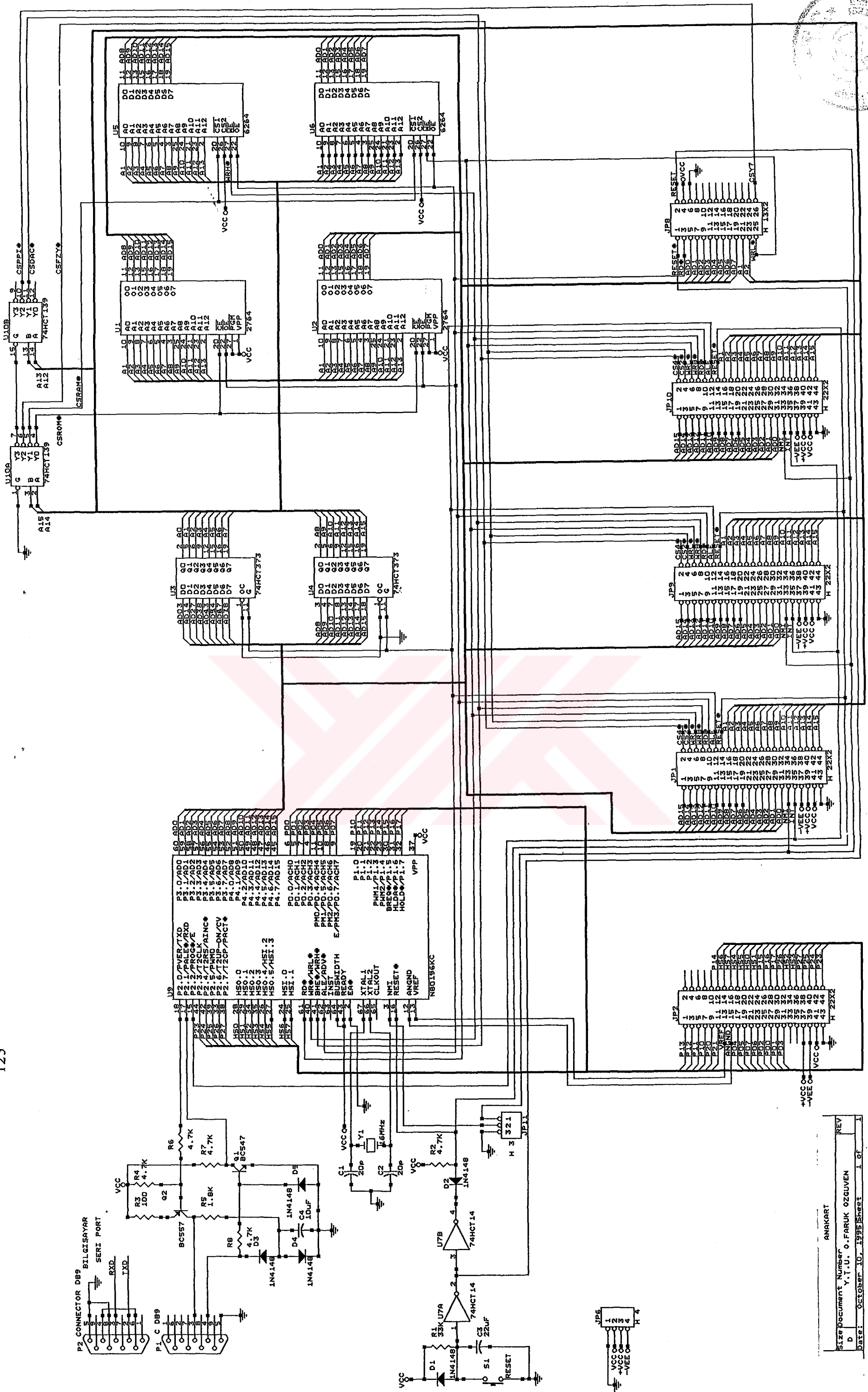


13.MİKRODENETLEYİCİLİ SİSTEMİN ANAKART DONANIMI VE DİĞER ÜNİTELER

Ana kartta mikrodnetleyici, adres tutma işlemi için tutma tümdevreleri, RAM ve EPROM bellekleri, tümdevre seçme işlemi için seciçi devre, sistemi sıfırlama devresi, seri giriş ve çıkış için arabirim devresi ve genişleme yuvalarından oluşur. (Şekil 13.1) Genişleme yuvaları üç çeşit olarak tasarlanmıştır. Devrede birinci tür genişleme yuvasından üç tane vardır. Bunlara A1-A15 adres yolu, AD0-AD15 veri yolu, tek ve simetrik besleme kaynağı uçları ve ayrıca kontrol uçları bağlanmaktadır (Şekil13.2.a). Dolayısıyla istenilen amaca göre kullanılacak kartlar bu yuvalara bağlanabilir. İkinci tür genişleme yuvası bu sistemde gösterge ve tuş takımını denetlemek için kullanılır (Şekil 3.2.b). Bu yüzden adres ve veri yolu ile beraber kontrol uçlarının sadece bu işlem için işe yaracak kısımları bağlanmıştır. Üçüncü tür genişleme yuvası ise kullanılması muhtemel port uçlarını ve diğer işlevleri olan bir takım uçları içerir (Şekil. 13.2.c)

Mikrodnetleyicili sistem dinamik adres/veri yolunu kullanan 16 bitlik yapıya sahiptir. (Şekil 10.8). Yani adres ve veri yolu 16 bit olarak çalışır.Bu yüzden CCR kaydedicisinin sıfırlama işlemi sırasında EPROM 'dan 2018H adres gözünden okuduğu tümdevreyi ayarlama baytının (CCB) 2.biti lojik 1 olarak belirlenir. Bu durumda mikrodnetleyicinin BUSWIDTH bacağının durumu yol genişliğini belirler ve 16 bitlik yol için lojik 1 durumuna getirilir.Bu durumda sistem bir defada 16 bit okuduğu için daha hızlı çalışma özelliğine ulaşır.Bu adres/veri yolları port 3 ve port 4'ü paylaşmalı olarak kullanır.Bu portların çıkışında 16 bitlik adres yolunu tutmak için iki tane 74HCT373 tümdevresi kullanılarak sağlanmıştır.74HCT373 tümdevrelerinden biri adresin alt baytını ve diğeri de üst baytını tutmak için kullanılır. Girişlerdeki geçerli adresi tutma işleminin

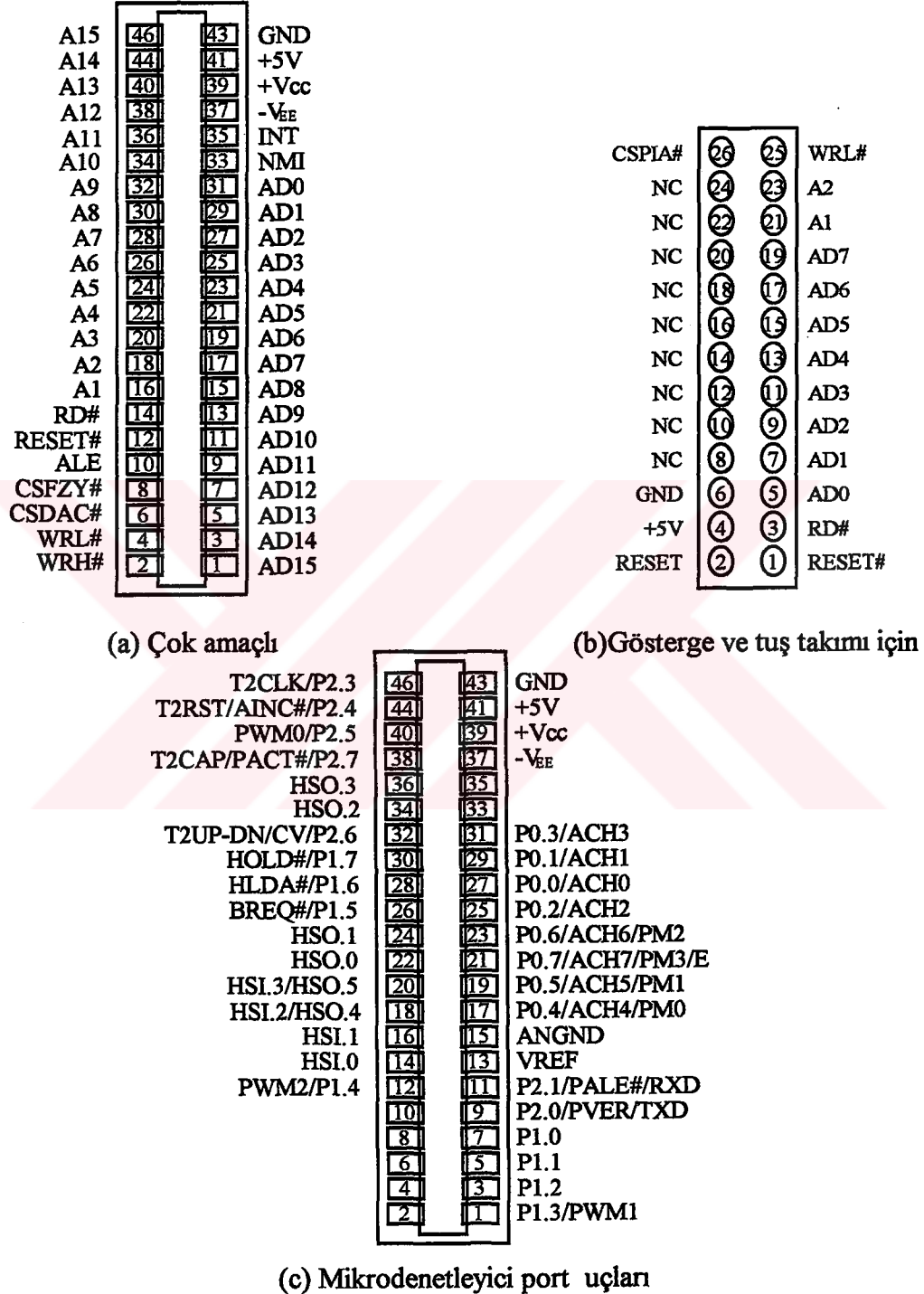




Size	Document Number	REV
D	Y.T.U. O.FARUK OZGUVEN	1
Date:	October 10, 1995	Sheet 1 of 1

Şekil 13.1. Anakart şeması

gerçekleştirilmesi mikrodenetleyicinin ALE bacağı ile sağlanır. Geçerli adres oluştuğunda ALE lojik 0 olur. Bu durumda tutma tümdevrelerinin



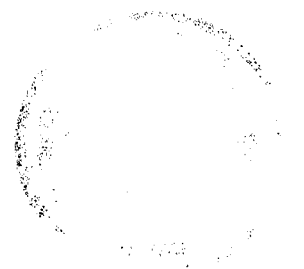
Şekil 13.2. Genişleme yuvası bacak bağlantıları

çıkışına o andaki adres bilgisi ulaşır. Daha sonra adres/veri yolunda veri bilgisi oluşursa ALE lojik 1 olacağı için tutma tümdevrelerinin girişi kapatılmış olur. Böylece sadece adres bilgisi tutulmuş olur.

Burada mikrodnetleyici programı dış EPROM bellekten okuduğu için EA# bacağı lojik 0 durumundadır. EPROM 'lar veri yolunun alt ve üst baytları için iki tane olarak kullanılır. Bu belleklerin adres yolları paralel ve veri yolları üst ve alt bayt olmak üzere seri yapıya sahiptir. OE# bacakları birbirine paralel bağlanarak mikrodnetleyicinin RD# bacağı ile denetlenir. Yani okuma işleminde üst ve alt baytlar bir defada veri yoluna alınır.

Yazılır ve okunur (RAM) belleklerin adres ve veri yolu bağlantıları EPROM'lar ile aynı yapıdadır. RAM okunurken EPROM 'da olduğu gibi üst ve alt baytlardan oluşan 16 bitlik veri bir defada okunur. RAM 'e 16 bitlik bir bilgi yazılması işlemi okuma işleminde olduğu gibi veri bir defada yazılmaz. Yani 16 bitlik verinin üst ve alt baytları ilgili belleklere ayrı bir şekilde yazılır. Alt belleğe alt bayt yazılırken mikrodnetleyici WRL# işaretini ve üst belleğe üst bayt yazılırken WRH# işaretini üretir. WRL# işareti alt belleğin WR# bacağına, WRH# işareti ise üst belleğin WR# bacağına bağlanmıştır.

Her adres bir baytlık bellek gözünü gösterir. Adres yolunun A0 bacağı kullanılmadığı için alt belleklerdeki gözler çift adresler ve üst belleklerdeki gözler ise tek adresler ile temsil edilirler. Mikrodnetleyicideki bir adresin EPROM ve RAM 'deki adres karşılığı farklıdır. Örneğin sıfırlama işlemi sırasında denetleyici 2018H bellek gözüne gider ve ilk komutu bu bellek gözünden alır. Fakat denetleyicinin adres yolunun A0 bacağı bağlı olamadığı ve A1 bacağı bellek ünitelerinin A0 bacaklarına bağlı olduğu için 2018H bellek adresi 1040H adresine denk gelir. Bu adres tablo 13.1 'deki gibi hesaplanır.



Tablo 13.1. Mikrodenetleyicideki adresin dış ünitedeki karşılığı

Adres	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	Hex. adr.
Mikroden.	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	
	2			0				8				0					2080H
Karşılığı	K14	K13	K12	K11	K10	K9	K8	K7	K6	K5	K4	K3	K2	K1	K0	-	
Dış ünite	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	-	
	1			0				4				0					1040H

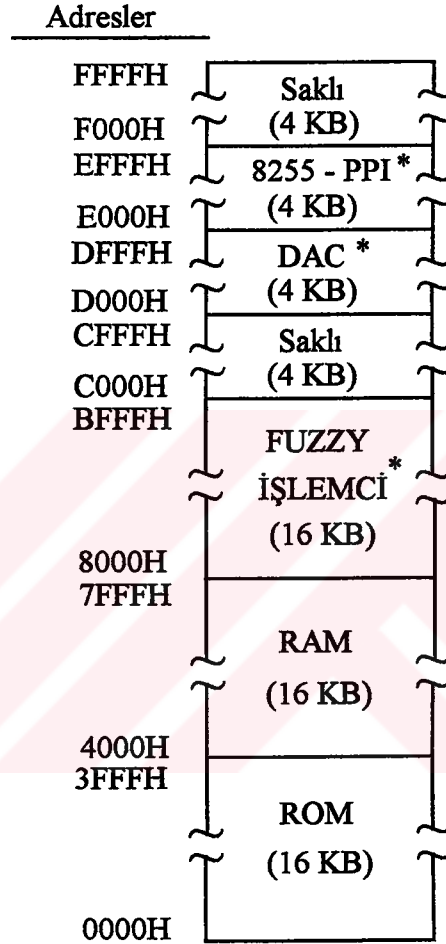
Bu tablodan da anlaşılacağı gibi mikrodenetleyicideki adres sola bir bit ötelenmektedir. Yani bu adresin dış ünitedeki karşılığı, iki sayısına bölünmesidir.

Sıfırlama devresinde 74HCT14 evirici tümdevresi kullanılmıştır. Bu eviriciler girişe bağlanan anahtar ile sıfırlama işlemi yapıldığı zaman mikrodenetleyici dışındaki ilgili diğer ünitelerin sıfırlamaları için kuvvetlendirme işlemini yaparlar. Her iki evirici çıkışında sıfırlama işaretinin lojik iki durumu mevcuttur. Dış ünitenin sıfırlanma durumuna göre aktif işaret alınabilir. Mikrodenetleyicinin sıfırlama bacağına bağlanan 4.7K Ω 'luk direnç evirici çıkışındaki gerilimi yukarı çekmek için kullanılmıştır.

Sistemde 8XC196KC kullanıldığından saat darbesi frekansı, kristal kullanılarak 16 MHz olarak belirlenir. Şayet denetleyici olarak 8XC196KD kullanılırsa saat darbesi frekansı 24 MHz olmalıdır.

Seri haberleşme için kullanılan arabirim devresi gönderme ve alma işlemi için kullanılan iki tranzistör mevcuttur(Elector, 1987). Gönderme işlemi için kullanılan BC557 tranzistörü ise denetleyiciden çıkan işareti kuvvetlendirerek çıkış hattına verir. Aynı zamanda bu tranzistör gönderilen işaretin evriğinin alınmasına neden olur. Çünkü bu işaret bilgisayar tarafından bir kez daha evriği alınır. Böylece normal işaret elde edilir. BC 547 tranzistörü bilgisayar tarafından

evriği alınarak gönderilen işaretin evriğini alarak mikrodenetleyiciye ulaşmasını sağlar. Bilgisayarın mikrodenetleyiciye gönderdiği işaret +9V ile -9V arasında değişir. Arabirim devresindeki BC547 ve diyotlar gelen bu işareti 0V ile 5V arasında sınırlar.



* Sadece çift adresler kullanılır.

Şekil 13.3. Anakart bellek bölgeleri

Adres seçme işleminde içinde iki tane 2 hattı 4 hatta çeviren çoğullayıcı bulunan 74HCT139 tümdevresi kullanılır. İlk çoğullayıcı 64 KBaytlık bölgeyi 16 Kbaytlık bölgelere ayırır (Şekil 13.3). Bu yüzden çoğullayıcının girişine A14 ve A15 adres uçları bağlanmıştır. İlk 16 KBaytlık bölgeye (0000H-3FFFH) program belleği olarak kullanmak

için herbiri 8 Kbayt olan iki EPROM yerleştirilmiştir. Sistem sıfırlandığında mikrodenetleyici ilk komutu 2080H adresinden alır. Dolayısıyla EPROM için ayrılan bölge bu adresi ve kesme isteği vektör adreslerini kapsamalıdır.

İkinci 16 Kbaytlık (4000H-7FFFH) bölgede herbiri 8 Kbayt olan iki tane 6264 RAM tümdevresi vardır. Bu belleğe yazılacak programın kodları 4000H adresinden başlayarak yazılabilir. Yalnız 7F00H ile 7FFFH adresleri arasındaki bellek gözleri geliştirme seti programı için bazı verilerin saklanması için kullanılır. 7FF0H ile 7F05H adresleri arasındaki bellek gözleri o andaki adres ve bu adres gözündeki 8 bitlik verinin onaltılı sayı değerinin yedi parçalı gösterge değerine çevrilmiş verilerini saklamak için kullanılır. Ayrıca 7F0AH, 7F16H ve 7F1CH adres gözleri tuş takımında bazı tuşlar için ileride yapılacak geliştirmeler için kullanılabilir. Bu nedenle sistem ilk açıldığında bu bellek gözlerine geridön (RET) komutunun kodu olan F0H yazılır.

Üçüncü bölge 8000-BFFFH adresleri arasındaki bellek gözlerini kapsar. Bölüm 12 'de anlatıldığı gibi bu bölge fuzzy işlemci için ayrılmıştır. Adresteki verilerin sadece çift adresli olanları kullanılır.

Dördüncü bölge olan C000H ile FFFFH adres bölgesi ikinci çoğullayıcı tarafından dörder KBaytlık bölgelere ayrılır. Bu çoğullayıcı çıkışındaki birinci ve dördüncü 4 KBaytlık bölgeler kullanılmamıştır. İkinci 4 KBaytlık bölge (D000H-DFFFH) Analog-Digital ve Digital-Analog çevriciler için kullanılır. Bu çeviriciler bu bölgeye yansımali olarak yerleştirilebilir. Ayrıca bölge için kullanılan seçici uç genişleme yuvalarına bağlıdır. İstenilirse başka amaç için kullanılabilir. İkinci çoğullayıcının çıkışında belirlenen üçüncü 4 KBaytlık bölge (E000H-EFFFH) gösterge ve tuş takımı devresi için ayrılmıştır.



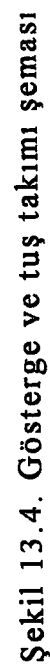
13.1. Gösterge ve tuş takımı devresi:

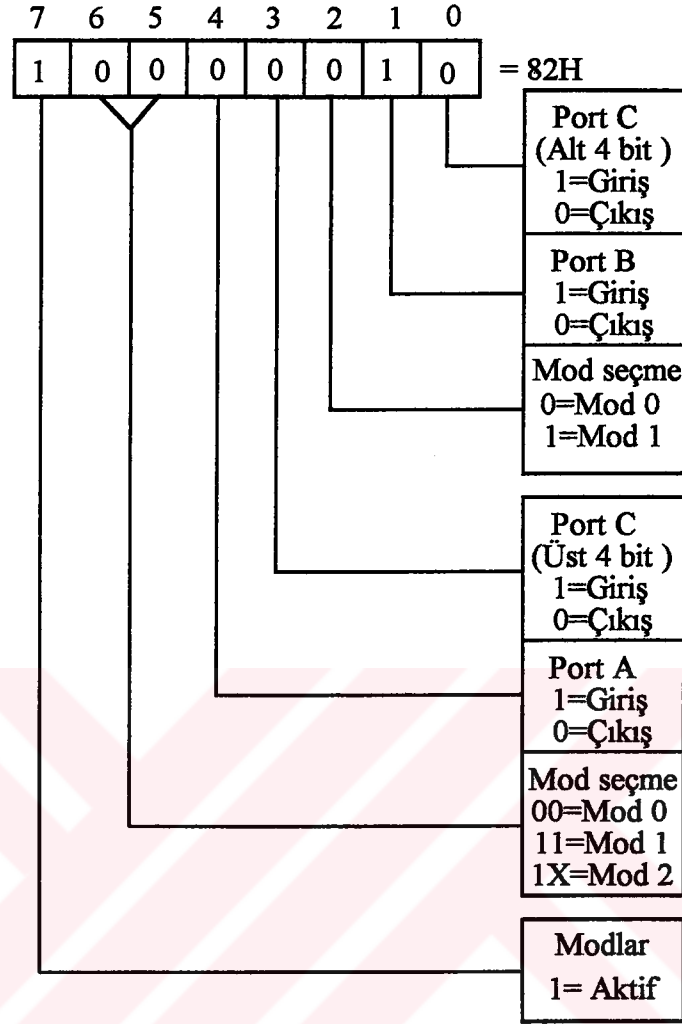
Bu devrede kullanılan 8255 paralel giriş çıkış arabirim tümdevresi yansımali olarak yerleştirilmiştir. 8255 fuzzy işlemciye benzer şekilde çift adreslerde bulunur. Yani 16 bitlik veri yolunun alt baytı kullanılır. Bu yüzden WR# bacağına denetleyicinin WRL# ucu bağlanır(Şekil 13.4). 8255 arabirim tümdevresinin üç giriş çıkış portu ve ayrıca bir kontrol kaydedicisi vardır. Portlara ve kaydedicilere 8255'in A0 ve A1 bacakları şartlanarak ulaşılır. Bu iki bacak mikrodenetleyicinin A1 ve A2 adres uçlarına bağlıdır. Tablo 13.2. ilgili adreslerde bulunan portları ve kaydedicileri gösterir.

Tablo 13.2. 8255 çevre birimi kaydedicilerinin adresleri

A2	A1	ADRES	FONKSİYONU
0	0	E000H	A portu
0	1	E002H	B portu
1	0	E004H	C portu
1	1	E006H	Kontrol

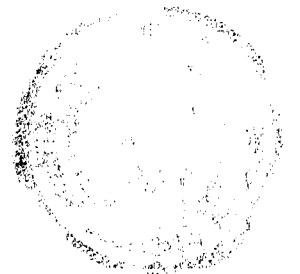
Gösterge ve tuş takımı devresinde A portu (PA0-PA7) gösterge parçaları, C portunun altı biti (PC0-PC5, 6 gösterge) göstergelerin katodları ve aynı zamanda tuş takımının kolonları (PC0-PC4, 5 kolon), B portunun beş biti ise tuş takımının satırları (PB0-PB4, 5 satır) için kullanılır. Ayrıca C portunun 7. biti hoparlörü sürer. Bu yüzden A ve C portu çıkış ve B portu giriş olarak şartlandırılır (Brey, 1987), (Şekil 13.5).





Şekil 13.5. Kontrol kaydedici bitlerinin fonksiyonları

A portunun gösterge parçalarına, C portunun göstergenin katotlarına ve tuş takımının kolonlarına, B portunun tuş takımının satırlarına bağlanma durumlarındaki ilgili uçlara karşılık gelen port uçları tablo 13.3. ve tablo 13.5. 'de gösterilir. Ayrıca A portu için göstergede elde edilecek 0H-FH arasındaki sayılar için gösterge şekline çevrilmiş değerler tablo 13.4. 'da gösterilir.



Tablo 13.3. C portunun katot ve sutunlarına karşılık gelen port uçları

PC5	PC4	PC3	PC2	PC0	PC0
Katot6	Katot5	Katot4	Katot3	Katot2	Katot1
X	Kolon5	Kolon4	Kolon3	Kolon2	Kolon1

Tablo 13.4. 0H - FH arasındaki sayıların gösterge karşılıkları

	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0	
	e	d	n	c	g	f	a	b	
0	1	1	0	1	0	1	1	1	D7H
1	0	0	0	1	0	0	0	1	11H
2	1	1	0	0	1	0	1	1	CBH
3	0	1	0	1	1	0	1	1	5BH
4	0	0	0	1	1	1	0	1	1DH
5	0	1	0	1	1	1	1	0	5EH
6	1	1	0	1	1	1	1	0	DEH
7	0	0	0	1	0	0	1	1	13H
8	1	1	0	1	1	1	1	1	DFH
9	0	1	0	1	1	1	1	1	5FH
A	1	0	0	1	1	1	1	1	9FH
B	1	1	0	1	1	1	0	0	DCH
C	1	1	0	0	0	1	1	0	C6H
D	1	1	0	1	1	0	0	1	D9H
E	1	1	0	0	1	1	1	0	CEH
F	1	0	0	0	1	1	1	0	8EH

Tablo 13.5. B portunun satırlara karşılık gelen port uçları

PB4	PB3	PB2	PB0	PB0
Satır5	Satır4	Satır3	Satır2	Satır1

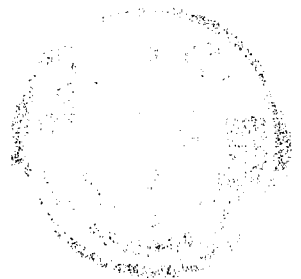
13.2.Dijital Analog (D/A) Çevirici Devre

Bu devre tümdevre seçici, dijital veriyi tutma ve tutulan veriyi analoga çeviren tümdevre ünitelerinden oluşur (Şekil 13.6). Adres seçici devrede 74HCT139 tümdevresi içerisindeki 2 hattı 4 hatta

çeviren iki çoğullayıcıdan biri kullanılır. Bu ünite D000H-DFFFH arasındaki 4 KBaylık adres bölgesini birer Kbaytlık dört bölgelere ayırır. D/A çevirici bu bölgelerden ilk bölgede (D000H-D3FFH) aktif duruma geçer. Fuzzy işlemci 12 bitlik olduğu için D/A çeviricide 12 Bitlik olarak seçilmiştir. Bu yüzden çevirici girişine bağlanan veri tutma devreside 12 bitliktir. Veri tutma işlemi 74HCT373 (8 bit) ve 74HCT75 (4 bit) tutma tümdevreleri ile gerçekleştirilir. Mikrodenetleyicinin WRH# bacağı sayesinde 16 bitlik verinin üst 8 bitinin alt 4 biti 74HCT75 tümdevresine yazılır. Ayrıca WRL# bacağı sayesinde 16 bitlik verinin alt 8 bitide 74HCT373 tümdevresine yazılır. D000H-D3FFH adresleri arasındaki herhangi bir adres gözündeki veriyi D/A çeviriciye ulaştırabilmek için WRL# ve WRH# bacakları ile 74HCT139 seçici devresinin Y0 çıkışı iki ayrı NOR kapısından geçirilir ve ayrı ayrı tutma tümdevrelerinin G (İzin verme) bacağına bağlanırlar. Böylece tümdevre seçme sinyali ürettiği zaman, WRL# ve WRH# sinyallerinden birisi var ise gerekli tutma tümdevresi gelen veriyi tutar ve D/A çevirici girişlerine verilmiş olur.

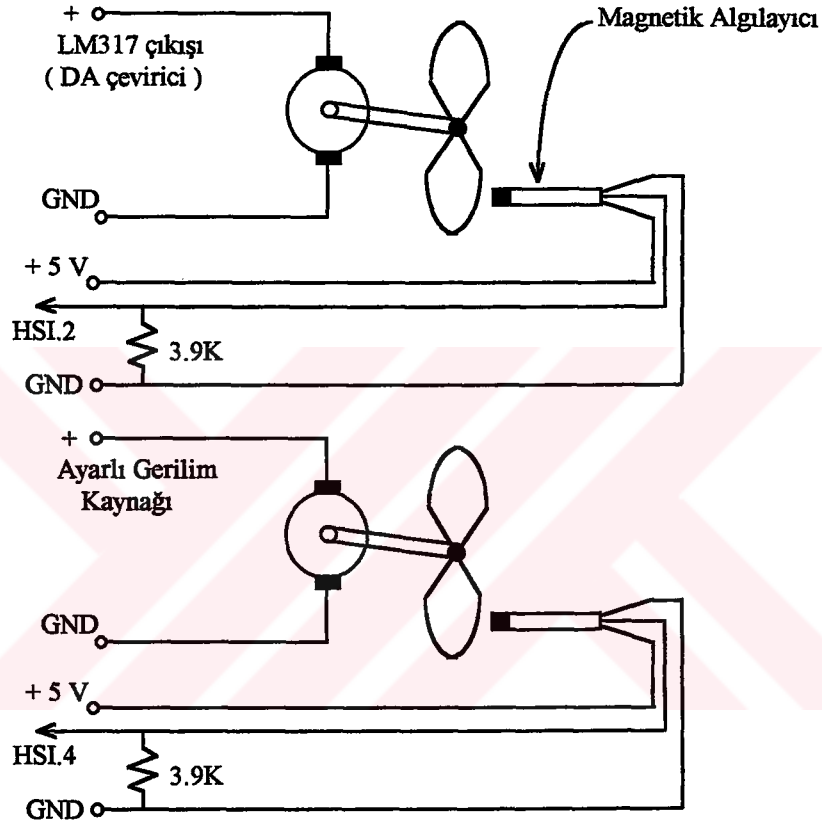
D/A çevirici olarak DAC 1200 tümdevresi kullanılmıştır. Tümdevrenin besleme gerilimi ± 12 Volttur. Çıkışında ± 10.24 Volt elde edilir. Eğer çıkışın simetrik olması istenmez ise D/A çevirici çıkışı 0-10.24 Volt olarak ayarlanabilir. Yani 12 bitin hepsi lojik 1 ise çıkış 0 volt ve 12 bitin hepsi lojik 0 ise çıkış 10.24 Volt olur. Bu durumda bir bitlik değişim $10.24V/4096 = 2.5$ mV değerine karşılık gelir.

Çıkışın tam ölçekleme ve ofset ayarı için devre üzerinde iki ayrı trimpot mevcuttur. Devre ± 12 V ve ayrıca 5 V gerilim kaynağı ile beslenir. D/A çıkışındaki gerilim ile motoru sürebilmek için LM 317 gerilim regülatörü ilave edilmiştir.



13.3. Motorlar ve Magnetik Algılayıcılar

Sistemde iki ayrı 12 Voltluk DC motor ve magnetik algılayıcı vardır(Şekil 13.6).Motorlardan birinin hızı mikrodnetleyici tarafından kontrol edilir. Diğer motorun hızı sabit ve isteğe bağlı olarak değiştirilir.

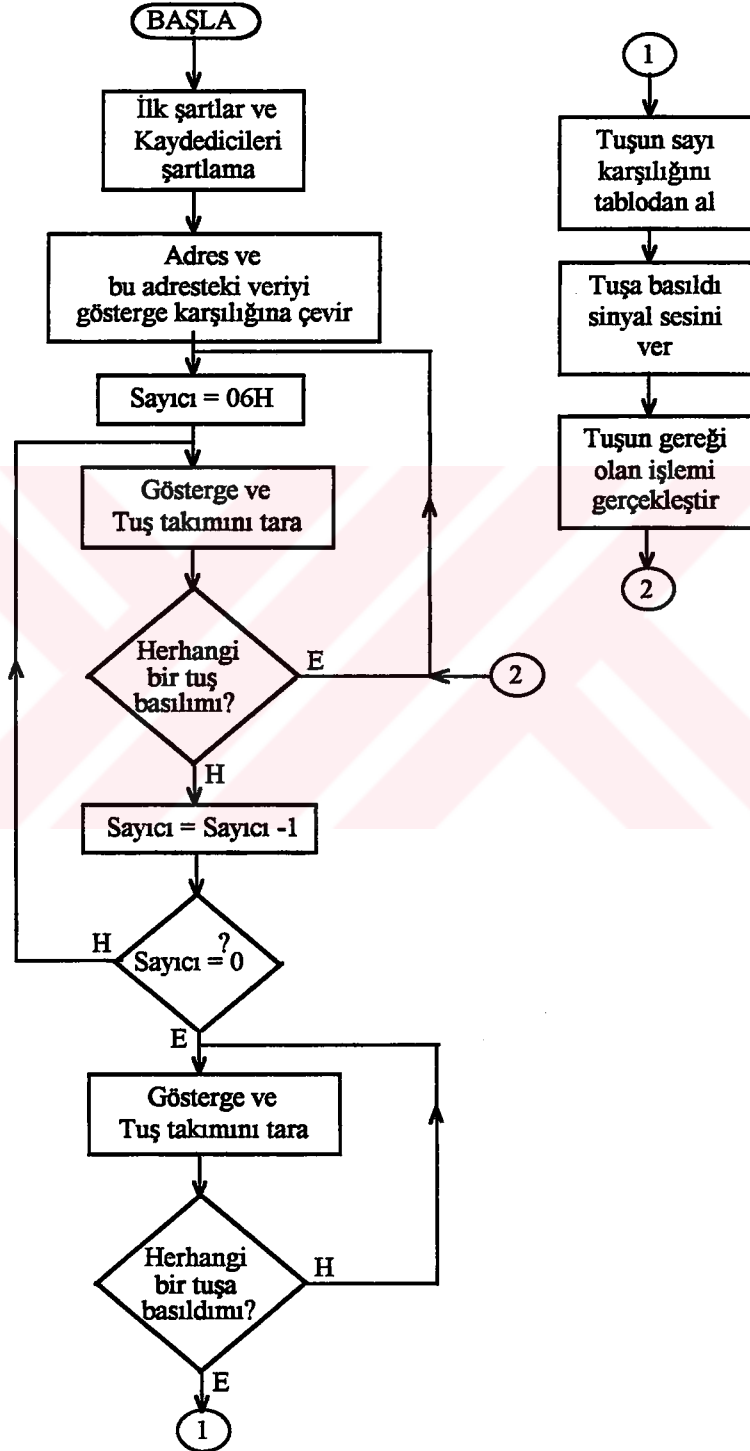


Şekil 13.6. Motorların ve magnetik algılayıcıların bağlantıları

Motorların dönüş hızları magnetik algılayıcılar tarafından mikrodnetleyiciye ulaştırılır.Hızı kontrol edilen motorun dönüş hızı bilgisi mikrodnetleyicinin HSI.2 bacağına ve hızı sabit olan motorun dönüş hız bilgisi ise HSI.3 bacağına verilir. HSI.2 ve HSI.3 girişlerinden gelen darbelerin çıkan ve düşen kenarlarına göre zamanlayıcı1 'deki sayma sonuçları değerlendirilir.

14. PROGRAMLAR

14.1.Gösterge ve tuş takımını kullanan geliştirme seti programı

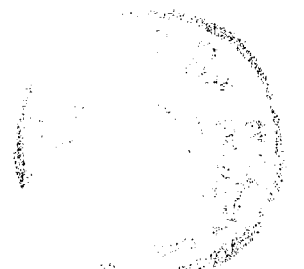


Şekil 14.1. Geliştirme seti programı akış diyagramı

İlk şartlar verilirken ve kaydediciler şartlanırken önce Dpencere 32 bayta ve Ypencere ise 0 nolu pencereye şartlanır. Ayrıca Yığın Gösteren Kaydedici 0200H adresine şartlandırılır. 8255 'in A portu gösterge parçalarına ve C portu katotlara bağlı oldukları için her iki port çıkış, B portu tuş takımının satır bilgisini okuduğu için giriş olarak şartlandırılır. Seri haberleşme için saniyedeki karakter hızı, haberleşme modu ayarlanır. Aynı zamanda ileride ilaveler için boş bırakılan iki tuşa basıldığı zaman gidilen adreslere geridön (RET) komutunun kodu konulur.

RAM 'e yazılacak programların kodları 4000H adresinden başlanarak yazılabilir. Bu adresi ve bu adresteki bilgiyi göstergede görebilmek için gösterge karşılığı olan değere çevrilir. İlgili bayt bu değere çevrilirken üst ve alt dört bit ayrı ayrı alınır. Daha sonra bu dört bitin gösterge karşılığı EPROM tablosundan alınır. Bu bilgilerin saklama bölgesi olan 7F00H-7F05H adreslerinde saklanır. 7F05H adresi en sağdaki göstergeye ait verinin saklandığı ve 7F00H adresi en soldaki göstergeye ait verinin saklandığı bellek gözüdür.

Esas gösterge ve tuş takımı taramaya geçmeden önce ilk tarama yapılır. Bu ön tarama herhangi bir tuşa basılmış ve gereği olan işlem yapıldıktan sonra eğer aynı tuş halen basılı ise yine aynı işlemler yapılmasına engel olur. Yani tuşa basılırsa gereği olan işlem bir defa yapılır. Sonra eğer halen ilgili tuş basılı ise gerçek taramaya geçilmez. Yine bir işlem yaptırabilmek için tuşu bırakmak gerekir. Yanlış bırakıldıktan sonra geçici durumlardan dolayı bir takım istenmeyen darbeler oluşur. Böylece bu durum birden fazla aynı işlemlerin yapılmasına neden olur. Oluşan bu sakıncayı ortadan kaldırmak için sayıcı kullanılır. Programın akışına göre ancak en son alınacak tuş basıldı darbesinden sonra sayıcı birer alzalarak sıfırlanır. Sayıcı sıfırlandıktan sonra esas gösterge ve tuş taramasına geçilir.



Bu tarama işleminde herhangi bir tuşa basılırsa tuşun sayısal olarak karşılığı EPROM tablosundan alınır. Tuş tarama ve display tarama işlemi aynı anda gerçekleşir. Gösterge taramada önce en sağdaki göstergeden başlanır. İlgili göstergelerin, gösterge parçalarına A portundan veri gönderilir ve aynı göstergenin katodu C portunun ilgili biti ile lojik 0'e çekilir. C portunun çıkışlarına evrik alma ve sürücü tümdevresi bağlıdır. Bu durumda ilgili gösterge bir süre yanık olarak bekletilir. Her gösterge tarandığında programdaki tuş sayıcısı ise satırları sayma işlemini gerçekleştirir. Gösterge en sağdan taramaya başlanıldığı için tuş takımında aynı şekilde taranır. Programdan geri dönülürken tuşa basılmış ise bu tuşun sayıcı tarafından sayısal değeri saklanır. Bu durumdaki değerler kullanılırsa tuşların isteğe göre yerlerini değiştirmek mümkün olmaz. Bu yüzden alınan sayısal değer asıl değeri EPROM'da başlangıç adresi 3F10H olan tablodan alınır. Böylece istenilen tuşa isteğe göre numara vermek mümkün olur. Eğer basılan tuşun değeri 10H 'e eşit veya bu değerden büyük ise komut anlamına gelir.

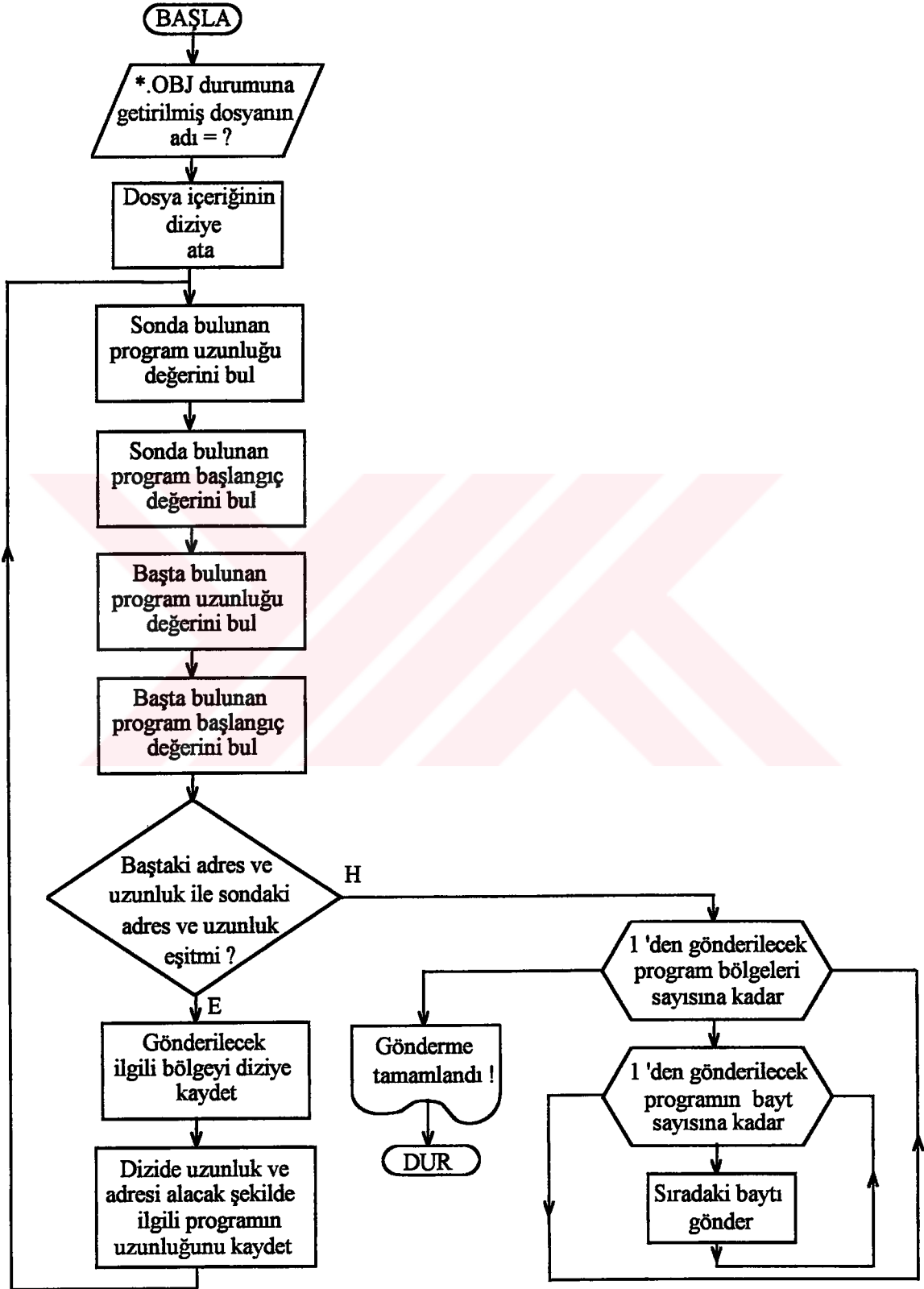
Tuşa basıldığını kullanıcıya bildirmek için sinyal sesi oluşturulur. Sinyal sesi C portunun 7. bitinden elde edilir. Çıkış tranzistör üzerinden hoparlöre bağlıdır. Ses programda elde edilen kare dalgadan elde edilir.

