SAÜ Fen Bilimleri Enstitüsü Dergisi
3.Cilt 1.Sayı (1999) 8-15

MİKRODENETLEYİCİLİ KISMİ İKLİMLENDİRME KONTROL SİSTEMİ II : YAZILIM

Erkan İmâl[1], Vedat Kıray[2], Kutay Erşan[2]

[1]Gazi Üniversitesi, Teknik Eğitim Fakültesi, Elektrik Eğitimi Bölümü, Teknikokullar, 06500, Ankara, TÜRKİYE.

[2]Dumlupınar Üniversitesi, Teknik Eğitim Fakültesi, Elektrik Eğitimi Bölümü, Simav, Kütahya, TÜRKİYE.

Özet- Bu makalede, bir oda içindeki havanın sıcaklık ve kalitesini denetleyen üç girişli ve dört çıkışlı bir sistemin denetiminde kullanılan bulanık mantık çıkarım algoritmasına ait bir akış diyagramı ve bu algoritmadan bir kısım program örnekleri verilmektedir. 87C51 mikrodenetleyicisi için *assembly kodu*nda yazılan bu algoritma, onaltı giriş ve onaltı kontrol çıkışına izin verdiğinden, genel olarak çok girişli ve çok çıkışlı sistemler için de kullanılabilecek yapıdadır. Giriş parametrelerine ait üyelik fonksiyonları üçgen ve yamuk şeklinde, kontrol çıkışları ise, birim üyelik fonksiyonu olarak tanımlanmakta ve her bir çıkış için en fazla sekiz üyelik fonksiyonu olabilmektedir.

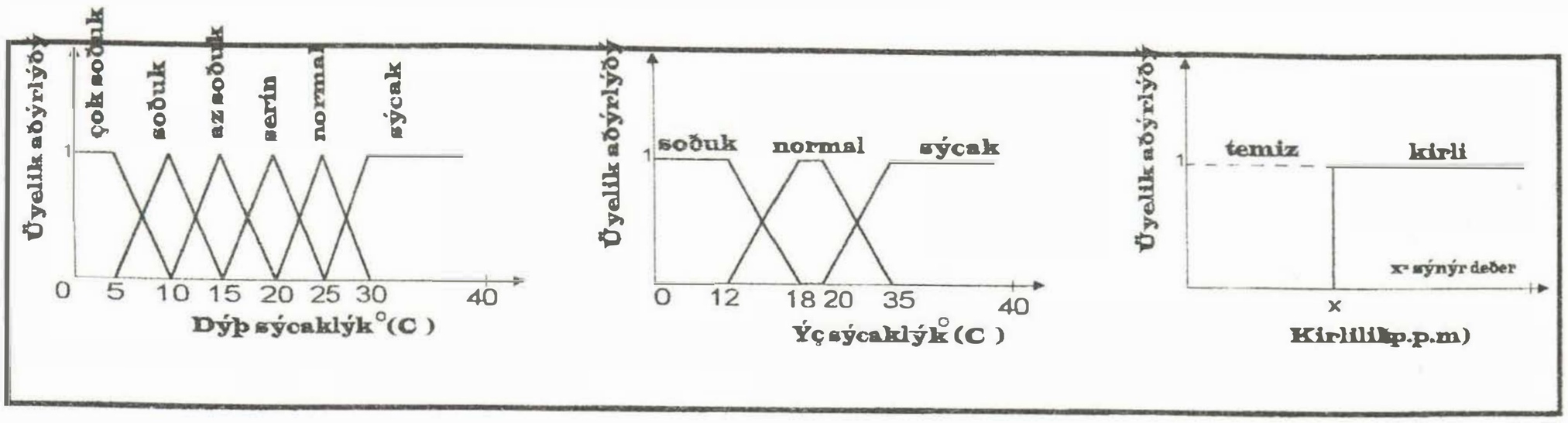
Abstract- In this article, a flowchart of fuzzy logic inference algorithm which is used in the control of a system controlling the air temperature and quality in a room with three inputs and four outputs, and some exemplary programs from it are given. Since this algorithm written in the *assembly code* for 87C51 microcontroller allows sixteen inputs and sixteen control outputs, generally, it has a structure so that may also be used for systems with multiple inputs and multiple outputs. Membership functions belonging to input parameters are defined in the form of triangle and trapezoid, as for the control outputs that are defined as the unity membership function, and eight membership functions are able to be for each output at most.

