

34758

YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

**RASTGELE SAYISAL İŞARETLERİN
ÖLÇÜMÜNÜ VE ÜRETİMİNİ SAĞLAYAN
DEVRELER VE SİSTEMLER**

34758

Elektronik Yük. Müh. Tuncay UZUN

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

Tez Savunma Tarihi : 13 Ocak 1994
Tez Danışmanı : Prof.Dr. Sezgin ALSAN (M.Ü.)
Jüri Üyeleri : Prof.Dr. Atilla ATAMAN (Y.T.Ü.)
: Prof.Dr. İhsan GÖK (M.Ü.)

İSTANBUL, Ocak 1994

**YÜKSEKÖĞRETİM KURULU
DOKÜMANTASYON MERKEZİ**

1000



İÇİNDEKİLER


ÖZET

SUMMARY

ÖNSÖZ

1. NÜKLEER ENSTRÜMANTASYONDA KARŞILAŞILAN İŞARET TÜRLERİ	1
1.1. İŞARETİN ANALOG ÖZELLİKLERİ	1
1.1.1 Genlik Ayırıcılar	2
1.2. İŞARETİN ZAMAN EKSENİ ÖZELLİKLERİ	2
1.2.1. Periyodik Olmayan Poisson Darbeler	2
2. NÜKLEER ENSTRÜMANTASYONDA KULLANILAN BAZI ELEKTRONİK DEVRELER	4
2.1. ORANDAN SAYISALA DÖNÜŞTÜRÜCÜ (RDC)	4
2.2. ZAMANDAN SAYISALA DÖNÜŞTÜRÜCÜLER (TDC)	7
2.2.1. 25ps Ayırıcılık, 12-bit, 64 kanal TDC	7
2.2.2. Ölü Zamansız Zamanında Sayısala Dönüştürücü Uygulaması İçin 1.2GHz Ötelemeli Kaydedici Tümdevresi	10
2.2.3. Zamandan Sayısala Dönüştürücülerin Kalibrasyonu	12
2.3. ZAMAN BELLEK HÜCRESİ (TMC)	14
3. BİLGİSAYAR İLE RASTGELE SAYISAL İŞARETLERİN ÖLÇÜMÜNÜ SAĞLAYAN SİSTEM	16
3.1. TASARLANAN ÖLÇÜM SİSTEMİNİN TANITIMI	16
3.2. IBM PC UYUMLU KİŞİSEL BİLGİSAYAR İLE GERÇEKLEŞTİRİLEN ÖLÇÜM SİSTEMİ	18
3.2.1. Ölçüm Sisteminin Donanımı	18
3.2.2. Ölçüm Sisteminin Yazılımı	27

4.	RASTGELE İŞARET ÜRETİMİ	45
4.1.	RASTGELE İŞARET ÜRETİMİNDE KULLANILAN DEVRELER	45
4.1.1.	Yalancı Rastgele İşaret Üreteçleri	48
4.2.	TASARLANARAK GERÇEKLEŞTİRİLEN YALANCI RASTGELE İŞARET ÜRETECİ	55
4.2.1.	Üreteç Sisteminin Donanımı	55
4.2.2.	Üreteç Sisteminin Yazılımı	66
5.	TASARLANAN ÜRETECİN ÖLÇÜMÜ VE DEĞERLENDİRİLMESİ	102
6.	SONUÇ REFERANSLAR ÖZGEÇMİŞ	116 118



Bu tezin oluşumu ve gerçekleşmesi sırasında, fikir ve eleştirileriyle katkısı bulunan değerli hocam ve tez yöneticim Sayın Prof.Dr.Sezgin ALSAN'a teşekkür ederim.

ÖZET

Bu tezde, rastgele sayısal işaretlerin ölçümü için elektronik ve bilgisayarlı ölçme devreleri tasarlanmış ve gerçekleştirilmiştir. Ayrıca bu ölçme işlemindeki işaretler analiz edilmiş ve bu işaretlerle eş özellikte yapay işaretleri üreten sayısal elektronik devreler tasarlanmış ve gerçekleştirilmiştir.

Gerçekleştirilen ölçüm sistemi, yazılım ile desteklenmiş programlanabilir sayıcı/zamanlayıcı donanımı içerir. Bu ölçüm sistemi, zaman aralığı rastgele değişen sayısal işaretleri, 200ns ile 20µs arasında değişen ayırıcılık ile ölçmeyi ve depolamayı sağlar.

Gerçekleştirilen diğer sistem, 100ns ile 10µs arasında değişen ayırıcılık ile zaman aralığı 100ns ile 0.1s arasında değişen yalancı rasgele sayısal işaretler üretir. Üreteç sisteminin karakteristiği, üreteç yazılımı ile istenen parametreler için tanımlanabilir.

Son bölümde, tasarlanan ve gerçekleştirilen yalancı rastgele sayısal işaret üreteç sisteminin çıkış işareti, tasarlanan ve gerçekleştirilen ölçüm sistemi ile ölçülmüştür. Ölçülen bilgiler, matematik ve istatistik analiz yazılımları ile analiz edilmiştir.

SUMMARY

In this thesis, electronic and computer circuits have been designed and realised for measuring random digital signals. Additionally, these random signals have been analysed in measuring process and a digital electronic circuit has been designed and realised to emulate these signals.

The realised measuring system mentioned above includes a programmable counter/timer hardware system supported with the software programmes. This measuring system measure and stores the signals whose time interval can be changed in the range of 200 ns to 20 μ s resolution.

The realised pseudo-random digital signal generator system generates the signals having 100 ns to 0.1 s time interval with range of 100 ns to 10 μ s resolution. The characteristic of the realised generator system can be defined for the required parameters using supported software.

Finally in the last part of the thesis, output signal of the designed and realised pseudo-random digital signal generator system is measured with the designed and realised measuring system. The measured data has also been analysed using mathematical and statistical software analysis programmes and the results are shown.

ÖNSÖZ

Elektronik ölçme işlemlerinde karşılaşılan bazı elektriksel işaretlerin periyodik olmaması nedeniyle klasik elektronikte kullanılan geleneksel ölçmeler, tam doğru yanıt vermemektedir. Örneğin, nükleer elektronikte isn zaman tabanlı bir frekansmetre ile yapılan ölçmelerde, bir ölçmenin diğeri ile aynı sayısal sonuç vermesi beklenmez. Bu bir yanılgı değildir. Bu şekildeki olayların değerlendirilebilmesi için Poisson dağılımının incelenmesi ve ölçmelerin bu özellikleri dikkate alınarak yapılması gerekmektedir.

Son yıllarda bilgisayarlı ölçme düzenlerinin gelişmesiyle, bu tip ölçü cihazları, yazılım ve donanım zenginliğinin sağladığı olanaklar nedeniyle bir çok alanda kullanılmaktadır. Bu nedenle gerçekleştirilen tasarımlarda bilgisayar yardımı ile, yazılım ve donanım geliştirmesi yapılacak, tasarım parametreleri ve ölçme işlemi gerçekleştirilecektir. Elde edilen bilgiler yine bilgisayarda oluşturulan hazır yazılım paketleri ve tasarlanan programlar ile incelenecektir.

Yapılan ölçümlerde hızlı ve doğru yanıt vermek için bazı seçenekler vardır. Bunlar hızlı ölçüm yapmak veya o ana kadar yapılan ölçümlerde alınan sonuçları değerlendirerek tahminde bulunmaktır. Bu tezde, eldeki teknolojik olanaklar çerçevesinde hızlı ölçüm hedef alınmakla birlikte, yapılan ölçümün doğruluğu bağımsız olarak incelenmiştir.

1. NÜKLEER ENSTRÜMANTASYONDA KARŞILAŞILAN İŞARET TÜRLERİ

Nükleer enstrümantasyon, nükleer tesislerde, nötron akısını ve radyasyon seviyelerini algılamak, ölçmek ve göstermek için kullanılan alet ve cihazlardır. Bu algılama ve ölçüm sonucunda, sistem kontrol edilebilir ve anormal durumlarda, çalışanlar uyarılabilir.

Nötron algılayıcıların çıkışı, elektriksel darbe veya akım işaretleri olup ortalama nötron akısı ile orantılıdır. Alçak güçlü araştırma reaktörlerinde, nötron akısı kontrol için tek parametre olabilir. Güç reaktörlerinde, termal güç en önemli parametredir. Bu nedenle yüksek güçte nötron akısına daha az önem verilir. Ancak başlangıç aşamasında nötron akısı ölçümü mutlaka gereklidir [1].

Algılayıcılardan gelen işaretler genlik ve zaman ekseninde değişen bilgiler taşırlar. Ölçüm düzeninin bu işaretleri ayrı ayrı değerlendirmesi gerekir. Nötronlara karşılık gelen büyük darbeleri, gama ışınları tarafından oluşturulan küçük darbelerden, genlik özellikleriyle ayrılarak değerlendirilebilir. Nötron akısı küçük ise darbelerin geliş hızı düşük, aksi halde yüksektir. Bu bilgiler zaman eksenini özellikleri ile ayrılarak değerlendirilebilir.

1.1. İŞARETİN ANALOG ÖZELLİKLERİ

Darbe modunda çalışabilen algılayıcıların çıkış darbelerinin genliklerinden gamma ayrımı yapılabilir.

Akım modu çalışmada elektriksel darbeler toplanarak bir doğru akım elde edilir.

1.1.1. Genlik Ayırıcılar

Genlik ayırıcı devreler, dedektör çıkışında oluşan darbelerin yüksekliklerini ayıran diskriminatör devreleridir [2]. Darbe yüksekliği ayırması, nötron işaretini toplam işarettten ayırmak için yapılır. Kuvvetlendirici sadece belirlenen bir yükseklikten büyük olan darbeleri kuvvetlendirir. Diskriminatör devresine bağlı olarak tetikleme devresinin çıkışında, değişik genlik ve genişlikte darbeler bulunur [3]. Sistemin daha sonraki devrelerinde, bilgiler analog veya sayısal olarak ifade edilerek incelenir [2].

1.2. İŞARETİN ZAMAN EKSENİ ÖZELLİKLERİ

Darbe modunda çalışabilen algılayıcılar, her nötron etkileşmesi ayrı bir darbe oluşturacak şekilde yeterince uzun bir zaman aralığı üretir. Bu modda çalışma sadece alçak seviyede nötron akısı ölçümlerinde kullanılır. Akı seviyesi darbe modunda çalışılmadığı yüksek bir seviyeye erişirse algılayıcı akım modunda çalışır. Çoğu algılayıcıların darbe modu çalışmasında, sayım hızı 10^5-10^7 sayım/sn'nin altındadır [3].

1.2.1. Periyodik Olmayan Poisson Darbeler

Darbe çıkışlı algılayıcılar, darbe arası süreleri Poisson dağılımı fonksiyonuna bağlı bir dağılım gösteren rastgele darbe arası sürelerle sahip darbe şeklinde çıkış işareti üretir.

Poisson olasılık ve dağılım fonksiyonunun özellikleri ve tanımı:

Ele alınan sistemin içinde rastgele oluşumlar bulunsun ve sistemde aşağıda belirtilen özellikler geçerli olsun:

- Sistem zaman ekseninin anlamlı dilimlerine yada yer veya belirli bir alanın alt kesimlerine göre incelenmektedir.

- Bu alt dilimlerde, biri diğeri izleyen olaylar bağımsızdır.

- Alt dilimlemede öyle bir aralık tanımlanabilir ki, bu aralıkta ilgilenilen olay bir defa ortaya çıkabilir, birden fazla ortaya çıkması mümkün değildir.

- Tanımlanan olayın bir defa ortaya çıkma olasılığı değişmemektedir.

Yukarıda belirtilen özelliklerin geçerli olduğu bir sistemde, x ilgilenilen olayın ortaya çıkma sayısı iken x'in olasılık fonksiyonu, $\lambda > 0$ olmak üzere, Dk.(1.1)'deki şekilde olur.

$$p(x) = \left\{ \begin{array}{l} \frac{e^{-\lambda} \lambda^x}{x!}, x=0,1,2,\dots \\ 0, \text{ diğ er durumlarda.} \end{array} \right\} \quad (1.1)$$

Bu fonksiyona "Poisson Dağılımı", x'e Poisson dağılmış rastlantı değişkeni denir [4,5]. Poisson dağılımının aritmetik ortalama ile varyansı aynı olup bu değer dağılımın parametresi olan λ 'ya eşittir. Poisson teoreminin en önemli uygulaması: zaman eksenindeki rastgele noktaların olasılığının incelenmesi gerektiğinde; poisson olasılık fonksiyonunun deney adedinin büyük, zaman aralığının periyot ile karşılaştırıldığında küçük olması halindeki gelişimidir [6].