Tuşun gereği olan işlem olarak araya bir bayt ilave etme (insert), aradan bir bayt silme (delete), adresi bir azaltma, adresi bir arttırma, seri porttan alma ve bir programı çalıştırma işlemleri vardır. Insert işleminde o anda göstergede görülen adresten itibaren sonraki adresleri bir bayt ileri öteleme işlemini gerçekleştirir. Bu sırada diğer veriler bozulmaz. Delete işlemi göstergede görülen veriyi siler. Bu adresten bir sonraki adresdeki veriler bir bayt geri kaydırılır. Adres arttırma işleminde göstergede görülen adres bir artar ve bu adresdeki veri gösterilir. Adres azaltma işleminde adres bir azalır ve adresdeki veri

gösterilir. RS 232 tuşuna basılırsa göstergeler sönük duruma geçer ve bilgisayardan seri porttan gönderilecek verileri bekler. Bu durumda alınan ilk iki bayt gönderilecek bayt sayısını, sonraki iki bayt bu verilerin yazılacağı adresin başlangıç adresidir. Sonra gönderilen veriler en son adrese yazılacak bayttan başlar.

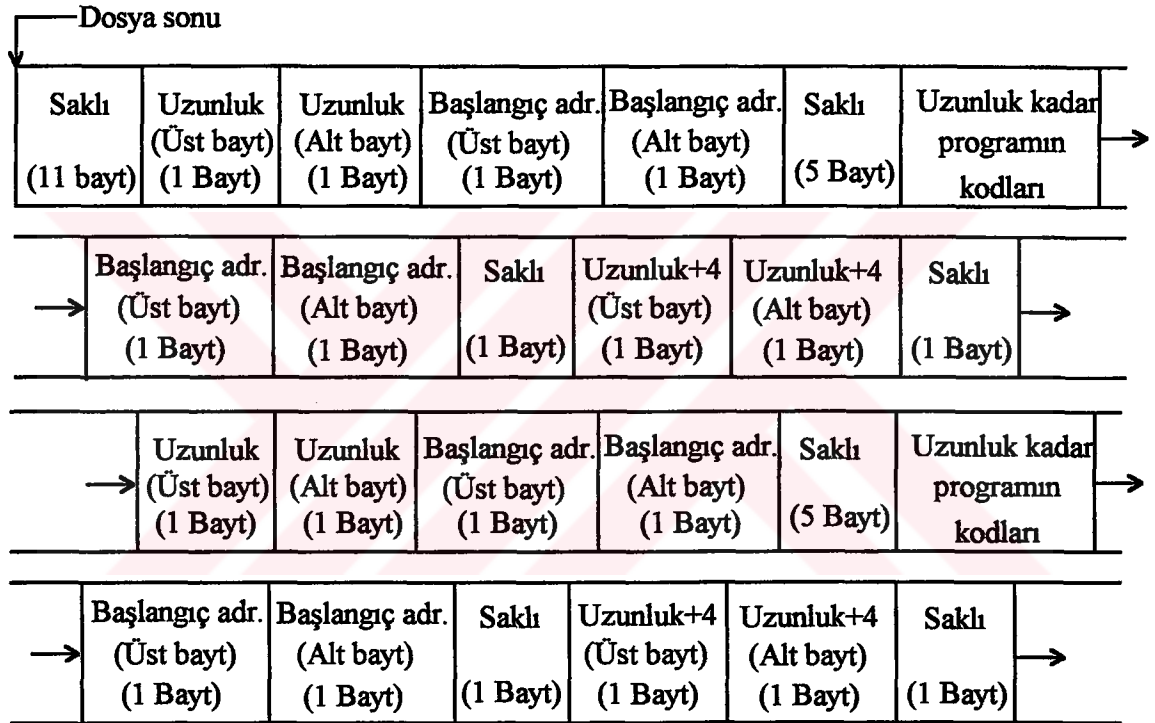


14.2. Programın seri porttan geliştirme setine gönderilmesi



Şekil 14.2. Programların seri porttan geliştirme setine gönderen prog.

Programın başında *.OBJ olarak verilen program herhangi bir editör programında yazılan makina dili *.SRC olarak kaydedilen program derlendikten sonra oluşur. Derleme işlemi ASM96 programı tarafından yapılır. *.OBJ dosyasının içinde gönderilecek ayrı ayrı adreste yazılan programların başlangıç adresleri , program uzunlukları mevcuttur (Şekil 14.3).



Şekil 14.3. *. OBJ dosyasının içeriğinde ayrı ayrı bellek bölgelerine yazılacak olan programların format durumu

İlgili *.OBJ dosyası ilk adresten itibaren dosya sonuna kadar bayt bayt okunur. Okunan her bayt, bayt() dizisine atanır. Dizinin boyutu başta bilinmediği için boyut ayırma işlemi sonradan yapılır.

Programın uzunluğu dosyanın sonundan 11 bayt önceki bayttır. Önce uzunluğun üst baytı sonra ise uzunluğun alt baytı vardır, (Şekil 14.3). Bu iki bayt alınır ve bir araya getirilerek programın

uzunluđu bulunur. Sonraki iki bayt ise programın geliştirme setinde yazılacağı bellek başlangıç adresinin üst ve alt baytı vardır. Bu iki bayt bir araya getirilerek adres başlangıçı hesaplanır. Daha sonra program uzunluđu+5 bayt sonra tekrar aynı bellek bölgesine yazılacak programın başlangıç adresi iki bayt olarak görülür. Bu iki bayt tekrar okunur ve tekrar aynı programın başlangıç adresi hesaplanır. Bir saklı bayttan sonra aynı programın uzunluğunun 4 fazlası olan iki bayt vardır. Bu uzunluk okunur ve tekrar uzunluk hesaplanır.

Önceden ve sonradan okunan uzunluk ve adresler karşılaştırılır. Eğer eşitlik var ise bu aradaki veriler programın kodlarıdır ve bu kodlar gndr() dizisine alınır. Bu işlem sırasında önce uzunluğun üst baytı sonra alt baytı alınır. Aynı işlem adres başlangıçı için de yapılır. Son olarak adres başlangıçından 5 bayt sonra başlayan program kodları diziye atanır.

Her defasında bulunan bu program bölgeleri sayılır. Ayrıca her programın bölgesinin uzunluđu ve başlangıç adreside geliştirme setine gönderileceği için program bölgesi uzunluğuna 4 ilave edilerek bu uzunluklar bir dizide saklanır.

Eğer bir program bölgesi için önceden ve sonradan okunan adresler ve uzunluklar eşit değil ise alınacak program bölgeleri bitmiş demektir ve programları gönderme işlemine geçilir. Önce kaç tane program bölgesi gönderileceği seri porttan geliştirme setine iletilir. Sonra program bölgesinin uzunluđu ve ardından başlangıç adresi ve bu program bölgesi için son olarak, sondan başlayarak program kodları gönderilir. Geliştirme setindeki seri port programı gönderilen uzunluğa ve adrese göre gönderilen verileri ilgili adreslere yazar.

Ayrıca gönderme işleminden önce bilgisayarın seri portunun istenen amaçlara göre ayarlanması gerekir. Bu şartlamayı sağlayan kaydedicilerin adresi ve bitlerinin fonksiyonları, gönderme işlemi



sırasında hat durumlarını gösteren durum kaydedicisi bitlerinin görevleri Ek 1 'de gösterilir.

14.4. Bilgi tabanı verilerinin fuzzy işlemciye gönderilmesi



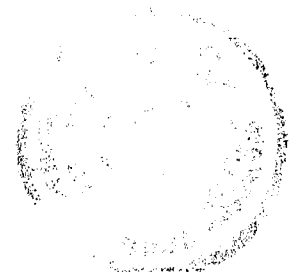
Şekil 14.4. Veri tabanı bilgilerinin fuzzy işlemciye gönderen programın akış diyagramı

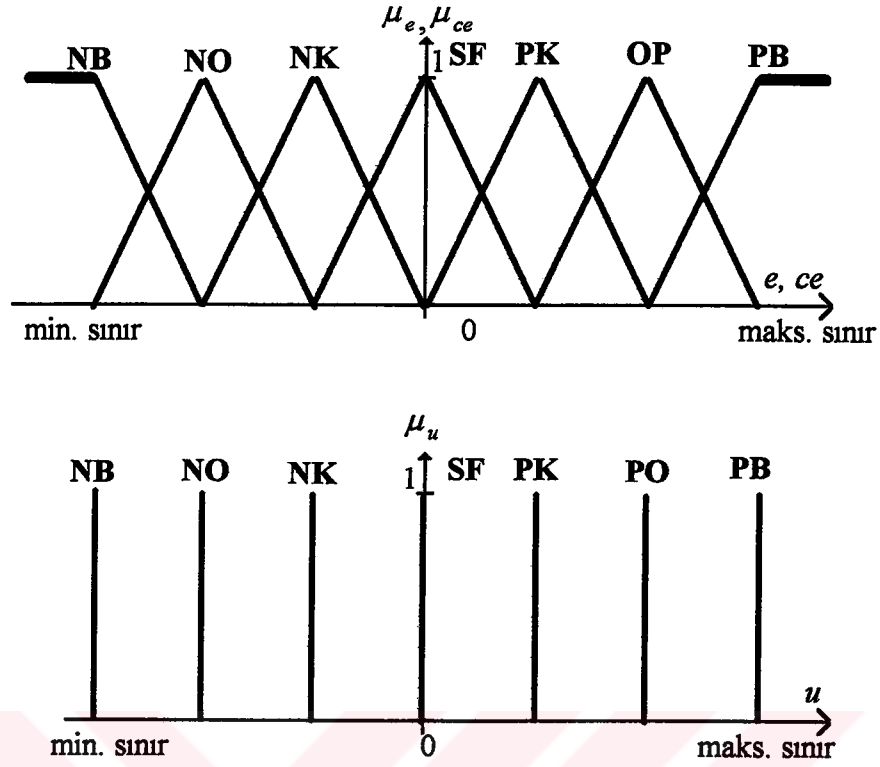
En bařta haberleřme hızı 4800 Baud'a ayarlanır. Çünkü mikrodenetleyicinin seri port haberleřme hızıda 4800 Baud'a ayarlanmıřtır. Dięer seri portu kullanan programlarda bu ayarlama yapılmaktadır.

*.BIN dosyası fuzzy çıkarım kartı için kullanılan FS 30-AT yazılım programından elde edilir. Bu programda önce kurallar yazılır (tablo 14.1.). Daha sonra üyelik fonksiyonları ve netleřtirme yöntemi seçilir (řekil 14.5). Bu işlemlerden sonra kuralların ve üyelik fonksiyonlarının bir araya getirilmesi ile derleme işlemi yapılır. Böylece *. BIN dosyası oluşur. Bu dosyada belirli adreslerde veri tabanı için gerekli bilgiler bulunur.

*.BIN dosyasındaki FBH-3D6H adresleri arasındaki veri baytları okunur ve diziye kaydedilir. Bu veri baytlarının kendinden sonra gelen baytlar ile yerleri deęiřtirilir. 3D7H-14A4H adresleri arasındaki veriler yerleri deęiřtirilmeden diziye kaydedilir. Bu bilgiler daha önce üzerinde çalışma yapılan simulasyon programının analizinden sonra belirlenmiřtir. Daha sonra oluşan yeni dizi seri port vasıtasıyla deęiřtirme setine buradan fuzzy işlemciye gönderilir.

Gönderme işlemi sırasında önce kaç deęiřik adres bölgesi kullanılacaęının sayısı gönderilir. Bu durumda yalnız bir adres bölgesi mevcut olduęu için önce 01H sayısı gönderilir. Daha sonra gönderilecek bayt sayısı (uzunluk=2755H) ve bařlangıç adresi (80A8H) gönderilir. Her bayt gönderilmeden önce gönderme kaydedicisinin boş olup olamaması kontrol edilir. Bu işlem bilgisayarın durum kaydedicisinin 7. biti yardımıyla saęlanır. Eęer bit lojik 0 ise gönderme kaydedicisi dolu aksi durumda gönderme kaydedicisinin boş olduęu anlařılır. Ek 1 'de bu kaydedici ile ilgili bilgi verilir.



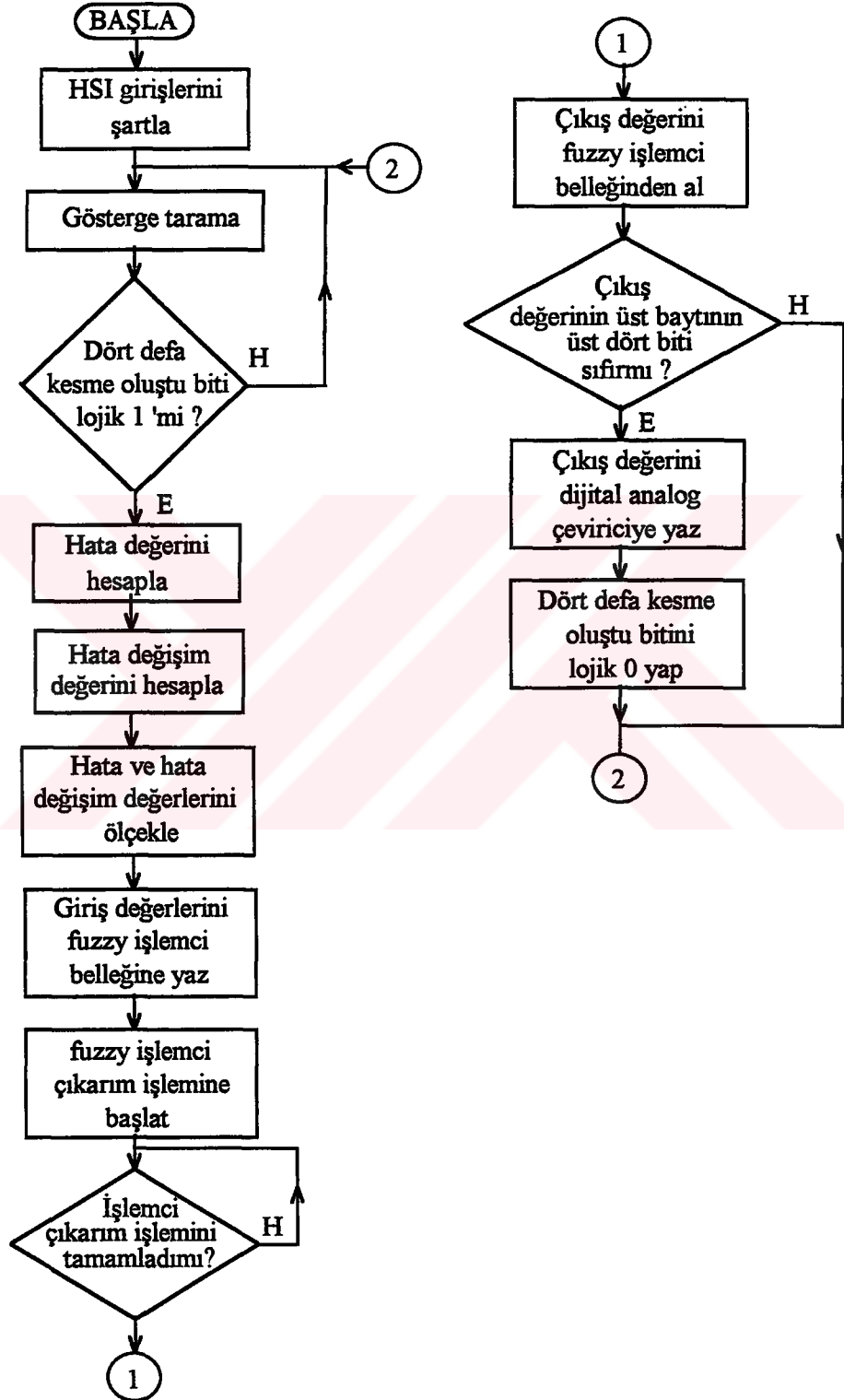


Şekil 14.5. Uygulamada kullanılan e , ce ve u için üyelik fonksiyonları.

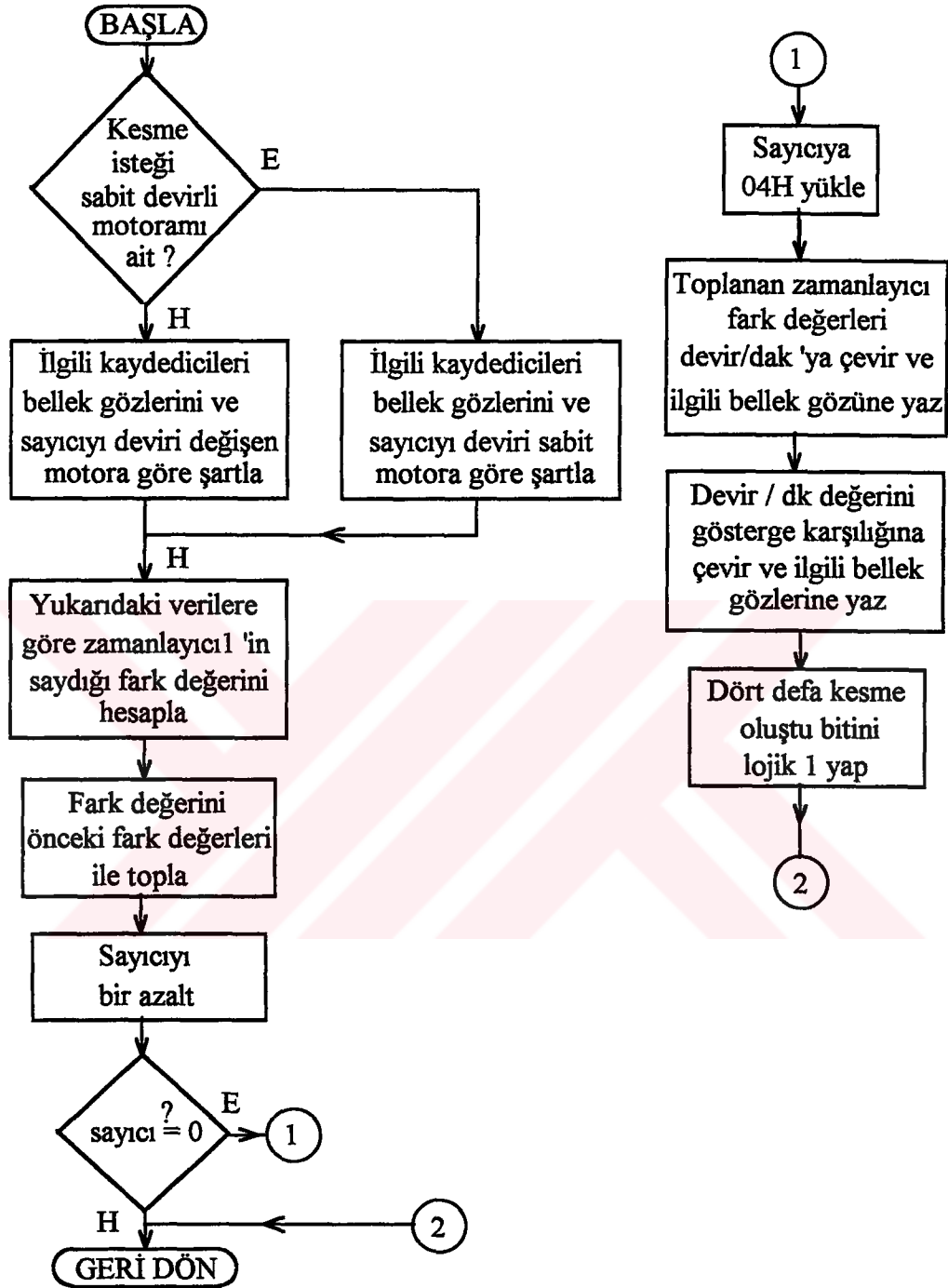
Tablo 14.1. Uygulamada kullanılan kuralların gösterilimi

$\frac{ce}{e}$	NB	NO	NK	SF	PK	PO	PB
PB	NB	PO	PB	PB	PB	PB	PB
PO	NB	*	PB	PB	PB	PB	PB
PK	NB	*	*	PK	PB	PB	PB
SF	NB	NB	NK	SF	PK	PB	PB
NK	NB	NB	NK	NK	*	*	PB
NO	NB	NB	NB	NB	NB	*	PB
NB	NB	NB	NB	NB	NB	NB	PB

14.5. Mikrodenetleyiciyi ve fuzzy işlemciyi kullanarak fuzzy lojik kontrol uygulama programı



Şekil 14.6. Uygulama programının akış diyagramı



Şekil 14.7. Kesme isteği programının akış diyagramı

Programın en başında HSI girişlerinden HSI.2 ve HSI.3 bacaklarına izin verilir (Şekil 14.6). Bu girişlerin çıkan ve düşen

kenar tetiklemeleri zamanlayıcı1'deki değerlerin tutma kaydedicisine yüklenmesine neden olur.

Deviri değiştirilen motorun devir sayısının göstergede gösterilmesi gösterge tarama ile sağlanır. Ayrıca devir sayısı sabit olan motorun dönüş hızını göstergede görebilmek için tuş takımı kullanılabilir. Dört defa kesinti isteği oluşmaz ise sürekli olarak gösterge taranır. Şayet dört defa kesme isteği oluşursa programın akışı değişir. Herhangi bir motor bir devir döndüğünde dört defa kesme isteği oluşur. Çünkü bir devirde dört kenar tetiklemesi vardır. Her iki motorun bir devirde programın akışını değiştirebilmeleri için her motor için ayrı bir sayıcı kullanılır. Bu sayıcılar kesme isteğini sayarlar ve herhangi biri sıfır olursa dört defa kesme isteği oluştu biti lojik 1 olur ve tekrar ilgili sayıcıya 04H yüklenir. Sonuçta bu bit kontrol edilir. Eğer bit lojik 1 ise programın akışı değişir. Programın akışı değiştiği zaman fuzzy işlemci ile ilgili işlemlere geçilir.

Önce iki motor arasındaki farkı veren hata hesaplama işlemi yapılır. Yani hızı sabit motorun devir sayısından hızı değiştirilen motorun devir sayısı değiştirilir. Yani $e_n = w_{sbt} - w_{deg}$ formülü ile ifade edilir. İkinci giriş değişkeni olan hatanın değişim değeri ise o andaki hatadan bir önceki hatanın değeri çıkarılarak hesaplanır. Yani $ce_n = e_n - e_{n-1}$ ifadesi ile hesaplanır. Bu işlem yapıldıktan sonra önceki hata değerine o andaki hata değeri ($e_{n-1} = e_n$) atanır.

Yukarıdaki gibi hesaplanan hata ve hatanın değişimi değerleri fuzzy işlemcinin belleğine yazılmadan önce ölçeklenmesi gerekir. Çünkü işlemçi 0 ile 4097 arasındaki sayılar ile işlem yapar. Fakat hata ve hatanın değişim değerleri motor devirlerine göre minimum ve maksimum sınır değerlerine sahiptir (Şekil 14.5). Bu yüzden minimum sınır değeri fuzzy işlemcide 0 'a ve maksimum sınır değeri ise 4094 'e karşılık gelir . Hata ve hatanın değişiminin sıfır değeri işlemcide 2047

sayısına denk düşer. Bu ölçekleme her iki giriş değeri için aşağıdaki gibi yapılır.

$$|maks.sınır| = |min.sınır| \quad (14.1)$$

$$Ölçekleme = 2047 + giriş\ değeri \frac{2047}{maks.sınır}$$

Elde edilen ölçeklenmiş giriş değerleri fuzzy işlemcinin belleğine yazılır. Daha önce anlatıldığı gibi giriş bilgilerinin yazıldığı ilk adres 8004H adresidir. Önce ilk giriş değeri olan hata değerinin üst baytı, sonraki çift adrese alt baytı yazılır. Bu adreslerden hemen sonraki ilk çift adrese hata değişim değerinin üst baytı ve sonraki çift adrese alt baytı yazılır. Her iki durumda üst baytların üst dört biti sıfır olması gerekir. Giriş bilgileri yazıldıktan sonra fuzzy işlemçi çıkarım işlemine başlatılır. Bu işlem 8000H adresindeki bellek gözünün 4.biti lojik 1 yapılarak sağlanır. İşlemciyi çıkarım işlemine başlattıktan sonra çıkarım sonucunu bulduğu yine aynı adresteki aynı bit ile anlaşılır. Eğer bu bit halen lojik 1 ise çıkarım işlemi tamamlanmamış demektir. Yani fuzzy işlemci meşgul durumda demektir. İlgili bit lojik 0 ise çıkarım işlemi tamamlanmış demektir.

Sıra sonuçta oluşan çıkış değerini okumaya gelmiştir. Çıkış değeri bellekten 8024H adresinden alınır. 16 bit olarak bellekte bulunan değerın üst baytı ilk adreste ve alt baytı bir sonraki çift adreste bulunur. Alınan çıkış değerinin üst baytının, üst dört biti sıfırdan farklı ise çıkarım işlemi sonucunda alınan değer geçerli değildir. Yani verilen giriş değerlerine karşılık çıkarım sonucunda elde edilecek bir değer mevcut değildir. Bu yüzden digital analog çeviriciye herhangi bir bilgi yazılmaz ve tekrar başa dönülür.

Elde edilen çıkış değeri 0 ile 4094 arasındadır. 0 değeri 1.25 volt değerine (DAC çıkışında LM 317 olduğu için) ve 4094 değeri ise

10.65 volta karşılık gelir. Fakat elde edilen dijital değerlerin evriği alındıktan sonra dijital analog çeviriciye verilmesi gerekir. Çünkü çevirici kendi içinde bir evrik alma işlemi daha yapar. Fuzzy işlemciden elde edilen çıkış değerinin ölçeklenmesine gerek yoktur. Çünkü dijital analog çeviricinin ile fuzzy işlemcinin işlem kapasiteleri aynı ve 12 bittir.

Son olarak dört kesme isteği sonunda bir devir için işlem yapıldığı için dört defa kesinti biti tekrar aynı işlemlerin yapılmaması için lojik 0 yapılır ve tekrar gösterge tarama işlemine dönülür.

14.6.Kesme isteği alt programı:

Bu program 7000H adresinden itibaren başlar. HSI girişlerinde oluşan çıkan ve düşen kenar tetiklemelerinde programın akışı kesme isteği alt programına yönelir. (Şekil 14.7).

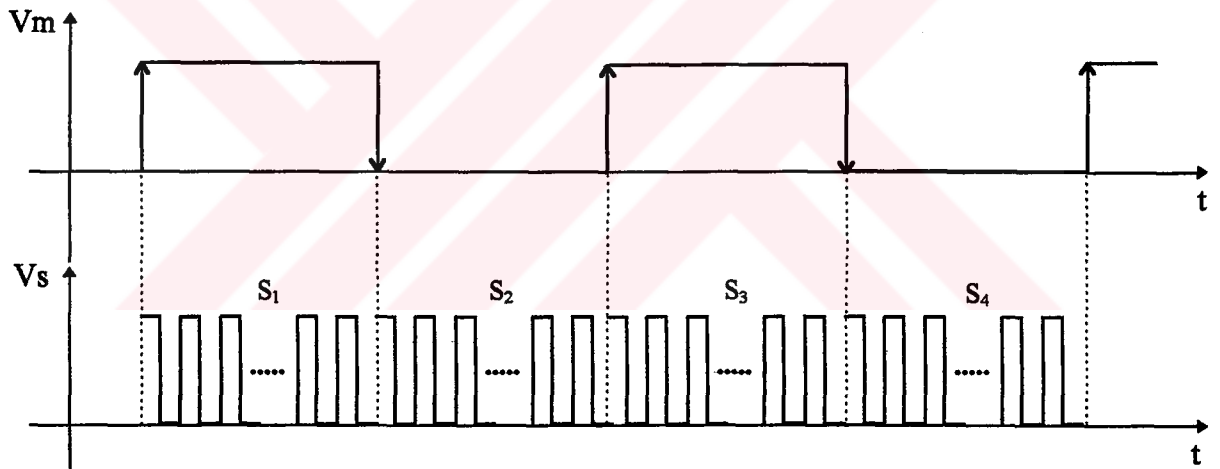
Bu programın başında oluşan kesme isteğinin hangi motora ait olduğu tespit edilir. Bu şekilde aynı kesme programı ile ayrı iki motorun devir sayısı ölçülür. HSI durum kaydedicisinin 4. biti (HSI.2) ve 6. biti (HSI.3) kesme isteğinin hangi motordan geldiğini gösterir. Bu karşılaştırma işleminden sonra sabit hızlı motor veya değişken hızlı motor için verilerin alınacağı veya yazılacağı ilgili bellek gözleri kaydedicilere yüklenerek ayarlanır.

Devir ölçümü için zamanlayıcı1 'deki değer alınır ve saklanır. Kesme isteği gelene kadar zamanlayıcı içeriden aldığı saat darbeleri ile ileri sayar. Her saat darbesi periyodu $1\mu s$ 'dir. HSI girişlerinden kesme isteği gelirse zamanlayıcı1'de o andaki değer ile başta saklanan değer arasındaki fark hesaplanır. Hesaplanan bu değer motorun bir devrinin yarısında lojik 1 'de veya lojik 0 'da kalma zamanı içinde sayılan değeri oluşturur. Magnetik algılayıcı önünden geçen pervane iki taraflı olduğu için bir devirde oluşan darbelerde iki tane lojik 1 zamanı ve iki tanede lojik 0 zamanı vardır. Bu durumda bir devirde iki çıkan ve iki düşen

kenar vardır. Böylece lojik 1 'de sayılan değerler ile lojik 0 'da sayılan değerler toplanır ve bir devirde sayılan değer bulunmuş olur. Böylece kenar tetiklemelerini sayan sayıcıda sıfıra gelmiş olur. Programın bu bölümünde tekrar kenar tetiklemelerini saymak için sayıcıya 04H sayısı yüklenir. Sonra bir devirde sayılan değer devir/dk 'ya çevrilir ve aşağıdaki gibi hesaplanır (Şekil 14.8.).

$$\text{Devir} / dk = \frac{60.10^6}{\text{Bir devirde sayılan değer}} (\text{Sayıcı darde Periyodu} = 1\mu s) \quad (14.2)$$

Hesaplanan bu değer göstergede gösterilecek şekle çevrilerek gösterge için ayrılan bellek bölgesine yazılır. Son olarak ana programda dört defa kesme isteğinin oluştuğunu anlayabilmek için dört defa kesme oluştu biti lojik 1 yapılır.



Şekil 14.8. Devir ölçmede kullanılan yöntem

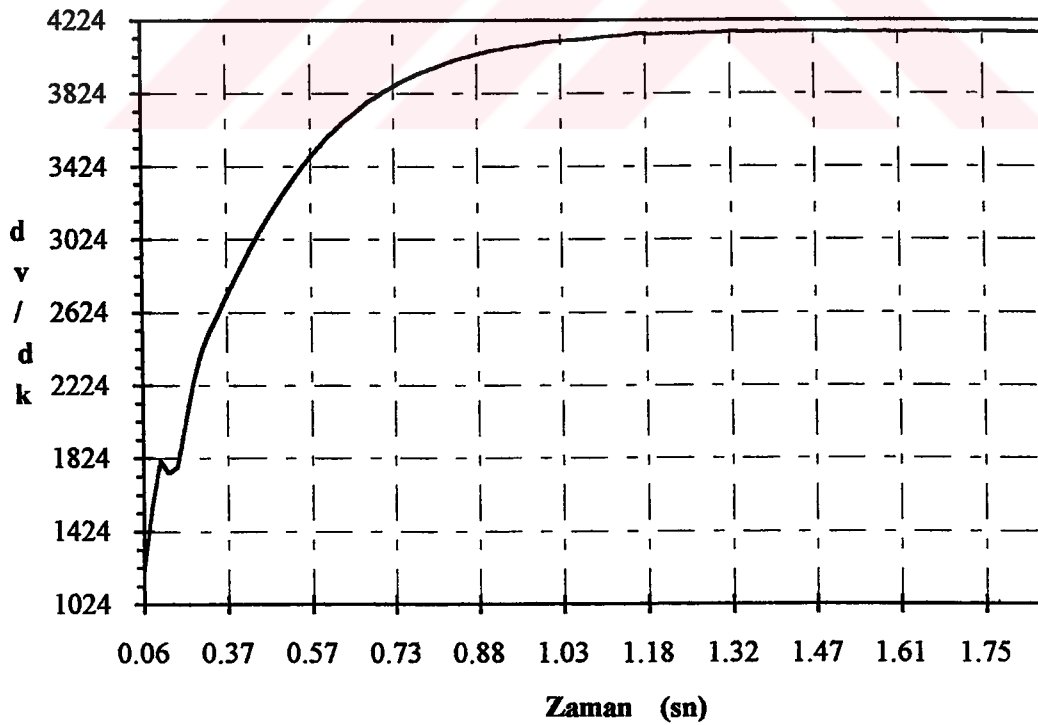
$$S_T = S_1 + S_2 + S_3 + S_4$$

$$\text{Devir} / dk = \frac{60.10^6}{S_T} \quad (14.3)$$

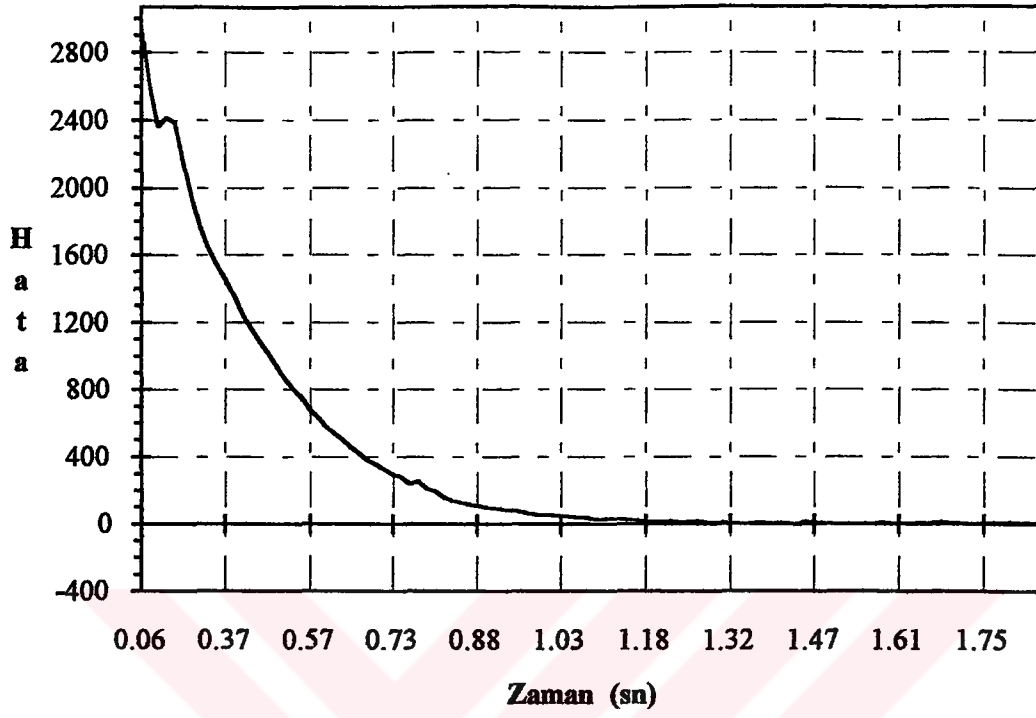
15.ELDE EDİLEN EĞRİLERİN ANALİTİK YORUMU:

Bu bölümde incelenecek olan eğrilerin tamamı motorun hızı kontrol edilirken gerçel zamanda çizilerek elde edilmiştir.

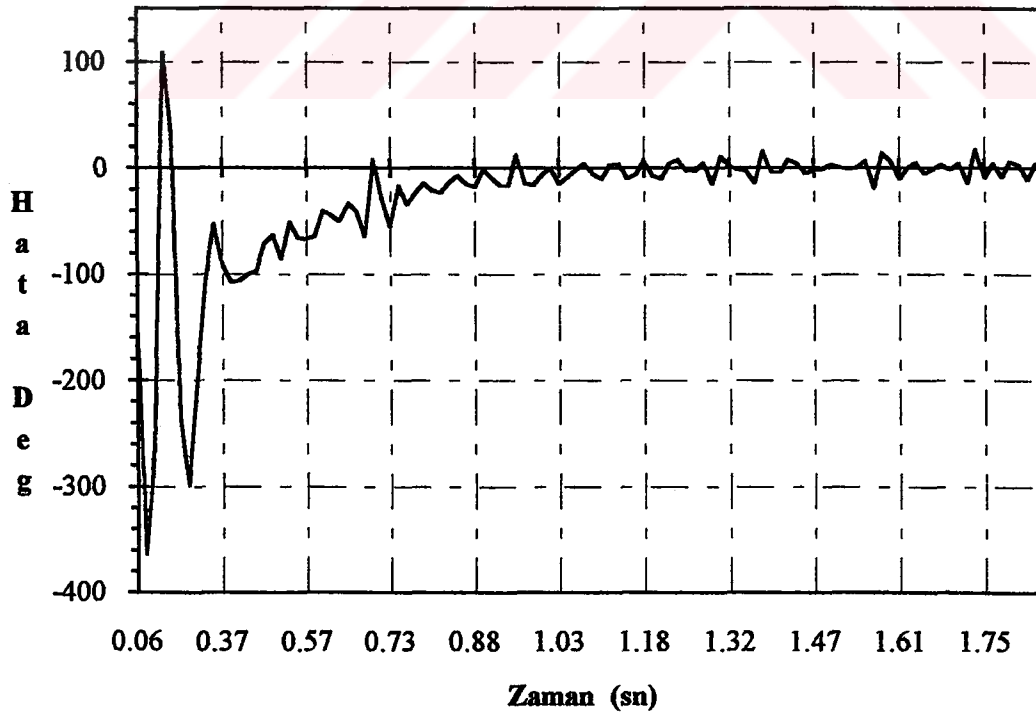
Şekil 15.1, 15.2, 15.3, 15.4, 15.6, 15.7 ve 15.8'deki eğriler devri kontrol edilen motorun devir, hata ve hatanın değişimi parametrelerinin zamana karşı değişen grafiklerini göstermektedir. Devri kontrol edilen motorun jeneratör olarak kullanılan devri kontrol edilen motora bir şaft ile bağlanmadığı ve devir ölçmek için magnetik algılayıcılar önünden geçen metal kanat boyu küçük olan pervane ile yapılan ölçümleri içerir. Eğride ve motorlarda en az devir 1024 dv/dk ve en fazla devir 4124 dv/dk olmaktadır. En az devirin 1024 dv/dk olmasının nedeni ölçüm için kullanılan zamanlayıcı modülünün 16 bitlik yani 0 'dan 65535'e kadar sayması durumundan kaynaklanır. En yüksek devrin 4124 dv/dk olmasının nedeni ise devri ayarlanan motora verilen gerilimin en fazla 9V-10V olmasından kaynaklanır. Motor için mikrodenetleyici kontrollü sürme gerilimi üreten DAC en



Şekil 15.1



Şekil 15.2



Şekil 15.3

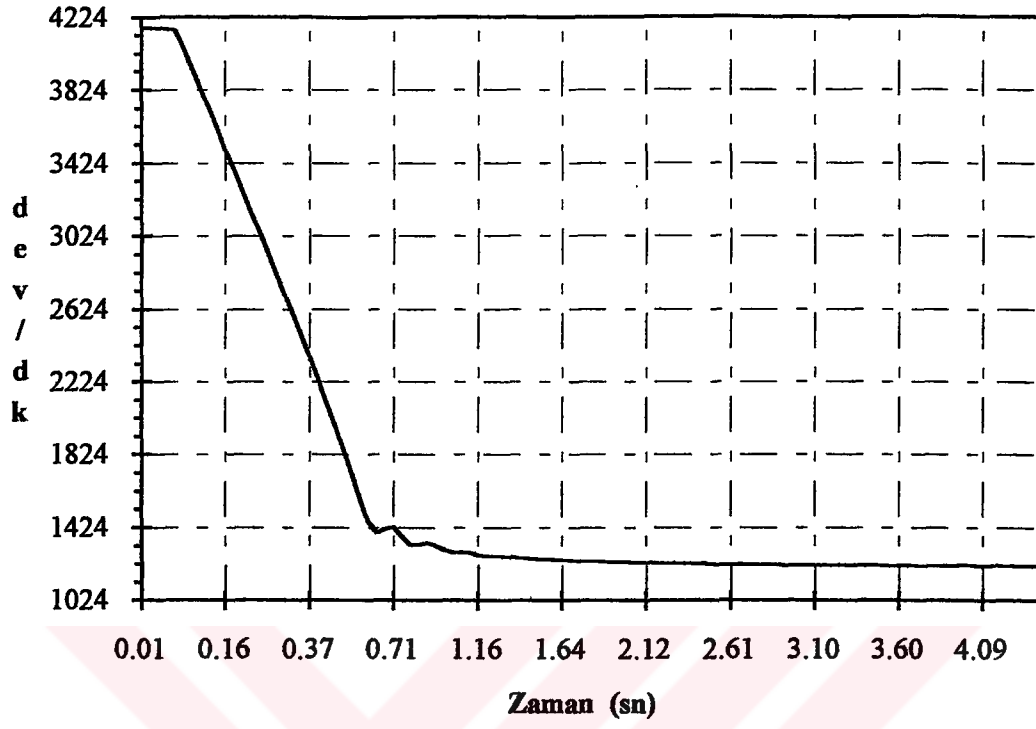
fazla 10.25V çıkış vermesi en fazla devri belirleyen etkenlerden birisidir. Yazılan makine dili program kullanılan mikrodenetleyicinin ve fuzzy işlemcinin hızlı olmasından dolayı diğer hesaplama işlemlerini yaparken daha yüksek hızlı devirleride ölçme yeteneğine sahiptir.

Şekil 15.1'de zamana karşılık olarak çizdirilen devir eğrisi görülmektedir. Hızı sabit olan motor 4124 dv/dk hızlı dönmektedir ve hızı kontrol edilebilen motor ise 1024 dv/dk hızla dönmektedir. Program çalıştırıldığı anda kontrol edilen motorun devri 1.32 sn'de diğer motorun hızına ulaşmakta ve bu hızda sabit kalmaktadır. Bu geçiş süresi daha sonraki eğrilerde görüleceği üzere hata, hatanın değişimi ve fuzzy çıkış ölçeklemesi değiştirilerek en az süreye indirmek mümkündür. Bu durumlar ilerideki eğrilerde çeşitli ölçekleme değerleri için yorumlanmıştır. Kullanılan sistemin fiziksel özelliklerinden ve kontrol algoritmasından dolayı kalıcı değere yaklaşım üstel olarak gerçekleşir. Eğri üzerinden görüleceği üzere 1824 dv/dk hız değerinde eğrinin bir çentik oluşturduğu görülmektedir. Bunun nedeni ise kurallarla ilgilidir. Çünkü bu kurallara göre ilk anda ilk devir ile ulaşılacak devir arasında çok büyük fark vardır. Bu sebeple motor çok hızlı bir şekilde eğim yüksek olarak hızını arttırmaya başlar. Fakat ulaşılması istenen devirde salınım olmaması için devir aniden biraz düşürülerek eğimin azalması sağlanır.