I. GİRİŞ DEĞERLERİNİN BULANIK FORMA DÖNÜŞTÜRÜLMESİ VE ÜYELİK FONKSİYONLARININ TANIMLANMASI

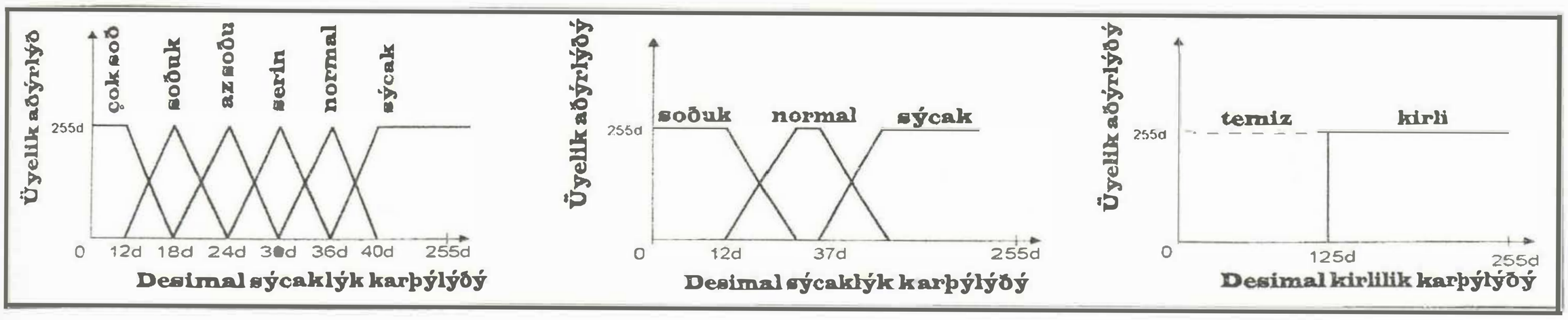
Bulanık mantık kontrolünde ilk basamak olarak, giriş bilgileri (gerçek sayılar) bulanık forma dönüştürülür. Algılayıcılardan gelen ve giriş bilgilerini oluşturan analog değerlerin, bulanık formdaki karşılıklarını tespit etmek için her bir girişe ait üyelik fonksiyonlarının tespit edilmesi gerekmektedir. Üyelik fonksiyonları, girişlere ait değer uzayını bölgelere ayırmada ve üyelik ağırlıklarını hesaplamada, hassasiyeti etkileyici önemli bir rol oynar. Soğuk, sıcak, yavaş, hızlı, düşük, yüksek gibi dilsel ifadelerle isimlendirilen bölgeler, sadece bir üyelik fonksiyonu veya birkaç üyelik fonksiyonunun birleşimiyle sınırlandırılabilir. Üyelik fonksiyonları, girişe ait her değerin 0 ile 1 arasında bir üyelik ağırlığı kazanmasını sağlarlar. Eğer algılayıcıdan gelen bir giriş değerinin üyelik ağırlığı 1 ise, o bölgenin tam üyesidir. Üyelik ağırlığı 1'den 0'a doğru azaldıkça o bölgenin daha az üyesi olmaktadır[1,2].

Bu çalışmada, üyelik fonksiyonlarına ait denklemlerin hepsi doğrusaldır. Çünkü lineer olmayan denklemler ile çalışmak sekiz bit'lik mikroişlemcilerde ancak zaman gecikmeli olarak yapılabilmektedir. Şekil 1'de, bu çalışma için oluşturulan, dış sıcaklık, iç sıcaklık ve hava kirlilik algılayıcılarına ait üyelik fonksiyonları görülmektedir. Bu fonksiyonların program ortamında oluşturulabilmesi için, şekil üzerindeki değerlerin desimal karşılıkları kullanılmıştır. Kullanılan mikrodenetleyici sekiz bit'lik olduğu için, gerçek değerlere karşılık gelen, 0-255 arasındaki desimal değerlerin kullanılması gerekmektedir. Şekil 1'deki gerçek değerlerin desimal karşılıkları, şekil 2'de gösterilmiştir.

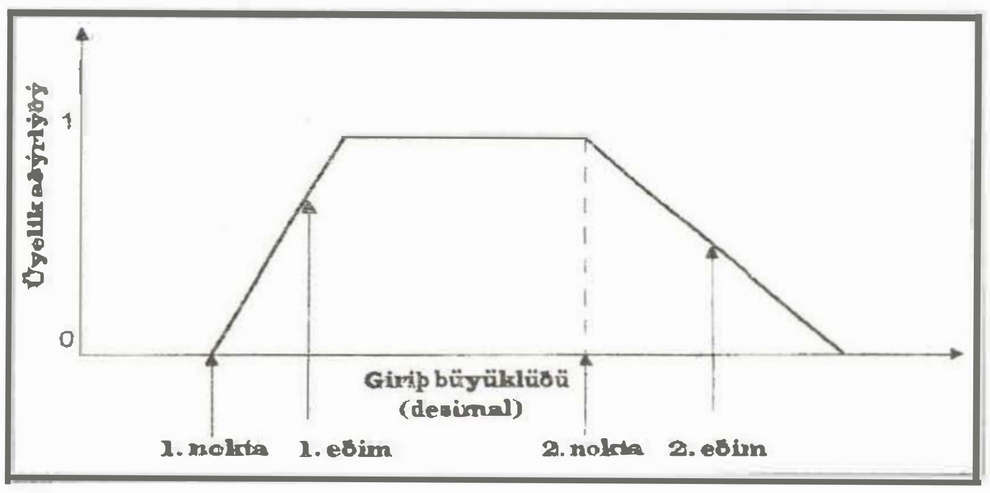
Program ortamında, her bir bölge tanımlanırken, dört değer kullanılmıştır. Bunlar, şekil 3'te görüldüğü gibi; 1. eğimin başlangıç noktası ve eğim oranı, 2. eğimin başlangıç noktası ve eğim oranı şeklindedir. Program'da, eğimi 1'den büyük değerler 1 kabul edilerek, yamuk şekilli bölgelerde, tavan genişliği için ayrıca bir değer girilmemektedir. Kontrol çıkışları için oluşturulan üyelik fonksiyonları, girişlere ait üyelik fonksiyonlarına göre daha farklıdır. Yani, birim fonksiyonu kullanılarak, çıkışta en fazla sekiz adet bölge oluşturulabilmektedir (şekil 4).