2. NÜKLEER ENSTRÜMANTASYONDA KULLANILAN BAZI ELEKTRONİK DEVRELER

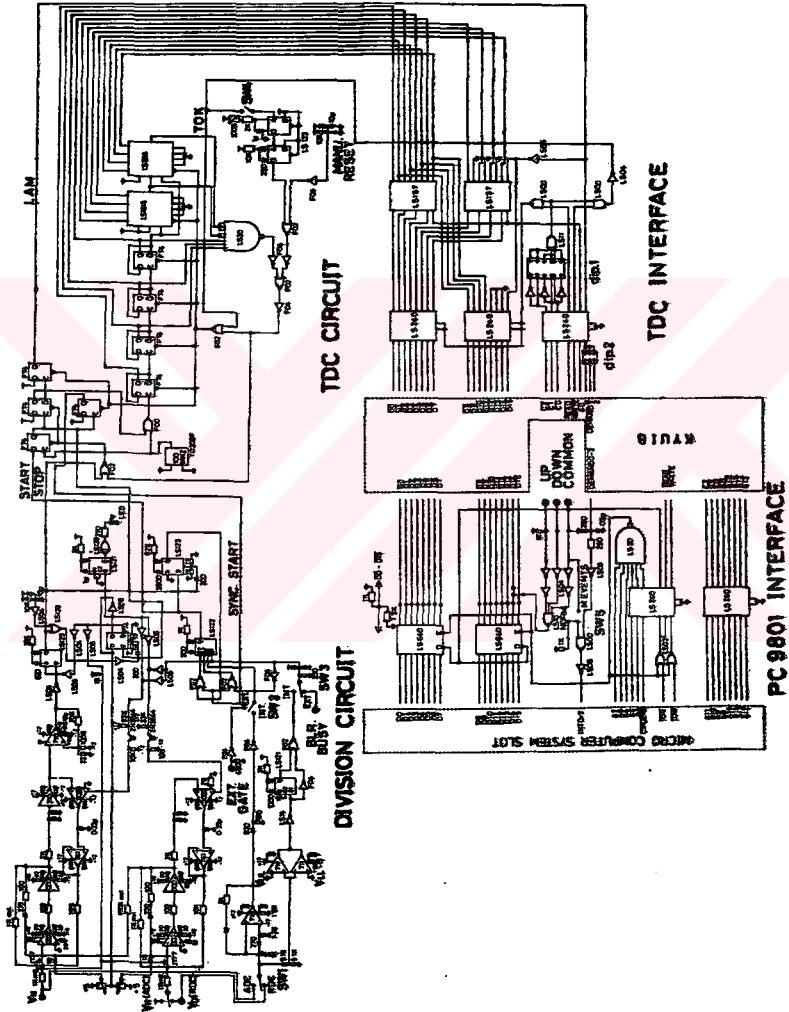
Yüksek enerji fiziği ve nükleer fizik deneylerinde zamandan sayısal işarete dönüştürücü (Time-to-Digital Converter) TDC modülü en gerekli bilgi toplama birimlerinden biridir [7]. İki giriş işaretinin darbe yüksekliklerinin oranını veren orandan sayısala dönüştürücüler (Ratio-to-Digital Converter) RDC devreleri, zamanı bellekte depolama (Time Memory Cell) TMC devreleri ve zamandan genliğe dönüştürücü (Time-to-Amplitude Converter) TAC devreleri gibi elektronik devreler nükleer enstrümantasyon ve deneylerde kullanılmaktadır. Bunlardan bazıları hakkında yapılan araştırmalar örneklerle aşağıda anlatılacaktır.

2.1. ORANDAN SAYISALA DÖNÜŞTÜRÜCÜ (RDC)

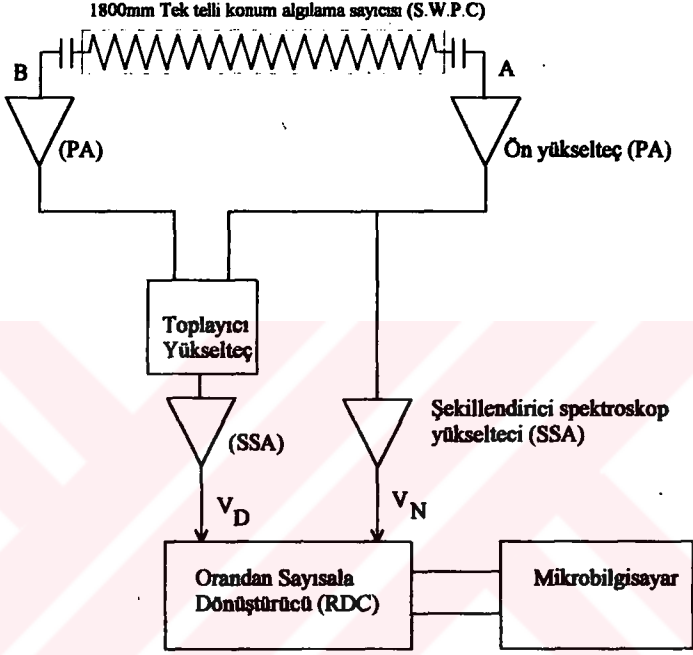
TDC modülleri orandan sayısala dönüştürücü (Ratio-to-Digital Converter) RDC modüllerinde de kullanılır.

RDC modülünde düzgün-tarama tipinde analog bölme devresi ve zamandan sayısala dönüştürücü TDC modülünden oluşan, iki giriş işaretinin darbe yükseklikleri arasındaki oranı sayısal olarak verir [8,9]. Düzgün tarama tipi analog bölme devresi Westphal [10] tarafından tasarlanmıştır. Bu devrede 1:20 oranında dinamik sınır içinde %0.1 doğruluk veren yüksek keskinlikte işlemsel yükselteçler kullanılmıştır.

Donanım veya yazılım tarafından kullanılan iki adet (VN ve VD girişleri) Analogdan sayısala dönüştürücü (ADC) ile bilgi sayısala çevrilmiştir [11,12].



Şekil 2.1 Oradan Sayısal Dönüştürücünün Elektriksel Devresi



Şekil 2.2 Orandan sayısal dönüştürücü için test deneyinin blok diyagramı.

RDC sistemi analog bölme devresi, TDC, bilgisayar arabirimi ve 8-bit mikrobilgisayarı içerir[9]. Bölme doğruluğu tam skalada %0.1'den daha iyi, 1:20 oranında dinamik sınır için düzgünlük %0.05'den daha iyidir. RDC aynı zamanda, sabit payda seviyesiyle 12-bit Wilkinson (düzgün-tarama tipi) ADC gibi çalışabilir.

START işareti diskriminatör devresi ve tepe dedektörü devresi tarafından üretilir. Bu işaret 100MHz saat üreticinin darbeleri ile uyumlu hale getirilir ve SYNC. START işareti üretilir. SYNC. START, TDC devresini tetikler

ve boşalma zamanını saymayı başlatır. Boşalmanın sonunda sıfır-geçiş karşılaştırıcısı STOP işaretini üretir. STOP işareti ile TDC saymayı durdurur. STOP işaretinden türetilen LAM işareti bilgisayara bilgi transferinin başlamasını bildirir. Bu işaret 16-bit mikrobilgisayara kesme (INTerrupt) INT işaretini üretir. Mikrobilgisayarın merkezi işlem birimi CPU bu işaret ile 16-bit bilgiyi alır (READ) ve yazma (WRITE) işareti (TOK şeklinde adlandırılmış) ile dönüştürücüyü (RDC) sıfırlar [8].

2.2. ZAMANDAN SAYISALA DÖNÜŞTÜRÜCÜLER

Yapılan araştırmalarda karşılaşılan zamandan sayısala dönüştürücülerde, teknolojik gelişmelere bağlı olarak ihtiyaç duyulan hız ve yetenekte çeşitli elektronik devrelerle karşılaşılmıştır. Bu devrelerden bazıları dikkate değer bulunarak, elektronik devreleri verilerek kısaca incelenmiştir.

2.2.1. 25ps Ayırıcılıklı, 12-Bit, 64 Kanal TDC

Tasarlanan 64 kanal zamandan sayısala dönüştürücü tam skala zaman sınırı 100ns, 12-Bit'in sağladığı ayırıcılık ile 25ps ayırıcılığa sahiptir [13]. Özellikleri aşağıda sıralandığı şekildedir:

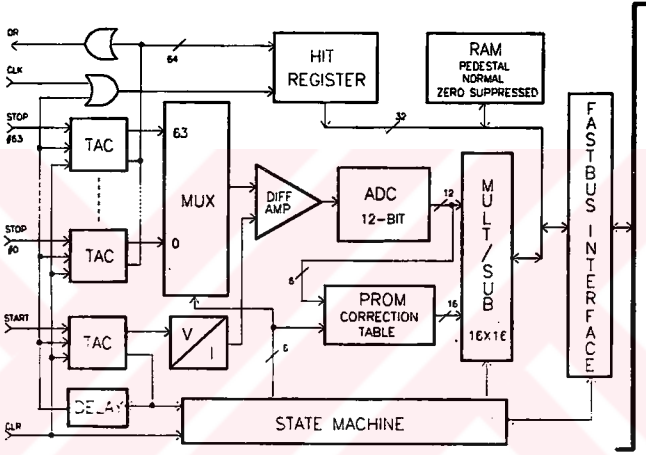
a. 64 zamandan genliğe dönüştürücü TAC, bir başarılı yaklaşım tipi (successive approximation) örneksel-sayısal dönüştürücünün girişine çoğullanmıştır.

b. Her TAC yüzey montajlı elemanlar ile ayrı, soketli, çok katlı küçük PC kart şeklinde yapılmıştır.

c. Tabloda bakma metodu ile gerçekleştirilmiş hata tesbiti ve doğrulama özelliği.

d. Vuru kaydedici ve temel çıkarma devreleri gerçekleştirilmiş.

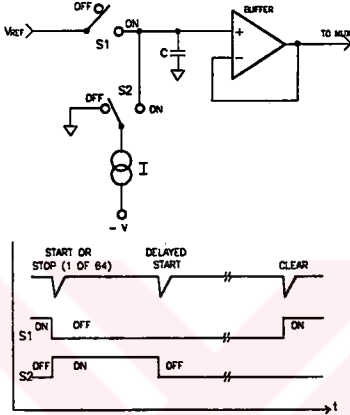
- e. Hızlı tetikleme amacıyla OR fonksiyonlu hızlı ECL mantık devreleri ile gerçekleştirilmiş STOP işareti.
- f. Vuru kaydediciye ihtiyaç duyulmaksızın sıfır bastırmalı blok veri transfer olanağı.



Şekil 2.3 Zamandan sayısal dönüştürücünün blok diyagramı.

Çalışma Prensibi:

Ortak START girişi ile 64 STOP girişi arasındaki zaman gecikmesi farksal olarak ölçülür. 65 işareti herbiri, ayrı bir dolma/boşalma tekniğini kullanan TAC devresine bağlanır [14]. STOP kanalı TAC çıkış gerilimleri çoğullanarak fark yükseltcecinin evirmeyen girişine uygulanır. START TAC gerilimi akıma dönüştürülerek doğrudan bu yükseltcecini eviren girişine uygulanır. STOP kanalındaki her ayrı çıkışı vuru kaydedicisine, dışarıdan veya içeriden bir saat ile bağlanır.



Şekil 2.4 Zaman dan genliğe dönüştürücünün basitleştirilmiş şeması ve zamanlama diyagramı.