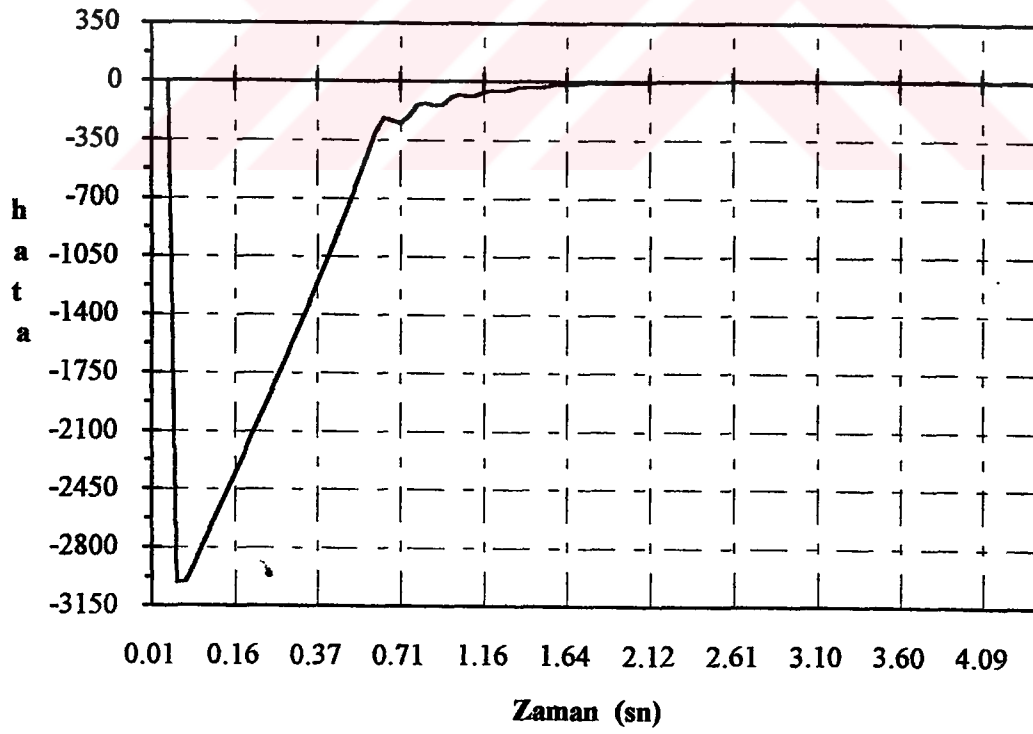
Şekil 15.2 hata zaman grafiğini göstermekte olup hata değeri 3100 dv/dk'dan başlayıp devir-zaman grafiğinde olduğu gibi 1.32 sn'de sıfıra düşmektedir. Bu eğride 2400 dv/dk değerinde yine bir çentik görülmektedir. Bu olay şekil 15.1 eğrisindeki aynı olaydan kaynaklanır.

Şekil 15.3'de hata değişimi veya hata hızı ile zaman grafiği görülmektedir. İlk anda hata değişimi olmadığı için sıfırdan başlamaktadır. 0.29 sn içerisinde eğri dört salınım yapmakta daha

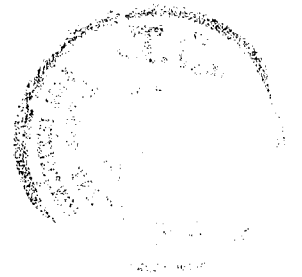




Şekil 15.4



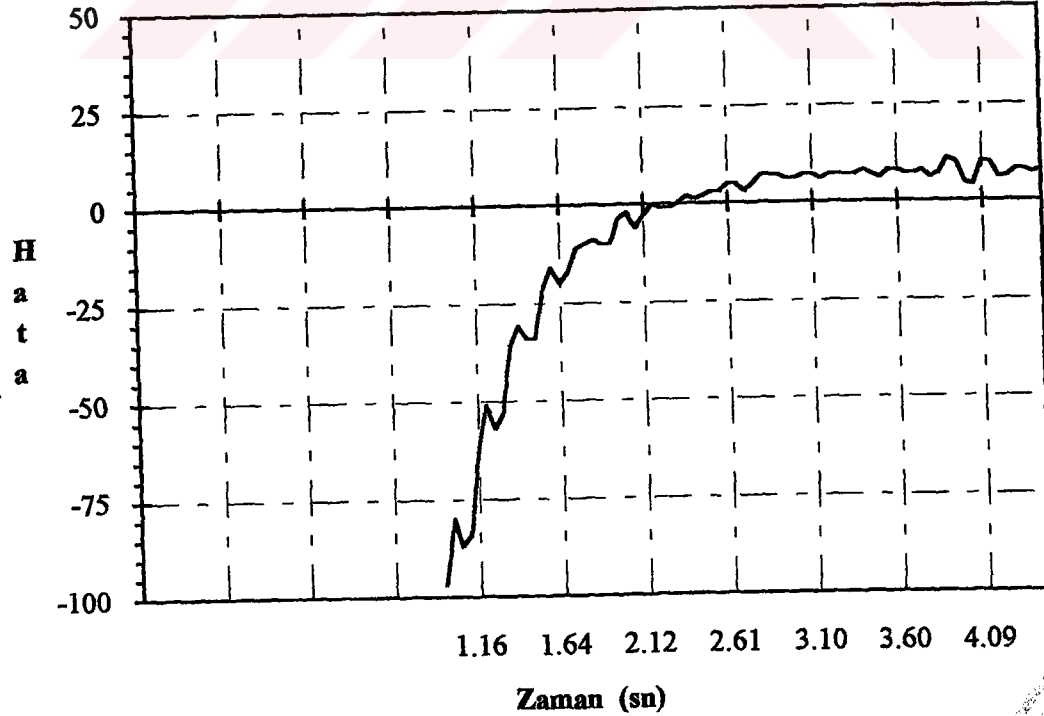
Şekil 15.5



sonra 0.35 sn'de salınımlar oldukça yavaşlamakta ve hata hızının değişimleri en aza inmiş bulunmaktadır. Eğriden görüleceği üzere daha önce anlatılan eğrilerde olduğu gibi 1.32 sn'de negatif değerlerden sıfıra çıkmakta ve sıfır yatay ekseninde çok küçük salınımlar yapmaktadır. Bunun nedeni ise hızın artmasından dolayı motorun titreşmesinden kaynaklanır. 0.35 sn'ye gelene kadar hata hızında meydana gelen salınımlar hızı kontrol edilen motorun 1824 dv/dk'daki devrinde oluşan çetikten meydana gelmektedir.

Şekil 15.4'deki eğri devir-zaman grafiğini gösterir. Burada hızı sabit olan motorun devri 1200 dv/dk'ya set edilmiş hızı kontrol edilen motorun devri ise 4100 dv/dk'dan başlayıp 1200 dv/dk'ya doğru lineer olarak azalmakta ve 0.60 sn'de hafif bir salınım ile motorun devri 1200 dv/dk hızına kilitlenmektedir.

Şekil 15.5 'de hata-zaman grafiğinde hata -3000 dv/dk'dan başlayıp 0.60 sn'de -200 dv/dk'ya çıkmakta ve şekil 15.4'deki eğride olduğu gibi küçük salınımlar ile hata değeri sıfıra gelmektedir. Şekil 15.6 bu salınımların büyük ölçekli olarak çizilmiş olan grafiğini göstermektedir.



Şekil 15.6



Şekil 15.7'de hata değişimi-zaman grafiği görülmekte olup hata değişimi 0.04 sn içerisinde 0'dan 3100 dv/dk'ya ulaşmakta ve tekrar sıfıra gelerek burada hatanın değişimi hemen hemen sabitleşmekte ve 0.60 sn'de hata grafiğinde olan salınımlardan dolayı hatanın değişimide salınımlara başlamakta ve 1.52 sn'lik bir yerleşme zamanına sahip olmaktadır.

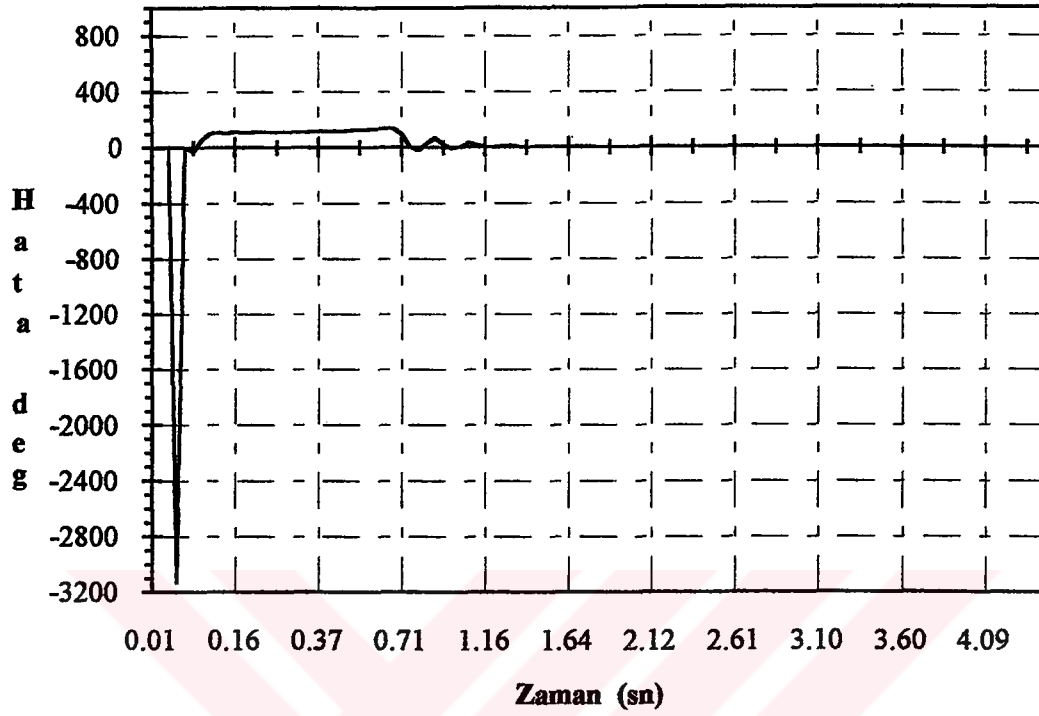
Şekil 15.9'dan şekil 15.38'e kadar olan eğriler devri kontrol edilen motora jeneratörün bağlanması (yük bağlı değil) sonucunda oluşan grafikleri gösterir. Bu durumda hızı kontrol edilen motorun devrini ölçmek için manyetik algılayıcının önünden geçen pervanenin kanat uzunluğu arttırılmıştır. Daha önce anlatılan eğrilerde ve şekil 15.9'dan şekil 15.26 arasındaki eğrilerde hata 20FFH, hatanın değişimi 3FEH ve fuzzy çıkış 600H değerleri ile ölçeklenmişlerdir.

Şekil 15.9 devir-zaman grafiğini göstermekte olup burada devir 1024 dv/dk değerinden başlamakta 2.01 sn'de 2565 dv/dk'ya çıkmakta ve bu devire kilitlenmektedir. Bu durum hızı kontrol edilen motor kalıcı duruma erişme süresinin 0.69 sn uzadığını gösterir.

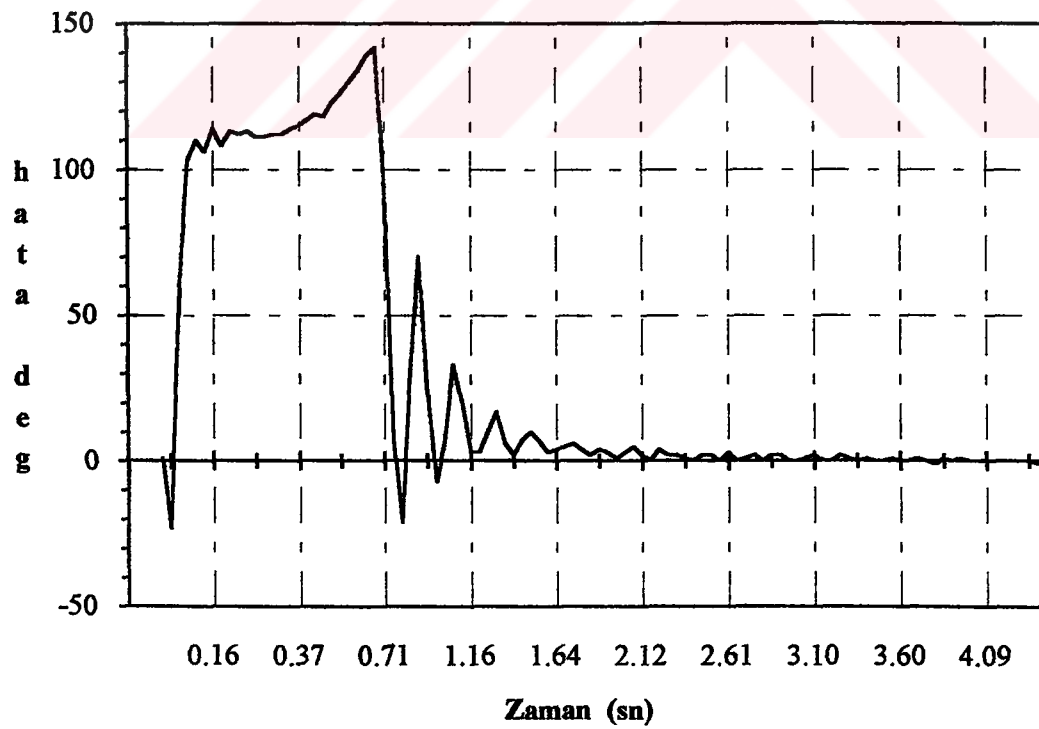
Şekil 15.11'de görülen hata değişimi eğrisinde şekil 15.9 ve şekil 15.10 eğrilerinde olduğu gibi hata hızının 2.01 sn'de sıfıra ulaştığı görülür. Eğride görülen salınımlar pervane kanadının bir önceki eğrilere göre artmasından dolayı motor yüksek devirlere çıkınca titreşimin artmasından kaynaklanır.

Şekil 15.12, 15.13 ve 15.14 hızı kontrol edilen motorun 1024 dv/dk'dan başlayıp 3650 dv/dk hızla dönen hızı sabit olan motorun devrine ulaşmasındaki devir, hata ve hatanın değişim eğrilerini gösterir. Dikkat edilecek olursa daha yüksek devirlere çıkıldığı zaman karalı duruma erişim süresi şekil 15.9'a göre azalmış ve şekil 1'deki gibi 1.32 sn olmuştur.

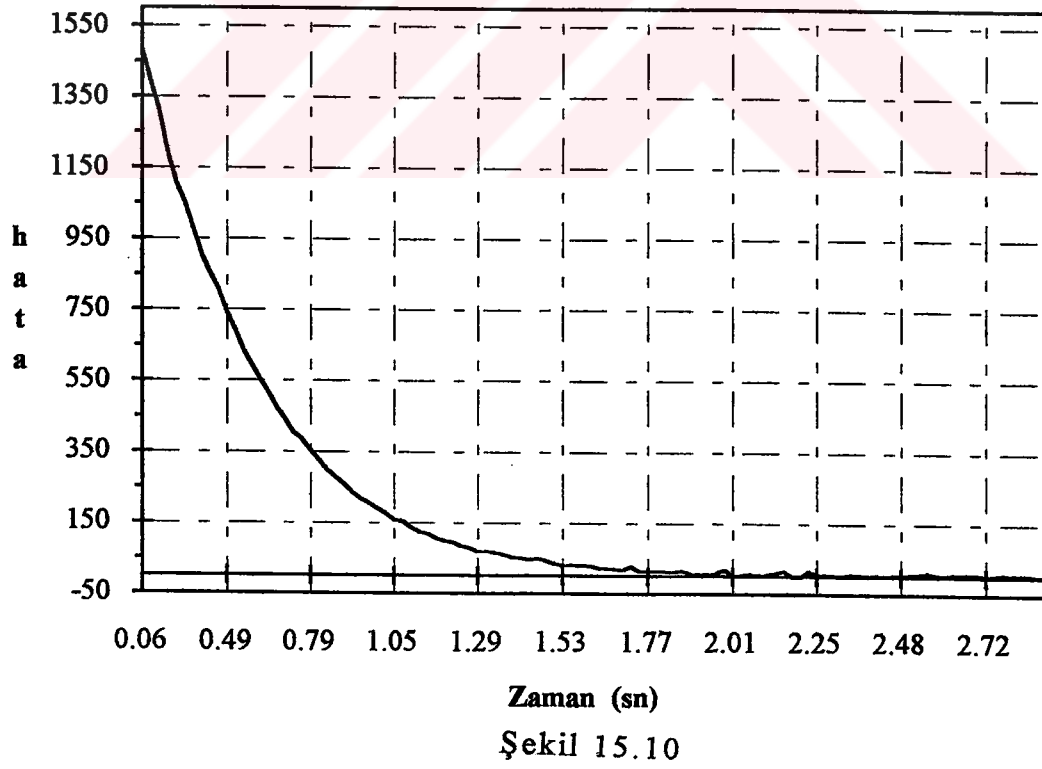
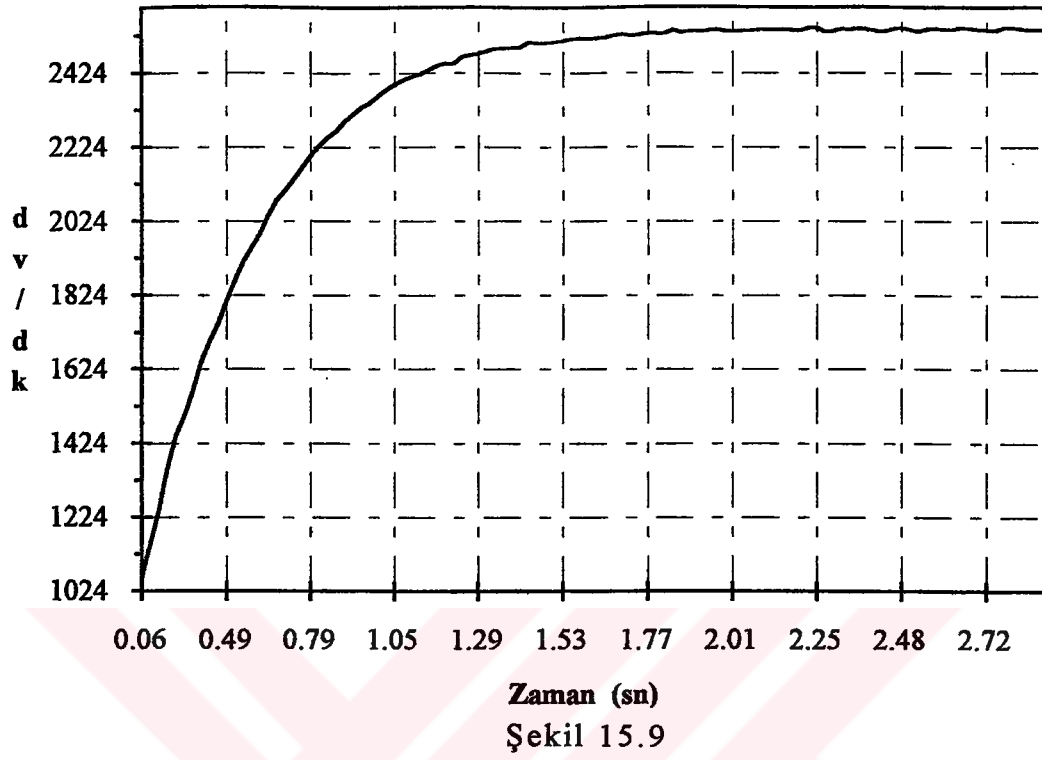


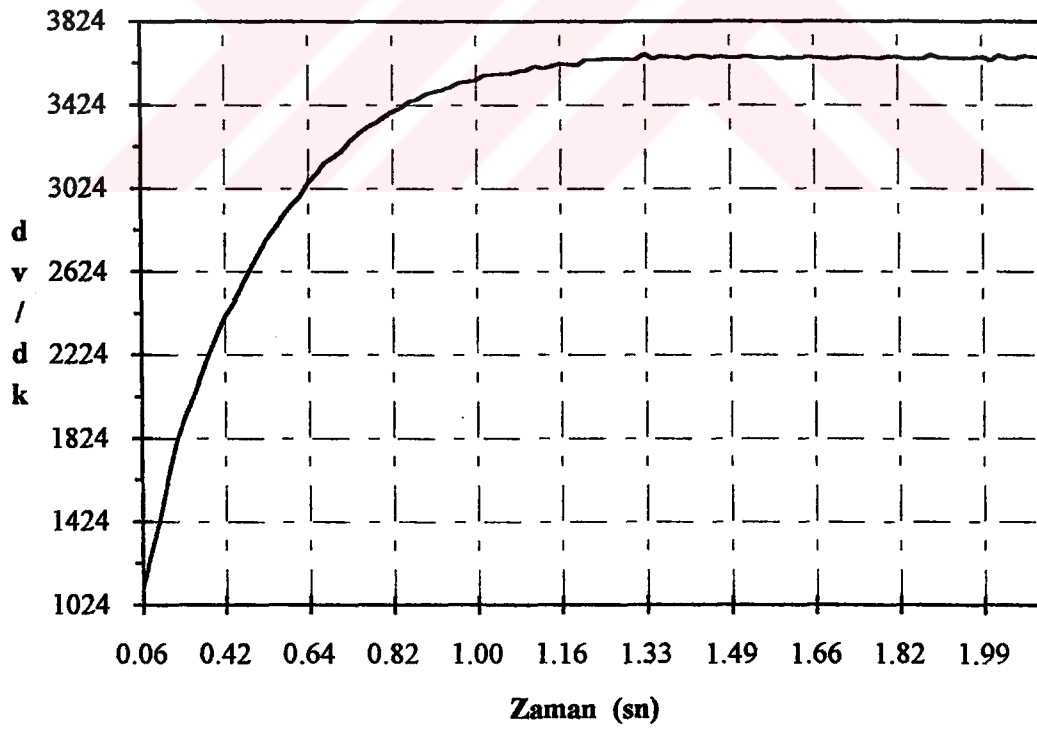
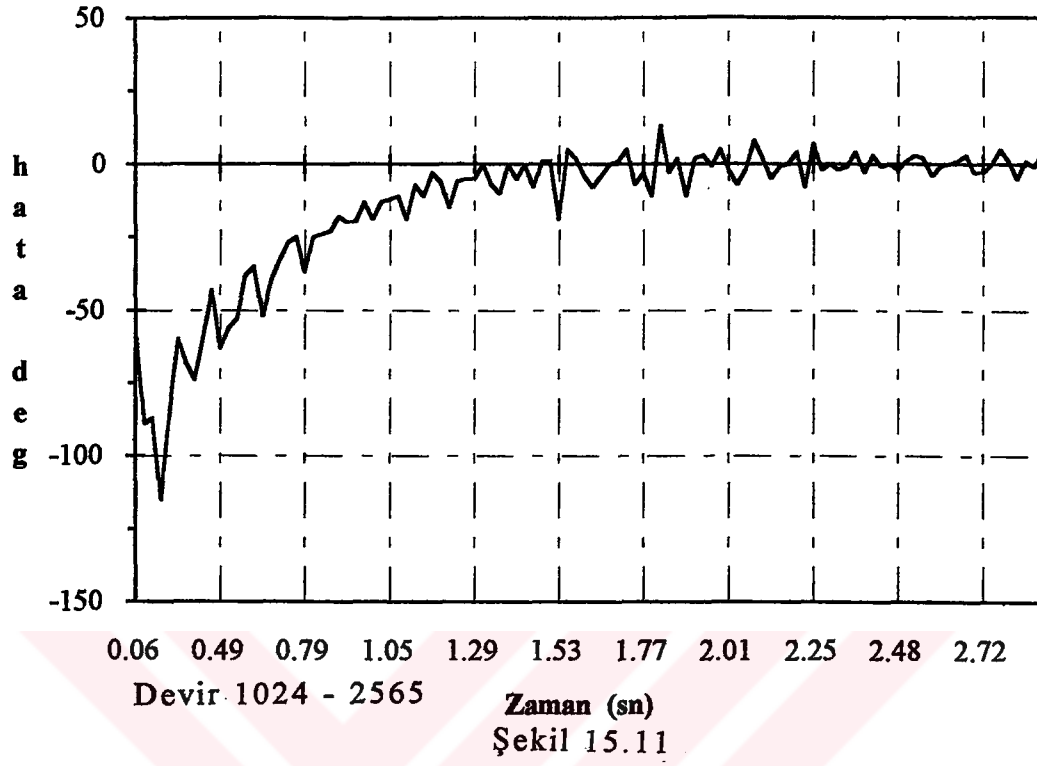


Şekil 15.7

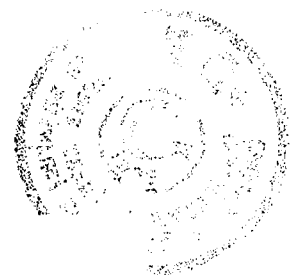


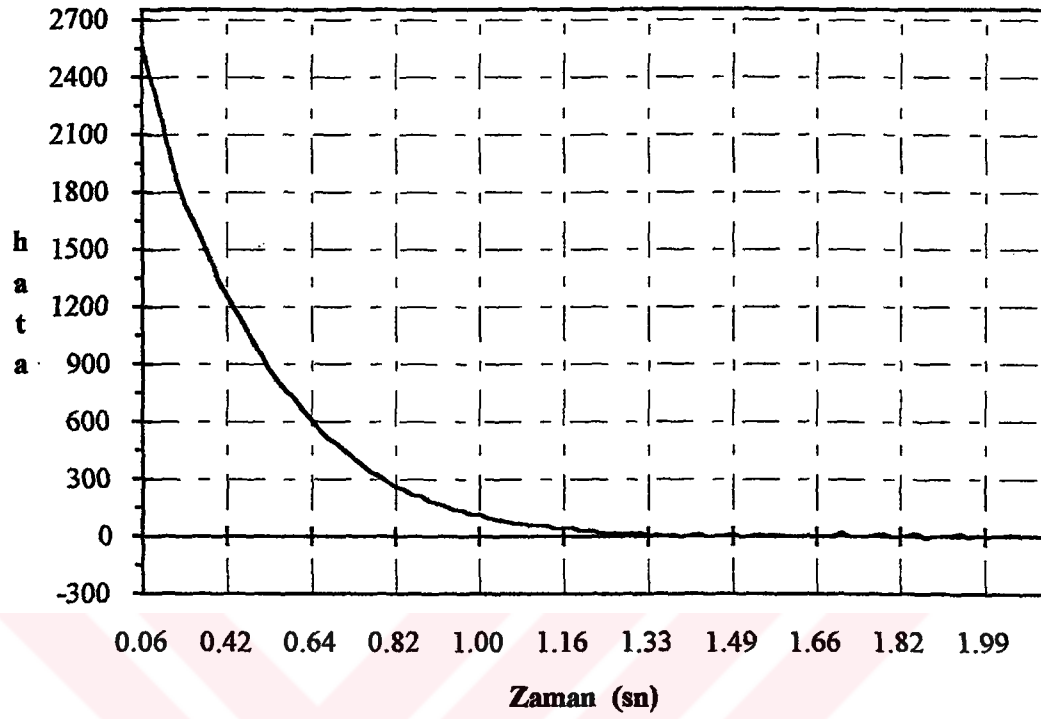
Şekil 15.8



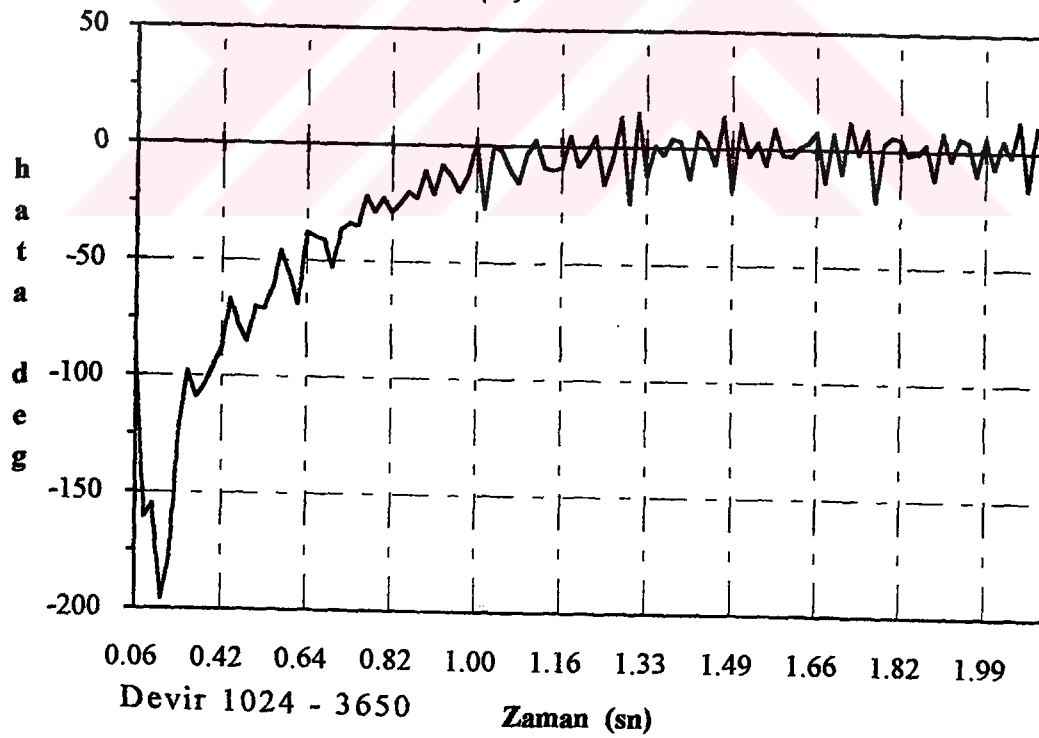


Şekil 15.12

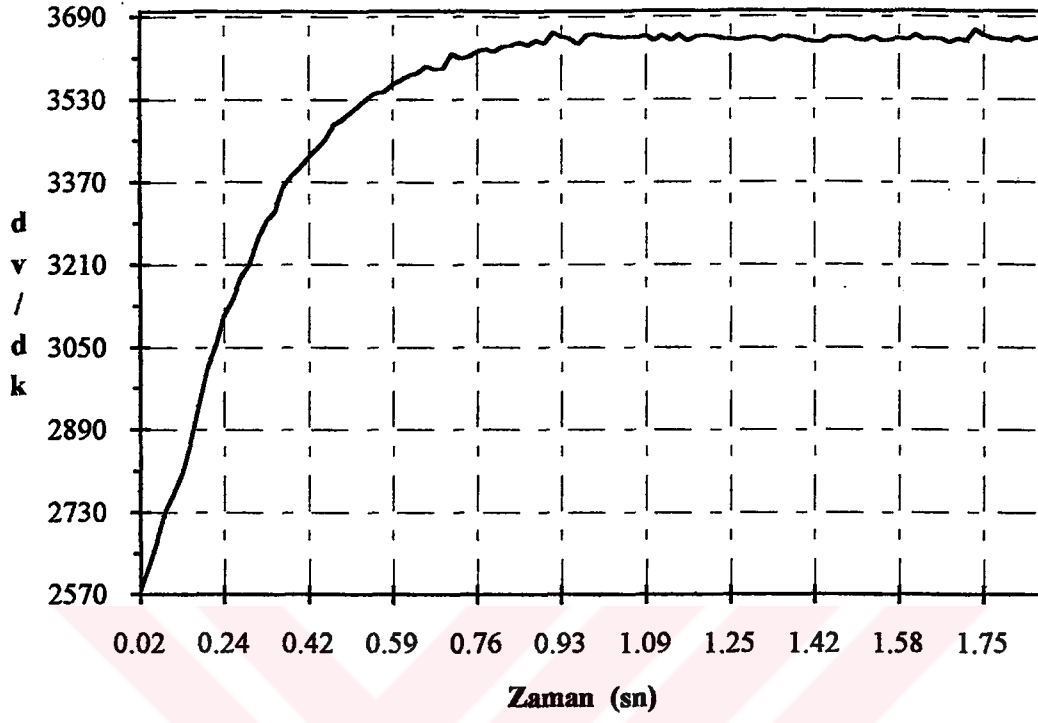




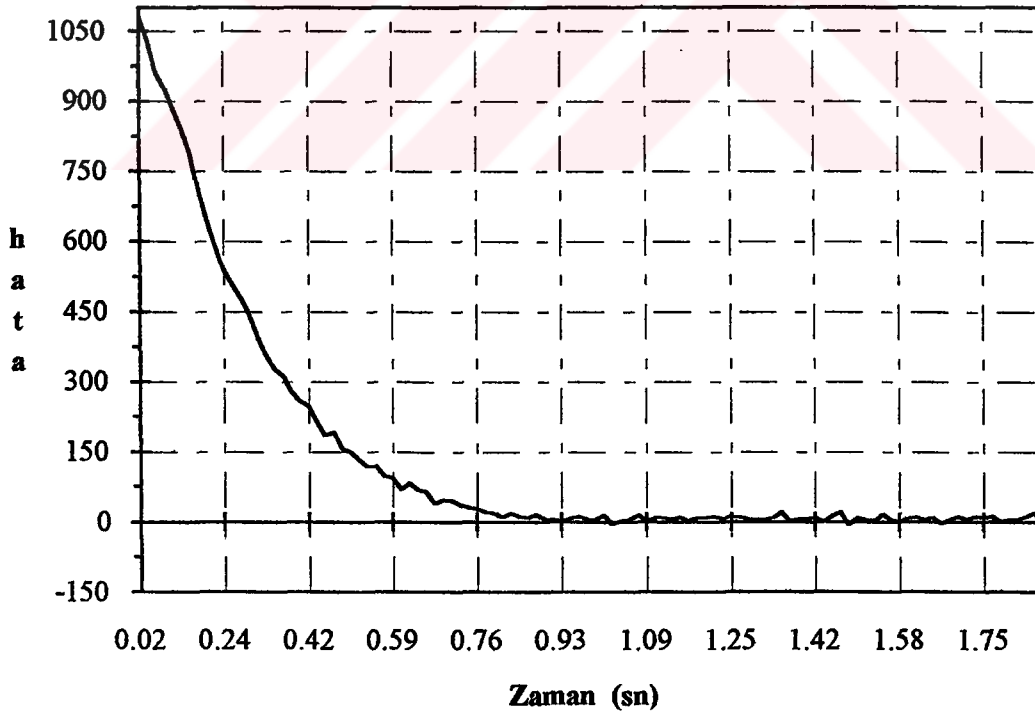
Şekil 15.13



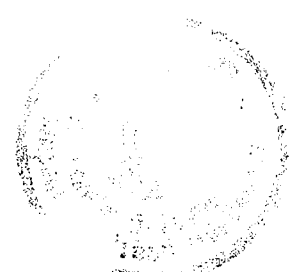
Şekil 15.14



Şekil 15.15



Şekil 15.16

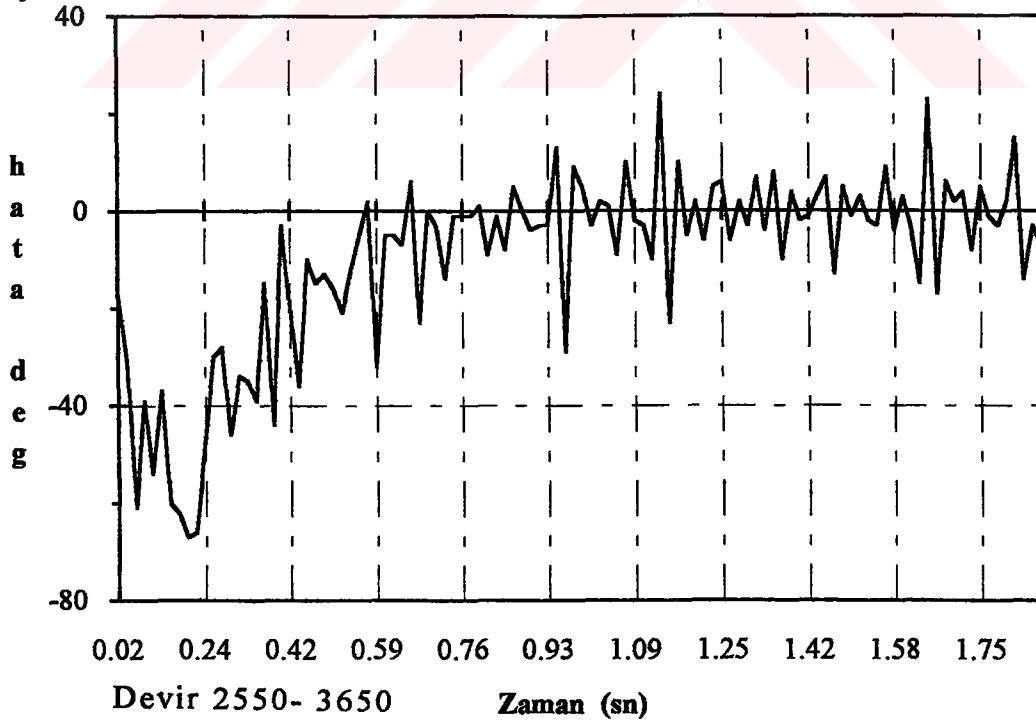


Şekil 15.15, 15.16 ve 15.17 devir kontrolu yapılan motorun hızı 2550 dv/dk'dan 3650 dv/dk'ya çıkarıldığındaki devir, hata ve hatanın değişimi eğrilerini göstermektedir. Devir daha fazla arttığı için motorun oluşturduğu titreşimler şekil 15.17'de hatanın değişimi eğrisinde olduğu gibi biraz daha artmıştır.

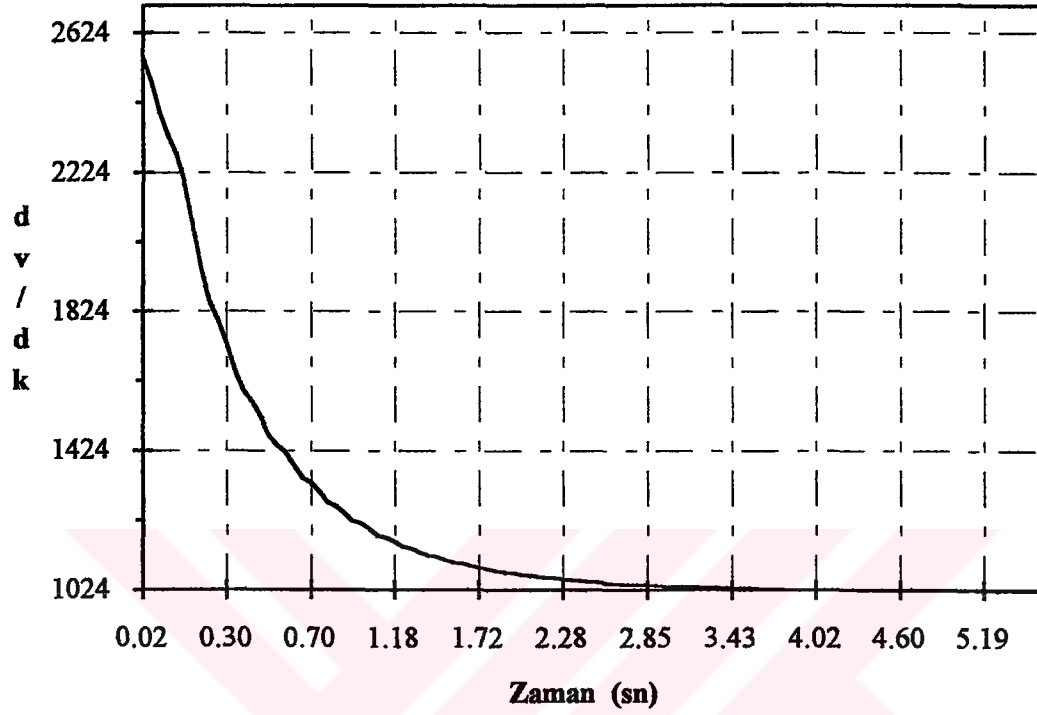
Şekil 15.18'den şekil 15.26'ya kadar olan eğriler motorun devrinin belirli bir değerden, hızı daha az sabit hızlı motorun devrine düşürülmesi durumundaki eğrileri gösterir.

Şekil 15.18'de devir-zaman grafiği 2565dv/dk'dan 1024 dv/dk'ya ulaştığını ve bu devirde kilitlenen bu motorun sürekli duruma erişme süresi 3.2 sn olmuştur.

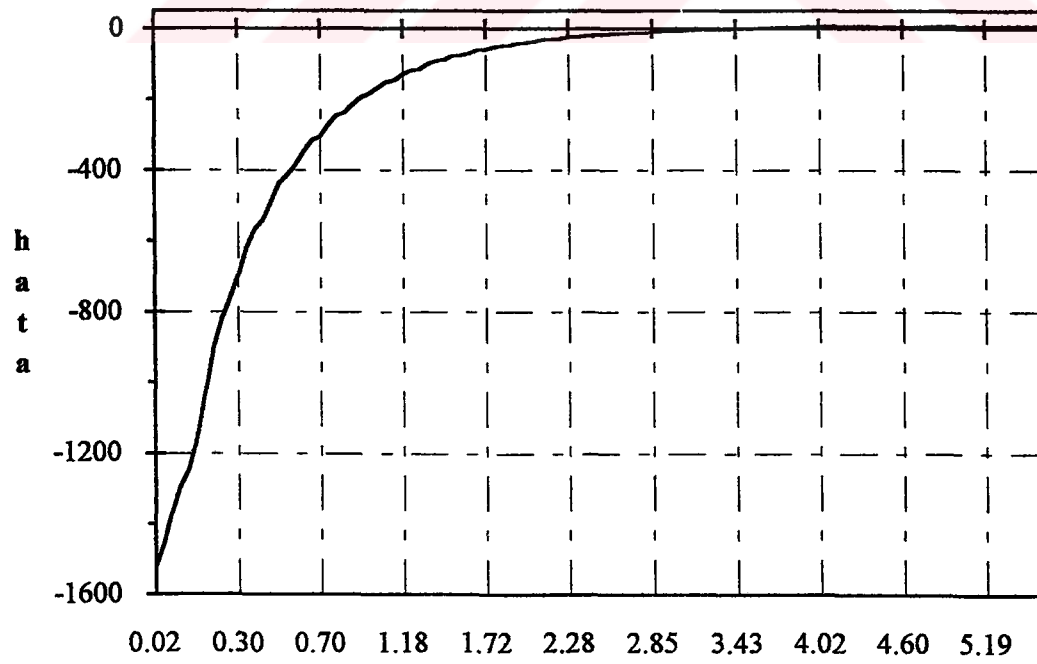
Şekil 15.20'de hata değişim-zaman eğrisinde hızı kontrol edilen motorun devri 1024 dv/dk'ya kilitlendiğinde salınım genliklerinin son derece küçük olması dikkat çekici olup bunun nedeni ise motor hızının düşmesinden ve dolayısıyla meydana gelen titreşimin azalmasından kaynaklanır.