Şekil 1- Giriş bilgilerinin bulanık forma dönüştürülmesi



Şekil 2- Girişlere Ait Üyelik Fonksiyonları



Şekil 3- Dört nokta ile bölge tanımlaması

	%30	%40	%50	%60	%70	%80	%100
0	450	600	750	900	1050	1200W	1500W
							Ağırlık
0	0.1	0.2	0.3	0.4	0.6	0.8	1

Şekil 4- Kontrol çıkışlarına ait üyelik fonksiyonları (Isıtıcı)

1.1. Kural Tabanının Oluşturulması

Kural tabanında, girişlerin alabileceği değerlere göre, herbir kontrol çıkışının alacağı değerler tespit edilir. Bu çalışma için oluşturulan ve 36 kuraldan meydana gelen kural tabanı, tablo 1'de görülmektedir. Dış ısı, iç ısı ve hava kalitesi girişlerinin alabileceği değerlere bağlı olarak, ısıtıcı, fan motoru, iç hava kanalı adım motoru ve dış hava kanalı adım motorunun alacağı çalışma değerleri tespit edilmiştir.

Programda, kural tabanının takibinde *tablolama yöntemi* kullanılmıştır. Yani, kural tabanı tablo şeklinde oluşturulmuş ve her bir satır incelenerek sonuca doğru gidilmiştir.

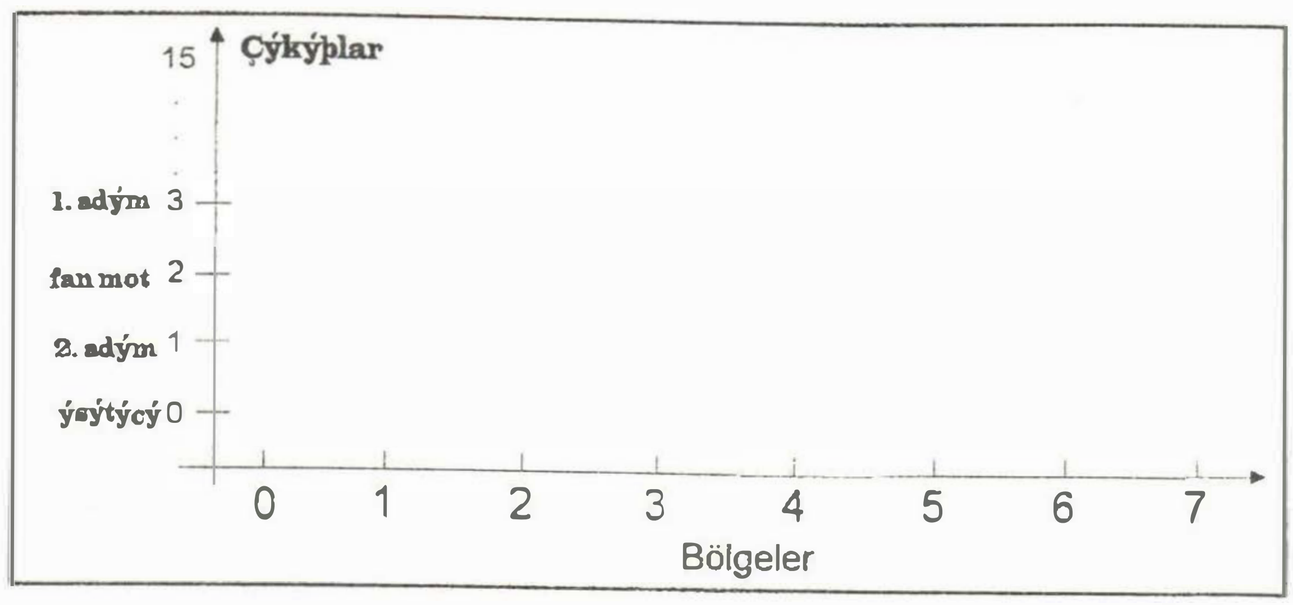
Tablo 1- Kural tabanının tablo olarak oluşturulması