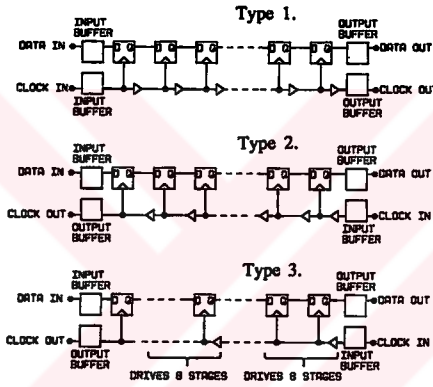
Bu arada START/STOP, analog çıkarmanın sonucu 12-Bit ADC ile sayısala dönüştürülür ve sayısal çarpma / çıkarmaya yönlendirilir [15]. Bunun sonucu RAM belleğin temel bölgesinde depolanır. Her ADC dönüşümünün üst bitleri kanal çoğullama bilgisi, aynı zamanda PROM'un adresindeki doğrulama faktörünü içerir. Bu faktörler 12-Bit ADC bilgisinin tamamı ile çarpılarak güncel yüklenmiş çıkarılan temel bilginin sonucunu verir. Doğrulanmış bilgi daima RAM belleğin 64 kelime normal bilgi bölgesinde depolanır. Taşmalar veya ölçülemeyecek kadar küçük olan bilgiler sonuç olarak RAM belleğin sıfır bastırma bölgesinde depolanır [16]. RAM belleğin temel, normal bilgi ve sıfır-bastırma bilgi bölgelerine bütün kapasite ile erişim sağlanabilir.

2.2.2. Ölü Zamansız Zamandan Sayısala Dönüştürücü Uygulaması İçin 1.2GHz Ötelemeli Kaydedici Tümdevresi

1.2GHz ötelemeli kaydedici, yüksek vuru hızı altında sürüklenme yatağının okunmasında, ölü zamansız TDC için geliştirilmiştir. Bellek derinliği, tetikleme kararı için yeterli gecikme zamanını elde etmek için 256 bitdir. Tümdevrenin ön üretimi, 1.2Watt'dan daha az güç harcaması ve 1.2GHz öteleme saatine kadar işlevini başarılı şekilde yapmayı sağlamıştır. Ölü zamansız TDC modülü tümdevre olarak geliştirme aşamasındadır [17].

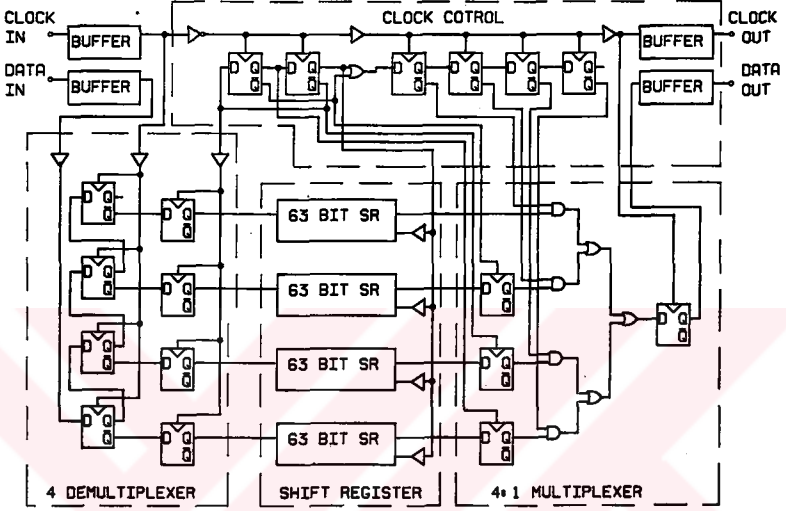
Yüksek enerji fizikçileri deneylerinde, sürüklenme yatağı için çok yüksek vuru hızı kapasitesine ihtiyaç duyarlar. İhtiyaç duyulan performansı elde etmek için yalnız yatağın değil, elektronik okumanın da iyileştirilmesi gereklidir. Elektronik okuma kısmında, sürüklenme zamanının ölü zamansız ölçülmesinin temeli, düşük gürültülü ve geniş bantlı yükselteç sistemlerinin karakteristiğinin ne kadar iyi olması gerekiyor ise o kadar iyi bir karakteristiğe sahip olmasıdır.[18] Çeşitli ölü zamansız TDC modülleri tasarlanmış ve çeşitli deneylerde kullanılmıştır [19]. Bunlardan bazıları ticari amaçla yapılmış olsada, 10-15cm/ms sürüklenme hızı için 200mm durum ayırıcılığından daha az olan ihtiyacı karşılamamaktadır [20]. Bunun için 1ns zaman ayırıcılığı gereklidir. Bu nedenle 1GHz öteleme saati ile çalışabilen yeni ötelemeli kaydedici test edilmiş ve tasarlanmıştır [17]. Statik Bellek (SRAM) veya klasik yolla ötelemeli kaydedici (SR) tasarlanabilir. Bellek derinliği aynı değerde olan SRAM tümdevresinin güç harcaması SR tümdevresinden daha iyidir. Çünkü SRAM tümdevrelerinde, veri transferinde transistörlerden küçük bir kısmı aktif

olur. Yüksek hızlı, düşük güçlü SRAM tümdevresi en az 5ns yazma çevrimine gerek duyar. Bu SR, en fazla 200MHz saat hızı ile çalışabilir demektir. 5 SR paralel bağlandığında, saat darbelerinin birbirine göre 1ns kaydırılmasıyla örnekleme frekansının 1ns'ye eşdeğer olması sağlanır.



Şekil 2.5 Ötelemeli kaydedici tasarımları.

Çeşitli SR tasarımları vardır. Şekil 2.5'de bunların basitleştirilmiş devre diyagramları bulunmaktadır. Birinci tip devrede saat darbesi ilk ikiliden uygulandığı için saat darbesinin gecikmesi önemli bir sorundur. Biriken farklar örnekleme periyodunu aşar ve yüksek saat hızlarında örneklenen bilgiden kayıp olur. İkinci tip devrede ilk devredeki sorundan kurtulmak için saat darbesi, son ikiliden uygulanmıştır. Burada ise saat darbesinin ilk ikiliye varışının sıcaklık ile değişimi önemlidir. Üçüncü, geliştirilmiş uyarlamada ise saat darbesinin sürüş yeteneği önemli bir problemdir.



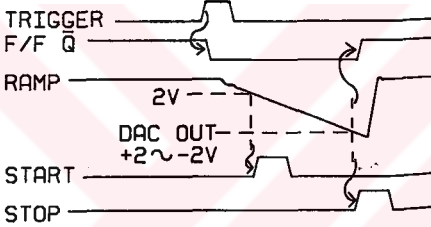
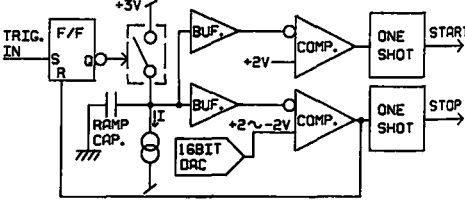
Şekil 2.6 Dördüncü tip tasarımı.

Şekil 2.6'da görülen dördüncü tip devrede, SR girişte ve çıkışta seriden paralele ve paralelden seriye dönüştürücü ile dörde bölünerek kullanılır. İçteki 64x4 SR saat frekansının dörtte birinde çalışır. Bu nedenle bu iç devrenin güç harcaması düşüktür [17].

2.2.3. Zamandan Sayısal Dönüştürücülerin Kalibrasyonu

Bazı (Time Of Fly) TOF sayıcıları TDC'lerde 100ns dinamik sınır içinde 50ps'den daha iyi keskinliğe gerek duyarlar [21]. Bu nedenle TDC devrelerinin kalibrasyonu için yüksek doğrulukta darbe üreticilerinin tasarımı gerekmektedir. Bunlardan biri START ve STOP çıkış darbeleri (13ps'den 200ns'ye kadar olan sınırlar) arasındaki zaman aralığının 15ps titremeden ($\pm 15ps = 30ps$ sapma genişliği)

daha düşük titreme ile düzgün, 14-bit doğruluğa sahip bir modüldür [7].



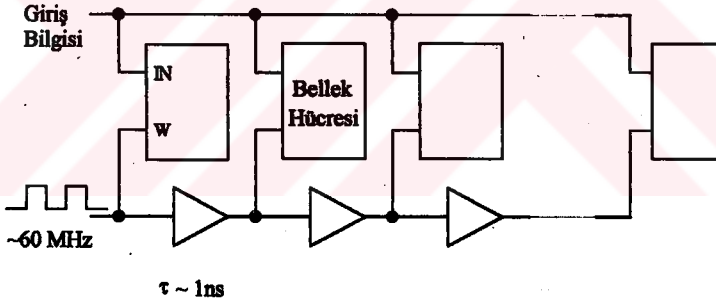
Şekil 2.7 Kalibrasyon devresinin blok diyagramı ve zamanlama çizelgesi.

Temel devre karşılaştırma devrelerine bağlanmış ayarlanabilir rampa üreticiyle yapılmıştır. "START" ve "STOP" kısımları Şekil 2.7'deki gibidir. Tetikleme işareti rampa kapasitesinin ilk boşalma evresini başlatan transistörü kapattıktan sonra ECL ikiliyi set eder. START kısmında karşılaştırıcının evirmeyen girişi START işareti için sabit çıkış zamanını +2V'da sabit tutar. STOP kısmında karşılaştırıcının evirmeyen girişine bağlanan 16-bit DAC'ın çıkışı (-2V'dan +2V'a) her tam skala çıkış kademesinde STOP işaretinin çıkış zamanlamasını belirler. Çıkış zaman aralığının sınırları, boşalma akımının genliği ile değişir. Rampa devresi ile karşılaştırıcı kısımları arasındaki tampon yükselteçler, çok düşük kutuplama akımına (FET girişli yükselteç), çok yüksek yükselme hızına (slew rate =

1000V/ms), geniş bir band genişliğine (100MHz'e kadar) ve 15-bit düzgünlüğe sahip olmalıdır.

2.3. ZAMAN BELLEK HÜCRESİ (Time Memory Cell TMC)

Yüksek enerji hızlandırma deneylerinde, yüksek hızlı ve yüksek yoğunluklu elektronik ölçmeye doğru tırmanılmaktadır. Bu çoklu vuru ölçüm modüllerinden birisi TDC modülüdür. Bunlar örnek olarak silisyum şerit dedektör ve tel yataklarında kullanılmaktadır. Genel olarak çoklu vuru TDC modülleri $1ns$ ayırıcılıktan daha iyisine ve ilk seviye tetikleyicisinde kullanıldığında, güçlendirme kabiliyetine ihtiyaç duyar [22]. Bir CMOS VLSI zaman bellek hücre (TMC) yongası oluşturulmuştur [23, 24, 25].

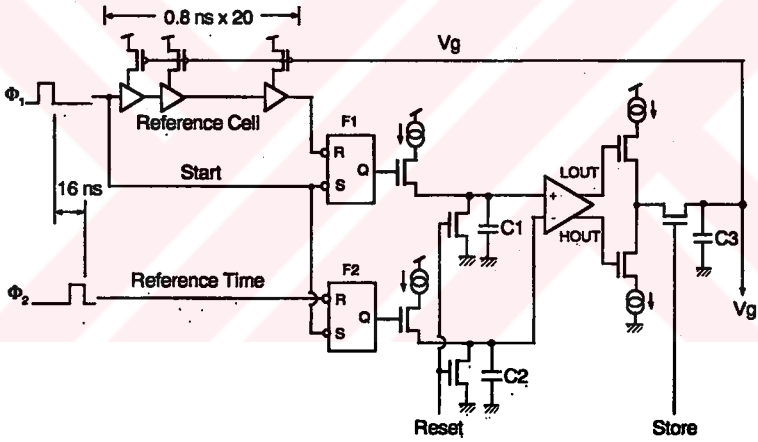


Şekil 2.8 Zaman bellek hücresinin temel yapısı.

TMC'nin ana elemanları, bellekler ve gecikme hatlarıdır (tampon serisi). Şimdiki tasarımlarla karşılaştırıldığında CMOS VLSI teknolojisi kullanılarak oluşturulan TMC tümdevresi çok düşük güç harcamasına sahiptir. TMC tümdevresinin yüksek doğruluğu, gecikme tamponları grubuna dışarıdan saat uygulanarak sağlanmıştır.

TMC modülünün temel yapısı, Şekil 2.8'de gösterildiği şekilde bağlanan bellek ve tampon birleşimidir.