Şekil 15.17

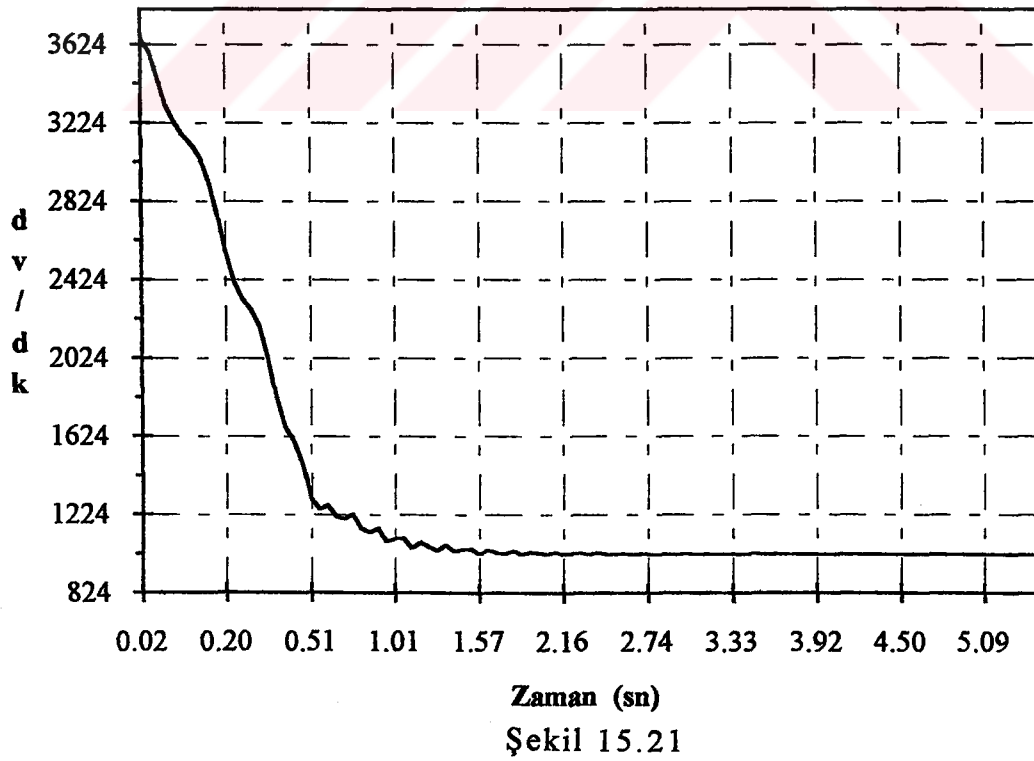
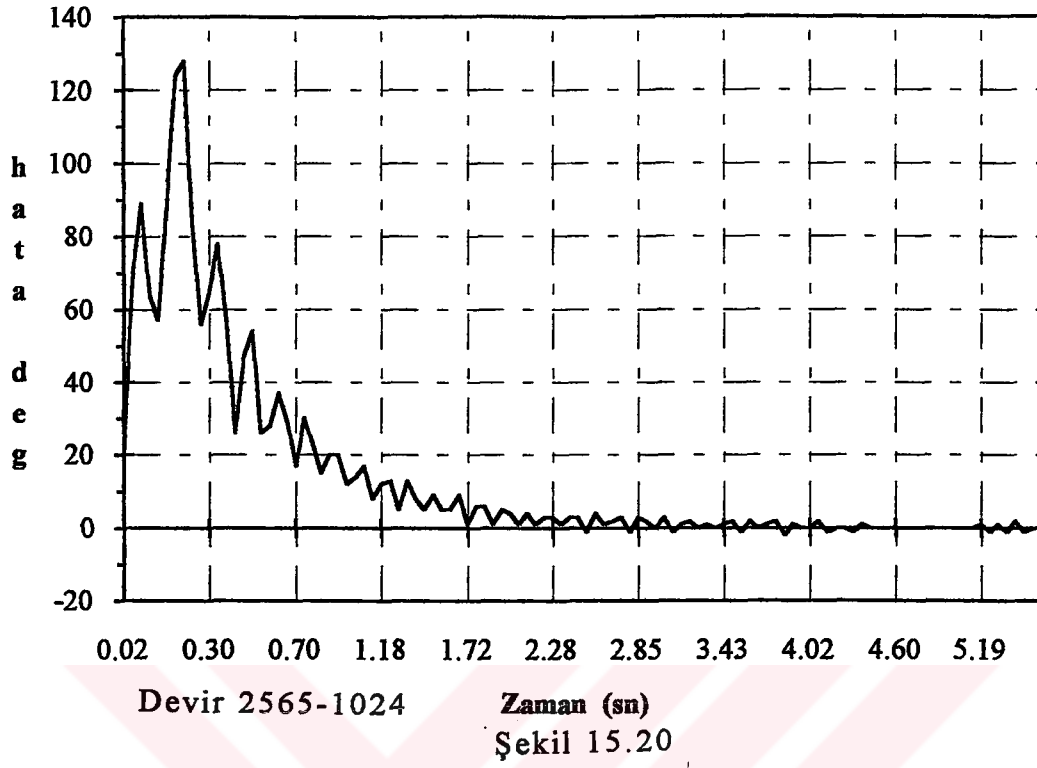


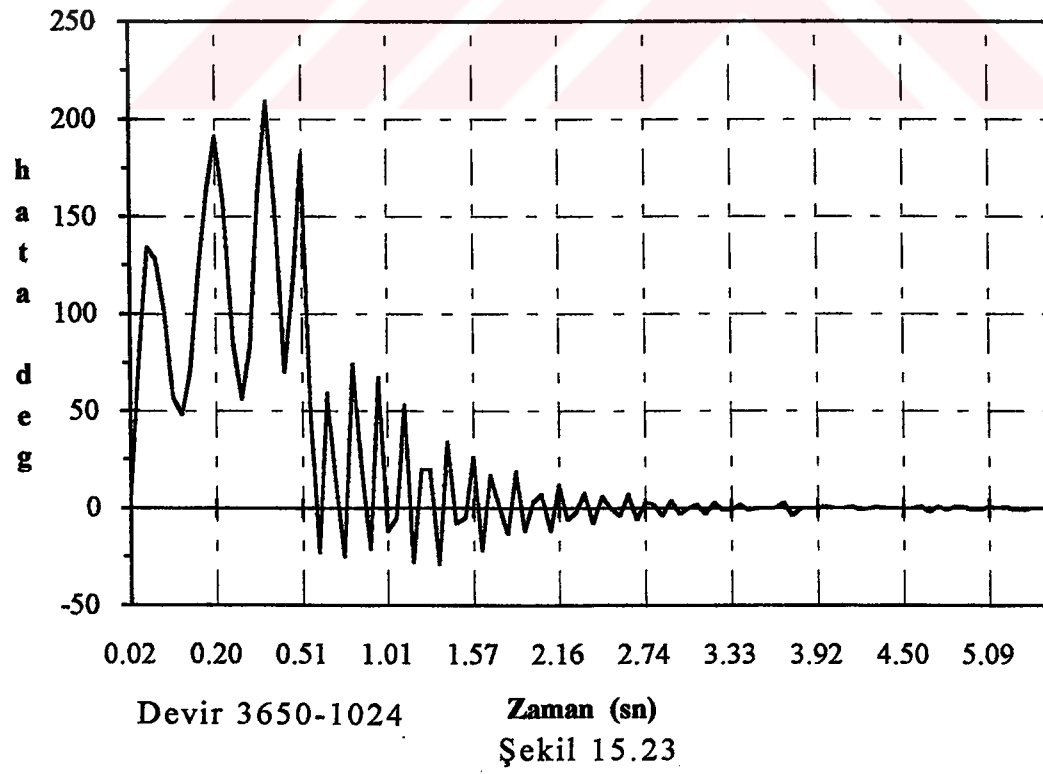
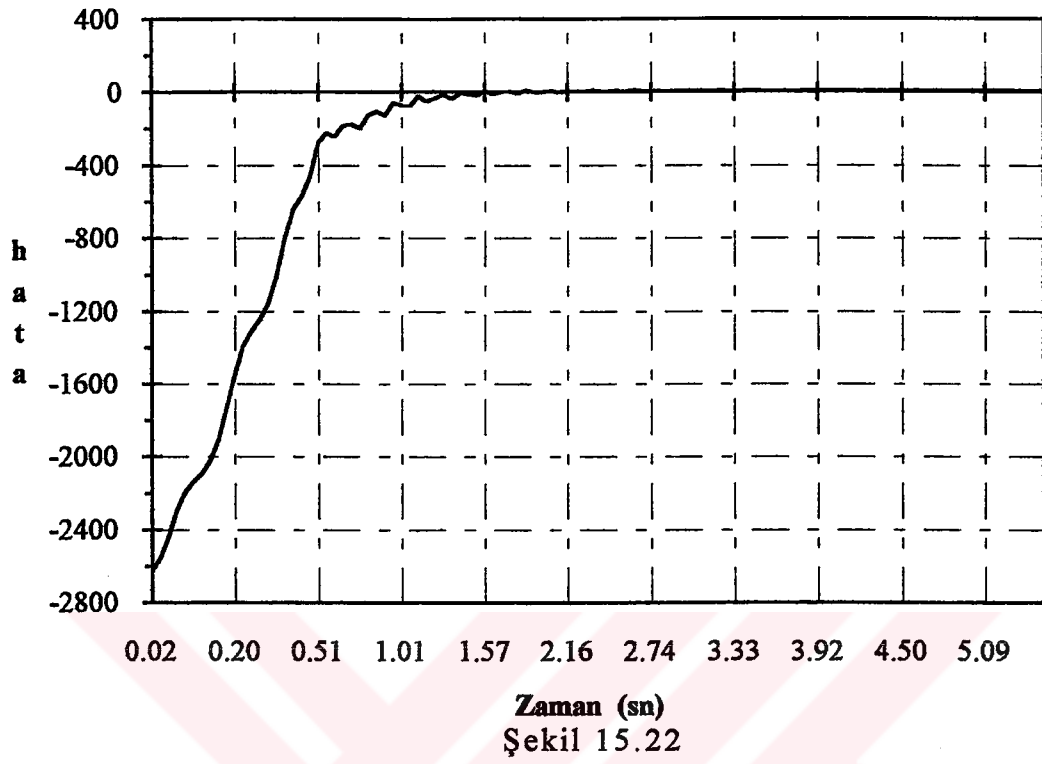
Şekil 15.18

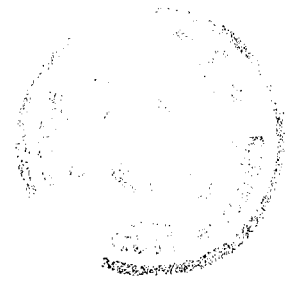
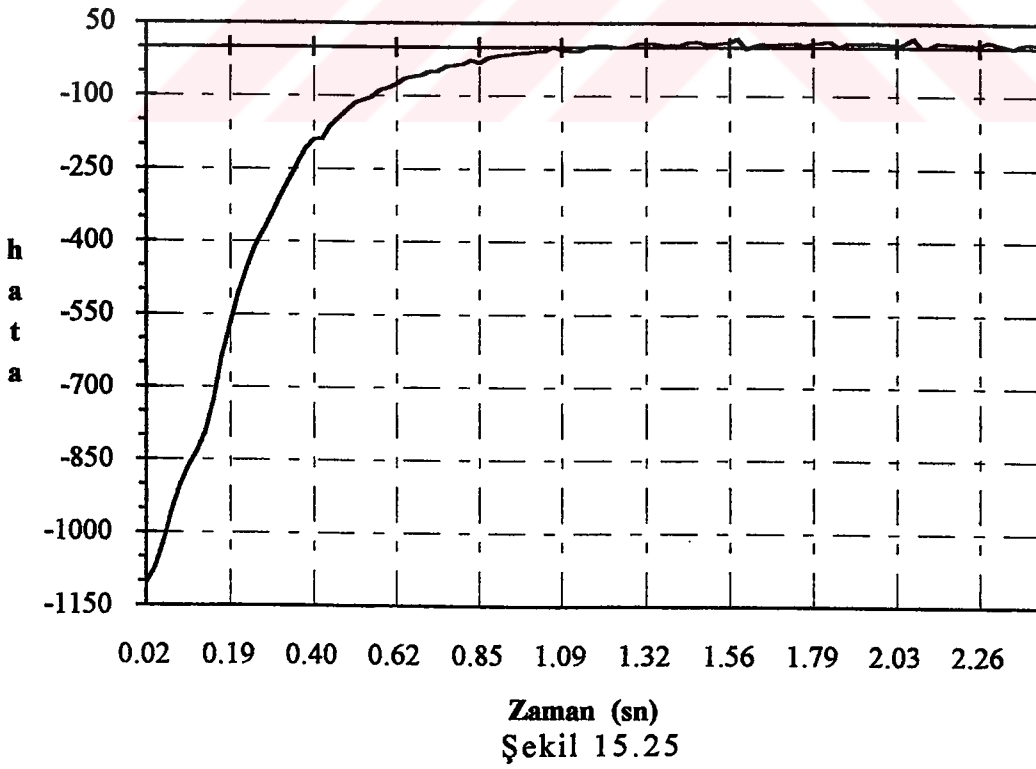
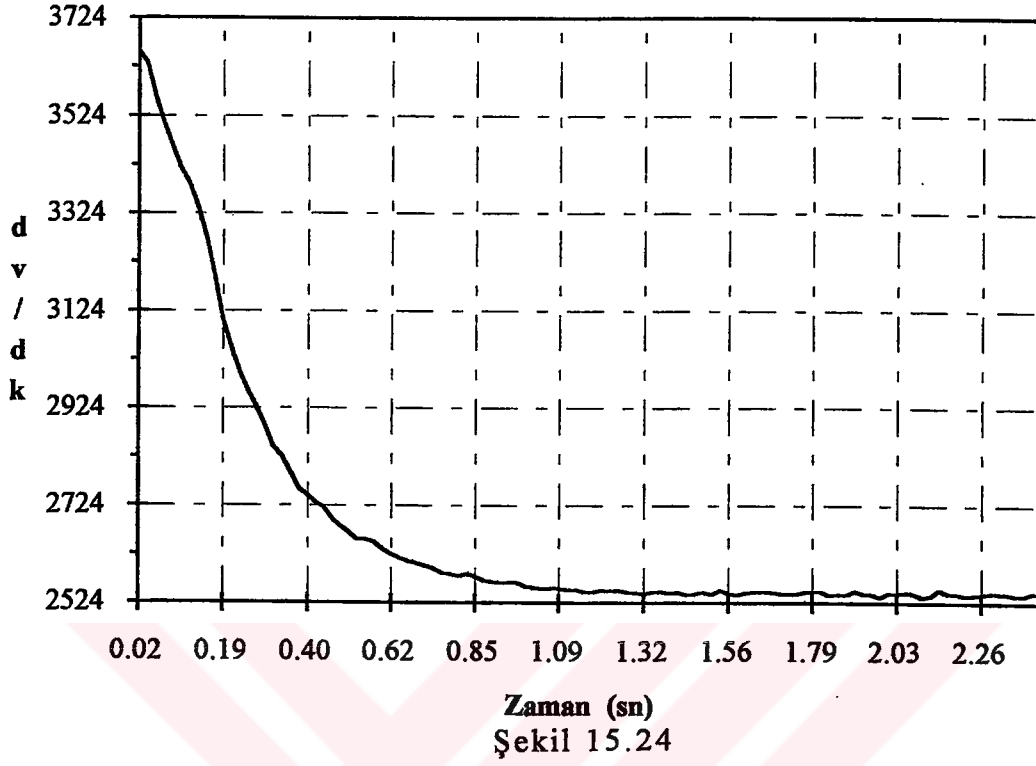


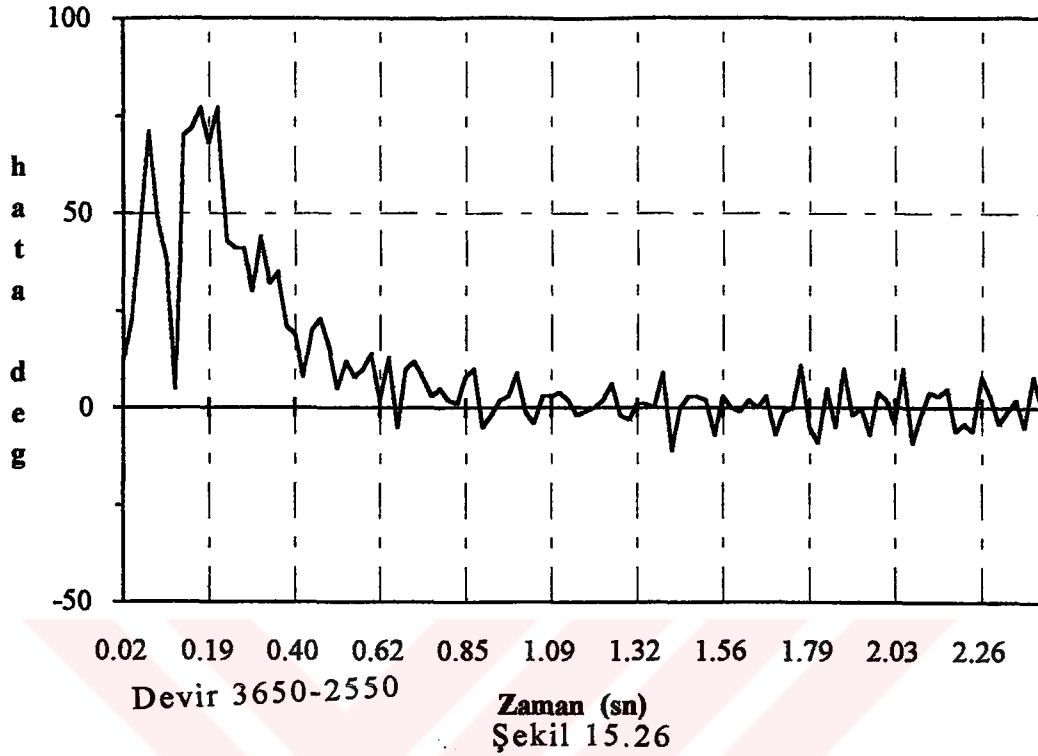
Şekil 15.19







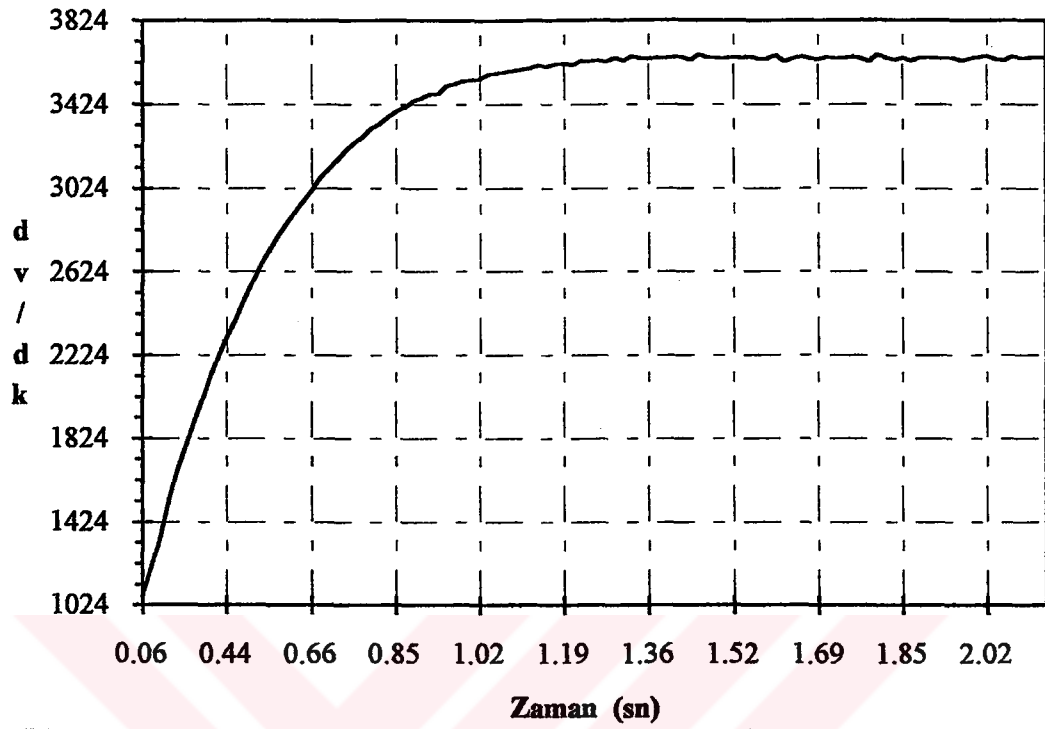




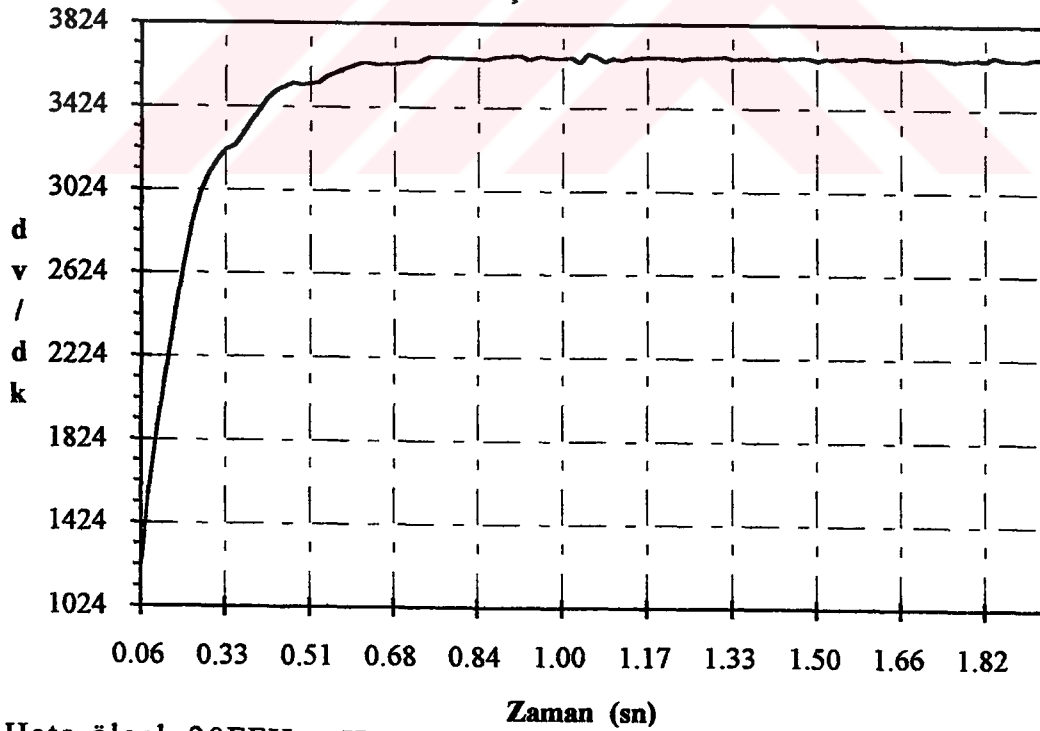
15.1. Tasarlanan Fuzzy lojik denetleyicinin çeşitli ölçek değerleri için performansının eğriler ile değerlendirilmesi:

Şekil 15.27'den şekil 15.37'ye kadar olan eğriler çeşitli ölçeklemelere karşın devir-zaman grafiklerini göstermektedir. Eğrilerden görüleceği üzere ölçekleme sınırlarını arttırdığımızda fuzzy algoritmada hassasiyetin azaldığını ve kalıcı hıza erişim sürelerinin uzadığını görebiliriz. Ölçekleme sınırlarını daralttığımızda ise hassasiyetin arttığını ve bununla beraber kalıcı hıza erişme süresinin azaldığını görebiliriz. Yani daha küçük ölçekleme sınırlarında sistemin kontrol edilme hızının artmasına neden olur. Fakat bu durum fuzzy işlemcideki ölçekleme ile karıştırılmamalıdır. Çünkü fuzzy işlemcinin ölçek alanının geniş olması (bit kapasitesi) hassasiyetin artması demektir. Fuzzy denetleyicili bir sistemde normal ekseninde hassasiyetin azalması durumunda kalıcı hıza salınımı daha az olan bir eğri ile yaklaşmak mümkün olur. Fuzzy denetleyici girişindeki değişmeler çıkışa az olarak yansır. Dolayısıyla erişim daha yavaş olur.

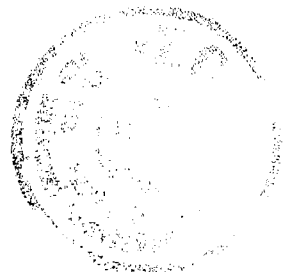


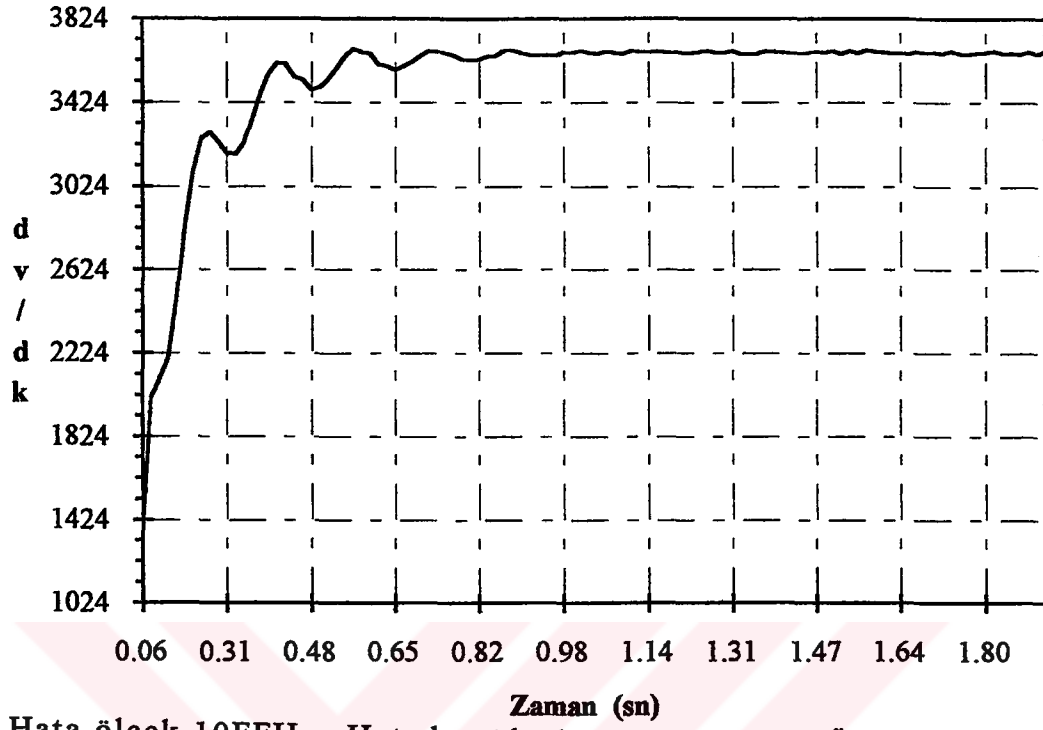


Şekil 15.27

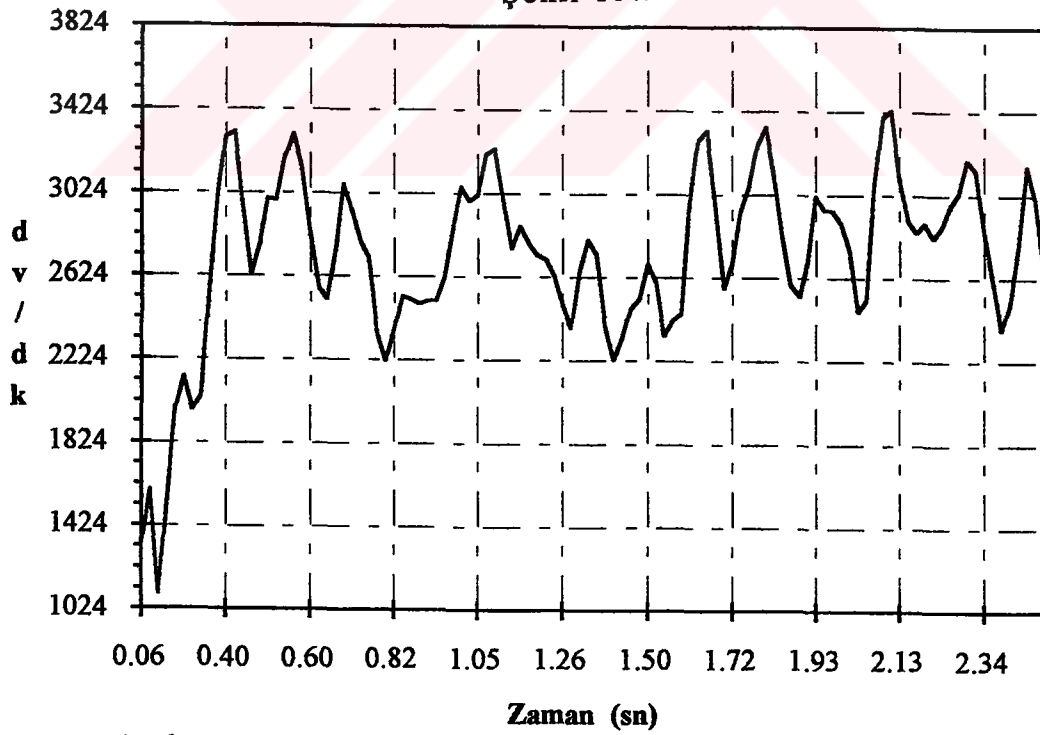


Şekil 15.28

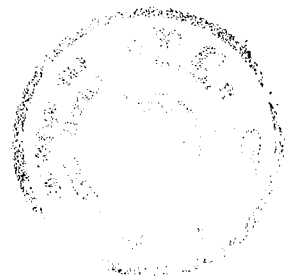


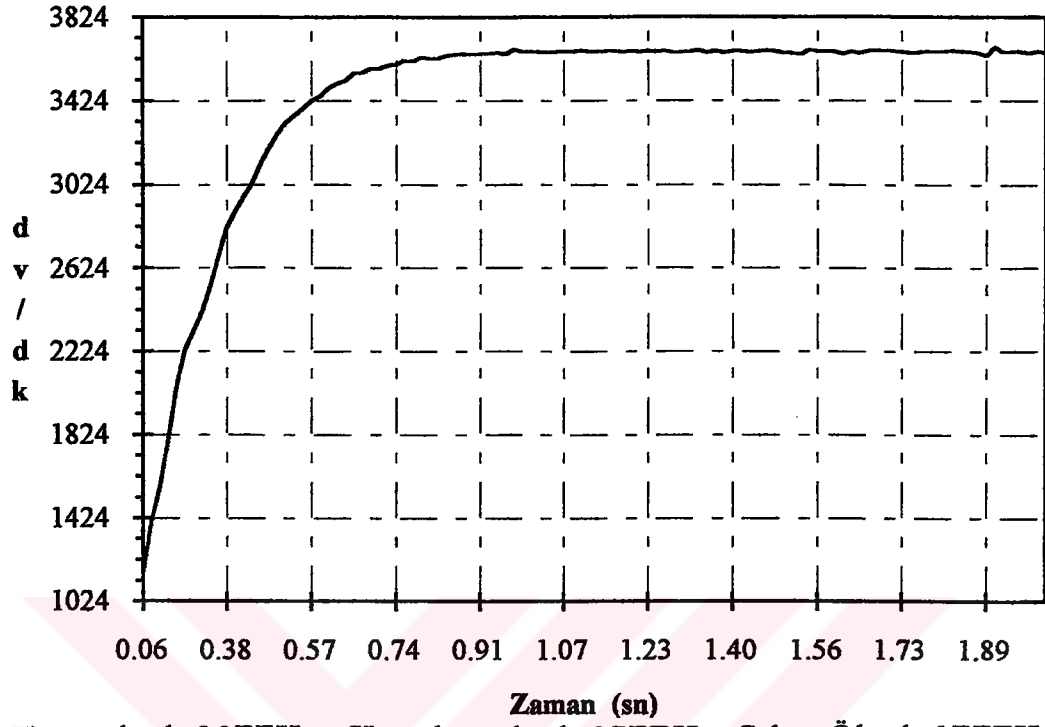


Şekil 15.29

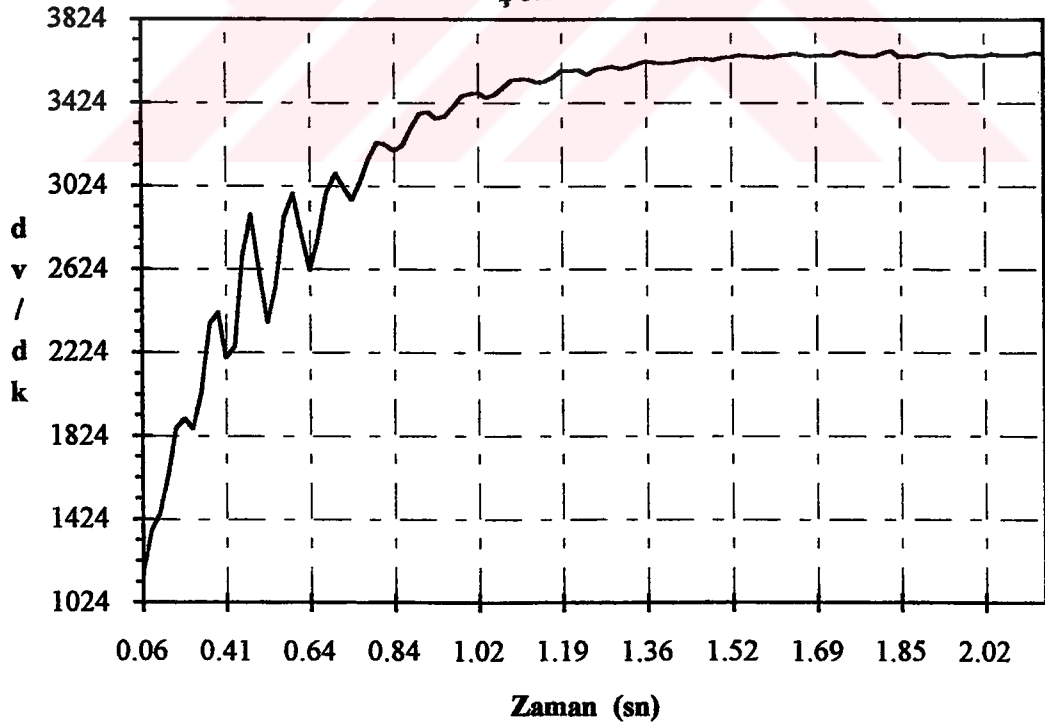


Şekil 15.30



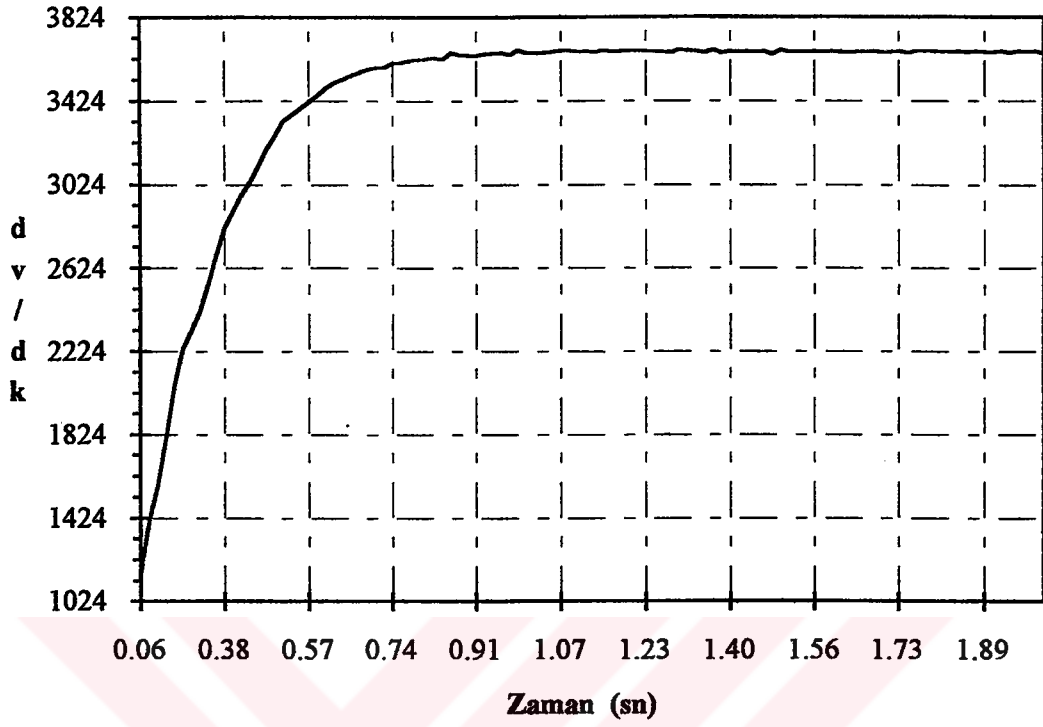


Şekil 15.31

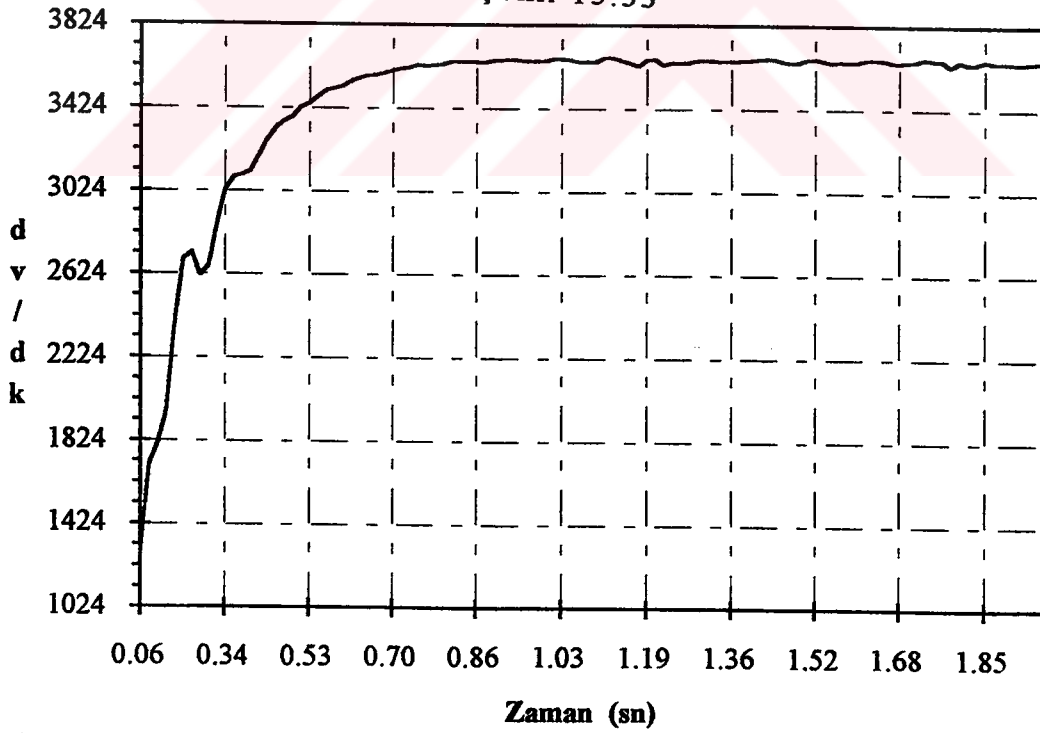


Şekil 15.32



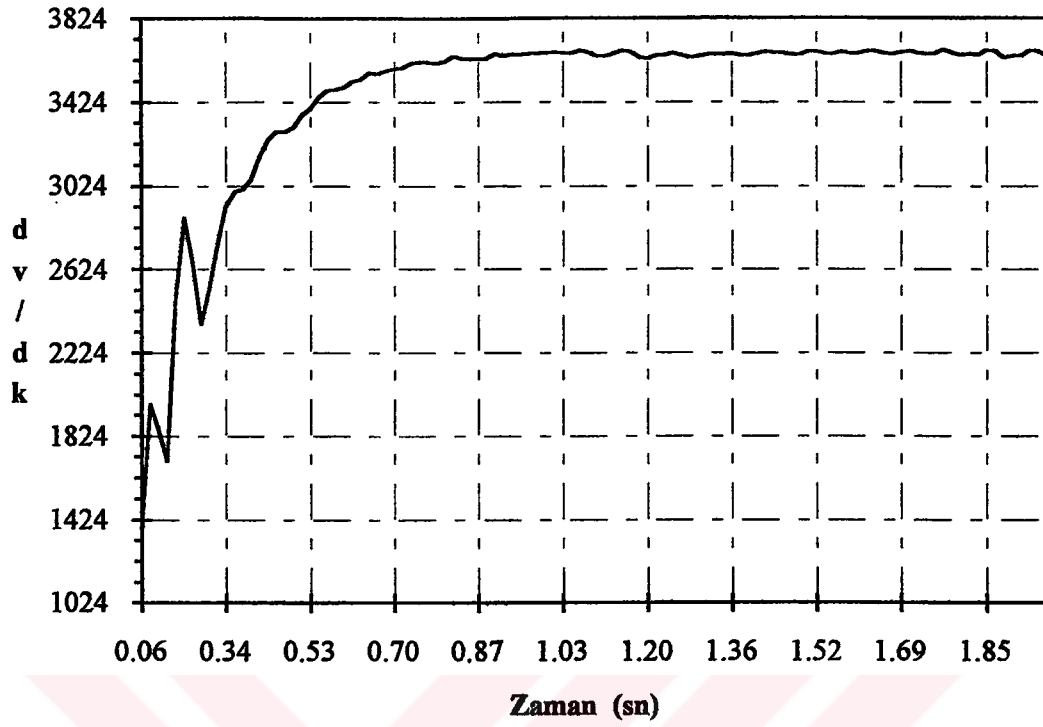


Hata ölçek 30FFH Hatadeg ölçek 07FEH Çıkış Ölçek 1000H
Şekil 15.33



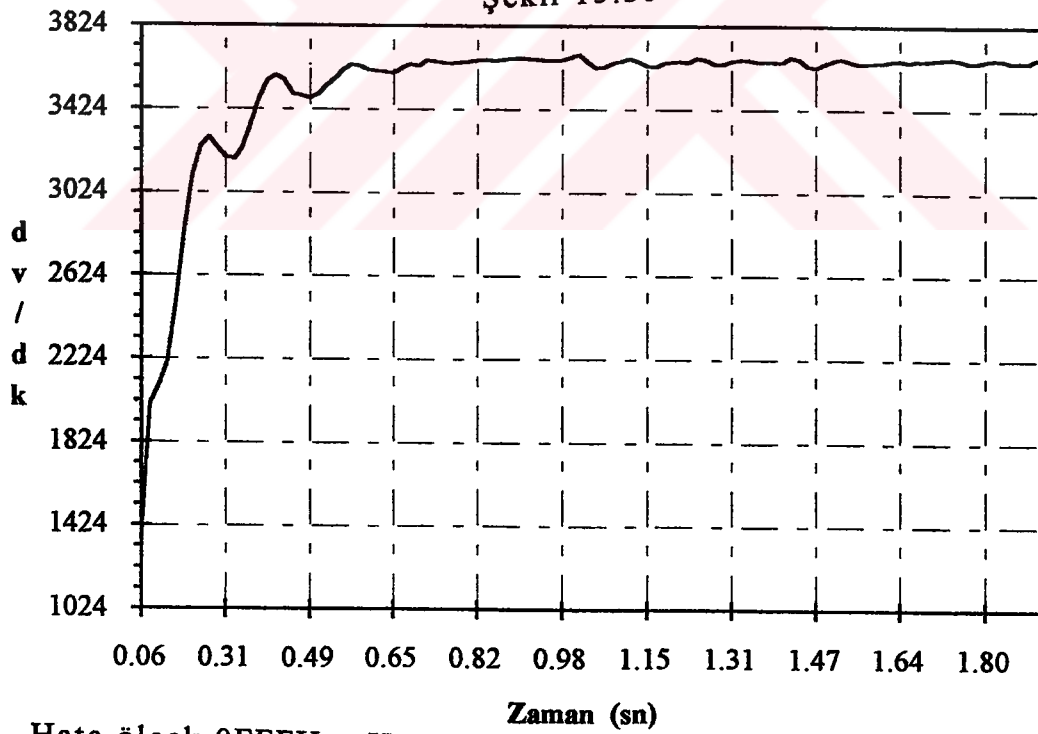
Hata ölçek 30FFH Hatadeg ölçek 07FEH Çıkış Ölçek 2000H
Şekil 15.34





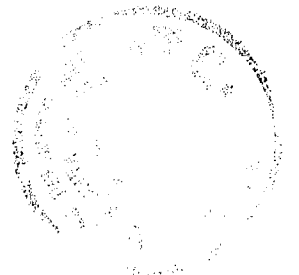
Hata ölçek 30FFH Hatadeg ölçek 07FEH Çıkış Ölçek 3000H

Şekil 15.35

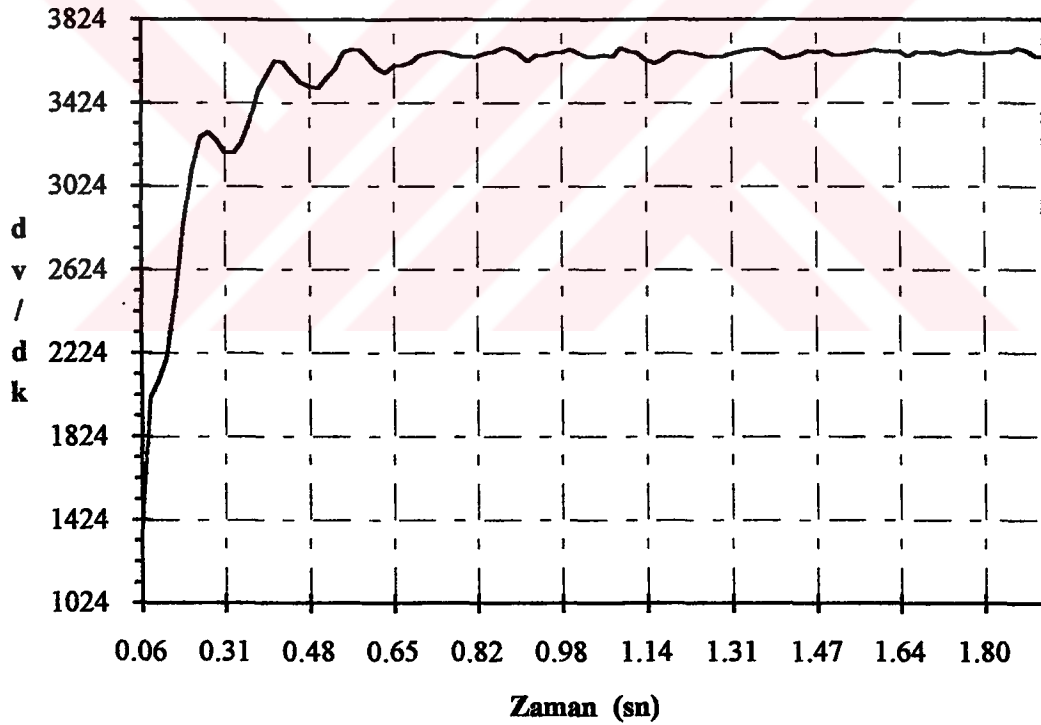


Hata ölçek 0FFFH Hatadeg ölçek 07FEH Çıkış Ölçek 3000H

Şekil 15.36

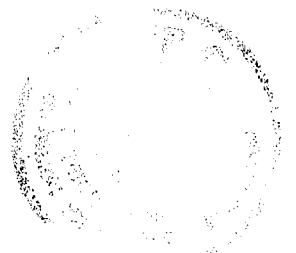


Hassasiyet arttırıldığı durumda ise kalıcı hıza erişim daha hızlı olacak fakat salınım daha da artar. Yani girişteki değişimler çıkışa daha büyük olarak yansır. Bu nedenle fuzzy lojik denetleyici tasarımcıları bu duruma dikkat etmeleri ve aynı zamanda ölçekleme sınırlarının belli bir değerin altına düşmesi durumunda sistemin kararsızlığa girmemesi için düşünülmesi gerekir. Şekil 30'da bu durum açıkça görülmektedir. Böyle bir sistemde fuzzy algoritma tasarlanırken dikkat edilecek diğer husus ise ölçeklemedeki hata giriş ölçeünün sistemin kontrolünde diğer ölçeklere göre daha etkili olduğu göz önüne alınmalıdır. Ayrıca diğer bir husus çıkış ölçeği hata ve hatanın değişimi ölçeklemesi işleminin tersi olarak sistem kontroluna etki yapmaktadır.