No	Dış ısı	İç ısı	Hava kali.	Isıtıcı	Fan	İ. hava ka.	D. hava ka.
1	Ç. So	So	Te	1 (%100)	1	1	0
2	Ç. So	So	Ki	1 (%100)	1	0	1
3	Ç. So	No	Te	0	0	1	0
4	Ç. So	No	Ki	0.4 (%60)	1	0	1
5	Ç. So	Sı	Te	0	1	0	1
6	Ç. So	Sı	Ki	0	1	1	1
7	So	So	Te	1 (%100)	1	1	0
8	So	So	Ki	1 (%100)	1	0	1
9	So	No	Te	0	0	1	0
10	So	No	Ki	0.3 (%50)	1	0	1
11	So	Sı	Te	0	1	0	1
12	So	Sı	Ki	0	1	1	1
13	A. So	So	Te	1	1	1	0
14	A. So	So	Ki	0.8 (%80)	1	0	1
15	A. So	No	Te	0	0	1	0
16	A. So	No	Ki	0.2 (%40)	1	0	1
17	A. So	Sı	Te	0	1	0	1
18	A. So	Sı	Ki	0	1	1	1
19	Se	So	Te	0.6 (%70)	1	1	0
20	Se	So	Ki	0.8	1	0	1
21	Se	No	Te	0	0	1	0
22	Se	No	Ki	0	1	0	1
23	Se	Sı	Te	0	1	0	1
24	Se	Sı	Ki	0	1	1	1
25	No	So	Te	0.6 (%70)	1	1	0
26	No	So	Ki	0.6 (%70)	1	0	1
27	No	No	Te	0	0	1	0
28	No	No	Ki	0	1	0	1
29	No	Sı	Te	0	1	0	1
30	No	Sı	Ki	0	1	1	1
31	Sı	So	Te	0.4 (%60)	1	0	1
32	Sı	So	Ki	0.3 (%50)	1	0	1
33	Sı	No	Te	0	0	1	0
34	Sı	No	Ki	0	1	0	1
35	Sı	Sı	Te	0	1	0	1
36	Sı	Sı	Ki	0	1	1	1

I.2. Çıkarım İşlemi

Program ortamında, algılayıcılardan alınan analog bilgilerin, hangi bölgelerin hangi oranda üyesi olduğu tespit edilir. Bu işlem için, üyelik fonksiyonlarının sınır değerlerinden ve eğimlerinden yararlanılır. Sonuçta, her giriş için, 0 ile 1 arasında üyelik derecesine sahip değerler elde edilir.

Elde edilen bu değerler kural tablosuna yerleştirilir. Kural tablosunun program ortamında oluşturulması, bölüm II.4'te, ayrıca anlatılacaktır. Kural tablosunun takibinde, bütün kural satırları sırasıyla çağrılarak girişlere ait değerlerin minumumu alınır. Bu programında, kontrol çıkışları bir matris gibi düşünülmüştür ve girişlerin minunum değeri, bu matris yapıda uygun yerde saklanır. Örneğin, 1. kuralda, girişlere ait minumum değer, ısıtıcının, fan motorunun, 1. adım motorun 7. bölgesine ve 2. adım motorun 0. bölgesine yerleştirilir. Matris yapı doldurulurken, aynı yere denk gelen birden fazla değer varsa bunların maksimumu alınmaktadır (Şekil 5).



Şekil 5- Kontrol çıkışlarının ve her kontrol çıkışına ait bölgelerin matris olarak gösterimi

I.3. Berraklaştırma ünitesi

Berraklaştırma işleminde, bulanık değer olarak tespit edilip, matrisel yapıda saklanan bu değerler yardımıyla, her bir kontrol çıkışı için, bulanık sonuç kontrol değeri tespit edilmektedir. Kontrol çıkışlarının her bölgesine karşılık gelen birer gerçek sayı mevcuttur. Bu gerçek sayılar ve bulanık çıkarım sonucu tespit edilen bulanık değerler, denklem 1'e yerleştirilerek, sonuç kontrol çıkışı bulunmaktadır. Denlem 1'de, U sonuç kontrol çıkışını, *wi* kontrol çıkışlarının her bölgesine karşılık gelen gerçek sayıları, $\mu(wi)$ bulanık çıkarım sonucunda tespit edilen sonuç bulanık değerleri göstermektedir. Örneğin, ısıtıcıya ait kontrol çıkışının 4., 5. ve 6. bölgeleri için, 0'dan farklı bulanık değerler bulunmuş olsun. Bu değerler, aynı bölgelere denk gelen gerçek değerlerle, denklem 1'de ilgili yerlere konularak *sonuç kontrol çıkışı* tespit edilmiş olur [2,3].

$$U = \frac{\sum_{i=1}^{n} wi.\mu(wi)}{\sum_{i=1}^{n} \mu(wi)} \qquad (1)$$

II. ODA İÇİ HAVA KALİTE VE SICAKLIK KONTROL SİSTEMİNE AİT 87C51 ASSEMBLER YAZILIMI

Kontrol sisteminin yazılımına ait akış diyagramı şekil 6'da görülmektedir. Assembler programlama dilinde kodlanan yazılım, aşağıda alt bölümler halinde incelenecektir.

II.1. Bulanık Mantık Çıkarım Programı

Embedded Systems Programming dergisinde 1991 yılında M.J. Sibigtroth tarafından [4] yayınlanan ve M68HC11 mikrokontrolörü için yazılmış olan bulanık mantık çıkarım programı, aşağıda görüldüğü gibi, Intel 87C51 assembler programına çevrilerek programın içine dahil edilmiştir.