Giren bilgi, her tamponun Δt aralığı ile geciktirilmiş yazma darbesiyle belleklere kaydedilir. Gecikme zamanının doğruluğu TMC modülünün çıkış anahtarıdır. Gecikme zamanının sabitliğini korumak için, gecikme hattındaki (tampon serisinde) toplam gecikme zamanını ölçen geri besleme devresi bunu dışarıdan uygulanan saat ile karşılaştırır (Şekil 2.9).



Şekil 2.9 Referans hücresinin ve geri besleme devresinin elektronik şeması.

3. BİLGİSAYAR İLE RASTGELE SAYISAL İŞARETLERİN ÖLÇÜMÜNÜ SAĞLAYAN SİSTEM

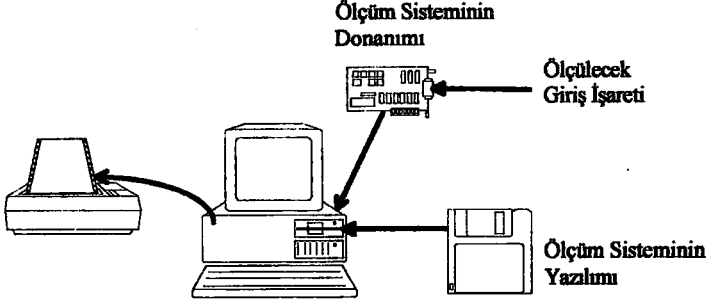
Periyodik olmayan veya çok uzun bir periyoda sahip olan yalancı rastgele sayısal işaretlerin frakans dağılımı değerlerinin ölçümü için, rastgele zaman aralığı ile gelen darbelerin darbe arası sürelerini ölçebilen bir elektronik düzen gereklidir.

Günümüzde bilgisayar hacimlerinin küçülmesi, güçlü ve taşınabilir ölçüm sistemlerinin gerçekleştirilmesine olanak sağlamıştır. Bu ölçüm sistemleri amaç ve ihtiyaca göre değişik özellik ve fonksiyonlara sahiptir. Bilgisayar kullanılan elektronik ölçüm sistemlerinin, istenen çözümü sağlaması için, yazılım desteğine ve belirli birkaç fonksiyona, gerekli özellikler ile sahip olması yeterlidir. Bununla birlikte amaca yönelik olarak tasarlanan ölçüm sistemleri, genel amaçlı olarak tasarlanan ölçüm sistemlerinden daha iyi ve hızlı sonuç vermektedir. Bu türden ölçüm sistemlerinin tasarlanabilmesi için, ölçülecek değerlerin karakteristik özelliklerinin çok iyi bilinmesi gerekir.

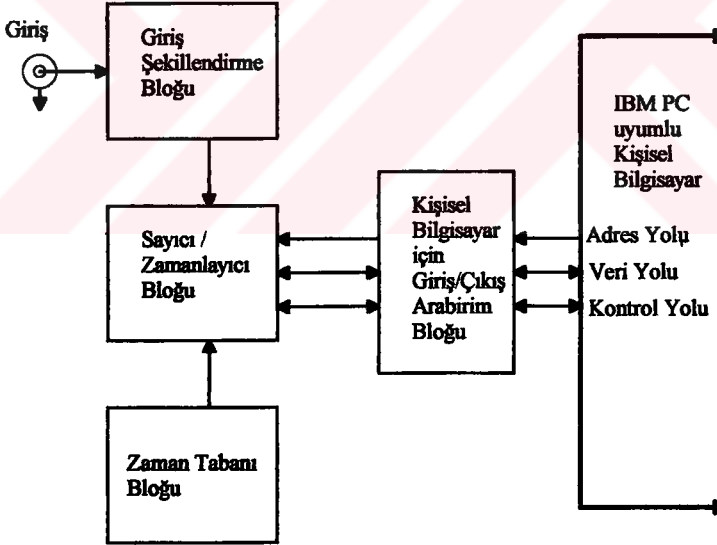
3.1. TASARLANAN ÖLÇÜM SİSTEMİNİN TANITIMI

Tasarlanarak gerçekleştirilen ölçüm sisteminin çalışma prensibi, yazılım ile desteklenmiş sayıcı/zamanlayıcı donanımına dayanır. Bu ölçüm sistemi darbe aralığı μs ile ns arasında değişen işaretlerin darbe arası sürelerini $200ns$ ile $20\mu s$ arasında değişen duyarlılıkla ölçerek kaydetmeyi sağlar.

Ölçüm sisteminin donanımı, IBM PC uyumlu kişisel bilgisayarlarda kullanılmak üzere bilgisayarın genişleme kanalına takılan bir bilgisayar kartından oluşmaktadır.



(a) Ölçüm sisteminin kurulması.



(b) Bilgisayar kartının blok diyagramı.

Şekil 3.1 Ölçüm sisteminin blok diyagramı.

Ölçüm sisteminin yazılımı ise, yine IBM PC uyumlu kişisel bilgisayarlarda kullanılan, gerçekleştirilen bilgisayar kartı için yazılmış bir ölçme ve veri aktarımı programı ile burada elde edilen verileri analiz eden hazır yazılımlardan oluşur.

3.2. IBM PC UYUMLU KİŞİSEL BİLGİSAYAR İLE GERÇEKLEŞTİRİLEN ÖLÇÜM SİSTEMİ

Ölçüm sisteminin donanım kısmı, IBM PC uyumlu kişisel bilgisayarlar için hazırlanmış bir kartın içindeki arabirimin desteklediği sayıcı/zamanlayıcı bloğundan, zaman tabanı üretici ve giriş ön yükseltecinden oluşmaktadır.

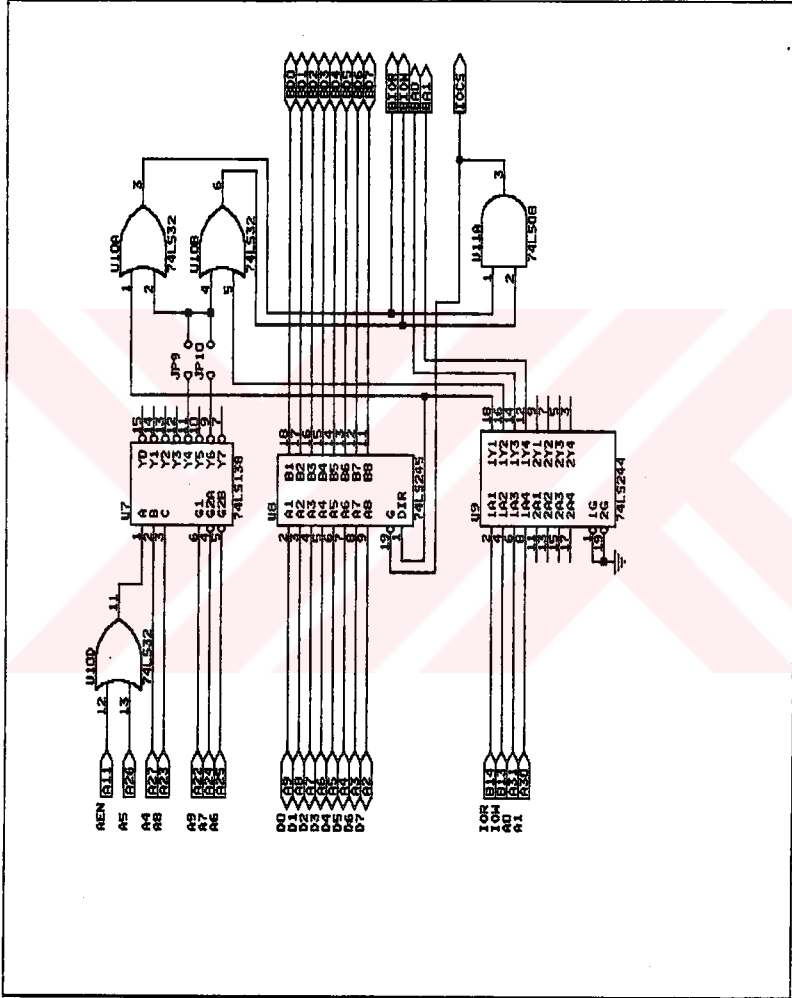
Ölçüm sisteminin yazılımı ise IBM PC kişisel bilgisayarının merkezi işlem birimi (CPU) 8086 makine dilinde yazılmış yakalama alt programı ile burada elde edilen verileri analiz eden hazır yazılım paketlerinden oluşur.

3.2.1. Ölçüm Sisteminin Donanımı

Giriş şekillendirme, sayıcı/zamanlayıcı saat ve zaman tabanı, IBM PC için giriş/çıkış arabirimi kısımlarından oluşan ölçüm sisteminin donanımının blok diyagramı Şekil 3.1'de görülmektedir.

Blok diyagramın ilk bloğunun girişi kişisel bilgisayarın (IBM PC) genişleme kanalında kullanılan bağlantı uçlarıdır [26,27,28]. Bu bağlantı uçları çift yüzlü 62 (2x31) noktadan oluşmaktadır. Bu blokta kişisel bilgisayarın (IBM PC) veri, adres ve denetim yolunun çevre birimlerince kullanılan bölümleri güçlendirilmiştir. Ayrıca seçime bağlı bir giriş/çıkış adres çözümlemesi yapılmıştır.

İkinci blokta, birinci bloktan alınan veri, adres, denetim yolu ve giriş/çıkış seçici kullanılarak kişisel



Şekil 3.2 Kişisel bilgisayar (IBM PC) için geliştirilmiş giriş/çıkış arabirimi.

bilgisayara (IBM PC), bir sayıcı/zamanlayıcı birimi bağlanmıştır. Bu birim zaman tabanı bloğundan referans zamanını ve giriş şekillendirme bloğundan, dışarıdan girilen ölçülecek işareti alır. Saat ve zaman bloğunda, sayıcı/zamanlayıcı bloğunun ölçüm için gerek duyduğu referans işareti üretilir.

Giriş şekillendirme bloğunda ise, girişe uygulanabilecek işaretin elektriksel özelliklerine ve bağlantı koşullarına uygunluk sağlanır.

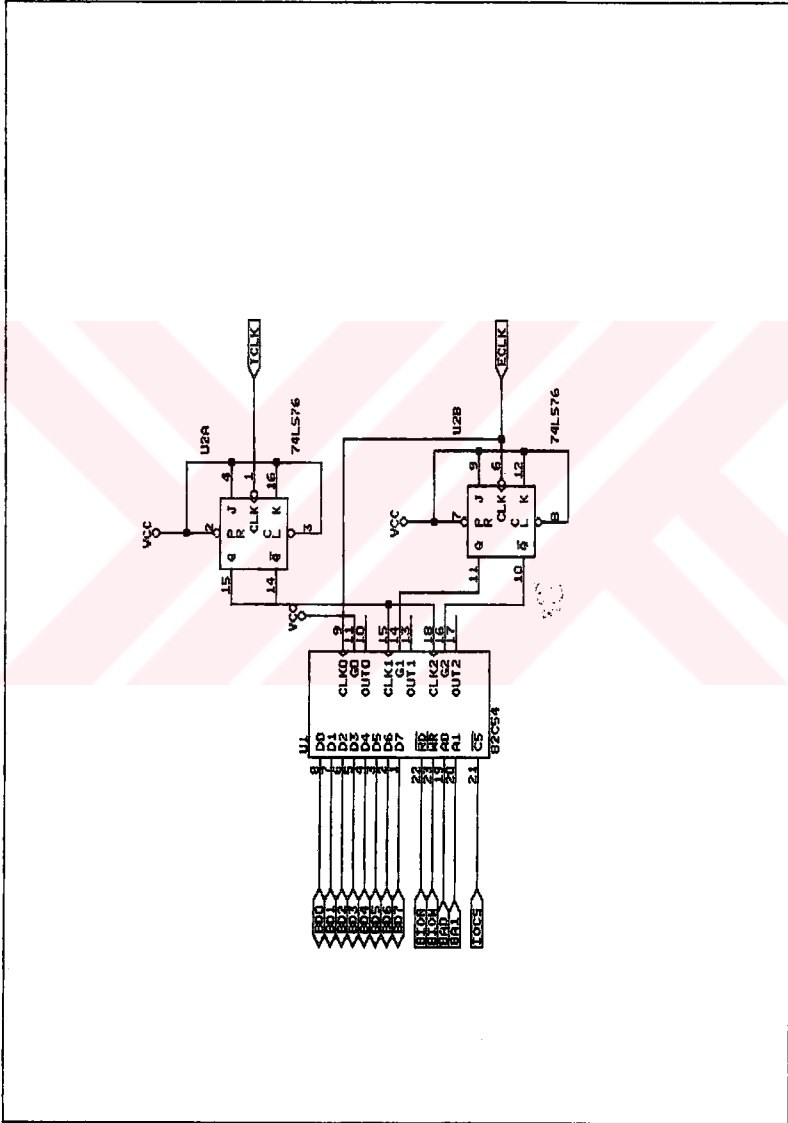
Kişisel bilgisayar (IBM PC) için geliştirilmiş giriş/çıkış arabiriminin ayrıntılı elektronik devre şeması Şekil 3.2'de verilmiştir. Burada görülen U8 tümdevresi, güvenli veri iletişimini sağlamak için, kişisel bilgisayarın (IBM PC) veri yolunu (D0-D7) güçlendirir. U9 tümdevresi, adres (A0,A1) ve denetim yolunu (IOR,IOW) güçlendirir. IOR, okuma, IOW ise yazma denetim ucudur [26, 27,28,29,30].