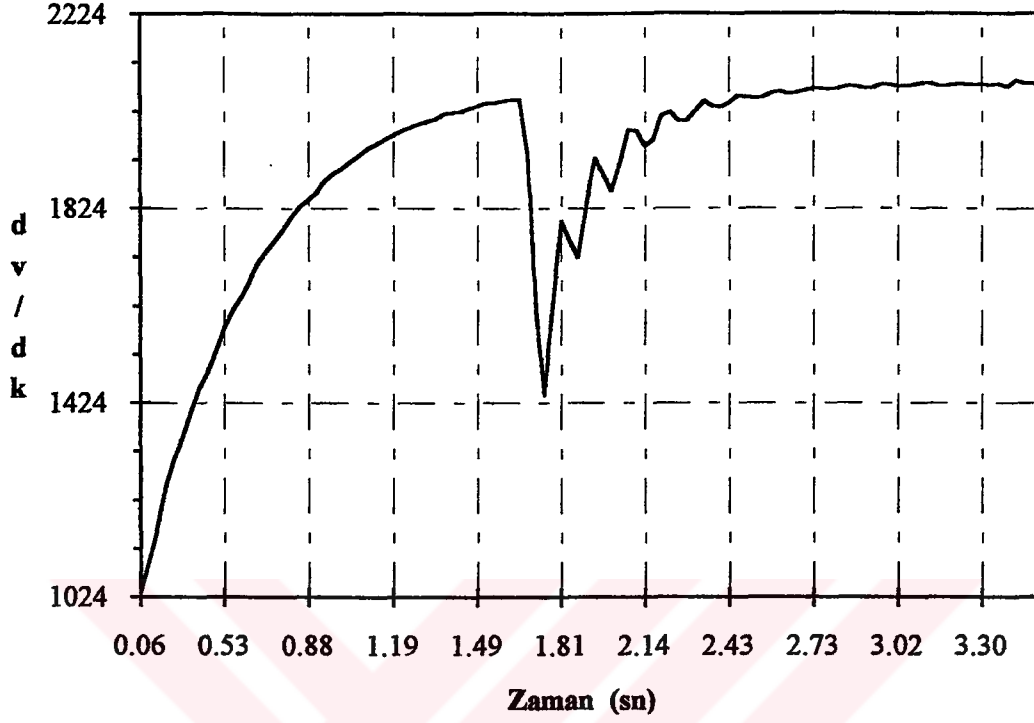


Hata ölçek 0FFFH Hatadeg ölçek 2000H Çıkış Ölçek 3000H

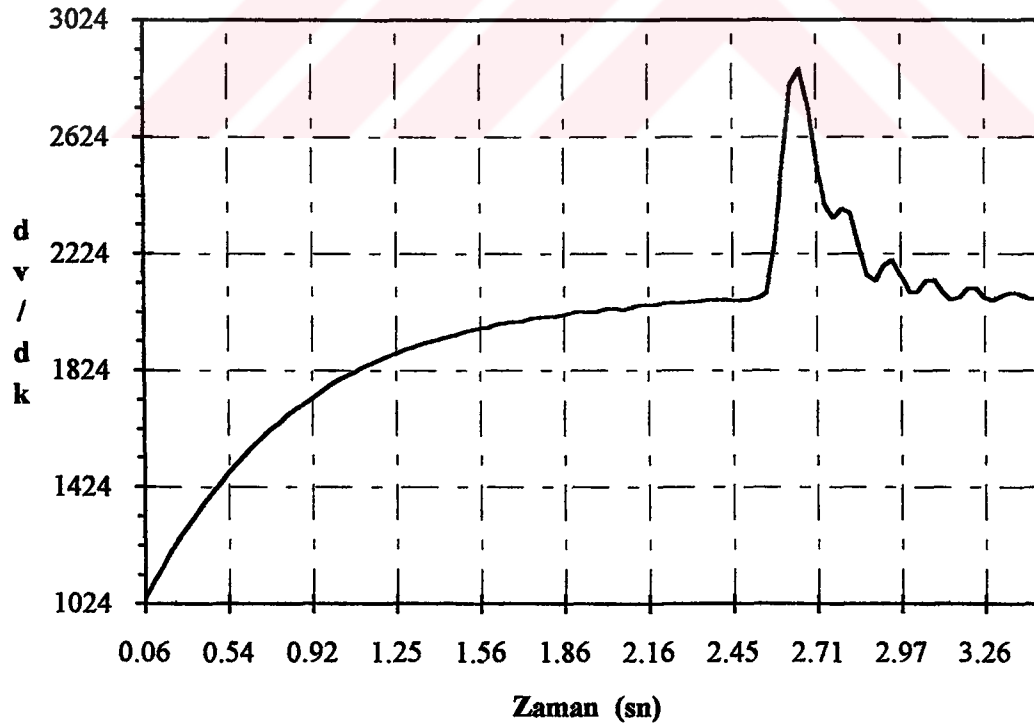
Şekil 15.37



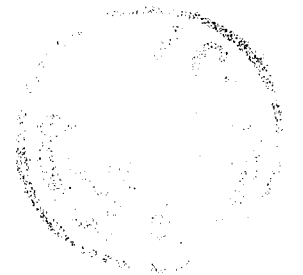
15.2. Sistemin Yükleneş ve Yükle Bırakılması Durumu:



Şekil 15.38

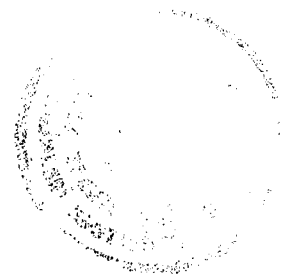


Şekil 15.39



Şekil 15.38 sistemin yüksüz durumdan yüklü duruma getirilmesi durumundaki eğriyi gösterir. Sistemin yüklenmesinde kullanılan jeneratör çıkışına yük olarak bağlanan omik karakterde 2.2Ω 'luk bir direnç özelliğindedir. Şekil 15.38'de görüleceği üzere hızı kontrol edilen motorun devri yüksüz durumda 1024 dv/dk'dan başlayıp 1.65 sn'de 2100 dv/dk kalıcı devir değerine ulaştığında sisteme bağlı olan yük devreye girmekte ve sistem 0.15 sn içerisinde motor devri 1430 dv/dk'ya düşmekte ve sistem kendini 1 sn içerisinde toparlayarak tekrar nominal devir değerine, ulaşması gereken devire kilitlenmiştir. Bu kilitleme süresini daha aza indirmek için önceden anlatıldığı gibi ölçekleme sınırlarını en iyi şekilde değiştirmek gerekir. Ayrıca devir bilgisi motor bir tur atmadan daha önce alınırsa sistemin performansı artar. Bu işlem bir shaft encoder ile sağlanabilir.

Şekil 15.39'daki devir zaman grafiğine bakıldığında yüklü durumda 1024 dv/dk'dan 2100 dv/dk'ya 2.57 sn'de ulaşmakta ve burada jeneratöre bağlı yük bırakıldığında motorun hızı 0.15 sn'de 2824 dv/dk'ya çıkmakta ve daha sonra kendini 0.61 sn'de toparlayıp nominal devire kilitlenmiştir.



16. SONUÇLAR

Daha önce aynı konuda yapılan çalışmalarda bilgisayara ilave edilen FB 30-AT kartı ve bu kart için kullanılan FS 30-AT yazılımı kullanıldı. Bu kart ve program vasıtasıyla simülasyon çalışmaları yapıldı. Çalışmalar sonucunda elde edilen eğriler incelendi, (Cansever ve diğerleri, 1993a; Cansever et. al.,1993b; Cansever and Özgüven, 1994, 1995). Fakat gerçek ortamda herhangi bir uygulama yapılmamıştı. Bu çalışma ile bilgisayardan bağımsız kendi başına kontrol edilecek her sisteme uyarlanabilen fuzzy lojik kontrol ünitesi tasarlanmış ve mikrodnetleyici ile uygulaması yapılmıştır.

Önce tasarlanan mikrodnetleyicili anakart, gösterge ve tuş takımı ile makina dilinde yazılan bir programın sadece kodlarını sistemin belleğine yazma imkanı elde edilmiştir. Bunun için monitor programı yazılmış ve yazılan programın çalıştırılması sağlanmıştır. Daha sonra bilgisayarda yazılan bir makina dili program derlendikten sonra oluşan *.OBJ dosyasının içerisindeki gerekli program kodları alınmış ve seri porttan mikrodnetleyicili anakartta gönderilmiştir. Böylece bilgisayarda yazılan böyle bir programın fazla zorluk çekilmeden anakartta çalışması sağlanmıştır. 8XC196KC/KD mikrodnetleyici ile çalışacak ve uygulama yapacak araştırmacılar için bir geliştirme seti tasarımı gerçekleştirilmiştir.

Anakart üzerinde bulunan genişleme yuvaları sayesinde, istenilen amaca göre yeni uygulamalar için gerçekleştirilen yeni ünitelerin sisteme ilave edilmesi uygun duruma getirilmiştir. Yani sistem geliştirilmeye açıktır.

Sistem üzerinde fuzzy lojik kontrol işlemini gerçekleştirebilmek için FP 3000 fuzzy işlemci kullanılarak bir ünite tasarlanmış ve uygulanmıştır. Çıkarım ve netleştirme işlemi bu ünite üzerinde gerçekleştirilmiştir. Böylece kontrol işleminde donanım kullanıldığı için

işlemlerin daha hızlı olması sağlanmıştır. Ayrıca ünite sayesinde bu işlem için yazılan yazılımın daha da azalması mümkün olmuştur.

DAC 1200 tümdevresi kullanılarak 12 bitlik dijital analog çevirici devresi tasarlanarak sisteme ilave edilmiştir. Kontrol uygulaması için iki DC motorlu sistem üzerinde çalışılmıştır. Fuzzy lojik kontrol tekniği kullanılarak isteğe göre değiştirilen sabit hızlı motorun devir sayısına diğer motorun devir sayısı eşitlenmiştir.

Fuzzy işlemcinin kullanımı için simülasyon programı analiz edilmiştir. Böylece işlemcinin bilgi tabanı, giriş ve çıkış verilerinin yazıldığı adresler bulunmuştur. Ayrıca işlemciyi çıkarıma başlatma ve ve çıkarımı sonuçlama işlemlerinin nasıl yapılacağı anlaşılmıştır. Fuzzy işlemcili ünitenin donanımı için yukarıda sözü geçen FB 30-AT kartı

Tablo 16.1. Diğer kontrol algoritmaları ile Fuzzy Lojik Kontrolün karşılaştırılması

Diğer Kontrol algoritmaları	Fuzzy Lojik Kontrol
Sistemin transfer fonksiyonuna ihtiyaç duyar.	Sistemin transfer fonksiyonuna ihtiyaç duyulmaz.
Sistem karmaşık duruma geldikçe gerçekleştirme gücü oluşur ve maliyet artabilir.	Uzman kişinin tecrübelerinden yararlanılır ve kontrol kurallarına ihtiyaç vardır.
Çoğunlukla bilgisayara bağımlı olarak çalışırlar.	Kontrol olayı hızlıdır.
Tasarım ve gerçekleştirme uzun zaman alabilir.	Fuzzy işlemci ve mikrodenetleyici bir sistemle yalnız başına gerçekleştirilebilir.
Kesin ölçüm değerlerini kullanılır.	Kesin olmayan ifadeleri kullanabilir.
	Uygulamada kolaylık sağlar.
	Algılayıcılara ihtiyaç vardır.



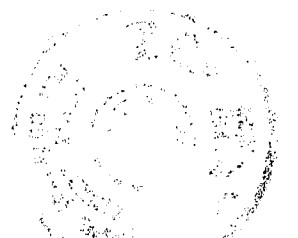
incelenmiş ve bu kartın donanım yapısı hakkında bilgi edinilmiştir. Bu şekilde işlemcinin hem yazılım ve hemde donanım bakımından mikrodenetleyicili sisteme uyarlama işlemi gerçekleştirilmiştir.

Elektronik devrelerin çizimi ve baskılı devrelerinin tasarımında bütünüyle elektronik devre çizim ve baskılı devre tasarımı yapan bir paket program kullanılmıştır, (OrCAD, 1989a, 1989b). Bu program elektronik bir devre için gerekli olan yazılım uygun bir şekilde kullanılmıştır. Böyle bir durumda daha karmaşık diğer devrelerin tasarlanıp gerçekleştirilebileceği görülmüştür.

Sonuçta elde edilen devre ile başka kontrol işlemlerini fuzzy lojik kontrolu kullanarak gerçekleştirmek mümkün olmaktadır. Kontrol edilen sisteme göre sadece kurallar ve üyelik fonksiyonları değiştirilerek yeni bilgi tabanı verileri oluşturulur. Daha sonra bu veriler fuzzy işlemcili üniteye gönderilir ve kontrol olayına geçilebilir. Ayrıca mikrodenetleyicinin içerisinde bulunan analog dijital çevirici, zamanlayıcı, darbe genişlik modölatör ünitesi, giriş çıkış portları gibi çevre birimlerinin bulunması sayesinde kontrol edilen sistemden alınan bilgilerin rahatlıkla devreye iletilmesi mümkün olmaktadır. Böylece genel amaçlı fuzzy lojik kontrol sistemi elde edilmiş olmaktadır.

Daha ilerisi için bu sisteme yeni geliştirmeler ilave edilebilir. Kurallar ve üyelik fonksiyonları tuş takımından verilir. Daha sonra bu sistem üzerinde derleme işlemi yapılarak fuzzy işlemci için gerekli olan bilgi tabanı verileri oluşturulabilir. Bilgi tabanı verileri fuzzy işlemci ünitesine yazıldıktan sonra kontrol işlemine geçilebilir.

Fuzzy lojik kontrol işleminde üyelik fonksiyonları seçiminde belirli bir yöntem yoktur. Ayrıca kuralları yazmak için uzman kişinin tecrübelerine ihtiyaç duyulur. Bu yüzden bu konudaki güçlüklerin aşılması için yapay sinir ağları ile fuzzy lojiğin bir araya getirilmesi çalışmaları devam etmektedir.



Fuzzy Lojik Kontrol işlemi diğer kontrol algoritmaları ile tablo 16.1 'deki gibi kısa olarak karşılaştırılabilir.

16.1. Çalışmanın maliyet hesabı ve benzer diğer üniteler ile karşılaştırılması:

Tablo 16.2. Çalışmanın maliyet hesap tablosu

Kartın Adı	Maliyeti (T.L.)
N80196KC Mikrodenetleyicili Geliştirme seti (Anakart)	9. 400. 000
FP3000 Fuzzy İşlemcili Kart	9. 500. 000
Keyboard ve Display kartı	3. 900. 000
Digital - Analog Çevirici (DAC) kartı	4. 700. 000
Toplam:	27. 500. 000

Tablo 16.3.Piyasada bulunan benzer ünitelerin fiyat tablosu

Kartın Adı	Maliyeti (T.L.)
N80196KC Mikrodenetleyicili Geliştirme seti	85. 000. 000
FB 10- AT Fuzzy İşlemcili Kart	120. 000. 000
Digital - Analog Çevirici (DAC) kartı	40. 000. 000
Toplam:	245. 000. 000

Not: 1 \$ = 170 000 T.L. alınarak hesap yapılmıştır.

Bu tezde bütün çalışmalar laboratuarda kendi imkanlar ile gerçekleştirilmiştir. Bu sistem üzerinde gerekli yazılımlar yapılmıştır. Dolayısıyla oluşturulan sistem en büyük yarar olarak piyasaya göre ekonomik yönden daha az maliyet getirmiştir. Sistem en küçük ayrıntısına kadar bilindiği için sistem üzerine yeni ilavelerin yapılması, geliştirilmesi mümkündür. Geliştirme seti yapısı bilindiğinden diğer mikroişlemciler ve mikrodenetleyiciler için geliştirme seti tasarlanabilir. Bu durum yine ekonomik yönden daha az maliyet getirecektir. Maliyetinin az ve sistemin uygun olmasından dolayı eğitimde kullanılabilecek özelliklere sahiptir.

http://tbsv.telemarkt.technopark.gmd.de/tt/www/preise_e.htm
inter-net adresinden alınan aynı cins , kalite ve fonksiyonları yerine getiren cihazların fiyatları aşağıda DM olarak verilmiştir. Bu nedenle donanım ve yazılım fiyatları ayrı ayrı verilmiş olup bunların toplamı (3. ve 4. sırada) 5400 DM gözükmekte yaklaşık 500.000.000 T.L. olmaktadır. Yapmış olduğumuz cihaz yaklaşık 17 kat benzer emsallerinden daha ucuzdur. Yurt içinden ilgili üniteleri satın almaya çalışırsak sadece donanımı 8 defa daha pahalı olduğu görülmektedir. Böylece hem yurtiçi ve hem yurtdışından benzer cihazlarla karşılaştırma yaptığımızda ekonomik yönden büyük avantajlara sahiptir.

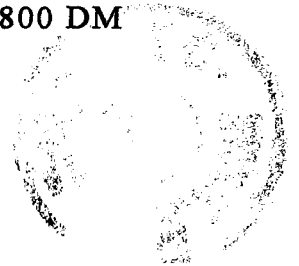
Price List 03/1996

Price list from March 1996.

All prices in DM; shipping costs and taxes (if applicable) not included.

Fuzzy Control Manager

- | | | |
|----------------|---|---------|
| 1.FCM-Standard | Fuzzy development system with universal ANSI-C-code output | 3100 DM |
| 2.FCM-3000 | Fuzzy development system with C-code output supporting the FP-3000 (without hardware) | 1800 DM |
| 3.FCM-SOFT | Reductioned Fuzzy development system for VMEbus board VMEFP3000 (without hardware) | 1800 DM |



4.FCM-TEAM	Fuzzy development system including FP-3000 microcontroller board	3600 DM
5.FCM-SPS	Fuzzy development system for fuzzy modul FZ001 (without hardware)	1800 DM
6.FCM-Modicon	Fuzzy development system for Quantum PLC (without hardware and "Concept" software module)	(inquire)
7.FCM-I3HD-CINT	Fuzzy development system for microcontrollers of 78K0-, 78K3-, V-series (integer C-code output)	1800 DM
8.FCM-I3HD-75X	Fuzzy development system for microcontrollers of 75X family (assembler source code output)	1800 DM
9.FCM-I3HD-17K	Fuzzy development system for microcontrollers of 17K family (assembler source code output)	1800 DM
10.FCM-PX-1500	Fuzzy development system for microcontrollers of MN1500 family (assembler source code output)	1800 DM
11.FCM-HC11	Fuzzy development system for microcontroller 68HC11 (assembler source code output)	1800 DM
12.FCM-8051	Fuzzy development system for microcontrollers of 8051 family (assembler source code output)	1800 DM
13.FCM- μ C	Fuzzy development system for several microcontrollers (integer-C-code output)	2500 DM
14.FCM-T800	Fuzzy development system for Transputer	2500 DM

Fuzzy denetleme işleminde kontrol edilen sistemden direkt olarak algılayıcılardan bilgi almak gerekir. Algılayıcılardan alınan bilgi analog ise mikrodenetleyici tarafından değerlendirilebilmek için ADC ile dijitalle çevirmek gerekir. Şayet direkt bilgi geliyorsa bu bilgilerin mikrodenetleyiciye ulaşması için Giriş/Çıkış portlarına ihtiyaç vardır. Ayrıca program içerisinde zamanlayıcı ve sayıcıya ihtiyaç olabilir. Yukarıdaki çalışmalarda bu işlemleri yerine getirmek için ayrı kartlar kullanılmıştır. Bu durum maliyetin artmasına neden olmaktadır. 80 196 mikrodenetleyicisi yukarıda



sözü geçen ünitelerin hepsini kendi içerisinde barındırmaktadır. Dolayısıyla maliyetin düşmesine neden olmaktadır.

Fuzzy işlemcinin kullandığı bellek ünitesi ROM olarak alınıp, kurallar ve üyelik fonksiyonunun derlenmesinden oluşan veriler bu bellek ünitesine yazılırsa sistem bilgisayara ihtiyaç duyulmadan denetleme işlemi yapılabilir.

KAYNAKLAR

Ascia, G., Catania, V., Puliafito, A., Vita, L., 1996."A Reconfigurable Parallel Architecture for Fuzzy Processor ", Information Sciences, Vol. 88: p.p. 299 - 315

Ascia, G., Catania, V., Russo, M., Vita, L., 1996." Rule Driven VLSI Fuzzy Processor", IEEE Micro, p.p. 62 - 74

Bander, W., and Khout, L.J., 1985. Probabilistic Versus Fuzzy Production Rules in Expert System, Int. J. Man. Machine Studies, Vol. 22:p.p. 347-353

Bellman, R.E. and Zadeh, L.A., 1970. Decision Making in a Fuzzy environment, Management Sci., Vol. 17 : p.p. 141-164.

Bezdek, J.C., 1981. Pattern Recognition with Fuzzy Objective Function Algorithms, Plenum Press.

Brae, M., and Rutherford, D.A., 1979. Theoretical and Linguistic Aspects of the Fuzzy Logic Controller, Automatica, Vol.15: p.p. 553-557.

Brey, B.B., 1987. The 8086/8088 Microprocessor.

Cansever, G., Engin, Ş.N., Özgüven, Ö.F., Uzam, M., 1993b. Fuzzy Logic Controller in Modern Control System. An Industrial Applications, International AMSE Conference Systems, London, England, Vol 2:p.p.45-67

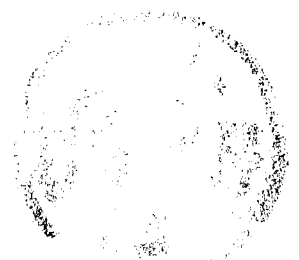
Cansever, G., Özgüven, Ö.F., 1994. Applications of Fuzzy Set Theory to Stabilization of an Inverted Pendulum by a High Speed Fuzzy Logic Controller, 7th Mediterranean Electrotechnical Conference, Vol. 3: p.p.1085-1088

Cansever, G., Özgüven, Ö.F., 1995. An Adaptive Fuzzy Controller Improving a Control System for Process Control. IECON'96 Conference TAIPEI - TAIWAN.

Cansever, G., Özgüven, Ö.F., Uzam, M., Engin, Ş.N., 1993a. Kontrol Sistemlerinde Bulanık Lojik: Bulanık Lojik Denetleyici, Elektrik Mühendisliği 5. Ulusal Kongresi, K.T.Ü.-Trabzon: p.p.57-62

Cheng, F. F., Yeh, S. N., 1993."Application of Fuzzy Logic in Speed Control of AC Servo System and an Intelligent Inverter", IEEE Trans. on Energy Conversion, Vol.8, No.2: p.p. 312 - 318

Corbet, T., Sepehri, N., Lawrence, P. D., 1996."Fuzzy Control of a Class of Hydraulically Actuated Industrial Robots", IEEE Trans. on Control Systems Technology, Vol.4, No.4: p.p. 419 - 426



Costa, A., Alessandro, G., Olivieri, M., 1996. "Hardware Design of Asynchronous Fuzzy Controllers", IEEE Trans. on Fuzzy Systems, Vol. 4., No.3: p.p. 328- 338

Eichfeld, H., Klimke, M., Menke, M., Nolles, J., Künemund, T., 1995. "A General-Purpose Fuzzy Inference Processor", IEEE Micro, p.p. 12 - 17

Eichfeld, H., Kunemund, T., Menke, M., 1996. "A 12b General-Purpose Fuzzy Logic Controller Chip", IEEE Trans. on Fuzzy Systems, Vol. 4, No.4: p.p. 460 - 475

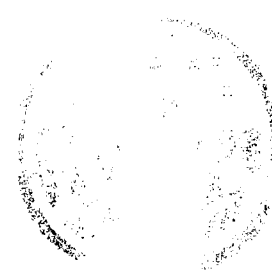
Elector, 1987. 8052AH BASIC microcontroller

El-Metwally, K. A., Hancock, G. G., Malik, O. P., 1996. "Implementation of a Fuzzy Logic PSS Using a Micro-controller and Experimental Test Results", IEEE Trans. on Energy Conversion , Vol. 11, No. 1: p.p. 91 - 96

Fattaruso, J. W., Mahant-Shetti, S. S., Barton, J. B., 1996. "A Fuzzy Logic Inference Processor", IEEE Journal of Solid-State Circuits, Vol.29, No.4: p.p.397- 402

Gains, B.R., 1976. Foundations of Fuzzy Reasoning , Int. J. Man-Machine Studies, Vol. 8: p.p 623-668

Guillemin, P., 1996. " Fuzzy Logic Applied to Motor Control", IEEE Trans. on Industry Applications, Vol. 32: p.p. 51-56



Guo, S., Peters, L., Surman, H., 1996. "Desing and Application of an Analog Fuzzy Logic Controller", IEEE Transaction on Fuzzy Systems, Vol.4, No.4: p.p. 429 - 438

Hara, K., Yokogawa, R., 1991. "Applications of Fuzzy Reasoning to Precision Inserting Operation", Fuzzy Sets and Systems, Vol. 39: p.p. 204 - 214

Haung, L.J., Tomizuka, M., 1990. A Self-Paced Fuzzy Tracking Controller for Two-Dimensional Motion Control, IEEE Trans. Sys., Man, Cybern., Vol.20: p.p. 1115-1124

Hesburg, T., Tomizuka, M., 1994. "Fuzzy Logic Control for Lateral Vehicle Guidance", IEEE Control Systems, p.p. 55 - 62

Hiloowala, R. M., Sharaf A. M., 1996. "A Rule Based Fuzzy Logic Controller for a PWM Inverter in a Stand Alone Wind Energy Conversion Scheme", IEEE Trans. on Industry Applications", Vol. 32, No.1: p.p. 57-65

Hollstein, T., Halgamuge, S. K., Glesner, M., 1996. "Computer - Aided Desing of Fuzzy Systems Based on Generic VHDL Specifications", IEEE Transaction on Fuzzy Systems, Vol.4, No.4: p.p. 403 - 417

Huang, T. C., Fellow, M.A.S., 1996. "High Performance Speed and Position Tracking of Induction Motors Using Multi-Layer Fuzzy Control", IEEE Trans. on Energy Conversion, Vol.11, No.2: p.p. 353 - 358



Huertas, J. L., Solano, S. S., Baturone, I., Barriga, A., 1996."Integrated Circuit Implementation of Fuzzy Controllers", IEEE Journal of Solid - State Circuits, Vol. 31, No.7: p.p. 1051 - 1058

Intel, 1988a. Embedded Control Applications

Intel, 1990. 8XC196KC User's Guide

Intel, 1991. 8XC196KC Advance Information

Intel, 1992. 8XC196KC/8XC196KD User's Manual

Intel, 1998b. MCS96 Macro Assembler User's Guide for DOS Systems

Kaufmann, A., 1975. Introduction to the Theory of Fuzzy Subsets, Academic Press ,Vol. I.

Kickert, W.J.M., 1976. Applications of a Fuzzy Controller to a warm Plant, Automatica, Vol. 12:p.p. 301-308

King, P.J., and Mamdani, E.H., 1977. The Application of Fuzzy Control System to Industrial Processes, Automatica, Vol.13: p.p.235-242.

Lai, M. F., Nakano M., Hsieh G.C., 1996. "Application of Fuzzy Logic in the Phase-Locked Loop Speed Control of Induction Motor Drive", IEEE Trans. on Industrial Electronics, Vol. 43, No. 6: p.p. 630 - 639

Lee, C.C., 1990a. Fuzzy Logic in Control Systems: Fuzzy Logic Controller, Part I, IEEE Trans. Sys., Man, Cybern., Vol. 20: p.p.404-418

Lee, C.C., 1990b. Fuzzy Logic in Control Systems: Fuzzy Logic Controller, Part II, IEEE Trans. Sys., Man, Cybern., Vol. 20: p.p.419-435

Lee, S., Bien, Z., 1994."Desing of Expandable Fuzzy Inference Processor ", IEEE Trans. on Consumer Electronics, Vol. 40, No.2: p.p. 171 - 175

Leung, K.S., and Lam, W., 1988. Fuzzy Concepts in Expert Systems, IEEE Computer, September:p.p. 43-56

Li, Y.F., and Lau, C.C., 1989. Development of Fuzzy Algorithm for Servo Systems, IEEE Control System Magazine, April 1989: p.p. 65-72

Liaw, C. M., Cheng, S. Y., 1995. " Fuzzy Two-Degrees-of-freedom Speed Controller for Motor Drivers", IEEE Trans. on Industrial Electronics, Vol. 42, No. 2: p.p. 209-216

Liaw, C.M., and Wang, J.B., 1991. Design and Implementation of a Fuzzy Controller for a High Performance Induction Motor Drive, IEEE Trans. Sys., Man, Cybern., Vol.21: p.p.921-929.

Lu, Y. S., Chen, J. S., 1994. "A Self-Organizing Fuzzy Sliding-Mode Controller Design for a Class of Nonlinear Servo Systems ", IEEE Trans. on Industrial Electronics, Vol. 41, No. 5: p.p. 492-502

Maeda, M., Maeda, Y., Murakami, S., 1991. "Fuzzy Drive Control of an Autonomous Mobile Robot", Fuzzy Sets and Systems, Vol. 39: p.p. 195-204

Maiers, J., and Sherif, Y.J., 1985. Applications of Fuzzy Set Theory, IEEE Trans. Sys., Man, Cybern., Vol. SMC-15: p.p 175-189

Mamdani, E.H., 1974. Applications of Fuzzy Algorithms for simple dynamic plant, Proc. IEEE, Vol. 121: p.p. 1585-1588

Mamdani, E.H., 1976. Advances in the Linguistic Synthesis of Fuzzy Controller, Int. J. Man. Machine Studies, Vol. 8:p.p. 669-678

Mamdani, E.H., and Assilian, S., 1975. An experiment in Linguistic synthesis with a Fuzzy Logic Controller, Int. J. Man. Mach. Studies, Vol. 7: p.p 1-13

Manaresi, N., Rovatti, R., Franchi, E., Guerrieri, R., Baccarani, G., 1996."A Silicon Compiler of Analog Fuzzy Controllers: From Behavioral Specifications to Layout", IEEE Trans. on Fuzzy Systems, Vol. 4., No.4: p.p. 418 - 428

Miyomato, S., and Nakayama, 1986. Fuzzy Information Retrieval Based on a Fuzzy Pseudthesaurus, IEEE Trans. Sys., Man, Cybern., Vol.SMC-2: p.p.278

Nakamura, K., Sakashita, N., Nitta, Y., Shimomura, K., Tokuda, T., 1993." Fuzzy Inference and Fuzzy Inference Processor", IEEE Micro, p.p. 37 - 48

NeuraLogix , 1992. "NLX230", Information Sheet

Omron, 1991a. Fuzzy Inference Board With FP 3000 Processor, Cat.No.ZC011

Omron, 1991b. Compact Digital Fuzzy Processor for Control, Cat.No.ZC012

OrCAD, 1989a. OrCAD Schematic Design Tools Users Guide.

OrCAD, 1989b. OrCAD PCB II User's Guide

Pal, S.K., and Majumder, D.D., 1986. Fuzzy Mathematical Approach to Pattern Recognition, Wiley, Halsted Press, Wiley.

Pedrycz, W., 1986. Fuzzy Control and Fuzzy Systems, Research Studies Press Ltd.

Reay, D.S., Mirkamezi-Moud M., Green, T. C., Williams, B. W., 1995."Switched Reluctance Motor Control Via Fuzzy Adaptive Systems", IEEE Control Systems, p.p. 8- 14

Ross, T.J., 1995. Fuzzy Logic with Engineering Applications, McGraw Hill, Inc.

Russo, G.V., Petta, C., Lopresti, D., Randazzo, N., 1996."Silicon Drift Detector Readout and On-Line Data Reduction Using a Fast VLSI Dedicated Fuzzy Processor ", Information Sciences, Vol. 95: p.p. 233 - 260

Rutherford, D.A., and Bloore, G.L., 1976. The Implementation of Fuzzy Algorithm for Control, Proc. IEEE, Vol.64:p.p. 572-573

SGS-Thomson Microelectronics, 1994."Weight Associate Rule Processor-W.A.R.P.1.1", Advanced Data

SGS-Thomson Mikroelectronics, 1996."AFE310 - AFE303", Information Sheet

Shenoi, S., Ashenoyi, K., Timmerman, M., 1995. "Implementation of a Learning Fuzzy Controller", IEEE Control Systems, p.p. 73 - 80

So, W. C., Tse, C. K., Lee, Y. S., 1996. "Development of a Fuzzy Logic Controller for DC/DC Converters: Desing, Computer Simulation, and Experimental Evaluation.", IEEE Trans. on Power Electronics, Vol. 11, No. 1: p.p. 24 - 31

Sousa ,G.C.D., Bose B.K., 1994. "A Fuzzy Set Theory Based Control of a Phase-Controlled Converter DC Machine Drive", IEEE Trans. on Industry Applications, Vol. 30: No.1, p.p. 34-44

Sousa, G. C. D., Bose, B. K., Cleland, J. G., 1995. " Fuzzy Logic Based On-Line Efficiency Optimization Controlof an Indirect Vector-Controlled Induction Motor Drive", IEEE Trans. on Industrial Electronics, Vol. 42, No. 2: p.p. 192 - 198

Stenz, R., Kuhn, U., 1995,"Automation of a Batch Distillation Column Using Fuzzy and Conventional Control", IEEE Trans. on Control Systems Technology, Vol.3, No.2: p.p. 171 - 176



Sugeno, M., 1985. A Introductory Survey of Fuzzy Control, Information Sciences, 36: p.p. 59-83

Sugeno, M., Murakami, K., 1985. An Experimental Study on Fuzzy Parking Control using a Model Car, Industrial Applications of Fuzzy Control, M. Sugeno, Ed. Amsterdam; North Holland: p.p. 125-138

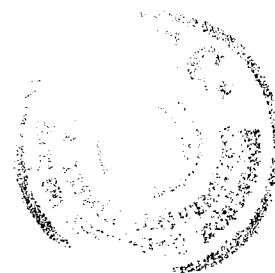
Surman, H., 1996. "Genetic Optimization of a Fuzzy System for Charging Batteries", IEEE Trans. on Industrial Electronics, Vol. 43, No. 5: p.p. 541-548

Suyitno, A., Fujikawa, J., Kobayashi, H., Yasuhiko, D., 1993. "Variable - Structured Robust Controller by Fuzzy Logic for servo motors", IEEE Trans. on Industrial Electronics, Vol. 40, No. 1: p.p. 80 - 88

Teeter, J. T., Chow, M. Y., Brickley J. J., 1996. " A Novel Fuzzy Friction Compensation Approach to Improve the Performance of a DC Motor Control System", IEEE Trans. on Industrial Electronics, Vol. 43, No. 1: p.p. 113 - 120

Togai, M., and Watanebe, H., 1986. Expert System on a Chip: An Engine for real-time approximate reasoning, IEEE Expert Syst. Mag., Vol 1: p.p. 55-62

Tong, R.M., 1977. A Control Engineering Review of Fuzzy System, Automatica, Vol. 13:p.p. 550-569



Tong, R.M., and Bonissone, P.P., 1980. A Linguistic Approach to Decision Making with Fuzzy Set, IEEE Trans. Sys., Man, Cybern., Vol. 10: p.p. 716-723

Tong, R.M., and Shapiro, D.G., 1985. Experimental Investigations of Uncertainty in a Rule-Based System for Information Retrieval, Int. J. Man. Machine Studies, Vol. 22:p.p. 265-282

Watson, S.R., Weisis, J.J., and Donell, M.L., 1979. Fuzzy Decision Analysis, IEEE Trans. Sys., Man, Cybern., Vol. SMC-9:p.p. 1-9

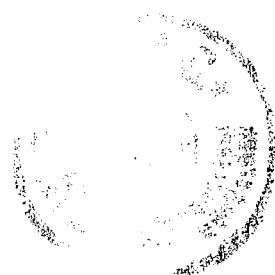
Whalen, T., and Schott, B., 1985. Alternative Logic for Approximate Reasoning in Expert System: a Comparative Study, Int. J. Man. Machine Studies, Vol. 22:p.p. 327-346

Wu, J. C., Liu, T.S., 1996."A Sliding - Mode Approach to Fuzzy Control Design", IEEE Trans. on Control Systems Technology, Vol.4, No.2: p.p. 141 - 150

Wu, Z. Q., 1990. "The Applications of Fuzzy Control Theory to an Oil-Fueled Annealing Furnace", Fuzzy Sets and Systems, Vol. 36: p.p. 145 - 156

Xie, W.X., 1984. An Information Measure for Fuzzy Sets, IEEE Trans. Sys., Man, Cybern., Vol. SMC-3

Yager, R., 1977. Multiple Objective Decision Making Using Fuzzy Set, Int - J. Man - Machine Studies. Vol. 9:p.p. 375-385.



Yamakawa, T., 1988. Fuzzy Microprocessors - Rule Chip and Defuzzifier Chip, Int. Workshop on Fuzzy System Applications, Lizuka, Japan, Aug. 1988: p.p. 51-52

Yamakawa, T., 1993. A fuzzy Inference Engine in nonlinear Analog Mode and Its Applications to a Fuzzy Logic Control, IEEE Trans. On Neural Networks. Vol.4:p.p. 496-522.

Yen, Y.J., Wang, F. J., Chen, Y. Y., 1993. "A Fuzzy Scheduling Controller for a Computer Disk File Track-Following Servo", IEEE Trans. on Industrial Electronics, Vol. 40, No. 2: p.p. 266-272

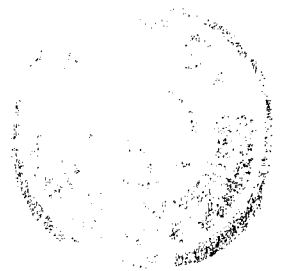
Zadeh, L.A., 1965. Fuzzy Sets , Information and Control, Vol. 8: p.p 338-353

Zadeh, L.A., 1973. Outline of A New Approach to the Analysis of Complex Systems and Decision Processes, IEEE Trans. Syst., Man, Cybern, Vol. SMC-3:p.p.28-44

Zadeh, L.A., 1978a. Fuzzy Sets Basis for a theory of Possibility, Fuzzy Sets and System, Vol 1:p.p. 3-28

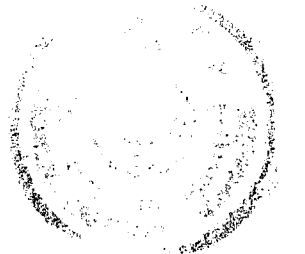
Zadeh, L.A., 1978b. A Meaning Representation Language for Natural Languages, Int. J. Man. Machine Studies, Vol. 10:p.p. 395-460

Zadeh, L.A., 1984. Making Computer Think Like People, IEEE Spectrum.



EK 1.

(ÖZEL AMAÇLI KAYDEDİCİLER)



Seri Port Alma İşlemi Kaydedicisi:

SBUF (RX) , 07H

Ypencere 0 -oku, Ypencere 15 - yaz

	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
Reset	0	0	0	0	0	0	0	0
Açıklama	Seri porttan alınan en son değeri içerir..							