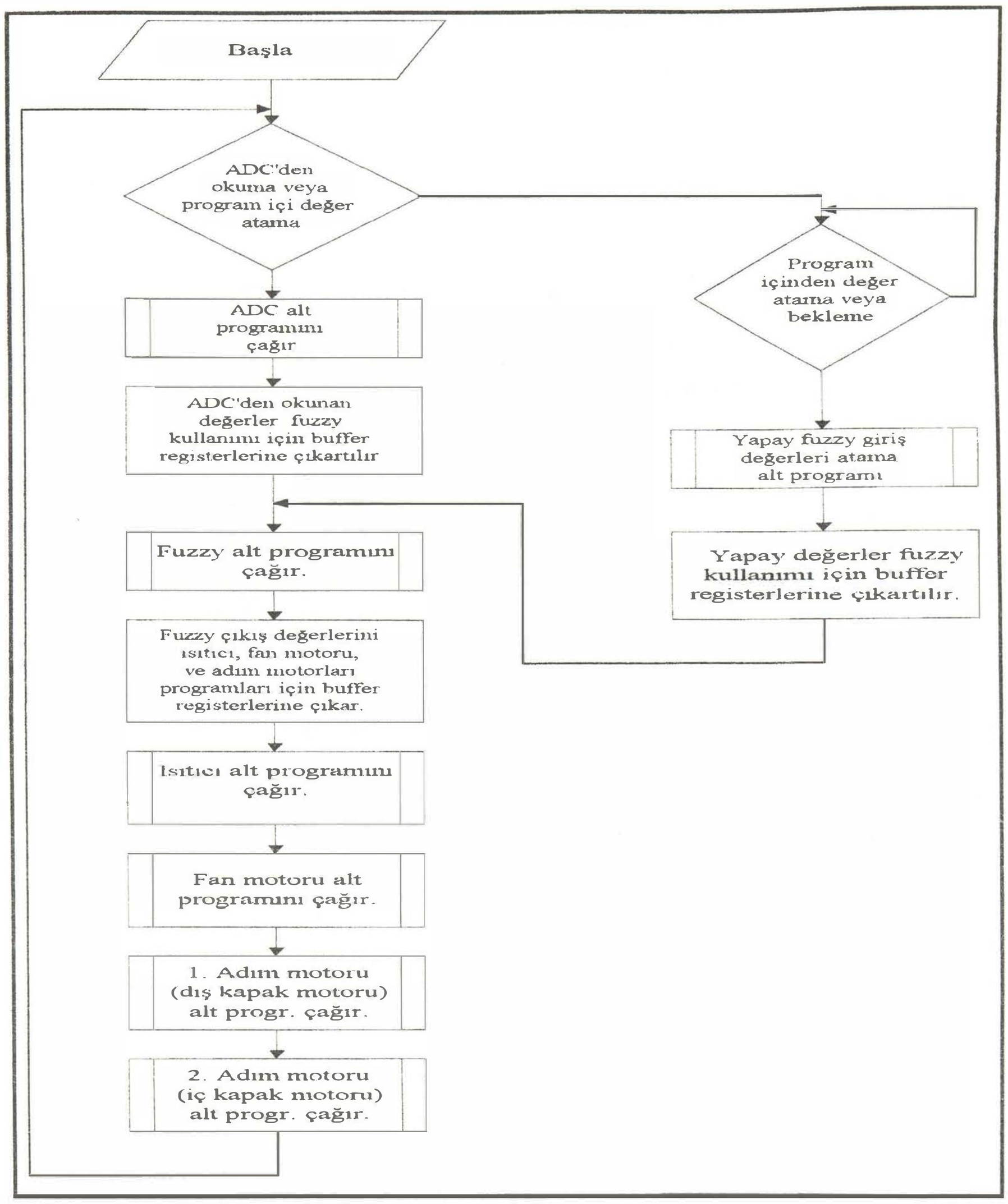
```
Fuzzy :

    mov   fuz_out,sp      ; Yığın değerini kaydet
    inc   fuz_out         ; Bir sonraki bölgeye işaretle
    mov   dptr,#fuz_outdef
    clr   a

    movc  a,@a+dptr       ; Çıkışlar değerlerini al
    rl    a               ; 8 ile çarp
    rl    a
    rl    a
    mov   r7,a            ; Fuzzy çıkış matris ölçüsü
    clr   a

    clr_out:
    push  acc             ; Matrisi 0 ile doldur.
    djnz  r7,clr_out
    mov   r3,a            ; Kural sayıcıyı 0'a kadar
yerleştir.

    rule_top: mov  if_min,#0ffh   ;max'a kadar IF
kuralını yerleştir.
    clr   skip            ; Atlama bit'ini başa al
    if_loop:  mov  a,r3
    inc   r3
    mov   dptr,#fuz_rules ; Kuralların başlaması
    movc  a,@a+dptr       ; Kural baytını al
```