Tablo 3.1 Giriş/çıkış adresinin seçimi.

GİRİŞ/ÇIKIŞ ADRESİ	JP9	JP10
(030X)_{HEX}	KAPALI	AÇIK
(031X)_{HEX}	AÇIK	KAPALI

Kişisel bilgisayarda (IBM PC), 0300H-031FH arasındaki giriş/çıkış adres bölgesi kullanıcı için boş bırakılmıştır [26,27,28]. U7 adres çözümlene tümdevresi, 0300H-030FH(=030X) veya 0310H-031FH(=031X) adres alanlarını bağlantı dizisi (JP9,JP10) ile seçilmesini sağlar(Tablo 3.1).

U10 tümdevresi, U7 ile birlikte sayıcı/zamanlayıcı bloğu giriş/çıkış okuma ve yazma denetim uçlarını



Şekil 3.3 Sayıcı/zamanlayıcı bloğunun elektronik devresi.

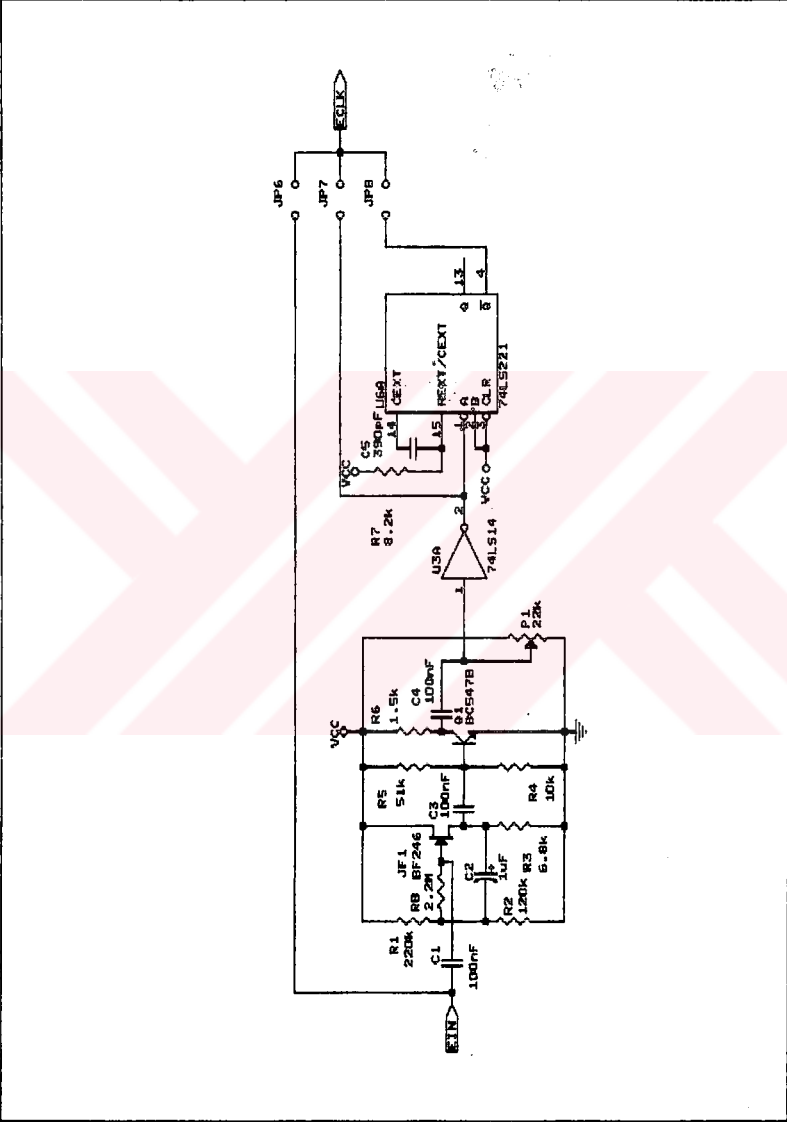
oluşturur. U11 ise yine sayıcı/zamanlayıcı bloğu için gerekli olan giriş/çıkış adresinin çözümünü sağlar.

Sayıcı/zamanlayıcı bloğunda, 10MHz'e kadar saat girişini kabul eden 82C54-2 programlanabilir sayıcı/zamanlayıcı tümdevresi (CTC, U1) ile JK flip-flop tümdevresi (U2) kullanılmıştır [31,32].

Bu bloğun ayrıntılı elektronik devresi Şekil 3.3'de görülmektedir. U1 tümdevresinin iç yapısı, çalışma şekli veri yolundan yazılım ile programlanan 16-Bit büyüklüğünde 3 sayıcı ve bunların giriş/çıkış uçlarından oluşmuştur.

Giriş şekillendirme bloğundan gelen ölçülecek darbeler (ECLK), U1 sayıcı/zamanlayıcı tümdevresinin 1 nolu sayıcısının saat girişine uygulanarak darbe arası süreleri ölçülen darbelerin sayısı sayılır. Bu işlemin amacı yazılım ile çalışma yapıldığı sırada gelen işaret olup olmadığının belirlenmesidir. Böyle bir durumda ölçülen işaret sayısı ile yazılım ile belirlenen ölçüm sayısı farklılık gösterecektir. Aradaki fark kadar sayıda darbe kaçırılmış, yanlış ölçülmüş olur. Yazılım, ölçüm işleminde yapılan hatayı bu şekilde tesbit eder. Bu işaret aynı zamanda U2B JK flip-flop tümdevresinin saat girişine uygulanarak, birbiri ardıda gelen darbelerin sayıcı/zamanlayıcı tümdevresinin 2 nolu ve 3 nolu sayıcılarında sırayla ölçülmesini, bu sayıcıların kapı (G2, G3) uçlarını denetleyerek düzenler.

Zaman tabanı bloğundan gelen referans saat işareti (TCLK) U2A JK flip-flop tümdevresi ile 2'ye bölünerek, darbe arası süreyi ölçen 1 ve 2 nolu sayıcılara saat (CLK1, CLK2) olarak verilir.



Şekil 3.4 Giriş ön yükselteç, schmitt tetikleme ve tek kararlı ikili devreleri.

Ölçülecek darbeler arası süre, belirlenen referans zamanıyla sayılarak Dk.(3.1)'deki gibi hesaplanarak bulunur.

$$\text{Darbe arası süre} = \text{Referans zamanı} \times \text{Sayıcı içeriği} \quad (3.1)$$

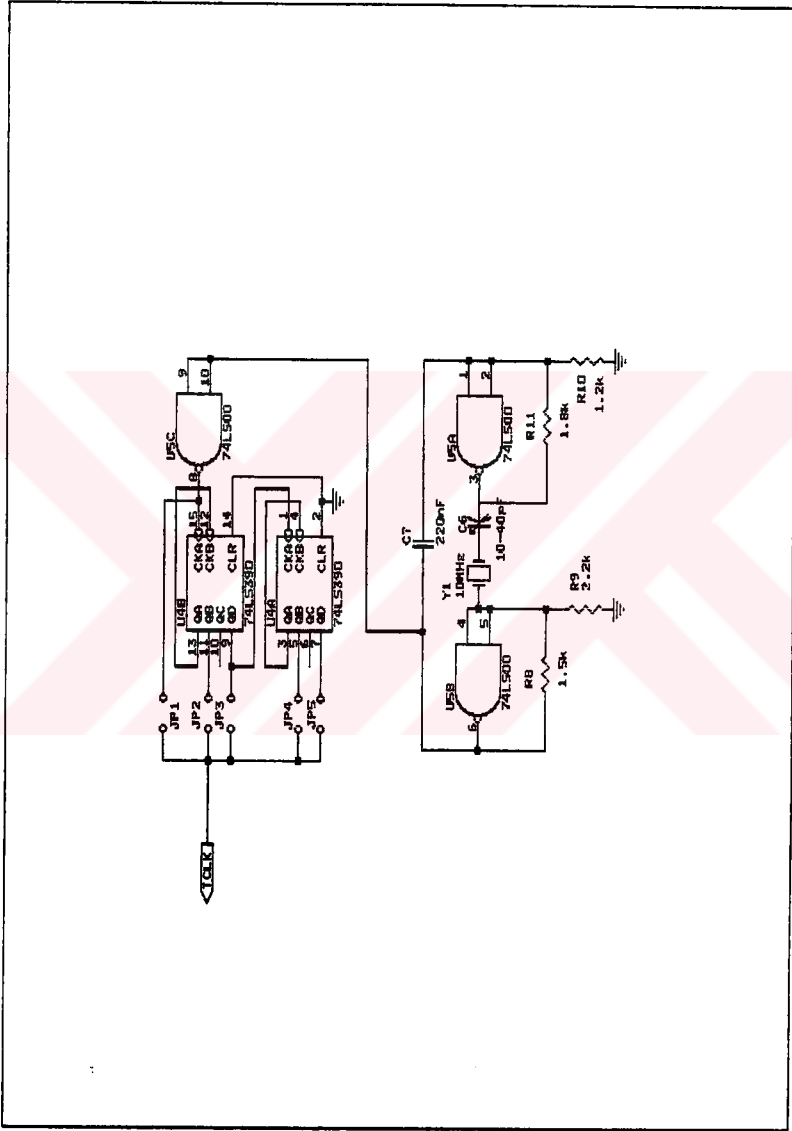
Şekil 3.4'de ise giriş şekillendirme bloğu görülmektedir. Ölçülecek darbe, giriş empadansı JFET kullanılarak (JF1) yükseltilmiş ($\approx 44M\Omega$) bir tampon devresine uygulanır. Sonra Q1 bipolar transistöründen oluşmuş ön yükseltece uygulanır. Bu yükselteç çıkışının DC seviyesi, P1 ayarlı direnci ile U3A schmitt tetikleyici tümdevresinin tetiklenme aralığına ayarlanabilir.

U3A'nın, U6A'nın çıkışındaki veya doğrudan girişe uygulanan TTL uyumlu lojik işaret seçime bağlı olarak çıkışa verilir. U6A tek kararlı ikilisi (MMV) ise yine seçime bağlı olarak, giriş darbe sürelerinin en alt düzeyinin çok küçük olması halinde, darbe yutucu ($\approx 3ms$) görevi yapar. Seçme işlemi Tablo 3.2'den yararlanılarak yapılabilir.

Tablo 3.2 Tek kararlı ikilinin veya doğrudan girişlerin seçilmesi.