Seri Port Gönderme İşlemi Kaydedicisi:

SBUF (TX) , 07H

Ypencere 0 - oku, Ypencere 15 - yaz

	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
Reset	0	0	0	0	0	0	0	0
Açıklama	Seri porttan gönderilecek veri yazılır							

Seri Port Denetleme Kaydedicisi:

SP_CON , 11H

Ypencere 0 -yaz, Ypencere 15 - oku

	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
Reset	0	0	0	0	0	0	0	0
Adı	-	-	-	TB8	REN	PEN	SER_MODE	
Fonks.	-	-	-	Gönderme işleminde 9. veri biti	Alma işlemine izin	Çift Eşitlik işlemine izin	Mod seçme işlemi	
Açıklama	Bu bitler daima sıfırdır ve saklı durumdadırlar.			Mod 2 ve 3 'de 9. veri biti. Gönderme işleminden sonra sıfırlanır. SP_CON.2=1 ise gönderme işleminde çift eşitlik biti olur.	1: P2.1 / RXD RXD 'ye izin . (Mod 1,2,3) 0: Alma işlemi durur. (Mod 1,2,3) 1: Alma işlemi başlar.(Mod 0) 0: Gönderme işlemi başlar. (Mod 0)	1: Gönderme işleminde TB8 çift eşitlik değeri, alma işleminde SP_STAT.7 biti Alma işlemi eşitlik hata biti olur.(Mod 1, 3) 0: Mod 2	0 0 : mod 0 0 1 : mod 1 1 0 : mod 2 1 1 : mod 3	

Seri Port Durum Kaydedicisi

SP_STAT, 11H

Ypencere 0 - oku, Ypencere 15 - yaz

	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
Reset	0	0	0	1	0	0	1	1
Adı	RPE / RB8	RI	TI	FE	TXE	OE	-	-
Fonks	Alma işlemi Eşitlik Hatası veya alınan bit 8	Alma isteği	Gönderme isteği	Çerçeve hatası	SBUF(TX) boş	Üzerine yazma hatası	-	-
Açıklama	SP_CON.2= 1 ise alınan eşitlik hatası SP_CON.2= 0 ise 9. veri biti	1:Alma işlemi tamam 0:Alma işlemi tamam değil	1:Durma biti gönderilmeye başladı (Gönderme işlemi tamam) 0: Gönderme işlemi tamamlanmadı	1: Alınan bilgide durma biti yok 0: Durma biti var	1:Gönderme kaydedicisi boş 0: Kaydedici dolu	1: Alınan veri okunmadan üzerine bilgi yazıldı 0: Yazılma yok	Bu bitler daima sıfırdır ve saklı durumda	

Karakter Hızı Kaydedicisi

BAUD_RATE, 0EH

Tüm pencerelerde oku, yaz

Kaydedicinin bulunduğu adrese ilk yazmadaki durum:

	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
Reset	0	x	0	0	0	0	0	0
Açıklama.	Karakter Hızı kelime değerinin en az ağırlıklı baytı belirler							

Kaydedicinin bulunduğu adrese ikinci yazmadaki durum:

	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
Reset	0	x	0	0	0	0	0	0
Açıklama	1:XTAL1 Saat darbeleri içeriden - f_{osc} 0:T2CLK Saat darbeleri dışarıdan							

Karakter Hızı kelime değerinin en fazla ağırlıklı 7 bitini belirler

Giriş / Çıkış Denetleme Kaydedicisi 1

IOC1, 16H

Ypencere 0 - yaz, Ypencere 15 - oku

	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
Reset	0	0	1	0	0	0	0	1
Adı	HSI_ INT	HSO5_ ENA	TXD_ SEL	HSO4_ ENA	T2OVF_ INT	T1OVF_ INT	EXTINT_ SRC	PWM_ SEL
Fonk	HSI kesme işlemi kaynağı	HSO.5/ HSL3	P2.0/ TXD	HSO.4 / HSL2	Zamanl.2 taşma kesme isteğine izin	Zamanl.1 taşma kesme isteğine izin	Dış kesme isteği kaynak	P2.5/ PWM çıkış
Açık lama	1: HSI FIFO doldu 0: HSI dan bilgi okuna- bilir. INT 02, 2004H ve INT_ MASK.2 =1 ise kesme isteğine izin verilir.	1:HSO.5 0:HSL3	1:TXD 0:P 2.0	1:HSO.4 0:HSL2	1:Taşma kesme isteği kaynağı olarak Zamanl.2'ye izin var 0:Taşma kesme isteği kaynağı olarak Zamanl.2'ye izin yok INT 00 2000H ve INT_MASK. 0=1 ise kesme isteğine izin verilir IOS1.4 kesme isteği durumu	1:Taşma kesme isteği kaynağı olarak Zamanl.1'e izin var 0:Taşma kesme isteği kaynağı olarak Zamanl.1'e izin yok INT 00 2000H ve INT_MASK. 0=1 ise kesme isteğine izin verilir IOS1.4 kesme isteği durumu	1: P0.7 0: P2.2 INT_ MASK.7= 1 ise kesme isteğine izin var. INT 07 , 200EH	0: P2.5 1:PWM

Pencere Seçme Kaydedicisi

WSR, 14H

Tüm pencerelerde oku, yaz

	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
Reset	0	X	X	X	0	0	0	0
Adı	HLDEN	W6	W5	W4	W3	W2	W1	W0
Fonks.	HOLD# /HLDA# işlemine izin	Pencere seçme	Pencere seçme	Pencere seçme	Pencere seçme bitleri			
Açıkla ma	1: İzin var 0: İzin yok	1: Düşey pencere 32 Bayt 0:Etkisiz	1: Düşey pencere 64 Bayt 0: Etkisiz	1: Düşey pencere 128 Bayt 0: Etkisiz	W4, W5 ve W6 bitleri lojik 0 ise seçilen düşey pencerede bu bitlere yazılacak sayının değerinde yatay pencereler seçilir. Eğer W4, W5 ve W6 bitlerinin herhangi biri lojik 1 ise bu bitler düşey pencereyi ve bu düşey pencerenin boyutunu seçer.			

Giriş / Çıkış Denetleme Kaydedicisi 0

IOC0, 15H

Ypencere 0 - yaz, Ypencere 15 - oku

	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
Reset	0	0	0	0	0	0	X	0
Adı	T2CLK_ SRC	HSI3_ ENA	T2RST_ SRC	HSI2_ ENA	T2RST_ ENA	HSI1_ ENA	SW_ T2RST	HSI0_ ENA
Fonk	Zamanl2 için saat darb. kaynağı	HSI girişi olarak HSL3 'e izin	Zamanl2 için dış sıfırlama. kaynağı	HSI girişi olarak HSL2 'e izin	Zamanl2 için dış sıfırlama- ya izini	HSI girişi olarak HSL1 'e izin	Zamanl2 için yazılımda sıfırlama	HSI girişi olarak HSL0 'e izin
Açıkla ma	1: HSL1 0: T2CLK ve IOC0.3=0 olmalı.	FIFO'ya sayıcı sonucunu yüklemek için tetikleme girişi 1: HSL3'e izin var 0: HSL3'e izin yok	1: HSL0 bacağı çıkan kenar 0: T2RST bacağı çıkan kenar IOC0.3=1 olmalı.	FIFO'ya sayıcı sonucunu yüklemek için tetikleme girişi 1: HSL2'e izin var 0: HSL2'e izin yok	1: İzin var 0: İzin yok	FIFO'ya sayıcı sonucunu yüklemek için tetikleme girişi 1: HSL1'e izin var 0: HSL1'e izin yok	1: zaman2 sıfırlanır 0: zaman2 sıfırlama işlemi olmaz	FIFO'ya sayıcı sonucunu yüklemek için tetikleme girişi 1: HSL0'e izin var 0: HSL0'e izin yok

Giriş / Çıkış Denetleme Kaydedicisi 2

IOC2, 0BH

Ypencere 0 - yaz, Ypencere 15 - oku

	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
Reset	0	0	0	0	0	0	X	0
Adı	CAM_CLR	LOCK_ENA	T2ALT_INT	AD_FAST	AD_TIME_ENA	SLOW_PWM	T2UD_ENA	FAST_T2_ENA
Fonk	Tüm CAM girişlerini sıfırlama	CAM girişlerini kilitleme-ye izin	Zamanl2 için taşma sınırlarını seçme	A/D için saat darbesi ayarına izin	AD_TIME kaydedici-ne izin	PWM için saat darbesi ayarına izin	Zamanl. 2 için ileri/geri saymaya izin	Zamanl. 2 için hızlı saymaya izin
Açıklama	1:sıfırlama var 0:sıfırlama yok	1:İzin var 0:İzin yok	1:7FFFH/8000H 0: FFFH/0000H Zamanl2 taşma kesme isteği için sınır seçer INT_MASK1.4 =1izin var INT 12 2038H	1:Normal mod Çevirim 89.5 durum zamanı 0:Hızlı mod Çevirim zamanı 156.5 durum zamanı	1:AD_TIME çevirim zamanını belirler. 0: ise 80C196KB uyumlu mod	1:İzin var PWM çıkış periyodu 512 durum zamanı 1:İzin yok PWM çıkış periyodu 256 durum zamanı	1: ise P2.6= 1 geri sayma P2.6=0 ileri sayma 0:ise Sürekli ileri sayma	1: Hızlı saymaya izin. Her darbeye bir sayma *HSO'da kullanm 0: Normal saymaya izin. Her sekiz darbeye bir sayma

Kesme İsteği Önleme Kaydedicisi 1

INT_MASK1, 13H

Tüm pencerelerde yaz, oku

	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
Reset	0	0	0	0	0	0	0	0
Adı	NMI_MASK	FIFO_MASK	EXTINT1_MASK	T2OVF_MASK	T2CAP_MASK	HSI4_MASK	RI_MASK	TI_MASK
Fonk	NMI	HSI FIFO dolu	EXTINT bacağı	Zamanl2 taşma	Zamanl2 tutma	HSI FIFO4	Alma işlemi	Gönderme
Açıklama	X: Her iki durumda NMI'ya izin var. INT 15, 203EH	1: İzin var INT 14, 203CH	1: İzin var. P2.2 bacağı EXTINT-(P0.7) veya EXTINT1 üretebilir. INT 13 203AH	1: İzin var. IOC2.5 taşma sınırını seçer. INT 12, 2038H	1: İzin var. INT 11, 2036H	1: HSI FIFO4 kesme isteğine izin var. INT 10, 2034H	1: İzin var ve INT_MASK1.0 =1, INT_MASK.6 =0 olmalıdır. INT 09, 2032H	1: İzin var ve INT_MASK1.1 =1, INT_MASK.6 =0 olmalıdır. INT 08, 2030H

Giriş / Çıkış Durum Kaydedicisi 1

IOS1, 16H

Ypencere0 - oku, Ypencere15 - yaz

	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
Reset	0	0	0	0	0	0	0	0
Adı	HSI_RDY	FIFO_FULL	T1_OVF	T2_OVF	SWTF3	SWTF2	SWTF1	SWTF0
Fonk	HSI tutma kaydedici veri hazır	FIFO altı sonuç girişi yapıldı	Zamanl1 taşma bayrağı	Zamanl2 taşma bayrağı	Yazılım Zamanl3 bayrağı	Yazılım Zamanl2 bayrağı	Yazılım Zamanl1 bayrağı	Yazılım Zamanl0 bayrağı
Açıklama	1: Veri hazır 0: Hazır değil Kesme isteği oluşturur	1: FIFO'ya altı veya daha fazla sonuç girildi. Kesme isteği oluşturur	1: Taşma kesme isteğini zamanl1 oluşturdu. INT_MASK.0 = 1 olmalı INT 00, 2000H	1: Taşma kesme isteğini zamanl2 oluşturdu. INT_MASK.0 = 1 olmalı INT 00, 2000H	Bu bitlerden herhangi biri lojik 1 ise ilgili zamanlayıcının sona ulaştığını gösterir ve INT 05, 200AH kesme isteğini oluştururlar.			

Ypencere 0 - oku, Ypencere 15 - bit 0,2,4,6 yaz

Ypencere 0 -oku, Ypencere 15 - yaz

Ypencere 0 -oku, Ypencere 15 - yaz

[illegible]

Kesinti Önleme Kaydedicisi

INT_MASK, 08H

Tüm pencerelerde yaz, oku

	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
Reset	0	0	0	0	0	0	0	0
Adı	EXTINT_MASK	SER_MASK	SWT_MASK	HSIO_MASK	HSO_MASK	HSIDAT_MASK	AD_MASK	TIMER_MASK
Fonk	EXTINT veya P0.7 kesme isteği	Seri port	Yazılım zamanl.	HSIO dış kesme isteği	HSO sonuç çıkışı	HSI veri okunabilir FIFO dolu	A/D çevrimi tamam	Zaman1/ Zaman2 taşma
Açık-lama	1:EXTINT kesme isteğine izin verilir. IOC1.1 kesme isteği kaynağını seçer. (P0.7 veya P2.2 bacağı)	1: Seri port kesme isteğine izin verilir. INT 06, 200CH INT_MASK1.0 =1, alma INT_MASK1.1 =1 gönder	1: Yazılım zamanla-yıcısı kesme isteğine izin verilir. INT 05, 200AH	1: HSI0 bacağı kesme isteğine izin verilir. INT 04, 2008H	1: HSO kesme isteğine izin verilir. INT 03, 2006H	1: HSI'da veri okunabilir kesme isteğine izin verilir. IOC1.7 kesme isteği kaynağını seçer. INT 02, 2004H	1: A/D çevrimi tamamladı kesme isteğine izin verilir. INT 01, 2002H	1: Taşma kesme isteğine izin verilir. IOC1.2=1 kaynak Zaman1 INT 00 IOC1.3=1 kaynak Zaman2 INT 00 / INT 12

Tümdevre Ayarlama Kaydedicisi:

Bit	Bit adı	Fonksiyonu	Reset	Tanımlama															
0	PD	Çekilen güçü azaltmak için izin	1	Mikrodenetleyicinin harcadığı güçü azaltmak için kullanılır. Bu bit lojik 1 ise güç azaltma moduna izin verilir. Eğer lojik 0 ise izin verilmez.															
1	BW0	Yol genişliğini denetleme	1	Dinamik veya 8 bitlik yol genişliğini seçer. 1=Yol genişliği BUSWIDTH bacağındaki konum ile denetlenir BUSWIDTH=1, yol 16 bitlik BUSWIDTH=0, yol 8 bitlik 0= Yol genişliği 8 bitlik olur ve BUSWIDTH bacağı yol genişliğini denetleyemez.															
2	WR	Yazma- Tutma modunu seçme	1	1= Geçerli adres ve standart yol modunda WR# ve BHE# sinyalleri üretilir. 0= Yazma - Tutma ve Yazma - Tutma ile beraber geçerli adres modunda WRL# ve WRH# sinyalleri üretilir.															
3	ALE	Geçerli adres tutma modunu denetleme	1	1= Yazma - tutma ve Standart yol modunda geçerli adresi tutmak için ALE sinyali üretilir. 0= ALE sinyali yerine ADV# sinyali üretilir ve dış belleği kolay yoldan seçmek için kullanılır.															
4	IORC0- IORC1	İç hazır olma durumunu denetleme	10	Yavaş dış üniteler ile uyumlu çalışmak için bekleme zamanını belirler. <table><tr><th>IRC1</th><th>IRC0</th><th>Maks. Bekleme Zamanı</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>2</td></tr><tr><td>1</td><td>0</td><td>3</td></tr><tr><td>1</td><td>1</td><td>READY bacağı ile denetlenir.</td></tr></table>	IRC1	IRC0	Maks. Bekleme Zamanı	0	0	1	0	1	2	1	0	3	1	1	READY bacağı ile denetlenir.
IRC1	IRC0	Maks. Bekleme Zamanı																	
0	0	1																	
0	1	2																	
1	0	3																	
1	1	READY bacağı ile denetlenir.																	
5	LOC0- LOC1	Kilitleme bitleri	00	İç program belleğinin korunmasını sağlar. <table><tr><th>LOC1</th><th>LOC0</th><th>koruma</th></tr><tr><td>0</td><td>0</td><td>okuma ve yazmayı önlemek için</td></tr><tr><td>0</td><td>1</td><td>okumayı önlemek için</td></tr><tr><td>1</td><td>0</td><td>yazmayı önlemek için</td></tr><tr><td>1</td><td>1</td><td>koruma yoktur.</td></tr></table>	LOC1	LOC0	koruma	0	0	okuma ve yazmayı önlemek için	0	1	okumayı önlemek için	1	0	yazmayı önlemek için	1	1	koruma yoktur.
LOC1	LOC0	koruma																	
0	0	okuma ve yazmayı önlemek için																	
0	1	okumayı önlemek için																	
1	0	yazmayı önlemek için																	
1	1	koruma yoktur.																	

Bilgisayardaki Hat Denetleme Kaydedicisi (3FBH)

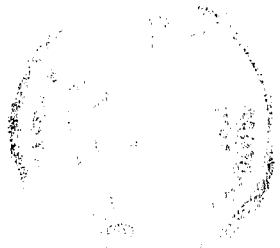
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
Adı	DLAB	-	-	EPS	PEN	STB	WLS1	WLS0
Açık- lama	Erişim bitini tutma	Kırma işlemi aktif	Eşitlik biti ilave etme	Çift eşitlik bitini seçme 0: tek 1: çift	Eşitlik işlemine izin 0: izin yok 1: izin var	Durma biti sayısı 0: 1 bit 1: 2 bit	Veri Uzunluğunu seçme 0 0 5 bit 0 1 6 bit 1 0 7 bit 1 1 8 bit	

Bilgisayardaki Hat Durum Kaydedicisi (3FDH)

	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
Adı	-	TSRE	THRE	BI	FE	PE	OE	DR
Açık- lama	Sürekli lojik 0	Verici ötelemeli kaydedici boşmu? 1: Boş 0: Dolu	Verici tutma kaydedici si boşmu? 1: Boş 0: Dolu	Kırma kesinti durumu	Çerçeve hatası	Eşitlik hatası	Üzerine yazma hatası	Bilgi hazır 1: Hazır 0: Değil

EK 2.

(KOMUTLAR)



Mnemonic	Operands	Operation (Note 1)	Flags						Notes
			Z	N	C	V	VT	ST	
ADD/ADDB	2	$D \leftarrow D + A$	✓	✓	✓	✓	↑	—	
ADD/ADDB	3	$D \leftarrow B + A$	✓	✓	✓	✓	↑	—	
ADDC/ADDCB	2	$D \leftarrow D + A + C$	↓	✓	✓	✓	↑	—	
SUB/SUBB	2	$D \leftarrow D - A$	✓	✓	✓	✓	↑	—	
SUB/SUBB	3	$D \leftarrow B - A$	✓	✓	✓	✓	↑	—	
SUBC/SUBCB	2	$D \leftarrow D - A + C - 1$	↓	✓	✓	✓	↑	—	
CMP/CMPB	2	$D - A$	✓	✓	✓	✓	↑	—	
MUL/MULU	2	$D, D + 2 \leftarrow D \times A$	—	—	—	—	—	—	2
MUL/MULU	3	$D, D + 2 \leftarrow B \times A$	—	—	—	—	—	—	2
MULB/MULUB	2	$D, D + 1 \leftarrow D \times A$	—	—	—	—	—	—	3
MULB/MULUB	3	$D, D + 1 \leftarrow B \times A$	—	—	—	—	—	—	3
DIVU	2	$D \leftarrow (D, D + 2) / A, D + 2 \leftarrow \text{remainder}$	—	—	—	✓	↑	—	2
DIVUB	2	$D \leftarrow (D, D + 1) / A, D + 1 \leftarrow \text{remainder}$	—	—	—	✓	↑	—	3
DIV	2	$D \leftarrow (D, D + 2) / A, D + 2 \leftarrow \text{remainder}$	—	—	—	✓	↑	—	
DIVB	2	$D \leftarrow (D, D + 1) / A, D + 1 \leftarrow \text{remainder}$	—	—	—	✓	↑	—	
AND/ANDB	2	$D \leftarrow D \text{ AND } A$	✓	✓	0	0	—	—	
AND/ANDB	3	$D \leftarrow B \text{ AND } A$	✓	✓	0	0	—	—	
OR/ORB	2	$D \leftarrow D \text{ OR } A$	✓	✓	0	0	—	—	
XOR/XORB	2	$D \leftarrow D \text{ (excl. or) } A$	✓	✓	0	0	—	—	
LD/LDB	2	$D \leftarrow A$	—	—	—	—	—	—	
ST/STB	2	$A \leftarrow D$	—	—	—	—	—	—	
XCH/XCHB	2	$D \leftarrow A, A \leftarrow D$	—	—	—	—	—	—	
LDBSE	2	$D \leftarrow A; D + 1 \leftarrow \text{SIGN}(A)$	—	—	—	—	—	—	3,4
LDBZE	2	$D \leftarrow A; D + 1 \leftarrow 0$	—	—	—	—	—	—	3,4
PUSH	1	$SP \leftarrow SP - 2; (SP) \leftarrow A$	—	—	—	—	—	—	
POP	1	$A \leftarrow (SP); SP + 2$	—	—	—	—	—	—	
PUSHF	0	$SP \leftarrow SP - 2; (SP) \leftarrow \text{PSW};$ $\text{PSW} \leftarrow 0000\text{H}; I \leftarrow 0$	0	0	0	0	0	0	
POPF	0	$\text{PSW} \leftarrow (SP); SP \leftarrow SP + 2; I \leftarrow \text{✓}$	✓	✓	✓	✓	✓	✓	
SJMP	1	$PC \leftarrow PC + 11\text{-bit offset}$	—	—	—	—	—	—	5
LJMP	1	$PC \leftarrow PC + 16\text{-bit offset}$	—	—	—	—	—	—	5
BR[indirect]	1	$PC \leftarrow (A)$	—	—	—	—	—	—	
TIJMP	3	$PC \leftarrow [A] + 2 * ([B] \text{ AND } C)$	—	—	—	—	—	—	
SCALL	1	$SP \leftarrow SP - 2;$ $(SP) \leftarrow PC; PC \leftarrow PC + 11\text{-bit offset}$	—	—	—	—	—	—	5
LCALL	1	$SP \leftarrow SP - 2; (SP) \leftarrow PC;$ $PC \leftarrow PC + 16\text{-bit offset}$	—	—	—	—	—	—	5

Mnemonic	Operands	Operation (Note 1)	Flags						Notes
			Z	N	C	V	VT	ST	
RET	0	$PC \leftarrow (SP); SP \leftarrow SP + 2$	-	-	-	-	-	-	
J (conditional)	1	$PC \leftarrow PC + 8\text{-bit offset (if taken)}$	-	-	-	-	-	-	5
JC	1	Jump if C = 1	-	-	-	-	-	-	5
JNC	1	jump if C = 0	-	-	-	-	-	-	5
JE	1	jump if Z = 1	-	-	-	-	-	-	5
JNE	1	Jump if Z = 0	-	-	-	-	-	-	5
JGE	1	Jump if N = 0	-	-	-	-	-	-	5
JLT	1	Jump if N = 1	-	-	-	-	-	-	5
JGT	1	Jump if N = 0 and Z = 0	-	-	-	-	-	-	5
JLE	1	Jump if N = 1 or Z = 1	-	-	-	-	-	-	5
JH	1	Jump if C = 1 and Z = 0	-	-	-	-	-	-	5
JNH	1	Jump if C = 0 or Z = 1	-	-	-	-	-	-	5
JV	1	Jump if V = 0	-	-	-	-	-	-	5
JNV	1	Jump if V = 1	-	-	-	-	-	-	5
JVT	1	Jump if VT = 1; Clear VT	-	-	-	-	0	-	5
JNVT	1	Jump if VT = 0; Clear VT	-	-	-	-	0	-	5
JST	1	Jump if ST = 1	-	-	-	-	-	-	5
JNST	1	Jump if ST = 0	-	-	-	-	-	-	5
JBS	3	Jump if Specified Bit = 1	-	-	-	-	-	-	5,6
JBC	3	Jump if Specified Bit = 0	-	-	-	-	-	-	5,6
DJNZ/ DJNZW	1	$D \leftarrow D - 1;$ If $D \neq 0$ then $PC \leftarrow PC + 8\text{-bit offset}$	-	-	-	-	-	-	5
DEC/DECB	1	$D \leftarrow D - 1$	✓	✓	✓	✓	↑	-	
NEG/NEGB	1	$D \leftarrow 0 - D$	✓	✓	✓	✓	↑	-	
INC/INCB	1	$D \leftarrow D + 1$	✓	✓	✓	✓	↑	-	
EXT	1	$D \leftarrow D; D + 2 \leftarrow \text{Sign}(D)$	✓	✓	0	0	-	-	2
EXTB	1	$D \leftarrow D; D + 1 \leftarrow \text{Sign}(D)$	✓	✓	0	0	-	-	3
NOT/NOTB	1	$D \leftarrow \text{Logical Not}(D)$	✓	✓	0	0	-	-	
CLR/CLRB	1	$D \leftarrow 0$	1	0	0	0	-	-	
SHL/SHLB/SHLL	2	$C \leftarrow \text{msb} \dots \text{lsb} \leftarrow 0$	✓	✓	✓	✓	↑	-	7
SHR/SHRB/SHRL	2	$0 \rightarrow \text{msb} \dots \text{lsb} \rightarrow C$	✓	✓	✓	0	-	✓	7
SHRA/SHRAB/SHRAL	2	$\text{msb} \rightarrow \text{msb} \dots \text{lsb} \rightarrow C$	✓	✓	✓	0	-	✓	7
SETC	0	$C \leftarrow 1$	-	-	1	-	-	-	
CLRC	0	$C \leftarrow 0$	-	-	0	-	-	-	

Mnemonic	Operands	Operation (Note 1)	Flags						Notes
			Z	N	C	V	VT	ST	
CLRV _T	0	$VT \leftarrow 0$	–	–	–	–	0	–	
RST	0	$PC \leftarrow 2080H$	0	0	0	0	0	0	8
DI	0	Disable All Interrupts ($I \leftarrow 0$)	–	–	–	–	–	–	
EI	0	Enable All Interrupts ($I \leftarrow 1$)	–	–	–	–	–	–	
DPTS	0	Disable all PTS Cycles ($PSE = 0$)	–	–	–	–	–	–	
EPTS	0	Enable all PTS Cycles ($PSE = 1$)	–	–	–	–	–	–	
NOP	0	$PC \leftarrow PC + 1$	–	–	–	–	–	–	
SKIP	0	$PC \leftarrow PC + 2$	–	–	–	–	–	–	
NORML	2	Left shift till msb = 1; $D \leftarrow$ shift count	✓	✓	0	–	–	–	7
TRAP	0	$SP \leftarrow SP - 2$; $(SP) \leftarrow PC$; $PC \leftarrow (2010H)$	–	–	–	–	–	–	9
PUSHA	1	$SP \leftarrow SP - 2$; $(SP) \leftarrow PSW$; $PSW \leftarrow 0000H$; $SP \leftarrow SP - 2$; $(SP) \leftarrow IMASK1/WSR$; $IMASK1 \leftarrow 00H$	0	0	0	0	0	0	
POPA	1	$IMASK1/WSR \leftarrow (SP)$; $SP \leftarrow SP + 2$ $PSW \leftarrow (SP)$; $SP \leftarrow SP + 2$	✓	✓	✓	✓	✓	✓	
IDL _{PD}	1	IDLE MODE IF KEY = 1; POWERDOWN MODE IF KEY = 2; CHIP RESET OTHERWISE	–	–	–	–	–	–	
CMPL	2	D-A	✓	✓	✓	✓	↑	–	
BMOV, BMOV _i	2	$[PTR_HI] + \leftarrow [PTR_LOW] +$; UNTIL COUNT = 0	–	–	–	–	–	–	

NOTES:

1. If the mnemonic ends in "B" a byte operation is performed, otherwise a word operation is done. Operands D, B, and A must conform to the alignment rules for the required operand type. D and B are locations in the Register File; A can be located anywhere in memory.
2. D, D + 2 are consecutive WORDS in memory; D is DOUBLE-WORD aligned.
3. D, D + 1 are consecutive BYTES in memory; D is WORD aligned.
4. Changes a byte to word.
5. Offset is a 2's complement number.
6. Specified bit is one of the 2048 bits in the register file.
7. The "L" (Long) suffix indicates double-word operation.
8. Initiates a Reset by pulling $\overline{RESET}$ low. Software should re-initialize all the necessary registers with code starting at 2080H.
9. The assembler will not accept this mnemonic.

MNEMONIC	DIRECT	IMMED	INDIRECT		INDEXED	
			NORMAL*(1)	A-INC*(1)	SHORT*(1)	LONG*(1)
ADD (3-op)	4/44	5/45	4/46	4/46	5/47	6/47
SUB (3-op)	4/48	5/49	4/4A	4/4A	5/4B	6/4B
ADD (2-op)	3/64	4/65	3/66	3/66	4/67	5/67
SUB (2-op)	3/68	4/69	3/6A	3/6A	4/6B	5/6B
ADDC	3/A4	4/A5	3/A6	3/A6	4/A7	5/A7
SUBC	3/A8	4/A9	3/AA	3/AA	4/AB	5/AB
CMP	3/88	4/89	3/AB	3/AB	4/8B	5/8B
ADDB (3-op)	4/54	4/55	4/56	4/56	5/57	6/57
SUBB (3-op)	4/58	4/59	4/5A	4/5A	5/5B	6/5B
ADDB (2-op)	3/74	3/75	3/76	3/76	4/77	5/77
SUBB (2-op)	3/78	3/79	3/7A	3/7A	4/7B	5/7B
ADDCB	3/B4	3/B5	3/B6	3/B6	4/B7	5/B7
SUBCB	3/B8	3/B9	3/BA	3/BA	4/BB	5/BB
CMPB	3/98	3/99	3/9A	3/9A	4/9B	5/9B
MUL (3-op)	5/(2)	6/(2)	5/(2)	5/(2)	6/(2)	7/(2)
MULU (3-op)	4/4C	5/4D	4/4E	4/4E	5/4F	6/4F
MUL (2-op)	4/(2)	5/(2)	4/(2)	4/(2)	5/(2)	6/(2)
MULU (2-op)	3/6C	4/6D	3/6E	3/6E	4/6F	5/6F
DIV	4/(2)	5/(2)	4/(2)	4/(2)	5/(2)	6/(2)
DIVU	3/8C	4/8D	3/8E	3/8E	4/8F	5/8F
MULB (3-op)	5/(2)	5/(2)	5/(2)	5/(2)	6/(2)	7/(2)
MULUB (3-op)	4/5C	4/5D	4/5E	4/5E	5/5F	6/5F
MULB (2-op)	4/(2)	4/(2)	4/(2)	4/(2)	5/(2)	6/(2)
MULUB (2-op)	3/7C	3/7D	3/7E	3/7E	4/7F	5/7F
DIVB	4/(2)	4/(2)	4/(2)	4/(2)	5/(2)	6/(2)
DIVUB	3/9C	3/9D	3/9E	3/9E	4/9F	5/9F
AND (3-op)	4/40	5/41	4/42	4/42	5/43	6/43
AND (2-op)	3/60	4/61	3/62	3/62	4/63	5/63
OR (2-op)	3/80	4/81	3/82	3/82	4/83	5/83
XOR	3/84	4/85	3/86	3/86	4/87	5/87
ANDB (3-op)	4/50	4/51	4/52	4/52	5/53	5/53
ANDB (2-op)	3/70	3/71	3/72	3/72	4/73	4/73
ORB (2-op)	3/90	3/91	3/92	3/92	4/93	5/93
XORB	3/94	3/95	3/96	3/96	4/97	5/97
PUSH	2/C8	3/C9	2/CA	2/CA	3/CB	4/CB
POP	2/CC	—	2/CE	2/CE	3/CF	4/CF

MNEMONIC	DIRECT	IMMED	INDIRECT		INDEXED	
			NORMAL	A-INC	SHORT	LONG
LD	3/A0	4/A1	3/A2	3/A2	4/A3	5/A3
LDB	3/B0	3/B1	3/B2	3/B2	4/B3	5/B3
ST	3/C0	—	3/C2	3/C2	4/C3	5/C3
STB	3/C4	—	3/C6	3/C6	4/C7	5/C7
XCH	3/04	—	—	—	4/0B	5/0B
XCHB	3/14	—	—	—	4/1B	5/1B
LDBSE	3/BC	3/BD	3/BE	3/BE	4/BF	5/BF
LBSZE	3/AC	3/AD	3/AE	3/AE	4/AF	5/AF

Mnemonic	Length/Opcode
PUSHF	1/F2
POPF	1/F3
PUSHA	1/F4
POPA	1/F5
TRAP	1/F7
LCALL	3/EF
SCALL	2/28–2F ⁽³⁾
RET	1/F0
LJMP	3/E7
SJMP	2/20–27 ⁽³⁾
BR[]	2/E3
TIJMP	4/E2
JNST	1/D0
JST	1/D8
JNH	1/D1
JH	1/D9
JGT	1/D2
JLE	1/DA
JNC	1/B3
JC	1/D8
JNVT	1/D4
JVT	1/DC
JNV	1/D5
JV	1/DD
JGE	1/D6
JLT	1/DE
JNE	1/D7
JE	1/DF
JBC	3/30–37
JBS	3/38–3F

Mnemonic	Length/Opcode
DJNZ	3/E0
DJNZW	3/E1
NORML	3/0F
SHRL	3/0C
SHLL	3/0D
SHRAL	3/0E
SHR	3/08
SHRB	3/18
SHL	3/09
SHLB	3/19
SHRA	3/0A
SHRAB	3/1A
CLRC	1/F8
SETC	1/F9
DI	1/FA
EI	1/FB
DPTS	1/EC
EPTS	1/ED
CLRVT	1/FC
NOP	1/FD
RST	1/FF
SKIP	2/00
IDLPD	1/F6
BMOV	3/C1
BMOVi	3/CD

NOTES:

3. The 3 least significant bits of the opcode are concatenated with the 8 bits to form an 11-bit, 2's complement offset.

MNEMONIC	DIRECT	IMMED	INDIRECT		INDEXED	
			NORMAL*	A-INC*	SHORT*	LONG*
ADD (3-op)	5	6	7/10	8/11	7/10	8/11
SUB (3-op)	5	6	7/10	8/11	7/10	8/11
ADD (2-op)	4	5	6/8	7/9	6/8	7/9
SUB (2-op)	4	5	6/8	7/9	6/8	7/9
ADDC	4	5	6/8	7/9	6/8	7/9
SUBC	4	5	6/8	7/9	6/8	7/9
CMP	4	5	6/8	7/9	6/8	7/9
ADDB (3-op)	5	5	7/10	8/11	7/10	8/11
SUBB (3-op)	5	5	7/10	8/11	7/10	8/11
ADDB (2-op)	4	4	6/8	7/9	6/8	7/9
SUBB (2-op)	4	4	6/8	7/9	6/8	7/9
ADDCB	4	4	6/8	7/9	6/8	7/9
SUBCB	4	4	6/8	7/9	6/8	7/9
CMPB	4	4	6/8	7/9	6/8	7/9
MUL (3-op)	16	17	18/21	19/22	19/22	20/23
MULU (3-op)	14	15	16/19	17/19	17/20	18/21
MUL (2-op)	16	17	18/21	19/22	19/22	20/23
MULU (2-op)	14	15	16/19	17/19	17/20	18/21
DIV	26	27	28/31	29/32	29/32	30/33
DIVU	24	25	26/29	27/30	27/30	28/31
MULB (3-op)	12	12	14/17	13/15	15/18	16/19
MULUB (3-op)	10	10	12/15	12/16	12/16	14/17
MULB (2-op)	12	12	14/17	15/18	15/18	16/19
MULUB (2-op)	10	10	12/15	13/15	12/16	14/17
DIVB	18	18	20/23	21/24	21/24	22/25
DIVUB	16	16	18/21	19/22	19/22	20/23
AND (3-op)	5	6	7/10	8/11	7/10	8/11
AND (2-op)	4	5	6/8	7/9	6/8	7/9
OR (2-op)	4	5	6/8	7/9	6/8	7/9
XOR	4	5	6/8	7/9	6/8	7/9
ANDB (3-op)	5	5	7/10	8/11	7/10	8/11
ANDB (2-op)	4	4	6/8	7/9	6/8	7/9
ORB (2-op)	4	4	6/8	7/9	6/8	7/9
XORB	4	4	6/8	7/9	6/8	7/9
LD, LDB	4, 4	5, 4	5/8	6/8	6/9	7/10
ST, STB	4, 4	—	5/8	6/9	6/9	7/10
XCH, XCHB	5, 5	—	—	—	8/13	9/14
LDBSE	4	4	5/8	6/8	6/9	7/10
LDBZE	4	4	5/8	6/8	6/9	7/10
BMOV	6 + 8 per word + 3 for each memory controller reference					
BMOVi	7 + 8 per word + 14 for each interrupt + 3 for each memory controller reference					
PUSH (int stack)	6	7	9/12	10/13	10/13	11/14
POP (int stack)	8	—	10/12	11/13	11/13	12/14
PUSH (ext stack)	8	9	11/14	12/15	12/15	13/16
POP (ext stack)	11	—	13/15	14/16	14/16	15/17

*Times for operands addressed as SFRs and internal RAM (0-1FFH)/memory controller references (200-0FFFFH)).

NOTE:

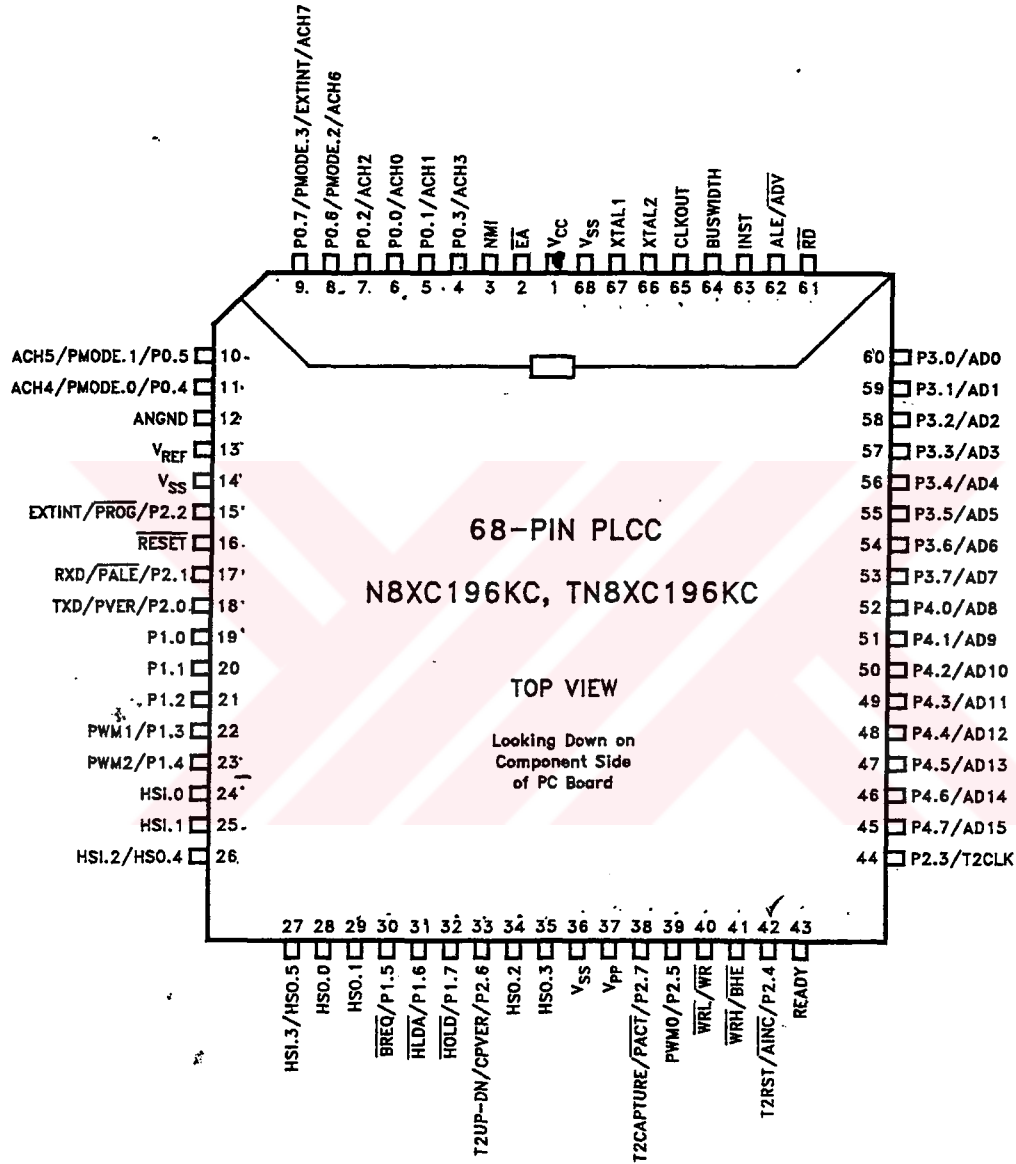
1. Execution times for memory controller references may be one to two states higher depending on the number of bytes in the prefetch queue.
2. INT stack is 0-1FFH and EXT stack is 200-0FFFFH.

MNEMONIC		MNEMONIC	
PUSHF (int stack)	6	PUSHF (ext stack)	8
POPF (int stack)	7	POPF (ext stack)	10
PUSHA (int stack)	12	PUSHA (ext stack)	18
POPA (int stack)	12	POPA (ext stack)	18
TRAP (int stack)	16	TRAP (ext stack)	18
LCALL (int stack)	11	LCALL (ext stack)	13
SCALL (int stack)	11	SCALL (ext stack)	13
RET (int stack)	11	RET (ext stack)	14
CMPL	7	DEC/DECB	3
CLR/CLRB	3	EXT/EXTB	4
NOT/NOTB	3	INC/INCB	3
NEG/NEGB	3		
LJMP	7		
SJMP	7		
BR [indirect]	7		
TIJMP	15 + 3 for each memory controller reference		
JNST, JST	4/8 jump not taken/jump taken		
JNH, JH	4/8 jump not taken/jump taken		
JGT, JLE	4/8 jump not taken/jump taken		
JNC, JC	4/8 jump not taken/jump taken		
JNVT, JVT	4/8 jump not taken/jump taken		
JNV, JV	4/8 jump not taken/jump taken		
JGE, JLT	4/8 jump not taken/jump taken		
JNE, JE	4/8 jump not taken/jump taken		
JBC, JBS	5/9 jump not taken/jump taken		
DJNZ	5/9 jump not taken/jump taken		
DJNZW	6/10 jump not taken/jump taken		
NORML	8 + 1 per shift (9 for 0 shift)		
SHRL	7 + 1 per shift (8 for 0 shift)		
SHLL	7 + 1 per shift (8 for 0 shift)		
SHRAL	7 + 1 per shift (8 for 0 shift)		
SHR/SHRB	6 + 1 per shift (7 for 0 shift)		
SHL/SHLB	6 + 1 per shift (7 for 0 shift)		
SHRA/SHRAB	6 + 1 per shift (7 for 0 shift)		
CLRC	2		
SETC	2		
DI	2		
EI	2		
DPTS	2		
EPTS	2		
CLRVT	2		
NOP	2		
RST	20 (includes fetch of configuration byte)		
SKIP	3		
IDLPD	8/25 (proper key/improper key)		

EK 3.