Şekil 6- Algoritmanın akış diyagramı

```
    mov    r2,a                                    mov    dptr,#fuz_indef ;  Üyelik  tanımlamalarına
    jb     acc.7,then_loop ;  MSB ise, THEN döngüsüne    başla
git                                                mov    a,r7
    jb     skip,if_loop        ;   İF bölümüne atla    rl     a
                                                   mov    r7,a
    mov    r0,#buffer      ;   Fuzzy  girişlerine başla   movc   a,@a+dptr
    anl    a,#01111000b    ;   Girişleri  değerlerini al  mov    r6,a
    rr     a                                           mov    a,r7
    rr     a                                           inc    a
    rr     a                                           movc   a,@a+dptr
    mov    r7,a                                        mov    dpl,a
    add    a,r0                                        mov    dph,r6
    mov    r0,a                                        mov    a,r2            ;   Kural byte'nı al
    mov    a,@r0           ;   Analog giriş değerini al   anl    a,#00000111b    ;   Bölge seç
    mov    r4,a                                        rl     a
```

```
rl    a
add   a,dpl
mov   dpl,a
jnc   no_flow1
inc   dph
no_flow1: clr   a
movc  a,@a+dptr  ; Bölge tanımlamada 1. noktayı al
setb  c
subb  a,r4
cpl   a
jc    not_seg0      ;   Sonuç pozitif =>  kısım > 0
sjmp  jam_zero      ;   Sonuç negatif =>  kısım = 0

   not_seg0:    mov   b,a    ;   Farkı b'ye kaydet
   mov   a,#2
   movc  a,@a+dptr  ;  Bölge  tanımlamada  2.
noktayı al
   setb  c
   subb  a,r4
   cpl   a
   jc    is_seg2       ;   Sonuç pozitif =>  kısım = 2
   mov   a,#1          ;   Sonuç negatif =>  kısım = 1
   movc  a,@a+dptr     ;   1. eğimi al
   jz    jam_ff        ;   eğim  0  ise  FF  değerini
yerleştir.
   mul   ab            ;   Eğim * fark = Üyelik
derecesi
   jb    ov,jam_ff     ;   Sonuç  FF^den büyük ise FF
değerini koy
   sjmp  have_grade
   is_seg2: mov   b,a  ;   Farkı b'ye kaydet
   mov   a,#3
   movc  a,@a+dptr     ;   2. eğimi al
   jz    jam_zero      ;   eğim  0 ise 0 değerini koy
   mul   ab            ;   eğim  ile fark çarpımını
hesapla
   jb    ov,jam_zero   ;   sonuç  FF'den  büyük  ise  0
değerini koy
   sjmp  have_grade
   jam_zero:    clr   a
   have_grade:  cjne a,if_min,$+3 ; IF bölümünün min
değeri ile sonucu karşılaştır.
   jnc   not_lower     ;   Daha  büyük ise  devam et
   mov   if_min,a      ;   Daha küçük  ise  İF bölümün
minimumunu cevapla
   not_lower:   jnz   if_loop
   setb  skip          ;   Sonuç sıfır ise atla modunu
yerleştir.
   sjmp  if_loop       ;   Bir  sonraki   İF  bölümü  ile
devam et

   then_loop:   jb    skip,not_higher
   anl   a,#00111111b
   add   a,fuz_out
   mov   r0,a
   mov   a,@r0         ; İlgili  fuzzy çıkışını  al
   cjne  a,if_min,$+3  ;  Sonuç  ile  İF bölümünün
minumumunu karşılaştır.
   jnc   not_higher    ; Daha küçük  ise  devam et
   mov   @r0,if_min    ;  Daha  büyük  ise,  fuzzy  çıkış
cevabı
   not_higher:  mov   a,r3
   movc  a,@a+dptr     ; Bir sonraki kural  byte'nı al
   jb    acc.7,test_last ; MSB  byte'ı 1 ise, kural sonunu
test et
   ljmp  rule_top      ; İF bölümü ise işleme başla
   test_last:   inc   r3
   cjne  a,#0ffh,then_loop ; THEN bölümü ise atla ve
bir sonrakini kontrol et

   defuzzy:  mov   if_min,#0   ;  Numarası  belirtilen
döngüyü al
   cog_loop: clr   a           ; Durulamayı başlat
   mov   sigma_fuzzy,a
   mov   sigma_fuzzy+1,a
   mov   sigma_product,a
   mov   sigma_product+1,a
   mov   sigma_product+2,a
   sum_loop:    mov   a,fuz_out
   add   a,if_min
   mov   r0,a
   mov   a,@r0         ;  Bulanıklaştırılmış çıkışı al
   mov   r7,a
   add   a,sigma_fuzzy+1 ;  Çıkışların toplamına ekle
   mov   sigma_fuzzy+1,a
   jnc   no_ovfl2
   inc   sigma_fuzzy
   no_ovfl2:  mov   dptr,#fuz_outdef
   mov   a,if_min
   inc   a
   movc  a,@a+dptr   ; Birim fonksiyon değerini  al
   mov   b,a
   mov   a,r7          ; Bulanıklaştırılmış  çıkışı al
   mul   ab            ;   Çarpma
   add   a,sigma_product+2
   mov   sigma_product+2,a
   mov   a,b           ;   Çıkışların toplamına  ekle
   addc  a,sigma_product+1
   mov   sigma_product+1,a
   jnc   no_ovfl3
   inc   sigma_product
   no_ovfl3:    inc   if_min ;  Bir  sonraki  birim
fonksiyon değerini işaretle
   mov   a,if_min
   anl   a,#7          ;   Çıkarılmış bölge
   jnz   sum_loop      ;   Son  değil  ise,  toplamaya
devam

   mov   a,sigma_fuzzy
   jnz   divide
   mov   a,sigma_fuzzy+1
   jz    save_out  ; Sıfıra bölünüyor ise, sıfır degerini al
```

```
divide:    mov    r4,#0
    mov    r5,sigma_product
    mov    r6,sigma_product+1
    mov    r7,sigma_product+2
    lcall  ?C_LPUSH            ; Bölüneni al.
    mov    r4,#0
    mov    r5,#0
    mov    r6,sigma_fuzzy
    mov    r7,sigma_fuzzy+1    ; Böleni al
    lcall  ?C_ULDIV   ; Üretilen değerlerin toplamları /
çıkışların toplamları
    lcall  ?C_LPULL            ; r7' deki bölmeyi al
    mov    a,r7
save_out: mov    r2,a          ; Durulanmış çıkışları al
    mov    a,if_min
    rr     a
    rr     a
    rr     a                   ; Çıkış  numarasını (değer) al
    mov    sigma_fuzzy,a
    add    a,#buffer-1  ; Durulanmış  çıkış-1'in başlangıcı
    mov    r0,a
    mov    a,r2
    mov    @r0,a          ; Durulanmış çıkışı yerleştir.
    mov    dptr,#fuz_outdef
    clr    a
    movc   a,@a+dptr          ; Çıkış değerlerini  al
    cjne   a,sigma_fuzzy,cog_loop

    dec    fuz_out
    mov    sp,fuz_out
    ret

fuz_indef: dw     fuz_in0
    dw     fuz_in1
    dw     fuz_in2
    dw     fuz_in3
    dw     fuz_in4
    dw     fuz_in5
    dw     fuz_in6
    dw     fuz_in7
    dw     fuz_in8
    dw     fuz_in9
    dw     fuz_in10
    dw     fuz_in11
    dw     fuz_in12
    dw     fuz_in13
    dw     fuz_in14
    dw     fuz_in15
```