SEÇİLEN	JP6	JP7	JP8
TTL uyumlu Sayısal Giriş	KAPALI	AÇIK	AÇIK
Analog doğrudan Giriş	AÇIK	KAPALI	AÇIK
MMV dolaylı Analog Giriş	AÇIK	AÇIK	KAPALI

Saat ve zaman tabanı bloğunun ayrıntılı elektronik devre şeması Şekil 3.5'de görülmektedir. Saat üretici, U5A ve U5B vedeğil kapıları ile 10MHZ kristalden oluşturulmuş bir kare dalga osilatördür. Osilatörün çıkışı U5C vedeğil kapısı ile güçlendirilerek zaman tabanı üreticine saat olarak verilir. Zaman tabanı üretici, kaskad bağlı 2 haneli

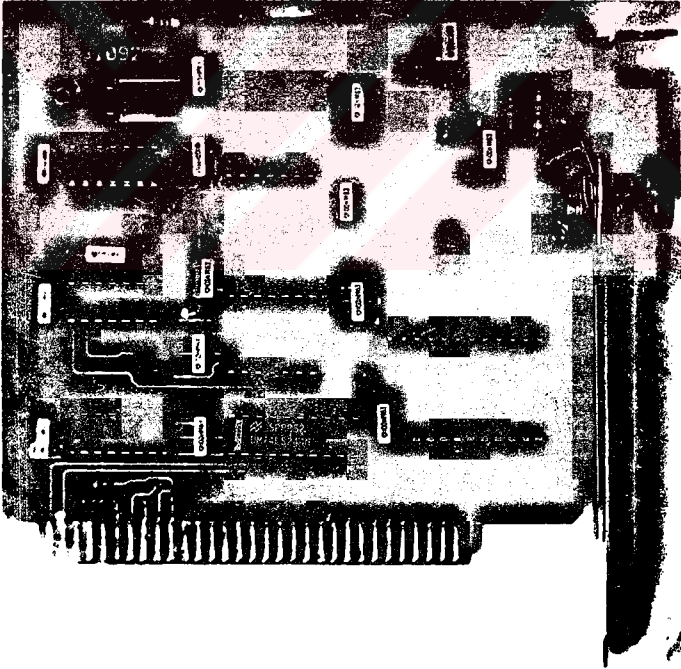


Şekil 3.5 Kristal osilatör ve zaman tabanı üretici.

BCD sayıcıdan oluşmaktadır. Zaman tabanı seçimi Tablo 3.3'de gösterilen biçimde bağlantı dizileri (JP1,JP2,JP3,JP4,JP5) ile yapılabilir.

Tablo 3.3 Zaman tabanı seçim tablosu.

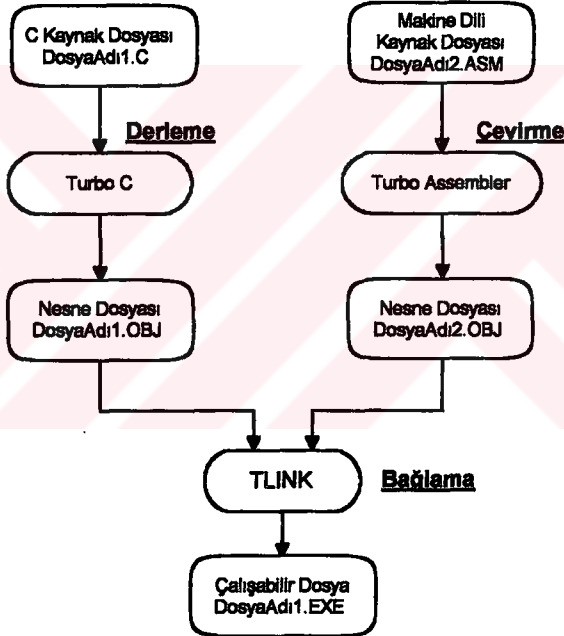
Zaman Tabanı	JP1	JP2	JP3	JP4	JP5
200ns	KAPALI	AÇIK	AÇIK	AÇIK	AÇIK
1µs	AÇIK	KAPALI	AÇIK	AÇIK	AÇIK
2µs	AÇIK	AÇIK	KAPALI	AÇIK	AÇIK
10µs	AÇIK	AÇIK	AÇIK	KAPALI	AÇIK
20µs	AÇIK	AÇIK	AÇIK	AÇIK	KAPALI



Şekil 3.6 Bilgisayar kartının görüntüsü.

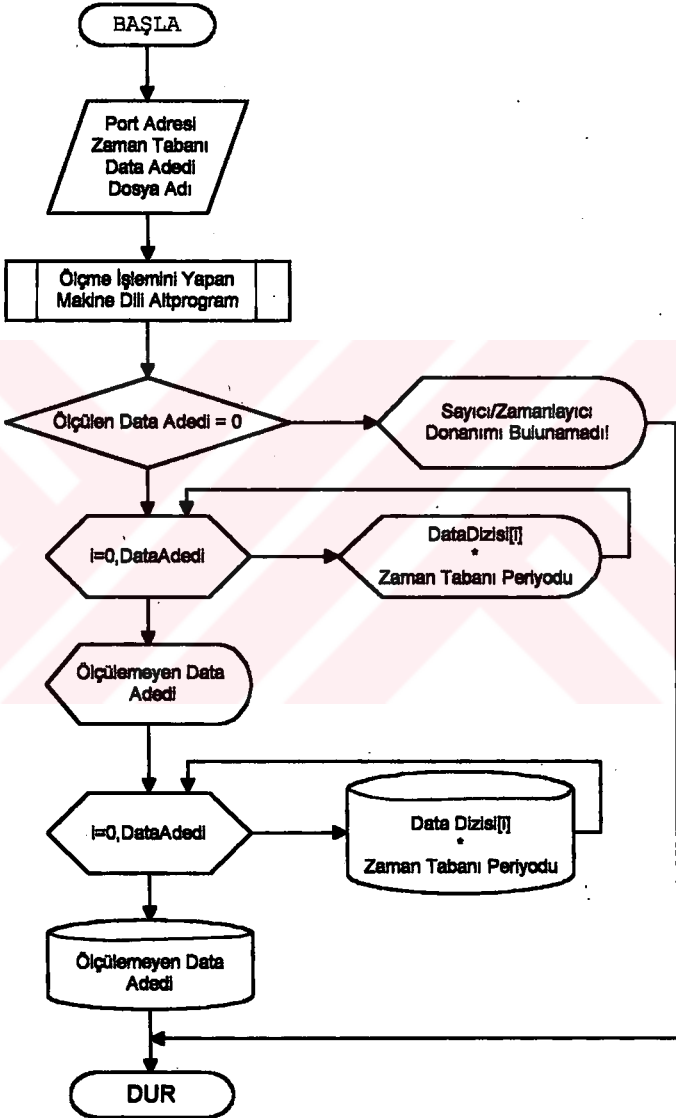
3.2.2. Ölçüm Sisteminin Yazılımı

Ölçüm sisteminin yazılımı, yapılacak ölçümün yazılım ve donanım özelliklerinin girildiği C dilinde yazılmış bir ana program ve bu programın içinden çağrılan ölçüm işlemini gerçekleştiren makine dili alt programdan meydana gelmiştir.



Şekil 3.7 Ölçüm yazılımının oluşturulması.

Ölçüm yazılımının IBM PC veya uyumlularında MSDOS disk işletim sistemi üzerinde bağımsız olarak çalışabilmesi için Şekil 3.7'de gösterilen akış diyagramı uyarınca C dilinde yazılan ana programın derlenmesi, makine dili programın çevrilmesi ve bağlayıcı ile her iki işlem sonucunda ortaya



Şekil 3.8 Ana programın akış diyagramı.

çıkan nesne tipli dosyaların birbirine bağlanması gerekir [39,40].

Ölçüm programının akış diyagramı Şekil 3.8'de gösterilmiştir. Programın başında, tasarlanan sayıcı/zamanlayıcı kartının kullanıldığı IBM PC veya uyumlu kişisel bilgisayarında boş olan bir giriş/çıkış port adresine şartlandırılması için port adresi girilir. Ölçülen darbe arası sürelerin gerçek değerlerinin hesaplanabilmesi için gerekli olan, ölçüm donanımında kullanılan zaman tabanı üreticinin seçilen sayım periyodu verilir. Daha sonra ölçülecek darbe arası sürelerin adedi ve bunların disk üzerinde kaydedileceği dosyanın adı girilir.

Ölçme işlemi, IBM PC veya uyumlu kişisel bilgisayar için yazılmış makine dili altprogram çağrılarak yapılır. Bu alt programın akış diyagramı ve açıklaması daha sonra yapılacaktır. Bu makine dili programın içinde ölçüm donanımının varlığı denetlenir. Eğer donanım yok veya hatalı ise ölçülen data adedi değişkenine 0 yerleştirilerek ana programa geri döndürülür. Bu durumda ekrana geçerli donanımın bulunmadığını belirten bir uyarı yazılarak programdan çıkılır. Donanımın doğrulanması halinde makine dili altprogramda alınan ölçümler DataDizisi adlı diziyeye yerleştirilir. DataDizisi dizinin son elemanı ölçülemeyen darbe adedini verir. Bu dizinin içeriği önce ekranda görüntülenir. Sonra başlangıçta verilen ad ile diske kaydedilir ve programdan çıkılır. Ana programın listesi Şekil 3.9'da görülmektedir. Program "Turbo C" kullanılarak yazılmış ve derlenmiştir [33,34,35,36].

Şekil 3.9 Ana programın listesi.

```
#include <stdio.h>

extern void Pdm(unsigned int far * DataPtr, unsigned int * DataAdedi, unsigned int
PortAdr);

unsigned int ztp,padr,i,DataS;
unsigned int DataDizisi[30004];
char dad[12];
FILE *datf;

main()
{
    clrscr();
    printf("*****\n");
    printf("**** Darbe Arası Süreleri Ölçerek Kaydeden Program ****\n");
    printf("*****\n");
    printf("\n");
    printf("**** Ölçüm İşleminde Kullanılan Donanımın Tanımları ****\n");
    printf("\n");
    printf("Giriş/Çıkış Port Adresi (30XH,31XH) = ");scanf("%X",&padr);
    printf("Zaman Tabanı Üreticinin Periyodu (us) = ");scanf("%d",&ztp);
    printf("\n");
    printf("**** Ölçülecek Değerler İçin Girilecek Tanımlar ****\n");
    printf("\n");
    printf("Ölçülecek Data Sayısı (1...30000) = ");scanf("%d",&DataS);
    printf("Çıkış dosyasının adı (*.*) = ");
    for (i=0; i<12; i++)
    { dad[i]=getch(stdin);
      if (dad[i]!='\b')
      { if (i>0) printf("\b\b");
        dad[i]='\x0';
        -i;
      }
    }
}
```

```
        if (i>0) { dad[i]='\x0';
                -i;
            }
    }
    else printf("%c",dad[i]);
    if (dad[i]!='r')
        { dad[i]='\x0';
          break;
        }
    }
    printf("\n");
/* Ölçme işlemini yapan Makine dili alt programın çağırılması */

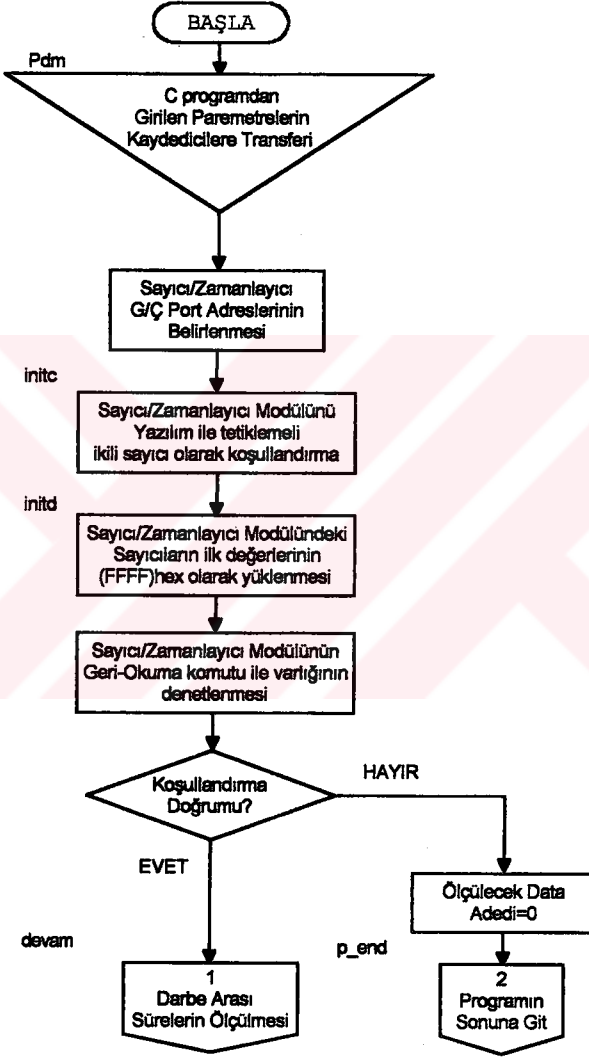
Pdm(DataDizisi, &DataS, padr);

if (DataS==0)
{
printf("\n*****
*\n");
printf("\n* Verilen Adreste Sayıcı/Zamanlayıcı (CTC) Donanımı Bulunamadı! *\n");
printf("\n*****\n");
}
else
{
for (i=0; i<DataS; i++)
{
printf("%5d)%9d\n",i,DataDizisi[i]*ztp);
}
printf("\nÖlçülmeyen data adedi = %d\n",DataDizisi[DataS]);
datf=fopen(dad,"wt");
```

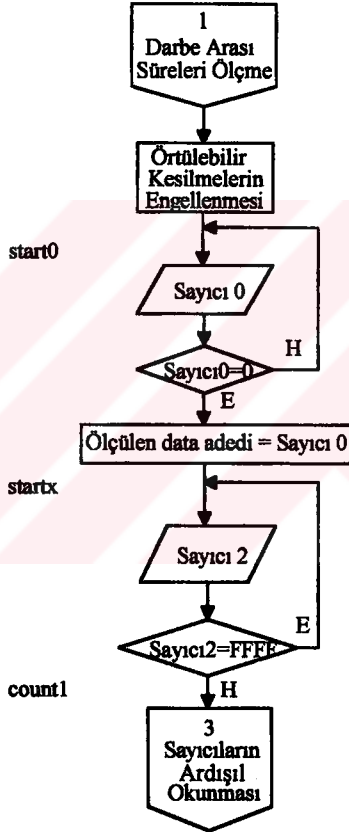
```
for (i=0; i<DataS; i++)
{
    fprintf(datf,"%d\n",DataDizisi[i]*ztp);
}
fclose(datf);
fprintf(datf,"%d,Ölçülemeyen data var!\n",DataDizisi[DataS]);
}
return 0;
}
```

Şekil 3.9 Ana programın listesi.