(8XC196KC ÜST GÖRÜNÜM)



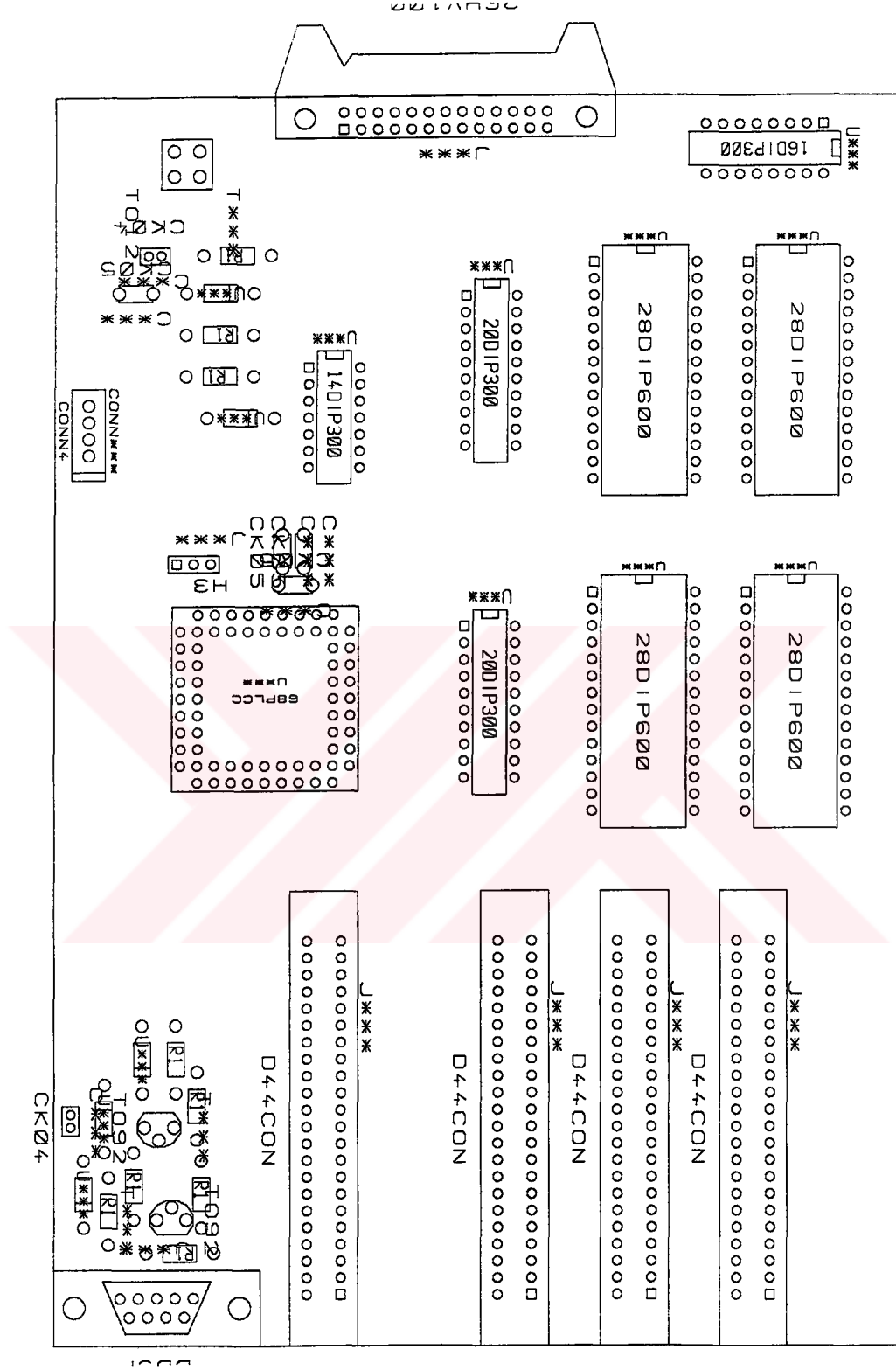


Şekil E.3.1. 8CX196KC'nin 68 bacaklı dış kılıfı

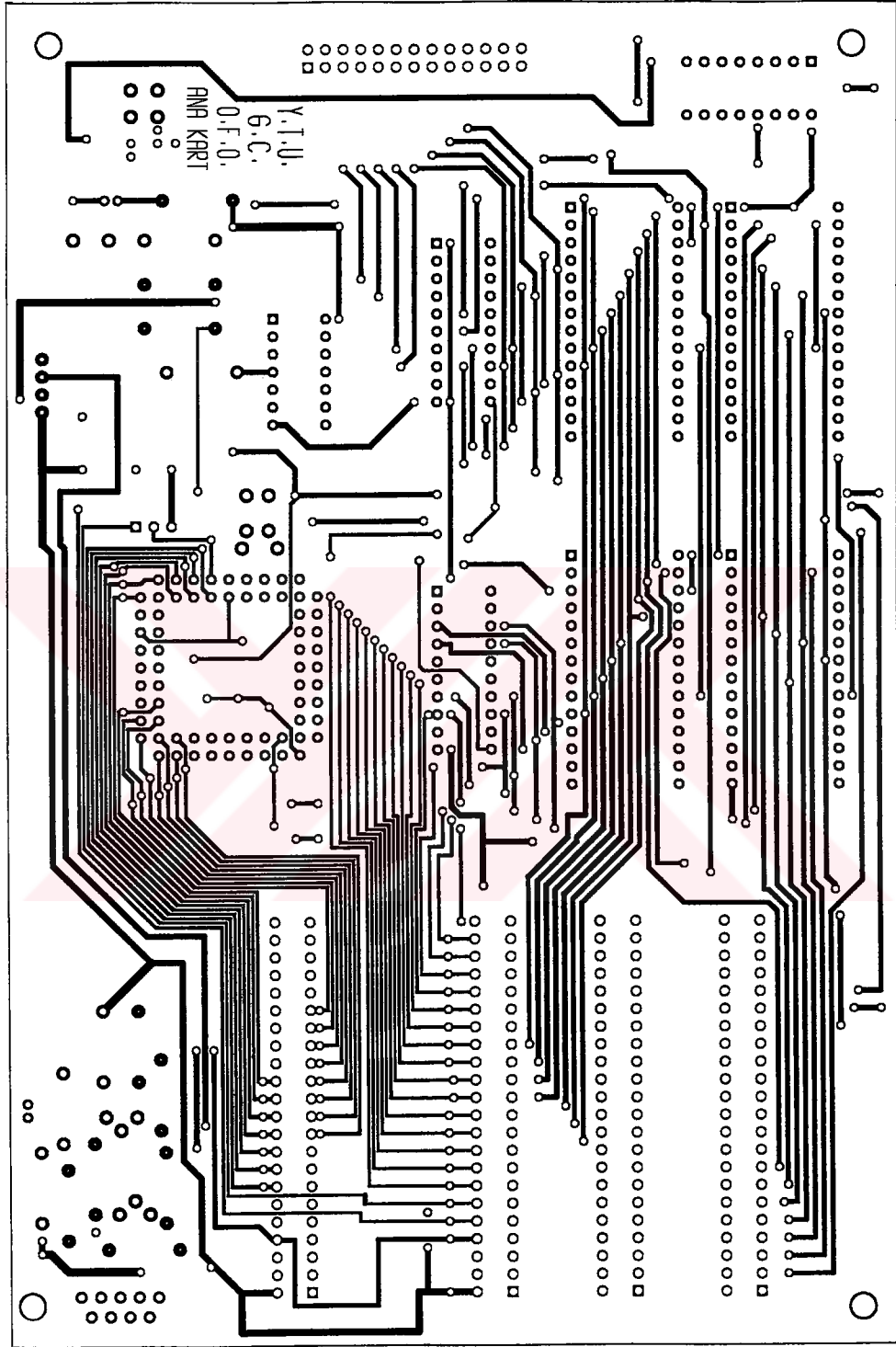
EK 4.

**(DEVRELERİN ÜST GÖRÜNÜMLERİ VE BASKILI
PLAKETLERİ)**

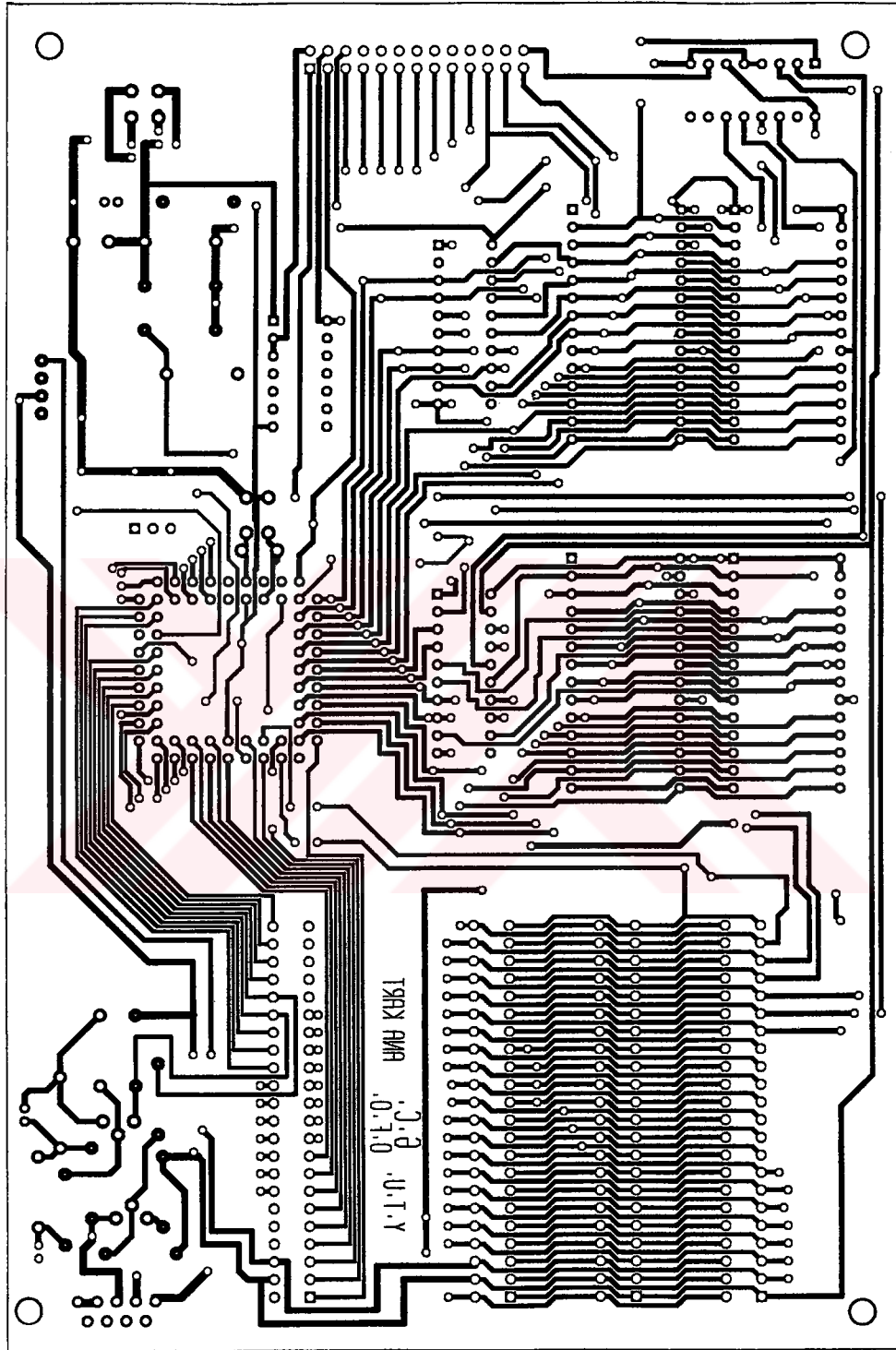




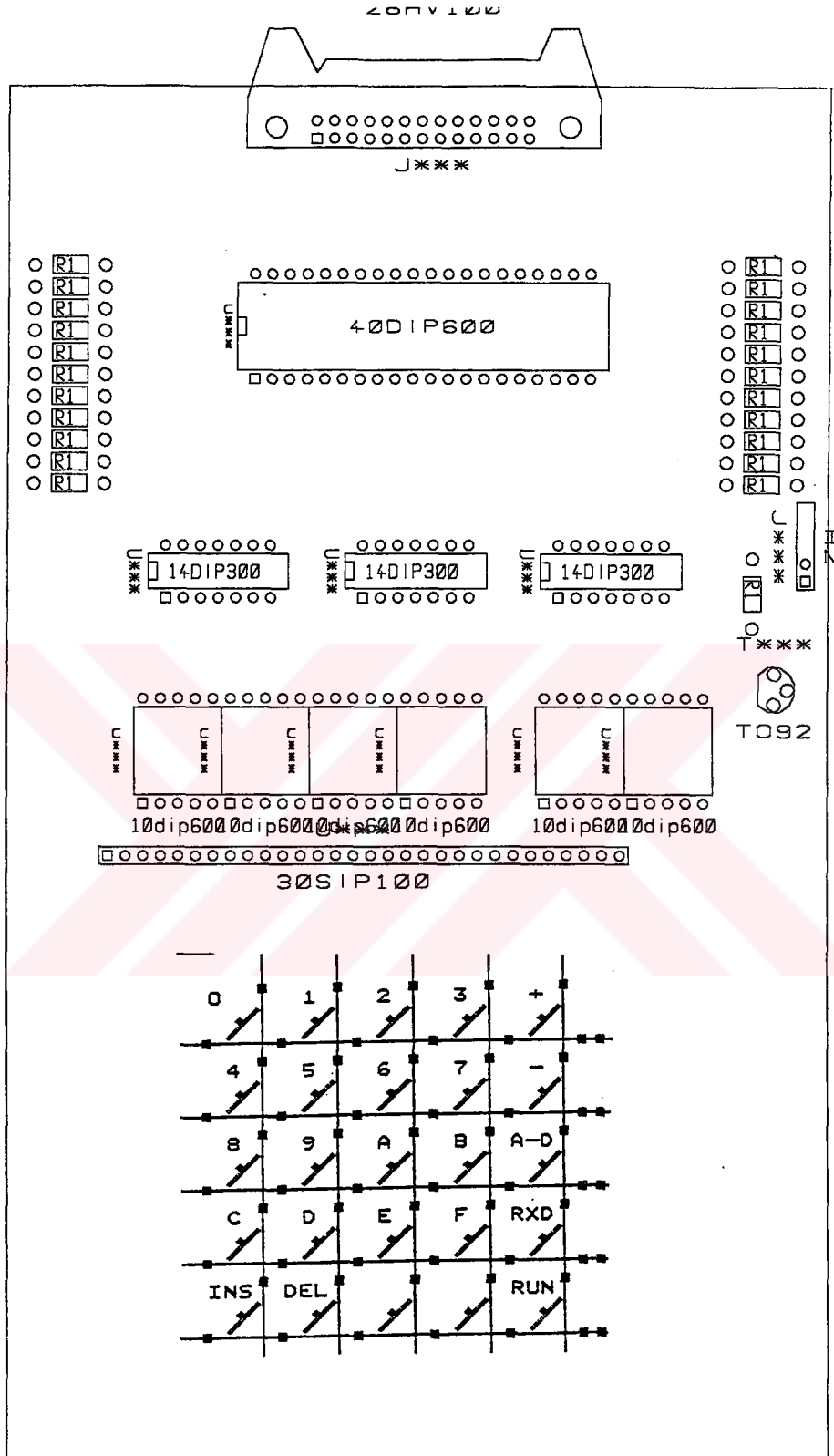
Şekil E.4.1. Anakart elemanlarının üst görünümü



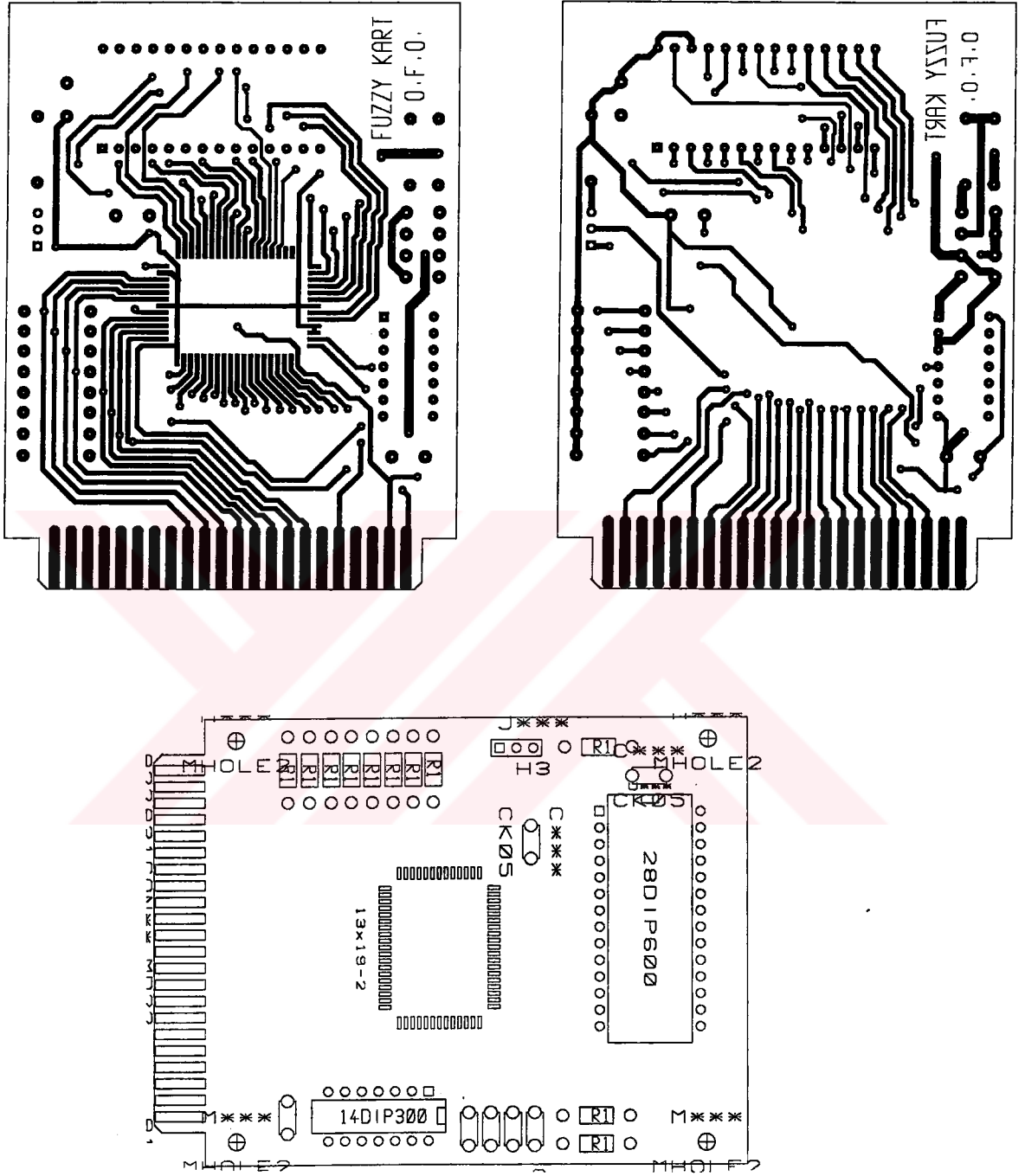
Şekil E.4.2. Anakart baskılı plaketa üst yüzey görünümü



Şekil E.4.3. Anakart baskılı plaketa alt yüzey görünümü



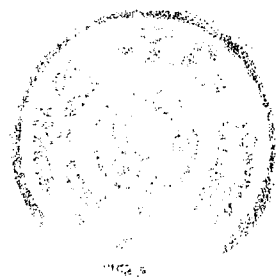
Şekil E.4.4. Gösterge ve tuş takımı devresi elemanlarının üst görünümü



Şekil E.4.5. Fuzzy kart elemanlarının, baskılı plaketinin üst alt yüzey görünümü

EK 5.

(SİSTEMLERİN PROGRAMLARI)



DOS 6.22 (038-N) MCS-96 MACRO ASSEMBLER, V1.2

SOURCE FILE: IKIMOTS.SRC

OBJECT FILE: IKIMOTS.OBJ

CONTROLS SPECIFIED IN INVOCATION COMMAND: PL(60) PW(80) PR

ERR	LOC	OBJECT	LINE	SOURCE STATEMENT
			1	
	001A		2	RSEG AT 1AH
	001A		3	AX: DSW 1
			4	
	001E		5	RSEG AT 1EH
	001E		6	CX: DSW 1
			7	
			8	
	0026		9	RSEG AT 26H
	0026		10	BSRR: DSB 1
			11	
	0030		12	RSEG AT 30H
	0030		13	HATA: DSW 1
	0032		14	FHATA: DSW 1
	0034		15	HATADEG: DSW 1
	0036		16	FHATADEG: DSW 1
	0038		17	CARP: DSL 1
	003C		18	BX: DSW 1
	003E		19	SONUC: DSW 1
	0040		20	GL: DSB 1
	0041		21	GH: DSB 1
	0042		22	ACC: DSB 1
	0044		23	DISPB: DSW 1
	0046		24	SAKLA: DSW 1
	0048		25	FARK: DSW 1
	004A		26	SAYI: DSW 1
	004C		27	SAY: DSB 1
	004E		28	TOPL: DSW 1
	0050		29	BOL: DSL 1
	0054		30	GOND: DSW 1
	0056		31	DEVIR: DSW 1
	0058		32	ONHATA: DSW 1
	005A		33	IGOND: DSW 1
	005C		34	IACC: DSB 1
	005D		35	DISPREG: DSB 1
			36	
	0015		37	IDCO EQU 15H
	0003		38	HSI_MODE EQU 03H
	0016		39	IOC1 EQU 16H
	0008		40	INT_MASK EQU 08H
	0004		41	HSI_TIME EQU 04H
	0006		42	HSI_STATUS EQU 06H
	0016		43	IOS1 EQU 16H
	000E		44	BAUDRATE EQU 0EH
	0067		45	BAUD9600L EQU 67H
	0080		46	BAUD9600H EQU 80H
	0009		47	CONSART EQU 09H
	0025		48	IOC1SART EQU 00100101B
	0007		49	SBUFTX EQU 07H
	0011		50	SER_STAT EQU 11H
	0011		51	SP_CON EQU 11H

ERR LOC	OBJECT	LINE	SOURCE STATEMENT
2159		52	BASTANTARA EQU 2159H
7F03		53	DISPBUF EQU 7F03H
20E2		54	DATASEG EQU 20E2H
D000		55	DACADR EQU 0D000H
E004		56	CPORT EQU 0E004H
E000		57	APORT EQU 0E000H
4500		58	OZEL EQU 4500H
0D5F		59	SINIR EQU 3423
2080		60	RESET EQU 2080H
2260		61	SERIOKU EQU 2260H
4600		62	RS232BUF EQU 4600H
		63	
4500		64	MOTOR2 EQU 4500H
4502		65	MOT2TOPL EQU 4502H
4504		66	MOT2SAK EQU 4504H
4506		67	MOT2SAY EQU 4506H
4507		68	VEYA2 EQU 4507H
		69	
4508		70	MOTOR1 EQU 4508H
450A		71	MOT1TOPL EQU 450AH
450C		72	MOT1SAK EQU 450CH
450E		73	MOT1SAY EQU 450EH
450F		74	VEYA1 EQU 450FH
		75	
003D		76	BXH EQU (BX+1)
			YTE
0032		77	FHATAL EQU FHATA
			YTE
0033		78	FHATAH EQU (FHATA+1)
			YTE
0036		79	FHATADEGL EQU FHATADEG
			YTE
0037		80	FHATADEGH EQU (FHATADEG+1)
			YTE
001A		81	AL EQU AX
			YTE
001B		82	AH EQU (AX+1)
			YTE
004E		83	TOPLL EQU TOPL
			YTE
004F		84	TOPLH EQU (TOPL+1)
			YTE
0050		85	BOLL EQU BOL
			ORD
0052		86	BOLH EQU (BOL+2)
			ORD
0038		87	CARPL EQU CARP
			ORD
003A		88	CARPH EQU (CARP+2)
			ORD
0054		89	GONDL EQU GOND
			YTE
0055		90	GONDH EQU (GOND+1)
			YTE
		91	
6000		92	CSEG AT 6000H
		93	

ERR	LOC	OBJECT	LINE	SOURCE STATEMENT
			94	
	6000	2841	95	SCALL ILKSART
			96	
	6002	286B	97	SCALL RS232SART
			98	
	6004	287F	99	SCALL BIRBAYT
			100	
	6006	2879	101	SCALL G
			102	
	6008	2893	103	SCALL HSISART
			104	
	600A	FB	105	EI
	600B		106	INTBEKLE:
	600B	A1007F44	107	LD DISPB,#7FOOH
	600F	B12041	108	LDB GH,#20H
	6012	B10440	109	LDB GL,#04H
	6015		110	TARA:
			111	
			112	; ORB GH,DENE
	6015	C70104E041	113	STB GH,CPORT
			114	; ANDB GH,#10111111B
			115	; XORB DENE,#01000000B
	601A	B24542	116	LDB ACC,[DISPB]+
	601D	C70100E042	117	STB ACC,APORT
	6022	B1F042	118	LDB ACC,#0FOH
	6025		119	BEKLE:
	6025	E042FD	120	DJNZ ACC,BEKLE
	6028	180141	121	SHRB GH,#01H
			122	
	602B	3C2602	123	JBS BSRR,4,AZALTTARA
	602E	2805	124	SCALL CIKHES
	6030		125	AZALTTARA:
	6030	E040E2	126	DJNZ GL,TARA
	6033	27D6	127	BR INTBEKLE
			128	
	6035		129	CIKHES:
	6035	B10042	130	LDB ACC,#00H
	6038	C70100E042	131	STB ACC,APORT
			132	
	603D	286B	133	SCALL ISLEM
			134	
	603F	911026	135	ORB BSRR,#10H
			136	
	6042	FO	137	RET
			138	;
			139	
	6043		140	ILKSART:
	6043	B10242	141	LDB ACC,#02
	6046	C7010E4542	142	STB ACC,MOT1SAY
	604B	C701064542	143	STB ACC,MOT2SAY
	6050	A1000054	144	LD GOND,#0000H
	6054	C3010A4554	145	ST GOND,MOT1TOPL
	6059	C301024554	146	ST GOND,MOT2TOPL
	605E	C7010F4554	147	STB GONDL,VEYA1
	6063	B18042	148	LDB ACC,#80H
	6066	C701074542	149	STB ACC,VEYA2
	606B	B1015D	150	LDB DISPREG,#01H

ERR	LOC	OBJECT	LINE	SOURCE STATEMENT
			151	
	606E	FO	152	RET
			153	
			154	;
	606F		155	RS232SART:
	606F	B1670E	156	LDB BAUDRATE,#BAUD96
Z			157	L ;Haberleme h
	6072	B1800E	157	LDB BAUDRATE,#BAUD96
			158	H ;9600 Baud
	6075	B10942	159	LDB ACC,#CONSART
			160	;P2.1/RXD'de RXD'ye izin ve
	6078	C41142	160	STB ACC,SP_CON
			161	;Mode 1 (asenkron hab.)
	607B	B12542	161	LDB ACC,#IOC1SART
			162	;HSI 'da bilgi okunabilir kes.
	607E	C41642	162	STB ACC,IOC1
			163	;tercih,HS05/HSI.3'den HSI.3
			164	;P2.0/TXD 'den TXD ye izin
			165	;HS04/HSI.2 'den HSI.2
			166	;T2OVF ve T1OVF kes.'ne set
			167	;INT_MASK.0=1 ile izin
	6081		167	G:
	6081	C40742	168	STB ACC,SBUFTX
			169	;Stat 'daki 5.bit reset ve
			170	;sonra set
	6084	FO	170	RET
			171	
	6085		172	BIRBAYT:
	6085	EFD8C1	173	LCALL SERIOKU
	6088	79A51A	174	SUBB AL,#0A5H
	608B	DF03	175	JE GERIDON
	608D	E7F0BF	176	LJMP RESET
	6090		177	GERIDON:
	6090	FO	178	RET
			179	;
	6091		180	GONDER:
	6091	B01142	181	LDB ACC,SER_STAT
			182	;Stat 'daki 5.bit reset ise
	6094	712042	182	ANDB ACC,#00100000B
			183	;SBUFX halen dolu
	6097	DFFB	183	JE GONDER
			184	
	6099	C4071A	185	STB AL,SBUFTX
	609C	FO	186	RET
			187	;
	609D		188	HSISART:
	609D	B15215	189	LDB IDCO,#01010010B
			190	;HSI.2 ve HSI.3 'e izin
	60A0	B15503	190	LDB HSI_MODE,#01010
			191	B ;Sonu DEN kenar
	60A3	B10408	191	LDB INT_MASK,#000000

RR LOC	OBJECT	LINE	SOURCE STATEMENT
			B ;Kesme iste'ine izin
60A6	91F826	192	ORB BSRR,#11111000B
60A9	F0	193	RET
		194	
60AA		195	ISLEM:
60AA	51602642	196	ANDB ACC,BSRR,#011000C
		197	B
		198	; JNE ISLEMYOK
60AE	E05D0A	199	DJNZ DISPREG,DISPYOK
60B1	B15A5D	200	LDB DISPREG,#5AH
60B4	A30100453E	201	LD SONUC,MOTOR2
60B9	2B0F	202	SCALL DISP
60BB		203	DISPYOK:
60BB	A30108453E	204	LD SONUC,MOTOR1
		205	; SCALL DISP
		206	
60C0	2B21	207	SCALL FUZZY
		208	
60C2	A30100461A	209	LD AX,RS232BUF
		210	; LD AX,MOTOR2
		211	
60C7	290A	212	SCALL IKIBAYTGON
60C9		213	ISLEMYOK:
		214	
60C9	F0	215	RET
		216	
60CA		217	DISP:
60CA	A1037F1E	218	LD CX,#DISPBUF
60CE	B13F1B	219	LDB AH,#3FH
		220	
60D1	4507003E54	221	ADD GOND,SONUC,#07H
		222	
60D6	B0541A	223	LDB AL,GONDL
60D9	EF06C0	224	LCALL DASEG
		225	
60DC	B0551A	226	LDB AL,GONDH
60DF	EF00C0	227	LCALL DASEG
		228	
60E2	F0	229	RET
		230	
60E3		231	FUZZY:
60E3	A30100453C	232	LD BX,MOTOR2
		233	; LD BX,#ODFOH
60E8	483E3C30	234	SUB HATA,BX,SONUC
60EC	332606	235	JBC BSRR,3,GEC
60EF	A0305B	236	LD ONHATA,HATA
60F2	71F726	237	ANDB BSRR,#11110111B
60F5		238	GEC:
		239	
		240	; LD AX,HATA
		241	; SCALL IKIBAYTGON
		242	
60F5	A0301A	243	LD AX,HATA
		244	
60F8	C30102461A	245	ST AX,4602H
		246	

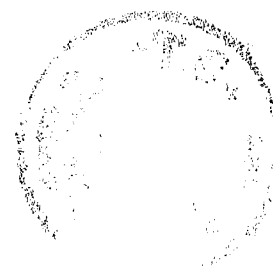
ERR	LOC	OBJECT	LINE	SOURCE STATEMENT
	60FD	A1FF2054	247	LD GOND,#20FFH
	6101	28A2	248	SCALL DLCEK
	6103	A03C32	249	LD FHATA,BX
			250	
			251	LD AX,BX
			252	SCALL IKIBAYTGON
			253	
	6106	48583034	254	SUB HATADEG,HATA,ONHATA
			A	
	610A	A03058	255	LD ONHATA,HATA
			256	
			257	LD AX,HATADEG
			258	SCALL IKIBAYTGON
			259	
	610D	A0341A	260	LD AX,HATADEG
			261	
	6110	C30104461A	262	ST AX,4604H
			263	
	6115	A1FE0354	264	LD GOND,#03FEH
	6119	288A	265	SCALL DLCEK
	611B	A03C36	266	LD FHATADEG,BX
			267	
			268	LDB AX,FHATADEG
			269	SCALL IKIBAYTGON
			270	
	611E	280F	271	SCALL GIRISYAZ
	6120	2822	272	SCALL CIKBASLMES
	6122	2831	273	SCALL CIKISAL
			274	
	6124	51F01B42	275	ANDB ACC,AH,#0F0H
	6128	D704	276	JNE SONUCYOK
			277	
			278	LD BX,AX
			279	SCALL IKIBAYTGON
			280	LD AX,BX
			281	
	612A	2891	282	SCALL TERSOLCEK
	612C	2842	283	SCALL SINKARYAZ
			284	
	612E		285	SONUCYOK:
	612E	FO	286	RET
			287	
	612F		288	GIRISYAZ:
	612F	C701048033	289	STB FHATAH,8004H
	6134	C701068032	290	STB FHATAL,8006H
	6139	C701088037	291	STB FHATADEGH,800BH
	613E	C7010A8036	292	STB FHATADEGL,800AH
	6143	FO	293	RET
			294	
	6144		295	CIKBASLMES:
	6144	B10442	296	LDB ACC,#04H
	6147	C701008042	297	STB ACC,8000H
	614C		298	MESGUL:
	614C	B301008042	299	LDB ACC,8000H
	6151	3A42F8	300	JBS ACC,2,MESGUL
	6154	FO	301	RET
			302	

ERR	LOC	OBJECT	LINE	SOURCE STATEMENT
			303	
	6155		304	CIKISAL:
	6155	B10542	305	LDB ACC,#05H
	6158		306	OKU:
	6158	B30124801B	307	LDB AH,8024H
	615D	B30124801B	308	LDB AH,8024H
	6162	B30126801A	309	LDB AL,8026H
	6167	B30126801A	310	LDB AL,8026H
	616C	E042E9	311	DJNZ ACC,OKU
	616F	FO	312	RET
			313	
			314	
	6170		315	SINKARYAZ:
	6170	641A56	316	ADD DEVIR,AX
			317	
	6173	410080561A	318	AND AX,DEVIR,#8000H
	6178	DF06	319	JE KARS
	617A	A1860156	320	LD DEVIR,#0186H
	617E	2016	321	BR YOLADEV
	6180		322	KARS:
	6180	B9860156	323	CMP DEVIR,#0186H
	6184	DB06	324	JC DEVBUYUK
	6186	A1860156	325	LD DEVIR,#0186H
	618A	200A	326	BR YOLADEV
	618C		327	DEVBUYUK:
	618C	B9780D56	328	CMP DEVIR,#0D78H
	6190	D304	329	JNC YOLADEV
	6192	A1780D56	330	LD DEVIR,#0D78H
	6196		331	YOLADEV:
			332	
	6196	C301064656	333	ST DEVIR,4606H
	619B	0356	334	NEG DEVIR
	619D	C30100D056	335	ST DEVIR,DACADR
	61A2	0356	336	NEG DEVIR
	61A4	FO	337	RET
			338	
	61A5		339	OLCEK:
	61A5	A1FF073C	340	LD BX,#2047
	61A9	FE4C1A3C38	341	MUL CARP,BX,AX
	61AE	FE8C5438	342	DIV CARP,BOND
	61B2	A0383C	343	LD BX,CARPL
	61B5	65FF073C	344	ADD BX,#2047
	61B9	710F3D	345	ANDB BXH,#0FH
	61BC	FO	346	RET
			347	
	61BD		348	TERSOLCEK:
	61BD	69FF071A	349	SUB AX,#2047
	61C1	A100063C	350	LD BX,#0600H
	61C5	FE4C3C1A38	351	MUL CARP,AX,BX
	61CA	FE8DFF0738	352	DIV CARP,#2047
	61CF	A0381A	353	LD AX,CARPL
	61D2	FO	354	RET
			355	
	61D3		356	IKIBAYTGON:
	61D3	EFBBFE	357	LCALL GONDER
	61D6	B01B1A	358	LDB AL,AX
	61D9	EFB5FE	359	LCALL GONDER

RR LOC OBJECT	LINE	SOURCE STATEMENT
61DC FO	360	RET
	361	
	362	
7000	363	CSEG AT 7000H
	364	
7000 F4	365	PUSHA
7001 CB1A	366	PUSH AL
	367	
7003 287C	368	SCALL MOTOR12
7005	369	INTBAS:
7005 28A9	370	SCALL YUKLE
	371	
7007 48464A48	372	SUB FARK,SAYI,SAKLA
700B DB0A	373	JC DEV
	374	
700D A1FFFF48	375	LD FARK,#OFFFFH
7011 684648	376	SUB FARK,SAKLA
7014 644A48	377	ADD FARK,SAYI
7017	378	DEV:
7017 B9100048	379	CMP FARK,#0010H
701B D32A	380	JNC YANLIS
	381	
701D A04A46	382	LD SAKLA,SAYI
7020 44484E4E	383	ADD TOPL,TOPL,FARK
	384	
7024 154C	385	DECB SAY
7026 D70B	386	JNE BIRDEVYOK
	387	
	388	;
	389	ANDB BSRR,#11101111B
7028 2848	390	SCALL CEVIR
702A 282C	391	SCALL KODLAYUK
	392	
702C B1024C	393	LDB SAY,#02
702F A100004E	394	LD TOPL,#0
7033	395	BIRDEVYOK:
7033 2816	396	SCALL BELLEKYAZ
	397	
7035 71505C	398	ANDB IACC,#01010000B
7038 8550005C	399	XOR IACC,#01010000B
703C D709	400	JNE YANLIS
703E A108455A	401	LD IGOND,#MOTOR1
7042 845C5C	402	XOR IACC,IACC
7045 27BE	403	BR INTBAS
7047	404	YANLIS:
7047 CC1A	405	POP AL
7049 F5	406	POPA
704A FO	407	RET
	408	
704B	409	BELLEKYAZ:
704B C35A0446	410	ST SAKLA,4[IGOND]
704F C75A064C	411	STB SAY,6[IGOND]
7053 C35A024E	412	ST TOPL,2[IGOND]
7057 FO	413	RET
	414	
7058	415	KODLAYUK:
7058 C25A4E	416	ST TOPL,[IGOND]

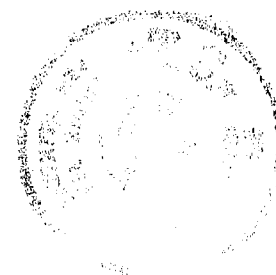
ERR	LOC	OBJECT	LINE	SOURCE	STATEMENT
	705B	B908455A	417	CMF	IGOND,#4508H
	705F	D710	418	JNE	SERIYUKYOK
			419		
	7061	6507004E	420	ADD	TOPL,#07
	7065	935A074F	421	ORB	TOPLH,7[IGOND]
			422		
	7069	C30100464E	423	ST	TOPL,RS232BUF
			424		
			425	; ANDB	AL,BSRR,#011000
			426	; JNE	SERIYUKYOK
			427		
	706E	71EF26	428	ANDB	BSRR,#11101111B
	7071		429	SERIYUKYOK:	
	7071	F0	430	RET	
			431		
			432	; CEVIR:	
	7072	A1008750	433	LD	BOLL,#8700H
	7076	A1930352	434	LD	BOLH,#0393H
	707A	8C4E50	435	DIVU	BOL,TOPL
	707D	A0504E	436	LD	TOPL,BOLL
	7080	F0	437	RET	
			438		
	7081		439	MOTOR12:	
	7081	B0065C	440	LDB	IACC,HSI_STATUS
	7084	A0044A	441	LD	SAYI,HSI_TIME
			442		
	7087	3E5C05	443	JBS	IACC,6,SONUCMOT2
	708A	3C5C13	444	JBS	IACC,4,SONUCMOT1
	708D	2020	445	BR	GERIDON1
	708F		446	SONUCMOT2:	
	708F	352608	447	JBC	BSRR,5,ILKYUKYOK
	7092	C30104454A	448	ST	SAYI,MOT2SAK
	7097	71DF26	449	ANDB	BSRR,#11011111B
	709A		450	ILKYUKYOK2:	
	709A	A100455A	451	LD	IGOND,#MOTOR2
	709E	200F	452	BR	GERIDON1
			453		
	70A0		454	SONUCMOT1:	
	70A0	362608	455	JBC	BSRR,6,ILKYUKYOK
	70A3	C3010C454A	456	ST	SAYI,MOT1SAK
	70AB	71BF26	457	ANDB	BSRR,#10111111B
			458		
	70AB		459	ILKYUKYOK1:	
	70AB	A108455A	460	LD	IGOND,#MOTOR1
	70AF		461	GERIDON1:	
			462		
	70AF	F0	463	RET	
			464		
	70B0		465	YUKLE:	
	70B0	A35A0446	466	LD	SAKLA,4[IGOND]
	70B4	B35A064C	467	LDB	SAY,6[IGOND]
	70B8	A35A024E	468	LD	TOPL,2[IGOND]
	70BC	F0	469	RET	
			470		
	4000		471	CSEG AT	4000H
	4000	A1FF0756	472	LD	DEVIR,#07FFH

ERR	LOC	OBJECT	LINE	SOURCE STATEMENT
	4004	0356	473	NEG DEVIR
	4006	C30100D056	474	ST DEVIR,DACADR
	4008	0356	475	NEG DEVIR
	400D	FO	476	RET
	400E		477	END

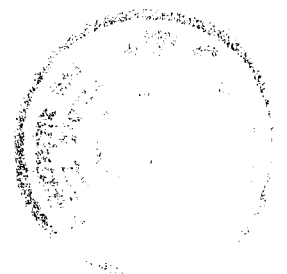


YMBOL TABLE LISTING

N A M E	VALUE	ATTRIBUTES
ACC	0042H	REG ABS BYTE
AH.	001BH	REG ABS BYTE
AL.	001AH	REG ABS BYTE
APORT	E000H	NULL ABS
AX.	001AH	REG ABS WORD
AZALTARA	6030H	CODE ABS ENTRY
BASTANTARA.	2159H	NULL ABS
BAUD9600H	0080H	NULL ABS
BAUD9600L	0067H	NULL ABS
BAUDRATE.	000EH	NULL ABS
BEKLE	6025H	CODE ABS ENTRY
BELLEKYAZ	704BH	CODE ABS ENTRY
BIRBAYT	6085H	CODE ABS ENTRY
BIRDEVYDK	7033H	CODE ABS ENTRY
BOL	0050H	REG ABS LONG
BOLH.	0052H	REG ABS WORD
BOLL.	0050H	REG ABS WORD
BSRR.	0026H	REG ABS BYTE
BX.	003CH	REG ABS WORD
BXH	003DH	REG ABS BYTE
CARP.	0038H	REG ABS LONG
CARPH	003AH	REG ABS WORD
CARPL	0038H	REG ABS WORD
CEVIR	7072H	CODE ABS ENTRY
CIKBASLMES.	6144H	CODE ABS ENTRY
CIKHES.	6035H	CODE ABS ENTRY
CIKISAL	6155H	CODE ABS ENTRY
CONSART	0009H	NULL ABS
CPORT	E004H	NULL ABS
CX.	001EH	REG ABS WORD
DACADR.	D000H	NULL ABS
DASEG	20E2H	NULL ABS
DEV	7017H	CODE ABS ENTRY
DEVBUYUK.	618CH	CODE ABS ENTRY
DEVIR	0056H	REG ABS WORD
DISP.	60CAH	CODE ABS ENTRY
DISPB	0044H	REG ABS WORD
DISPBUF	7F03H	NULL ABS
DISPREG	005DH	REG ABS BYTE
DISPYOK	60BBH	CODE ABS ENTRY
FARK.	0048H	REG ABS WORD
FHATA	0032H	REG ABS WORD
FHATADEG.	0036H	REG ABS WORD
FHATADEGH	0037H	REG ABS BYTE
FHATADEGL	0036H	REG ABS BYTE
FHATAH.	0033H	REG ABS BYTE
FHATAL.	0032H	REG ABS BYTE
FUZZY	60E3H	CODE ABS ENTRY
G	6081H	CODE ABS ENTRY
GEC	60F5H	CODE ABS ENTRY
GERIDON	6090H	CODE ABS ENTRY
GERIDON1.	70AFH	CODE ABS ENTRY
GH.	0041H	REG ABS BYTE

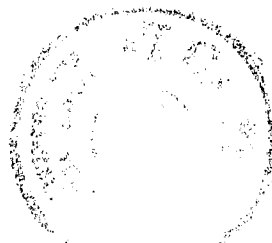


N A M E	VALUE	ATTRIBUTES
IRISYAZ.	612FH	CODE ABS ENTRY
L.	0040H	REG ABS BYTE
ONND.	0054H	REG ABS WORD
ONDER.	6091H	CODE ABS ENTRY
ONNDH	0055H	REG ABS BYTE
ONDL	0054H	REG ABS BYTE
HATA.	0030H	REG ABS WORD
HATADEG	0034H	REG ABS WORD
ISI_MODE.	0003H	NULL ABS
ISI_STATUS.	0006H	NULL ABS
ISI_TIME.	0004H	NULL ABS
ISISART	609DH	CODE ABS ENTRY
IACC.	005CH	REG ABS BYTE
IGOND	005AH	REG ABS WORD
IKIBAYTGON.	61D3H	CODE ABS ENTRY
ILKSART	6043H	CODE ABS ENTRY
ILKYUKYOK1.	70ABH	CODE ABS ENTRY
ILKYUKYOK2.	709AH	CODE ABS ENTRY
INT_MASK.	0008H	NULL ABS
INTBAS.	7005H	CODE ABS ENTRY
INTBEKLE.	600BH	CODE ABS ENTRY
IOCO.	0015H	NULL ABS
IOC1.	0016H	NULL ABS
IOC1SART.	0025H	NULL ABS
IOS1.	0016H	NULL ABS
ISLEM	60AAH	CODE ABS ENTRY
ISLEMYOK.	60C9H	CODE ABS ENTRY
KARS.	6180H	CODE ABS ENTRY
KODLAYUK.	7058H	CODE ABS ENTRY
MESGUL.	614CH	CODE ABS ENTRY
MOT1SAK	450CH	NULL ABS
MOT1SAY	450EH	NULL ABS
MOT1TOPL.	450AH	NULL ABS
MOT2SAK	4504H	NULL ABS
MOT2SAY	4506H	NULL ABS
MOT2TOPL.	4502H	NULL ABS
MOTOR1.	4508H	NULL ABS
MOTOR12	7081H	CODE ABS ENTRY
MOTOR2.	4500H	NULL ABS
OKU	6158H	CODE ABS ENTRY
OLCEK	61A5H	CODE ABS ENTRY
ONHATA.	0058H	REG ABS WORD
OZEL.	4500H	NULL ABS
RESET	2080H	NULL ABS
RS232BUF.	4600H	NULL ABS
RS232SART	606FH	CODE ABS ENTRY
SAKLA	0046H	REG ABS WORD
SAY	004CH	REG ABS BYTE
SAYI.	004AH	REG ABS WORD
SBUFTX.	0007H	NULL ABS
SER_STAT.	0011H	NULL ABS
SERIOKU	2260H	NULL ABS
SERIYUKYOK.	7071H	CODE ABS ENTRY
SINIR	0D5FH	NULL ABS
SINKARYAZ	6170H	CODE ABS ENTRY
SONUC	003EH	REG ABS WORD



N A M E	VALUE	ATTRIBUTES
SONUCMOT1	70A0H	CODE ABS ENTRY
SONUCMOT2	708FH	CODE ABS ENTRY
SONUCYOK.	612EH	CODE ABS ENTRY
SP_CON.	0011H	NULL ABS
TARA.	6015H	CODE ABS ENTRY
TERSOLCEK	61BDH	CODE ABS ENTRY
TOPL.	004EH	REG ABS WORD
TOPLH	004FH	REG ABS BYTE
TOPLL	004EH	REG ABS BYTE
VEYA1	450FH	NULL ABS
VEYA2	4507H	NULL ABS
YANLIS.	7047H	CODE ABS ENTRY
YOLADEV	6196H	CODE ABS ENTRY
YUKLE	70B0H	CODE ABS ENTRY

ASSEMBLY COMPLETED, NO ERROR(S) FOUND.