II.2- Girişlere Ait Üyelik Fonksiyonlarının Tanımlanması

Her bir girişe ait, üyelik fonksiyonları ile oluşturulan bölgeler, 4 değer ile tanımlanmıştır.

```
;    0. GİRİŞE AİT BÖLGELER

fuz_in0:   db     1,255,12,42 ; (0.bolge) 1-1.nokta, 2-
1.egim,3-2.nokta,4-2.egim
    db     12,42,18,42 ; (1. bölge )
    db     18,42,24,42 ; (2. bölge )
    db     24,42,30,42 ; (3. bölge )
    db     30,42,36,42 ; (4. bölge )
    db     36,42,254,255  ; (5. bölge )

;    1. GİRİŞE AİT BÖLGELER

fuz_in1:   db     1,255,12,9
    db     12,9,37,9
    db     37,9,254,255

;    2. GİRİŞE AİT BÖLGELER

fuz_in2:   db     1,255,128,255
           db     126,255,254,255
fuz_in3:
fuz_in4:
fuz_in5:
fuz_in6:
fuz_in7:
fuz_in8:
fuz_in9:
fuz_in10:
fuz_in11:
fuz_in12:
fuz_in13:
fuz_in14:
fuz_in15:
```

II.3. Çıkışlara Ait Bölgeler

Çıkışlara ait bölgeler birim üyelik fonksiyonları ile oluşturulmuştur. Bu sebeple her bölge tek değerle ifade edilmektedir.

```
fuz_outdef:    db     4   ; Çıkış adedi (4 tane)
    db     0,50,80,100,152,172,202,255   ; 1. çıkış işlem
değeri
    db     0,0,0,0,0,0,0,255          ; 2. çıkış işlem değeri
    db     0,0,0,0,0,0,0,255          ; 3. çıkış işlem değeri
    db     0,0,0,0,0,0,0,255          ; 4. cıkış işlem değeri
```

II.4. Kural Tablosu

Kural tablosu, program içinde oluşturulurken, bir kural satırındaki her giriş ve çıkış, 8 bit'lik binary bilgi ile ifade edilmiştir. 8 bit'lik sıralamada her bit'in anlamı aşağıda olduğu gibidi.

0. bit : Bit'in 0 olması, giriş bilgisi, 1 olması çıkış bilgisi olduğunu belirtir.

1-4. bit : 4 bit'lik hane, girişlere veya çıkışlara (giriş veya çıkış olduğu 0.bit'te belirtiliyordu) ait

sıralamayı, ikilik sayı sistemine göre belirtir ve giriş sayısının 16 ya kadar çıkmasına izin verir. Fakat bu uygulamada üç giriş kullanılmıştır.