Darbe arası süreleri ölçen makine dili altprogramın akış diyagramı Şekil 3.10'da dört kısım olarak gösterilmiştir. Altprogramın içinde kullanılan, ana programdan girilen giriş/çıkış port adresinin değeri, data adedi ve DataDizisi değişkenlerinin adresleri kaydedicilere transfer edilir. Giriş/ çıkış komutlarında kullanılmak üzere port adresi bellekte saklandığı yere yüklenir. Böylece makine dili altprogram istenilen giriş/çıkış port adresinde kullanılabilir hale gelir. Sayıcı/zamanlayıcı modülü yazılım ile tetiklemeli ikili sayıcı olarak koşullandırılır. Bu çalışma modunda modülde bulunan sayıcılar sürekli geriye doğru sayma işlemi yapar. Bu sayıcılardan sayıcı0 ölçülen darbe sayısını, sayıcı1 ve sayıcı2 ise ardışıl olarak zaman tabanı üreticinde üretilen saat periyodu ile darbe arası süreyi sayar. Bütün sayıcıların ilk değeri (FFFF)_{HEX} olarak yüklenir. Bu ana kadar yapılan işlemler, yapılması başlangıçta zorunlu olduğu için donanımın doğru olduğu kabul edilmiştir.

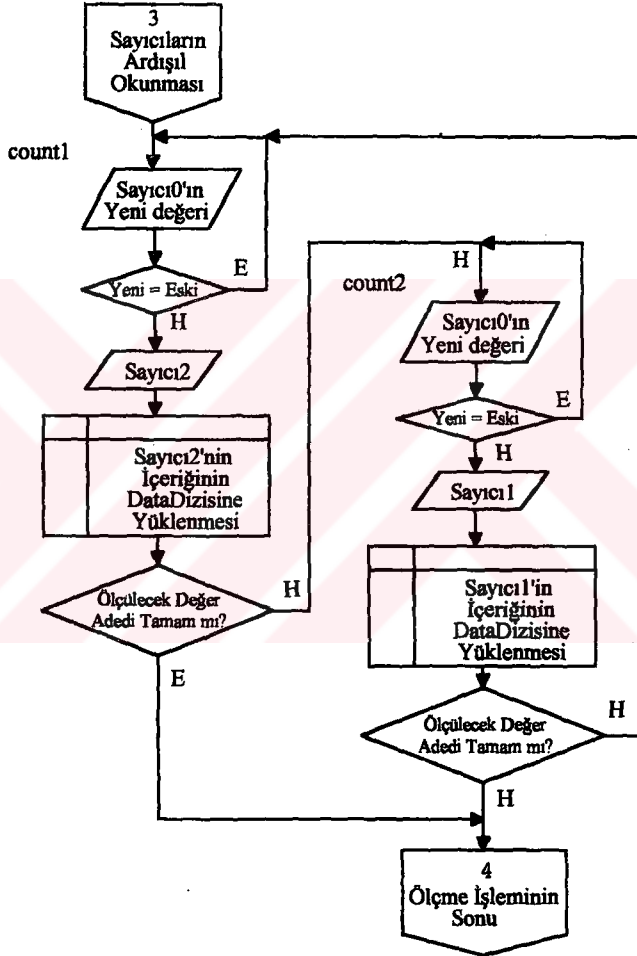


Şekil 3.10(1/4) Makine dili altprogramın akış diyagramı.



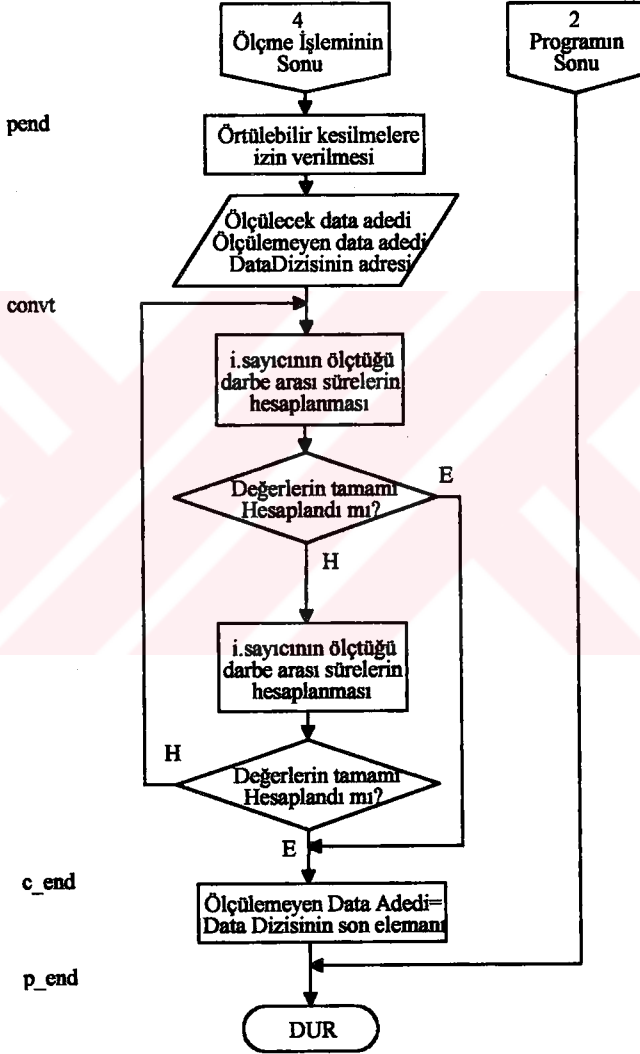
Şekil 3.10(2/4) Makine dili altprogramın akış diyagramı.

Donanımda kullanılan sayıcı/zamanlayıcı modülü, koşullandırma ve sayıcıların durumunu öğrenmek olanağını sağlar. Bu işlem özel bir geri-okuma komutu ile gerçekleştirilir. Bu yetenek kullanılarak sayıcı0'ın durumu0 sorulur ve koşullandırmasının belirlenen koşullandırma ile aynı olup olmadığı denetlenir. Eğer aynı değil ise donanımın doğru olmadığı belirlenmiş olur. Ana programdan transfer edilen ölçülecek data adedi (DataS) değişkeninin içeriğine 0 yüklenir ve ana programa geri döndürülür. Koşullandırma aynı ise donanımın doğruluğu belirlenmiş olur. İkinci kısımda, ölçüm işleminin yazılım destekli yapılması nedeniyle, programın kesintiye uğramaması için sistemin örtülebilir kesilmeler engellenmesi ile başlar. Sonra ilk darbe arası süreninin algılanması denetimi sayıcı0'ın okunması ile yapılır. Sayıcı0, ölçülecek darbe sayısını donanım ile gerçek anlamda sayar. İlk darbe yakalandığında sayıcı0'ın içeriği ölçülen data adedinin saklandığı kaydediciye yüklenir. Bu işlemler start0 ile startx arasında yapılır. Startx ile başlayan kısımda darbe arası süreleri ölçen sayıcılardan sayıcı2 okunarak, ilk darbenin gelmesi ile birlikte ilk sayan sayıcının hangisi olduğu belirlenir. Üçüncü kısımda, darbe arası sürelerin, özgün olarak geliştirilmiş olan ardışıl sayıcı yöntemi ile ölçümüne geçilir. Bu yöntem bir örnek sayıcısının denetimi altında, iki sayıcının ardışıl olarak gelen darbe arası süreleri, zaman tabanlı saatinin periyodu ile ardışıl olarak sayması prensibine dayanır.



Şekil 3.10(3/4) Makine dili altprogramın akış diyagramı.

İki sayıcının ardışıl sayması nedeniyle, bir sayının eksik yada fazla sayımı diğer sayıcıyı etkiler ve iki sayıcının toplam sayımı doğruluğu artırır. Ayrıca kullanılan bilgisayar, ölçüm donanım ve yazılım ile belirlenen duyarlık, bir sayıcı okunurken diğer sayıcının çalışmayı sürdürmesi nedeniyle artar. Burada darbenin algılanması, sayıcı0'ın okunan yeni değerinin eski değeri ile farklılık göstermesiyle gerçekleşir. Bu aynı zamanda sayıcı2'nin sayma işlemini bitirmesi anlamına gelir. Sayıcı2 okunur ve içeriği bellekte DataDizisine yüklenir. Ölçülecek değer adedinin tamamlanıp tamamlanmadığı denetlenir. Tamam değil ise, aynı işlem sayıcı1 için tekrarlanır. Tamam ise ölçme işleminin sonuna gidilir. Dördüncü ve son kısımda ise örtülebilir kesilmelere yeniden izin verilir ölçülen ardışıl değerlerin aralık değerlerine dönüştürme ve ölçülemeyen data adedinin DataDizisi'ne yükleme işlemleri yapılır. Ölçülecek data adedi, DataDizisi'nin adresi kaydedicilere yüklenir. Ölçülemeyen data adedi ise altprogramın sonunda geri alınarak DataDizisi'nin son elemanı olarak yerleştirilmek üzere yığına atılır. Sayıcıların ardışıl saymaları nedeniyle, her ölçüm kendinden iki sonraki ölçüm ile hesaplanarak dönüştürülür. Bu işlem ölçülen değerlerin tamamı dönüştürülene kadar yapılır. Dönüştürme işleminin sonunda değişen sistem parametreleri düzeltilerek altprogramdan çıkılarak ana programa geri dönülür. Makine dili programın listesi Şekil 3.11'de verilmiştir. Program "Turbo Assembler" kullanılarak yazılmış ve derlenmiştir [29,37, 38,39,40].



Şekil 3.10(4/4) Makine dili altprogramın akış diyagramı.