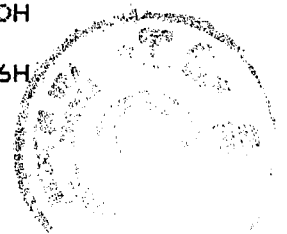
DOS 6.22 (038-N) MCS-96 MACRO ASSEMBLER, V1.2

SOURCE FILE: GEL4SERI.SRC

OBJECT FILE: GEL4SERI.OBJ

CONTROLS SPECIFIED IN INVOCATION COMMAND: PL(60) PW(80) PR

ERR	LOC	OBJECT	LINE	SOURCE STATEMENT
			1	
	001A		2	RSEG AT 1AH
	001A		3	AX: DSW 1
	001C		4	BX: DSW 1
	001E		5	CX: DSW 1
	0020		6	DX: DSW 1
	0022		7	EX: DSW 1
	0024		8	AG: DSW 1
	0026		9	BSRR: DSB 1
			10	;
			11	;
	0014		12	WSR EQU 14H
	0000		13	HWINDOWO EQU 00H
	0020		14	VWINDOWO EQU 20H
	001B		15	SP EQU 1BH:WORD
			16	;
	000E		17	BAUD_RATE EQU 0EH
	0009		18	SERI_SART EQU 09H
	0011		19	SERI_CON EQU 11H
	0011		20	SERI_STAT EQU 11H
	0007		21	SBUF_RX EQU 07H
	0007		22	SBUF_TX EQU 07H
	00CF		23	BAUD4800L EQU 0CFH
	0080		24	BAUD4800H EQU 80H
			25	
	E000		26	APORT EQU 0E000H ;
				gmentler
	E002		27	BPORT EQU 0E002H ;
r okuma				t
	E004		28	CPORT EQU 0E004H ;
				tot ve sutun
	E006		29	COMMAND EQU 0E006H
			30	;
	001A		31	AL EQU AX:BYTE
	001C		32	BL EQU BX:BYTE
	001E		33	CL EQU CX:BYTE
	0020		34	DL EQU DX:BYTE
	0022		35	EL EQU EX:BYTE
	0024		36	AGL EQU AG
			37	;
	001B		38	AH EQU (AX+1):BYTE
	0021		39	DH EQU (DX+1):BYTE
	0023		40	EH EQU (EX+1):BYTE
	0025		41	AGH EQU (AG+1):BYTE
			42	;
	7F0A		43	ADRAM EQU 7F0AH
	7F05		44	DISPBUF EQU 7F05H
	3F00		45	HEXDIS EQU 3F00H
	3F30		46	TUSFONKTAB EQU 3F30H
	0082		47	SARTLA EQU 82H
	7F16		48	RUNYAN EQU 7F16H



ERR	LOC	OBJECT	LINE	SOURCE	STATEMENT
		7F1C	49	RUNYANYAN	EQU 7F1CH
			50	:	
			51	:	
			52	:	
	2080		53		CSEG AT 2080H
			54	:	
	2080	B12014	55		LDB WSR,#VWINDOWO
	2083	B10014	56		LDB WSR,#HWINDOWO
	2086	A1000218	57		LD SP,#0200H
	208A	2813	58		SCALL ILKSART
	208C	A1006024	59		LD AG,#6000H
	2090	2837	60		SCALL DISPGDS
	2092		61	ANA:	
	2092	28B1	62		SCALL GERTARAMA
	2094	B01B1A	63		LDB AL,AH
	2097	28A2	64		SCALL DEGERI
	2099	29CF	65		SCALL BEEP
	209B	2914	66		SCALL ISLEM
	209D	27F3	67		BR ANA
			68	:	
			69	:	
	209F		70	ILKSART:	
	209F	B1821A	71		LDB AL,#SARTLA
	20A2	C70106E01A	72		STB AL,COMMAND
			73		
	20A7	B1FF26	74		LDB BSRR,#OFFH
			75		
	20AA	B1CF0E	76		LDB BAUD_RATE,#BAUD4
			77	OL	
	20AD	B1800E	77		LDB BAUD_RATE,#BAUD4
			78	OH	
	20B0	B1091A	78		LDB AL,#SERI_SART
	20B3	C4111A	79		STB AL,SERI_CON
			80		
	20B6	B1F01A	81		LDB AL,#OF0H
				;RET komutu	
	20B9	C7010A7F1A	82		STB AL,ADRAM
	20BE	C701167F1A	83		STB AL,RUNYAN
	20C3	C7011C7F1A	84		STB AL,RUNYANYAN
	20C8	F0	85		RET
			86	:	
			87	:	
	20C9		88	DISPGDS:	
	20C9	A1057F1E	89		LD CX,#DISPBUF
	20CD	B13F1B	90		LDB AH,#3FH
	20D0	B2241A	91		LDB AL,[AG]
	20D3	280D	92		SCALL DATASEG
	20D5	B0241A	93		LDB AL,AGL
	20D8	2808	94		SCALL DATASEG
	20DA	B0251A	95		LDB AL,AGH
	20DD	2803	96		SCALL DATASEG
	20DF	281B	97		SCALL NOKTA
	20E1	F0	98		RET
			99	:	
			100	:	
	20E2		101	DATASEG:	
	20E2	B01A22	102		LDB EL,AL



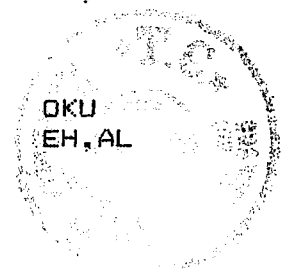
ERR	LOC	OBJECT	LINE	SOURCE STATEMENT
	20E5	2809	103	SCALL SEGSEKIL
	20E7	B0221A	104	LDB AL,EL
	20EA	18041A	105	SHRB AL,#04
	20ED	2801	106	SCALL SEGSEKIL
	20EF	F0	107	RET
			108	;
			109	;
	20F0		110	SEGSEKIL:
	20F0	710F1A	111	ANDB AL,#0FH
	20F3	B21A1A	112	LDB AL,[AX]
	20F6	C61E1A	113	STB AL,[CX]
	20F9	051E	114	DEC CX
	20FB	F0	115	RET
			116	;
			117	;
	20FC		118	NOKTA:
	20FC	38260F	119	JBS BSRR,0,NOKTADA
			120	;
			121	;
	20FF		122	NOKTAAD:
	20FF	A1057F1E	123	LD CX,#DISPBUF
	2103	B10222	124	LDB EL,#02
	2106	2824	125	SCALL NOKTASIL
	2108	B10422	126	LDB EL,#04
	210B	2810	127	SCALL NOKTAKOY
	210D	F0	128	RET
			129	;
			130	;
	210E	A1057F1E	131	NOKTADA: LD CX,#DISPBUF
	2112	B10222	132	LDB EL,#02
	2115	2806	133	SCALL NOKTAKOY
	2117	B10422	134	LDB EL,#04
	211A	2810	135	SCALL NOKTASIL
	211C	F0	136	RET
			137	;
			138	;
	211D		139	NOKTAKOY:
	211D	B21E1A	140	LDB AL,[CX]
	2120	91201A	141	ORB AL,#20H
				okta seg. bit 5
	2123	C61E1A	142	STB AL,[CX]
	2126	051E	143	DEC CX
	2128	E022F2	144	DJNZ EL,NOKTAKOY
	212B	F0	145	RET
			146	;
			147	;
	212C		148	NOKTASIL:
	212C	B21E1A	149	LDB AL,[CX]
	212F	71DF1A	150	ANDB AL,#0DFH
	2132	C61E1A	151	STB AL,[CX]
	2135	051E	152	DEC CX
	2137	E022F2	153	DJNZ EL,NOKTASIL
	213A	F0	154	RET
			155	;
			156	;
	213B		157	DEGERI:
	213B	B13F1B	158	LDB AH,#3FH



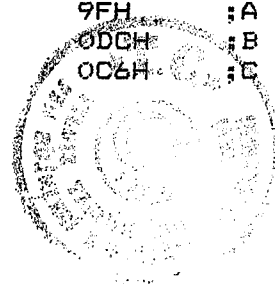
ERR	LOC	OBJECT	LINE	SOURCE	STATEMENT
	213E	75101A	159		ADDB AL,#10H
	2141	B21A1A	160		LDB AL,[AX]
	2144	F0	161		RET
			162	:	
			163	:	
	2145		164	GERTARAMA:	
	2145	B10623	165		LDB EH,#06
	2148		166	TUSUBIRAK:	
	2148	280F	167		SCALL BASTANTARA
	214A	3126F8	168		JBC BSRR,1,GERTARAMA
	214D	E023F8	169		DJNZ EH,TUSUBIRAK
			170	:	
	2150		171	TUSABAS:	
	2150	2807	172		SCALL BASTANTARA
	2152	EFB55D	173		LCALL GERIDON
	2155	3926F8	174		JBS BSRR,1,TUSABAS
	2158	F0	175		RET
			176	:	
			177	:	
	2159		178	BASTANTARA:	
	2159	910226	179		ORB BSRR,#02
	215C	B1001C	180		LDB BL,#00
	215F	A1057F1E	181		LD CX,#DISPBUF
	2163	B10120	182		LDB DL,#01
	2166	B10621	183		LDB DH,#06
	2169		184	TARAMA:	
	2169	B21E1A	185		LDB AL,[CX]
	216C	C70100E01A	186		STB AL,APOINT
	2171	C70104E020	187		STB DL,CPOINT
	2176	B1FF22	188		LDB EL,#OFFH
	2179		189	BEKLE:	
	2179	E022FD	190		DJNZ EL,BEKLE
	217C	111A	191		CLRB AL
	217E	C70100E01A	192		STB AL,APOINT
	2183	B0201A	193		LDB AL,DL
	2186	121A	194		NOTB AL
	2188	717F1A	195		ANDB AL,#7FH
	218B	C70104E01A	196		STB AL,CPOINT
	2190	B30102E01A	197		LDB AL,BPOINT
	2195	B10522	198		LDB EL,#05
	2198		199	BASILITUS:	
	2198	18011A	200		SHRB AL,#01
	219B	DB06	201		JC ONEMSIZ
	219D	B01C1B	202		LDB AH,BL
	21A0	71FD26	203		ANDB BSRR,#0FDH
	21A3		204	ONEMSIZ:	
	21A3	171C	205		INCB BL
	21A5	E022F0	206		DJNZ EL,BASILITUS
	21A8	190120	207		SHLB DL,#01
	21AB	051E	208		DEC CX
	21AD	E021B9	209		DJNZ DH,TARAMA
	21B0	F0	210		RET
			211	:	
			212	:	
	21B1		213	ISLEM:	
	21B1	99101A	214		CMFB AL,#10H
	21B4	D316	215		JNC TUSHEX

ERR	LOC	OBJECT	LINE	SOURCE STATEMENT
	21B6	79101A	216	SUBB AL,#10H
	21B9	A1303F1C	217	LD BX,#TUSFONKTAB
	21BD	741A1C	218	ADDB BL,AL
	21C0	A21C1E	219	LD CX,[BX]
	21C3	2801	220	SCALL FONKS
	21C5	FO	221	RET
			222	;
			223	;
	21C6		224	FONKS:
	21C6	C81E	225	PUSH CX
	21C8	FO	226	RET
			227	;
			228	;
	21C9		229	BASLA:
	21C9	C824	230	PUSH AG
	21CB	FO	231	RET
			232	;
			233	;
	21CC		234	TUSHEX:
	21CC	382603	235	JBS BSRR,0,DATADeg
	21CF	2810	236	SCALL ADDEG
	21D1	FO	237	RET
			238	;
			239	;
	21D2		240	DATADeg:
	21D2	B2241B	241	LDB AH,[AG]
	21D5	19041B	242	SHLB AH,#04
	21D8	901A1B	243	ORB AH,AL
	21DB	C6241B	244	STB AH,[AG]
	21DE	2EE9	245	SCALL DISPGOS
	21E0	FO	246	RET
			247	;
			248	;
	21E1		249	ADDEG:
	21E1	090424	250	SHL AG,#04
	21E4	901A24	251	ORB AGL,AL
	21E7	2EE0	252	SCALL DISPGOS
	21E9	FO	253	RET
			254	;
			255	;
	21EA		256	ADARTTIR:
	21EA	0724	257	INC AG
	21EC	2EDB	258	SCALL DISPGOS
	21EE	FO	259	RET
			260	;
			261	;
	21EF		262	ADAZALT:
	21EF	0524	263	DEC AG
	21F1	2ED6	264	SCALL DISPGOS
	21F3	FO	265	RET
			266	;
			267	;
	21F4		268	AD_DAT:
	21F4	950126	269	XORB BSRR,#01H
	21F7	2F03	270	SCALL NOKTA
	21F9	FO	271	RET
			272	;

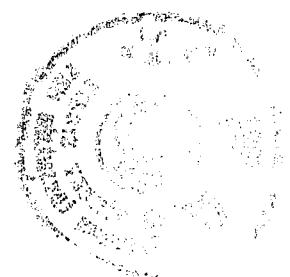
ERR	LOC	OBJECT	LINE	SOURCE STATEMENT
			273	;
	21FA		274	DEL:
	21FA C824		275	
	21FC		276	DELDEV:
	21FC B324011A		277	LDB AL,01H[AG]
	2200 C6251A		278	STB AL,[AG]+
	2203 B9FF7E24		279	CMP AG,#7EFFH
	2207 D7F3		280	JNE DELDEV
	2209 CC24		281	POP AG
	220B 2EBC		282	SCALL DISPGOS
	220D F0		283	RET
			284	;
			285	;
	220E		286	INSERT:
	220E C824		287	
	2210 B2251B		288	PUSH AG
	2213		289	LDB AH,[AG]+
	2213 B2241A		290	
	2216 B01A22		291	LDB AL,[AG]
	2219 B01B1A		292	LDB EL,AL
	221C B0221B		293	LDB AL,AH
	221F C6251A		294	LDB AH,EL
	2222 B9007F24		295	STB AL,[AG]+
	2226 D7EB		296	CMP AG,#7FOOH
	2228 CC24		297	JNE INSDEV
	222A 2E9D		298	POP AG
	222C F0		299	SCALL DISPGOS
			300	RET
			301	;
	222D		302	RS232:
	222D B0111A		303	
			304	LDB AL,SERI_STAT
	2230 2B2B		305	
	2232 B0221B		306	SCALL TEKBYTE
	2235		307	LDB AH,EL
	2235 2B1E		308	
	2237 A0221E		309	YENIADTR:
	223A 2B19		310	SCALL IKIBYTE
	223C A02220		311	LD CX,EX
	223F 2B04		312	SCALL IKIBYTE
	2241 E01BF1		313	LD DX,EX
	2244 F0		314	SCALL TRANSFER
			315	DJNZ AH,YENIADTR
			316	RET
			317	;
	2245		317	TRANSFER:
	2245 641E20		318	
	2248 0520		319	ADD DX,CX
	224A		320	DEC DX
	224A 2B0E		321	ALYAZ:
	224C C62022		322	SCALL TEKBYTE
	224F 0520		323	STB EL,[DX]
	2251 E11EF6		324	DEC DX
	2254 F0		325	DJNZW CX,ALYAZ
			326	RET
			327	;
	2255 2B09		328	IKIBYTE:
	2257 B01A23		329	SCALL
				LDB EH,AL



ERR	LOC	OBJECT	LINE	SOURCE STATEMENT
	225A		330	TEKBYTE:
	225A	2804	331	SCALL OKU
	225C	B01A22	332	LDB EL,AL
	225F	F0	333	RET
			334	;
			335	;
	2260		336	OKU:
	2260	B0111A	337	LDB AL,SERI_STAT
	2263	361AFA	338	JBC AL,6,OKU
	2266	B0071A	339	LDB AL,SBUF_RX
	2269	F0	340	RET
			341	;
			342	;
	226A		343	BEEP:
	226A	A12C0122	344	LD EX,#012CH
	226E	B1801E	345	LDB CL,#BOH
	2271		346	KARE:
	2271	C70104E01E	347	STB CL,CPDRT
	2276	B1FF1C	348	LDB BL,#OFFH
	2279		349	BEKDEV:
	2279	E01CFD	350	DJNZ BL,BEKDEV
	227C	95801E	351	XORB CL,#BOH
	227F	E122EF	352	DJNZW EX,KARE
	2282	F0	353	RET
			354	;
			355	;
			356	;
	7F0A		357	CSEG AT 7F0AH
	7F0A		358	GERIDON:
	7F0A	F0	359	RET
			360	;
			361	;
	7F16		362	CSEG AT 7F16H
	7F16		363	RUNY:
	7F16	F0	364	RET
			365	;
			366	;
	7F1C		367	CSEG AT 7F1CH
	7F1C		368	RUNYY:
	7F1C	F0	369	RET
			370	;
			371	;
	3F00		372	CSEG AT 3F00H
	3F00		373	HEXDISPLAY:
	3F00	D7	374	DCB OD7H ;0
	3F01	11	375	DCB 11H ;1
	3F02	CB	376	DCB OCBH ;2
	3F03	5B	377	DCB 5BH ;3
	3F04	1D	378	DCB 1DH ;4
	3F05	5E	379	DCB 5EH ;5
	3F06	DE	380	DCB ODEH ;6
	3F07	13	381	DCB 13H ;7
	3F08	DF	382	DCB ODFH ;8
	3F09	5F	383	DCB 5FH ;9
	3F0A	9F	384	DCB 9FH ;A
	3F0B	DC	385	DCB ODCH ;B
	3F0C	C6	386	DCB OC6H ;C



ERR	LOC	OBJECT	LINE	SOURCE STATEMENT
	3F0D	D9	387	DCB 0D9H ;D
	3F0E	CE	388	DCB 0CEH ;E
	3F0F	8E	389	DCB 8EH ;F
			390	;
			391	;
	3F10		392	CSEG AT 3F10H
	3F10		393	TUSDEGERI:
	3F10	12	394	DCB 12H ;+
	3F11	14	395	DCB 14H ;-
	3F12	16	396	DCB 16H ;AD-DA
	3F13	1C	397	DCB 1CH ;RUN st
	3F14	10	398	DCB 10H ;RUN
	3F15	03	399	DCB 03H ;3
	3F16	07	400	DCB 07H ;7
	3F17	0B	401	DCB 0BH ;B
	3F18	0F	402	DCB 0FH ;F
	3F19	1E	403	DCB 1EH ;RUN yan
	3F1A	02	404	DCB 02H ;2
	3F1B	06	405	DCB 06H ;6
	3F1C	0A	406	DCB 0AH ;A
	3F1D	0E	407	DCB 0EH ;E
	3F1E	20	408	DCB 20H ;RUN 2yar
	3F1F	01	409	DCB 01H ;1
	3F20	05	410	DCB 05H ;5
	3F21	09	411	DCB 09H ;9
	3F22	0D	412	DCB 0DH ;D
	3F23	18	413	DCB 18H ;DELETE
	3F24	00	414	DCB 00H ;0
	3F25	04	415	DCB 04H ;4
	3F26	08	416	DCB 08H ;8
	3F27	0C	417	DCB 0CH ;C
	3F28	1A	418	DCB 1AH ;INSERT
			419	;
			420	;
	3F30		421	CSEG AT 3F30H
	3F30		422	RUNADRES:
	3F30	C921	423	DCW BASLA ;
				UN
	3F32	EA21	424	DCW ADARTTIR ;
	3F34	EF21	425	DCW ADAZALT ;
	3F36	F421	426	DCW AD_DAT ;
				D-DA
	3F38	FA21	427	DCW DEL ;
				ELETE
	3F3A	0E22	428	DCW INSERT ;
				NSERT
	3F3C	2D22	429	DCW RS232 ;
				UN ust
	3F3E	167F	430	DCW RUNY ;
				UN yan
	3F40	1C7F	431	DCW RUNYY ;
				UN 2yan
	3F42		432	END



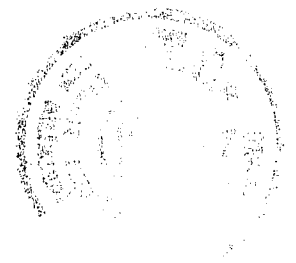
SYMBOL TABLE LISTING

N A M E	VALUE	ATTRIBUTES
AD_DAT.	21F4H	CODE ABS ENTRY
ADARTTIR.	21EAH	CODE ABS ENTRY
ADAZALT.	21EFH	CODE ABS ENTRY
ADDEG.	21E1H	CODE ABS ENTRY
ADRAM.	7FOAH	NULL ABS
AB.	0024H	REG ABS WORD
ABH.	0025H	REG ABS BYTE
AGL.	0024H	REG ABS
AH.	001BH	REG ABS BYTE
AL.	001AH	REG ABS BYTE
ALYAZ.	224AH	CODE ABS ENTRY
ANA.	2092H	CODE ABS ENTRY
APORT.	E000H	NULL ABS
AX.	001AH	REG ABS WORD
BASILITUS.	2198H	CODE ABS ENTRY
BASLA.	21C9H	CODE ABS ENTRY
BASTANTARA.	2159H	CODE ABS ENTRY
BAUD_RATE.	000EH	NULL ABS
BAUD4800H.	0080H	NULL ABS
BAUD4800L.	00CFH	NULL ABS
BEEP.	226AH	CODE ABS ENTRY
BEKDEV.	2279H	CODE ABS ENTRY
BEKLE.	2179H	CODE ABS ENTRY
BL.	001CH	REG ABS BYTE
BPORT.	E002H	NULL ABS
BSRR.	0026H	REG ABS BYTE
BX.	001CH	REG ABS WORD
CL.	001EH	REG ABS BYTE
COMMAND.	E006H	NULL ABS
CPORT.	E004H	NULL ABS
CX.	001EH	REG ABS WORD
DATADeg.	21D2H	CODE ABS ENTRY
DATASEG.	20E2H	CODE ABS ENTRY
DEGERI.	213BH	CODE ABS ENTRY
DEL.	21FAH	CODE ABS ENTRY
DELDEV.	21FCH	CODE ABS ENTRY
DH.	0021H	REG ABS BYTE
DISPBUF.	7F05H	NULL ABS
DISPGOS.	20C9H	CODE ABS ENTRY
DL.	0020H	REG ABS BYTE
DX.	0020H	REG ABS WORD
EH.	0023H	REG ABS BYTE
EL.	0022H	REG ABS BYTE
EX.	0022H	REG ABS WORD
FONKS.	21C6H	CODE ABS ENTRY
GERIDON.	7FOAH	CODE ABS ENTRY
GERTARAMA.	2145H	CODE ABS ENTRY
HEXDIS.	3F00H	NULL ABS
HEXDISPLAY.	3F00H	CODE ABS BYTE
HWINDOWO.	0000H	NULL ABS
IKIBYTE.	2255H	CODE ABS ENTRY
ILKSART.	209FH	CODE ABS ENTRY
INSDEV.	2213H	CODE ABS ENTRY



N A M E	VALUE	ATTRIBUTES
INSERT.	220EH	CODE ABS ENTRY
ISLEM	21B1H	CODE ABS ENTRY
KARE.	2271H	CODE ABS ENTRY
NOKTA	20FCH	CODE ABS ENTRY
NOKTAAD	20FFH	CODE ABS ENTRY
NOKTADA	210EH	CODE ABS ENTRY
NOKTAKOY.	211DH	CODE ABS ENTRY
NOKTASIL.	212CH	CODE ABS ENTRY
OKU	2260H	CODE ABS ENTRY
ONEMSIZ	21A3H	CODE ABS ENTRY
RS232	222DH	CODE ABS ENTRY
RUNADRES.	3F30H	CODE ABS WORD
RUNY.	7F16H	CODE ABS ENTRY
RUNYAN.	7F16H	NULL ABS
RUNYANYAN	7F1CH	NULL ABS
RUNYY	7F1CH	CODE ABS ENTRY
SARTLA.	0082H	NULL ABS
SBUF_RX	0007H	NULL ABS
SBUF_TX	0007H	NULL ABS
SEGSEKIL.	20F0H	CODE ABS ENTRY
SERI_CON.	0011H	NULL ABS
SERI_SART	0009H	NULL ABS
SERI_STAT	0011H	NULL ABS
SP.	0018H	NULL ABS WORD
TARAMA.	2169H	CODE ABS ENTRY
TEKBYTE	225AH	CODE ABS ENTRY
TRANSFER.	2245H	CODE ABS ENTRY
TUSABAS	2150H	CODE ABS ENTRY
TUSDEGERI	3F10H	CODE ABS BYTE
TUSFONKTAB.	3F30H	NULL ABS
TUSHEX.	21CCH	CODE ABS ENTRY
TUSUBIRAK	2148H	CODE ABS ENTRY
VWINDOWO.	0020H	NULL ABS
WSR	0014H	NULL ABS
YENIADTR.	2235H	CODE ABS ENTRY

ASSEMBLY COMPLETED, NO ERROR(S) FOUND.



```
DECLARE SUB oku (SONUC)
PRINT 60000 / 15.25: STOP
FOR I = 1 TO 50: COLOR I: PRINT "A"; I: NEXT I: STOP
CLS
OPEN "COM1:9600" FOR RANDOM AS #1
OUT &H3FB, &H3
OUT &H3FC, &H1

    oku SONUC
BEKLE:
    b = INP(&H3FD)
    PRINT b
    IF b MOD 2 = 1 THEN GOTO BEKLE

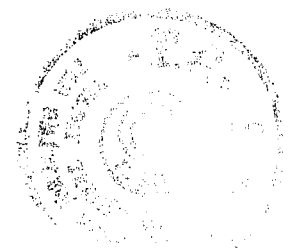
ALTEST:
    oku SONUC

    S1$ = HEX$(SONUC)
    topl = topl + 256 * SONUC
    oku SONUC
    A$ = S1$ + HEX$(SONUC) + " "
    topl = topl + SONUC
    DEVDAK = INT(60 / (topl * 1! * .000001))
    FOR k = 1 TO 50: NEXT k
    PRINT topl
    PRINT topl
    SCREEN 13
    COLOR 11
    LOCATE 10, 5: PRINT "DEVIR SAYISI(HEX)= "; HEX$(DEVDAK); " dev/d"
    PRINT
    COLOR 14
    LOCATE 12, 5: PRINT "DEVIR SAYISI(DEC)= "; DEVDAK; " dev/d"
    topl = 0
    PRINT "ff"
    GOTO ALTEST
    CLOSE #1

END

SUB oku (SONUC)

HAZIRDEGIL:
    b = INP(&H3FD)
    IF b MOD 2 = 0 THEN GOTO HAZIRDEGIL
    SONUC = INP(&H3FB)
    END SUB
```



```

    DECLARE SUB GONDER (bilgon)

    DIM BYTE(5036), A(5036)
    CLS

    OPEN "COM1:4800" FOR RANDOM AS #1

    OPEN "SAKLA.DAT" FOR OUTPUT AS #3

    K = 0
    OPEN "R", #2, "KTU.BIN", 1
    FIELD #2, 1 AS A$
    ADR = 8 * 16 ^ 3 + 10 * 16 + 6

    FOR I = &HFB TO &H3D6
    GET #2, I
    J = J + 1
    BYTE(J) = ASC(A$)
    NEXT I
    L = J

    FOR I = &H3D7 TO &H14A4
    GET #2, I
    J = J + 1
    BYTE(J) = ASC(A$)
    NEXT I
    CLOSE #2

    J = J + 1
    BYTE(J) = &H7F

    FOR I = 1 TO L STEP 2
    A = BYTE(I)
    BYTE(I) = BYTE(I + 1)
    BYTE(I + 1) = A
    NEXT I

    PRINT
    PRINT

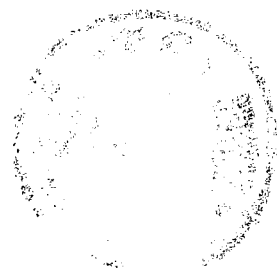
    M = 1
    FOR N = 1 TO 2
    FOR I = M TO L
    ADR = ADR + 2
    PRINT HEX$(ADR); "="; HEX$(BYTE(I)); " ";

    K = K + 1: A(K) = BYTE(I)

    PRINT #3, HEX$(ADR); "="; HEX$(BYTE(I)); " ";
    IF BYTE(I - 1) <> &HFF AND BYTE(I) <> &HFF AND BYTE(I - 1) = &H0 AND
YTE(I) = &H0 THEN
    C$ = INPUT$(1)
    END IF

    NEXT I
    PRINT
    PRINT
    M = L + 1

```



```

L = J
NEXT N
CLOSE #3

```

```

PRINT "J=": HEX$(J)
OUT &H3FB, &H17
FOR I = 1 TO 5
READ BILGI
GONDER BILGI
NEXT I

```

```

PLAY "L32 ML 02 E C"
CLS
LOCATE 10, 10
COLOR 11
PRINT "Fuzzy ilemciye Bilgi transferi....."

```

```

COLOR 12
LOCATE 12, 9
FOR I = 1 TO 32
PRINT "1";
NEXT I

```

```

COLOR 10
LOCATE 12, 9
BIRKARE = K / 33
TARAMA = BIRKARE
L = 0

```

```

FOR J = K TO 1 STEP -1
BILGI = A(J)
GONDER BILGI
BILGI = &H80
GONDER BILGI

IF L > TARAMA THEN
TARAMA = TARAMA + BIRKARE
PRINT "[ ";
END IF
L = L + 1

```

```

NEXT J

```

```

CLOSE #1
DATA &H01,&H27,&H55,&H80,&HAB
END

```

```

SUB GONDER (bilgon)
BEKLE:
BOSMU = INP(&H3FD)
IF BOSMU < &H40 THEN GOTO BEKLE
PRINT HEX$(bilgon); " ";
OUT &H3FB, bilgon
END SUB

```



```

DECLARE SUB byteword (i, byte(), adrveyauzun)
DECLARE SUB bytetopla (us, byte, basamaktop1)
DECLARE SUB adkars (adr, uzunluk, oanindis, byte(), kar)
DECLARE SUB diziol (baslindis, byte(), uzunluk, yk, gndr())

DECLARE SUB gonder (gndr(), uznlk(), say, sonuc)

DIM uznlk(100)

CLS
' $DYNAMIC

COLOR 9
PRINT "Gnderilecek program
INPUT ad$
CLS
ad$ = "c:\asm96\prog\" + ad$ + ".obj"
OPEN "R", #2, ad$, 1
FIELD #2, 1 AS a$

OPEN "COM1:4800" FOR RANDOM AS #1

OUT &H3FB, &H17

k = LOF(2)
REDIM byte(k), gndr(k)

FOR i = 1 TO k
GET #2, i
byte(i) = ASC(a$)
NEXT i
CLOSE #2

i = k - 11
k = i
devam:

byteword i, byte(), uzunluk

i = i - 2
byteword i, byte(), adres

i = i - 2
adkars adres, uzunluk, i, byte(), karar

IF karar = 1 THEN GOTO gonvekont

diziol k, byte(), uzunluk, yenidizin, gndr()
say = say + 1
uznlk(say) = uzunluk + 4

i = i - uzunluk - 12
k = i
uzunluk = 0: adres = 0

GOTO devam

```

gonvekont:

252

```
gonder gndr(), uznlk(), say, sonuc
IF sonuc = 0 THEN
BEEP
BEEP
PRINT "HATA VAR???"
ELSE
BEEP
COLOR 10
PRINT "GONDERME TAMAMLANDI!!!"
END IF
```

enson:

```
CLOSE #1
END
```

```
SUB adkars (adr, uzunluk, oanindis, byte(), kar)
```

```
  yeniind = oanindis - uzunluk - 5
```

```
  IF yeniind < 0 THEN
    kar = 1
    GOTO bitis
  END IF
```

```
  byteword yeniind, byte(), adres
  yeniind = yeniind - 3
```

```
  byteword yeniind, byte(), uzun
```

```
  IF adres <> adr OR uzun - 4 <> uzunluk THEN
    kar = 1
    GOTO bitis
  END IF
```

```
  kar = 0
```

bitis:

```
  END SUB
```

```
  SUB bytetopla (us, byte, basamaktop1)
    c = INT(byte / 16)
    d = byte - c * 16
    basamaktop1 = basamaktop1 + c * 16 ^ us + d * 16 ^ (us - 1)
    us = us - 2
  END SUB
```

```
  SUB byteword (i, byte(), adrveyauzun)
    us = 3
```

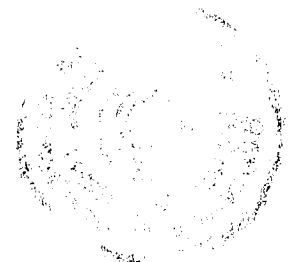
```
    bytetopla us, byte(i), adrveyauzun
    bytetopla us, byte(i - 1), adrveyauzun
```

```
  END SUB
```

```
  SUB diziol (baslindis, byte(), uzunluk, yk, gndr())
```

```
  FOR n = baslindis TO baslindis - uzunluk - 8 STEP -1
    IF n = baslindis - 4 THEN n = n - 5
```

```
    yk = yk + 1
```



```

gndr(yk) = byte(n)
NEXT n

END SUB

SUB gonder (gndr(), uznlk(), say, sonuc)
COLOR 11
OUT &H3F8, say

m = 1: l = 1
FOR j = 1 TO say
FOR i = 1 TO uznlk(j)
    k = k + 1
    IF gndr(k) <= 15 THEN
E$ = "0" + HEX$(gndr(k))
    ELSE
E$ = HEX$(gndr(k))
    END IF

LOCATE m, 1: PRINT E$;
    l = l + 3
    IF l > 75 THEN
        l = 1
        m = m + 1
        IF m > 23 THEN
            CLS
            m = 1
        END IF
    END IF
    OUT &H3F8, gndr(k)
    sumchk = sumchk + gndr(k)
NEXT i

bekle:
bosmu = INP(&H3FD)
IF bosmu < &H40 THEN GOTO bekle

NEXT i

b = INP(&H3F8)

IF b <> sumchk THEN
    sonuc = 0
END IF

NEXT j
sonuc = 1

son:
END SUB

```


' DCYUK.BAE

254

DECLARE SUB OKU (sonuc)

DIM hata(250), hatadeg(250), u(250)

' GOTO GRAFIK

'PRINT 6E+07 / 1024: STOP

KEY(1) ON

ON KEY(1) GOSUB dur

CLS

SCREEN 12

OPEN "COM1:9600" FOR RANDOM AS #1

OUT &H3FB, &H13

OUT &H3FC, &H1

' OKU sonuc

grafik:

CLS

SCREEN 12

COLOR 7

LOCATE 1, 1: PRINT "(Dev/dk)"

COLOR 3

' Y eksenı deęerleri

M = (4096 - 1024) / 6

I = 4096 + M

FOR K = 3 TO 25 STEP 3

LOCATE K, 3: PRINT USING "####"; I

I = I - M

NEXT K

COLOR 4

' X eksenı deęerleri

' I = 10.7

' FOR K = 12 TO 75 STEP 6

' LOCATE 28, K: PRINT USING "##.##"; I

' NEXT K

COLOR 7

'Pencere ve orijin alt kse

VIEW (1, 1)-(637, 430)

WINDOW (0, 0)-(637, 430)

I = 14



' Y ve X eksenini

255

```
LINE (50, 0)-(50, 415), I
LINE (35, 7)-(637, 7), I
```

' X eksenini ok isareti

```
LINE (632, 13)-(638, 7), I
LINE (632, 1)-(638, 7), I

LINE (632, 13)-(632, 2), I
```

' Y eksenini ok isareti

```
LINE (45, 410)-(50, 415), I
LINE (55, 410)-(50, 415), I

LINE (45, 410)-(55, 410), I

J = 2: K = 13
```

' Yatay izgaralar

```
FOR I = 55 TO 410 STEP 48
LINE (50, I)-(625, I), J, B, 257
LINE (48, I)-(51, I), K
NEXT I
```

' Dsey izgaralar

```
FOR I = 98 TO 637 STEP 48
LINE (I, 7)-(I, 390), J, B, 257
LINE (I, 5)-(I, 9), K
NEXT I
```

' NOKTASAY = 0
' yer = 11

```
IF IZ = 20 GOTO GEC
A$ = INPUT$(1)
```

GEC:

```
OUT &H3FB, &HA5
BEKL = 1
```

OKU sonuc

BEKLE:

```
B = INP(&H3FD)
IF B MOD 2 = 1 THEN GOTO BEKLE
```

DEVAM:

```
NOKTASAY = 0
yer = 11
s = 0
ztopl = 0
topl = 0
OKU sonuc

topl = topl + sonuc
```

```

OKU sonuc

topl = topl + 256 * sonuc

u(orsay) = topl
OND = topl

ztopl = ztopl + 600! / topl

topl = 0

orsay = 1
ONZ = 1
OND = (343 - 55) * (OND - 1024) / 3072

toporn = 120

DO

orsay = orsay + 1

OKU sonuc
topl = topl + sonuc

OKU sonuc

topl = topl + 256 * sonuc

BEKL = BEKL - 1

IF BEKL <> 0 GOTO YAZYOK
COLOR 9
LOCATE 2, 10: PRINT topl; " "; HEX$(topl)
BEKL = 20

YAZYOK:

ztopl = ztopl + 600! / topl

NOKTASAY = NOKTASAY + 1

IF NOKTASAY = 9 THEN

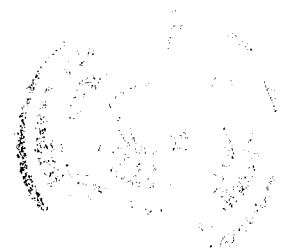
COLOR 9
LOCATE 28, yer: PRINT USING "##.##"; ztopl
yer = yer + 6
NOKTASAY = 0
s = s + 1

IF s = 12 THEN
IZ = 20
GOTO DEVAM
END IF
END IF

Z = orsay
D = topl

D = (343 - 55) * (D - 1024) / 3072

```



```
Z = INT(640 * Z / 120)

LINE (ONZ + 50, OND + 55)-(Z + 50, D + 55), 13

ONZ = Z
OND = D

top1 = 0

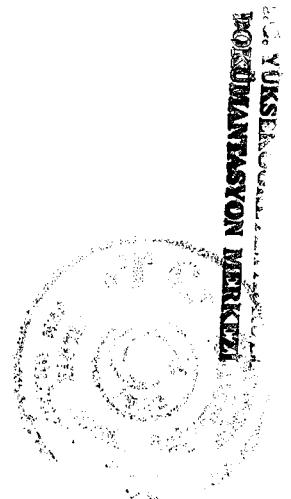
    LOOP WHILE toporn > orsay
dur:   CLOSE #1

COLOR 10

END

SUB OKU (sonuc)
HAZIRDEGIL:
    B = INP(&H3FD)
    IF B MOD 2 = 0 THEN GOTO HAZIRDEGIL
    sonuc = INP(&H3F8)

END SUB
```



ÖZGEÇMİŞ

Ö. Faruk ÖZGÜVEN Elektronik ve Haberleşme Yüksek Mühendisi
(1988 Yıldız Teknik Üniversitesi)

Doğum Tarihi ve Yeri 01. 11. 1963, Malatya

1981-1985 Yıldız Teknik Üniversitesi
Elektrik-Elektronik Fakültesi
Elektronik ve Haberleşme Mühendisliği Böl.

1978-1981 Malatya Ş.K.Ö Endüstri Meslek Lisesi
Elektronik Bölümü

1975-1978 Malatya Kubilay Ortaokulu

1970-1975 Malatya Hidayet İlkokulu

ALDIĞI GÖREVLER:

1989-1993 Araştırma Görevlisi
Yıldız Teknik Üniversitesi
Elektrik- Elektronik Fakültesi
Elektronik ve Haberleşme Mühendisliği
Devreler ve Sistemler Anabilim Dalı

1993-1994 Araştırma Görevlisi
Niğde Üniversitesi
Mühendislik Fakültesi
Elektronik Mühendisliği
Elektronik Anabilim Dalı

1994- Öğretim Görevlisi
İnönü Üniversitesi
Mühendislik Fakültesi
Elektrik-Elektronik Mühendisliği
Kontrol ve Kumanda Sistemleri Anabilim Dalı

Yabancı dil İngilizce