5-7. bit : 3 bit'lik hane bölge sıralamasını belirtir. Bahsedilen bölgeler, 1-4 no'lu bitlerde belirtilen giriş veya çıkışlara aittir.

$: Bu işaretin yazılımda herhangi bir anlamı yoktur, sadece anlamı farklı bitlerin arasını ayırarak görsel kolaylık sağlamak için kullanılır.

11111111 : Kuralların bittiğini belirtir.

```
fuz_rules:
        ;1. Kural
db      0$0000$000b     ;       0. girişin 0.bölgesi ise ve
db      0$0001$000b     ;       1. girişin 0.bölgesi ise ve
db      0$0010$000b     ;       2. girişin 0. bölgesi ise
db      1$0000$100b     ;       0. çıkışın 4. bölgesi then
db      1$0001$111b     ;       1. çıkışın 7. bölgesi then
db      1$0010$000b     ;       2. çıkışın 0. bölgesi then
db      1$0011$111b     ;       3. çıkışın 7. bölgesi.

        ;2. Kural
db      0$0000$000b     ;       0. girişin 0.bölgesi ise ve
db      0$0001$001b     ;       1. girişin 1.bölgesi ise ve
db      0$0010$000b     ;       2. girişin 0. bölgesi ise
db      1$0000$000b     ;       0. çıkışın 6. bölgesi then
db      1$0001$000b     ;       1. çıkışın 2. bölgesi then
db      1$0010$000b     ;       2. çıkışın 0. bölgesi then
db      1$0011$111b     ;       3. çıkışın 7. bölgesi.

        ;3. Kural
db      0$0000$000b     ;       0. girişin 0.bölgesi ise ve
db      0$0001$010b     ;       1. girisin 2.bölgesi ise ve
db      0$0010$000b     ;       2. girişin  0.bölgesi ise
db      1$0000$000b     ;       0. çıkışın 0.bölgesi then
db      1$0001$111b     ;       1. çıkışın 7.bölgesi then
db      1$0010$111b     ;       2. çıkışın 7.bölgesi then
db      1$0011$000b     ;       3. çıkışın 0.bölgesi.

        ;35. Kural
db      0$0000$101b     ;       0. girişin 5.bölgesi ise ve
db      0$0001$001b     ;       1. girişin 1.bölgesi ise ve
db      0$0010$001b     ;       2. girişin 1.bölgesi ise
db      1$0000$000b     ;       0. çıkışın 0.bölgesi then
db      1$0001$111b     ;       1. çıkışın 7.bölgesi then
db      1$0010$111b     ;       2. çıkışın 7.bölgesi then
db      1$0011$000b     ;       3. çıkışın 0.bölgesi.

        ;36. Kural
db      0$0000$101b     ;       0. girişin 5.bölgesi ise ve
db      0$0001$010b     ;       1. girişin 2.bölgesi ise ve
db      0$0010$001b     ;       2. girişin 1.bölgesi ise
db      1$0000$000b     ;       0. cıkışın 0.bölgesi then
db      1$0001$111b     ;       1. cıkışın 7.bölgesi then
db      1$0010$111b     ;       2. cıkışın 7.bölgesi then
db      1$0011$111b     ;       3. cıkışın 7.bölgesi.

        ;Kural sonu
db      1$1111$111b
end
```

IV. SONUÇ VE TARTIŞMA

Bu yazılım 8 bit'lik 87C51 mikrodenetleyicisi için hazırlanmıştır. 8 bit'lik klasik mikroişlemcilerle yapılan, bulanık kontrol çalışmalarına bir örnek teşkil edebilir.

Bu bulanık mantık çıkarımında kullanılan kural tabanı, tablo şeklinde oluşturulup, kurallar sırasıyla takip edilmiştir. Giriş parametrelerinin sayısı çok fazla olmayan sistemler için bu yöntem kullanılabilir. Kural sayısının artması durumunda, programdaki döngü süresi uzayacağı için kullanılırlığı azalır. Kontrol çıkışlarına ait üyelik fonksiyonları, programın basitleştirilmesi ve hız kazanması için birim fonksiyon olarak seçilmiştir, fakat hassasiyeti yüksek sistemler için bu yöntem yetersiz gelebilir.

V. KAYNAKLAR

[1] İmal, E., Kıray, V., "**Oda İçi Hava Kalite ve Sıcaklık Sistemi**", G.Ü. Fen Bilimleri Enstitüsü Dergisi, Cilt: 11, No: 1, Ocak 1998.

[2] İmal, E., Güven, **M.E., Kıray, V., "Oda İçi Hava Kalite Ve Sıcaklık Sisteminin Kontroluna Bulanık Mantığın Uygulanması", G.Ü.Fen Bilimleri Enstitüsü Dergisi, 1998.**

[3] Kıray, V., "**Oda İçi Hava Sıcaklığı ve Kalitesinin Mikrodenetleyici ve Bulanık Mantıkla Kontrolü**", Gazi Ünviversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 1997.

[4] Sibigtroth, M. J., "**Creating Fuzzy Micros**", Embedded Systems Programming, December 1991.