Şekil 3.11 Makine dili altprogramın listesi.

```
; Bu program TUNPDM kartı ile rastgele darbe arası süreleri
; gerçek zamanda ölçer ve C programda belirlenen değişkenlerin
; bulunduğu bellek alanında saklar saklar.
;
; Altprogramın C Programdan Çağırılma Biçimi:
; extern void PDM(unsigned int far * DataPtr, unsigned int * DataAdedi, ;
; unsigned int PortAdr);
```

```
; Input:
```

```
; unsigned int far * DataPtr      : Ölçülecek değerler dizisi ptr.
; unsigned int * DataAdedi       : Ölçülecek değer adedi ptr.
; unsigned int PortAdr          : CTC Giriş/Çıkış port adresi.
```

```
DOSSEG
.MODEL SMALL,C
.CODE
```

```
padrc0    dw    ?           ;CTC C0 port adresi
padrc1    dw    ?           ;CTC C1 port adresi
padrc2    dw    ?           ;CTC C2 port adresi
padrc     dw    ?           ;CTC CR port adresi
```

```
        PUBLIC Pdm
Pdm     PROC
        push  bp
        mov  bp,sp
        les  di,[bp+4]      ;dataların bulunduğu yer ES:DI
        mov  bx,[bp+8]     ;ölçülecek değer adedi ptr.
        mov  si,[bx]       ;ölçülecek değer adedi => SI reg.
```

```
add si,2
mov dx,[bp+10] ;Giriş/Çıkış Port adresi.
mov padrc0,dx ;CTC C0 Port adresi
inc dx
mov padrc1,dx ;CTC C1 Port adresi
inc dx
mov padrc2,dx ;CTC C2 Port adresi
inc dx
mov padrc,dx ;CTC CR Port adresi
inc dx
initc: mov dx,padrc ;CTC'nin C0, C1 ve C2 sayıcılarının
mov al,038h ;mode 4 yazılım ile tetiklemeli
out dx,al ;binary sayıcı olarak koşullandırılması.
mov al,078h
out dx,al
mov al,0b8h
out dx,al
initd: dec dx ;Sayıcıların ilk değerlerinin 0FFFFH
mov al,0ffh ;olarak yüklenmesi
out dx,al
out dx,al
dec dx
out dx,al
out dx,al
dec dx
out dx,al
out dx,al
mov dx,padrc
mov al,11100010b ;Read-Back Komutu (C0'ın Durumu?)
out dx,al
mov dx,padrc0
```

```
in  al,dx          ;C0'in durumunu oku
and  al,00001110b  ;Sayıcı prog. modu
cmp  al,00001000b  ;Mod 4?
jz   devam        ;CTC donanımı tamam.
mov  bx,[bp+8]     ;ölçülecek değer adedi ptr.
xor  ax,ax         ;0 -> ax
mov  [bx],ax       ;ax -> Ölçülecek değer adedi
jmp  p_end        ;programı bitir.
```

devam:

```
cli
start0: mov  dx,padrc  ;CTC'nin C0 sayıcısının içeriğinin
        mov  al,00h   ;okunmak üzere tutulması.
        out  dx,al
        mov  dx,padrc0 ;Ölçülecek ilk darbenin gelip
        in  al,dx     ;gelmediğinin kontrolü.
        mov  bl,al
        in  al,dx
        mov  bh,al
        inc  bx
        jnz  start0
```

```
dec  bx           ;CTC'nin C0 sayıcısının içeriği
mov  cx,bx       ;CX reg. saklanır.
```

startx:

```
mov  dx,padrc  ;CTC'nin C2 sayıcısının içeriğinin
mov  al,80h   ;okunmak üzere tutulması.
out  dx,al
mov  dx,padrc2 ;ilk sayan sayıcı C2 mi?
in  al,dx
mov  bl,al
in  al,dx
mov  bh,al
inc  bx
jz   count2    ;C2'de sayma işlemi yok
```

```
count1:    mov  dx,padrc    ;CTC'nin C0 sayıcısının içeriğinin
           mov  al,00h    ;okunmak üzere tutulması
           out  dx,al
           mov  dx,padrc0
           in   al,dx
           mov  bl,al
           in   al,dx
           mov  bh,al
           cmp  bx,cx
           jz   count1    ;yeni darbe gelmedi!
           mov  cx,bx
           mov  dx,padrc    ;CTC'nin C2 sayıcısının içeriğinin
           mov  al,080h    ;okunmak üzere tutulması
           out  dx,al
           mov  dx,padrc2
           in   al,dx
           mov  bl,al
           in   al,dx
           mov  bh,al
           mov  [di],bx    ;C2 sayıcısının içeriğinin saklanması.
           inc  di
           inc  di
           dec  si        ; bir sonraki değer
           jg   count2
           jmp  pend
count2:    mov  dx,padrc    ;CTC'nin C0 sayıcısının içeriğinin
           mov  al,00h    ;okunmak üzere tutulması
           out  dx,al
           mov  dx,padrc0
           in   al,dx
           mov  bl,al
```

```
in  al,dx
mov  bh,al
cmp  bx,cx
jz   count2      ;yeni darbe gelmedi!
mov  cx,bx
mov  dx,padrc    ;CTC'nin C1 sayısının içeriğinin
mov  al,040h     ;okunmak üzere tutulması
out  dx,al
mov  dx,padrc l
in  al,dx
mov  bl,al
in  al,dx
mov  bh,al
mov  [di],bx     ;C1 sayısının içeriğinin saklanması.
inc  di
inc  di
dec  si          ; bir sonraki değer
jg   count1
pend: sti        ; ölçme işleminin sonu
      ;zaman aralıklarının hesaplanması
mov  bx,[bp+8]   ; ölçülecek değer adedi ptr.
mov  si,[bx]     ; ölçülecek değer adedi => SI reg.
mov  bx,0FFFDh
sub  bx,cx
sub  bx,si       ; Ölçülemeyen data adedi.
push bx         ; Ölçülemeyen data yığına.
mov  di,[bp+4]  ; datanın başlangıç adr.
mov  ax,[di]    ; (Ci,ilk değer)=>ax
mov  cx,[di+2]  ; (Cj,ilk değer)=>dx
add  di,4       ; geçerli datanın başlangıç adr.
convt: mov  bx,ax ; eski değer => bx
```

```
sub  bx,word ptr [di] ; (eski deęer-yeni deęer)=>bx
mov  ax,[di]         ; yeni deęer => ax
mov  [di-4],bx      ; bx => yeni deęer
inc  di
inc  di
dec  si             ; bir sonraki deęer
jz   c_end         ; Cj için aynı işlem
mov  dx,cx         ; eski deęer => dx
sub  dx,word ptr [di] ; (eski deęer-yeni deęer)=>dx
mov  cx,[di]       ; yeni deęer => cx
mov  [di-4],dx     ; dx => yeni deęer
inc  di
inc  di
dec  si             ; bir sonraki deęer
jnz  convt
c_end: pop  cx
      mov  [di-4],cx ; ölçülemeyen data adedi.
p_end: pop  bp
      ret                ; Ölçülen deęerler data dizisinde.
Pdm   ENDP
      END
```

Şekil 3.11. Makine dili programın listesi.

4. RASTGELE İŞARET ÜRETİMİ

Rastgele işaretler, günlük yaşamda bulunan bir çok olayın ölçümü ve denetimi sırasında karşılaşılan problemlerden birisidir. Bu tür işaretler, elektronik müzik ritim cihazlarında ses, müzik oluşturucularda beyaz ve pembe gürültü üretici olarak, oda akustiğini test ve dengelemek amacı gibi çeşitli alanlarda kullanılabilir. Bu olayların ölçümü ve denetimi için, üzerinde çeşitli yöntemler ile inceleme ve araştırma işlemleri yapmak gerekebilir. Bununla birlikte, gerçek olaylar zinciri içinde bu konuda deney ve gözlem yapmak, güvenli ve kolay olmayabilir. Bu nedenle, çeşitli olayların ölçümünde karşılaşılan bu tür işaretlerin benzerlerinin, gerçek ortamdaki bağımsız olarak laboratuvar ortamında üretimi güvenlik ve kolaylık bakımından önemlidir. Bu konuda geliştirilen donanım ve yazılımların, üretimi yapılacak işaretin değişimine bağlı olarak uygun şekilde kullanılması yararlıdır.

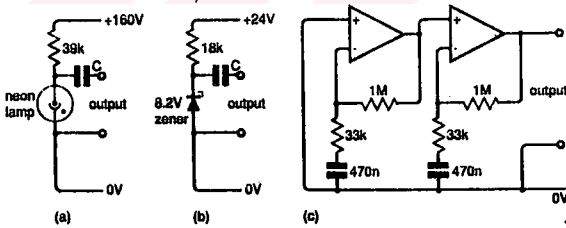
4.1. RASTGELE İŞARET ÜRETİMİNDE KULLANILAN DEVRELER

Rastgele işaret üretiminde kullanılan elektronik devreler, üretilmek istenen işaretin matematiksel ve elektriksel özelliklerine bağlı olarak, örneksel veya sayısal elektronik devreler olabilir. Bu elektronik devrelerin özelliklerinin, işaretin kullanım amacına bağlı olarak belirlenmesi yararlıdır.

Bu kısımda, yapılan araştırmalar sonucu karşılaşılan bazı rastgele işaret üreteçleri incelenmiştir.

Örneksel elektronik devreler ile yapılan rastgele işaret üreteçleri, genellikle gürültü üreteçleri olarak adlandırılır. Bu tür devrelerde rastgele işaret, devrede kullanılan elektronik devre elemanlarının, yapısal

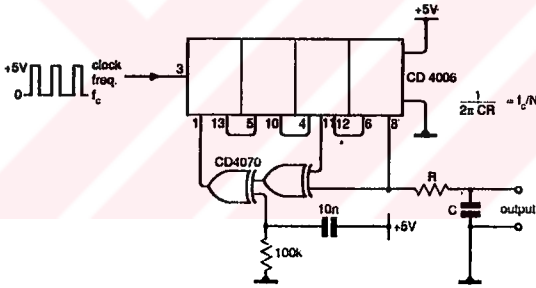
Özelliklerinden faydalanılarak oluşturulur. Bunlar, elektronik devrelerin gürültüye karşı davranışlarını ölçmek amacı ile kullanılır. Şekil 4.1'de bunlardan geçmişte kullanılan bazı gürültü üreteçleri görülmektedir [41]. Bu elektronik devrelerin ürettiği gürültü, ses frekansı bandında frekans dağılımı "beyaz" ve genlik dağılımı "normal" olarak bilinen iki karakteristik özelliğe sahiptir. Şekil 4.1(a)'da görülen neon lamba kaynaklı gürültü üreteci, ses bandında 20kHz'e kadar çok iyi çalışmakla birlikte, çalışmak için yüksek gerilime gerek duyduğundan kullanışlı değildir. Şekil 4.1(b) ve (c)'deki devreler ise daha kullanışlıdır. Şekil 4.1(b)'deki devre, uygun zener diyot seçilerek çok yüksek frekans (VHF) veya çok daha yüksek frekans (UHF) bandında çalışabilir. Şekil 4.1(c)'deki işlemsel yükselteç kullanılan devre, ses bandının alt ve üst sınır frekanslarındaki keskinlik sağlaması yönünden uygundur.



Şekil 4.1 Üç değişik tip gürültü üreteci.

Bütün bu elektronik devreler, elektronik devre elemanlarının, üretici firmaların garanti etmedikleri yapısal özelliklerine bağımlı olan gürültü değerleri

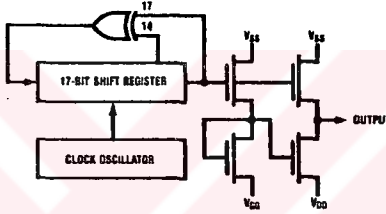
nedeniyle, diğer başka çözümlere oranla yeterince iyi değildir. Daha iyi çözüm yalancı rastgele ikili dizisi ("PRBS"=Pseudo Random Bit Sequence) üreticidir. Bu tip yalancı rastgele işaret temelli üretelerde, üretilen işaretin ilgilenilen frekans veya genlik dağılımı, devrede kullanılan elektronik devre elemanlarının yapısal özelliklerinden bağımsızdır. Şekil 4.2'de bu tip bir gürültü üretici görülmektedir [41]. Aynı temele dayanılarak yapılan tümdevre şeklindeki üreteler daha kullanışlıdır. Bunlardan bilinen ikisi "National Semiconductor" firmasının ürettiği MM5837(17 katlı), MM5437(23 katlı) tümdevrelerdir [42].



Şekil 4.2 Yalancı rastgele ikili dizisi temeline dayanan gürültü üretici.

Bu tümdevrelerden 17 katlı ötelemeli kaydedici kullanan birincisi, $2^{17}-1 = 131,071$ saat darbesi uzunluğunda periyoda sahiptir. Diğeri ise 23 katlı ötelemeli kaydedici kullanarak daha uzun bir periyoda ($2^{23}-1 = 8,388,607$) erişmiştir. Tümdevre şeklindeki bu üretelerde, saat osilatörü ve ötelemeli kaydedici bloğunun

geri besleme devresi tümleşik olarak bulunur. Bu nedenle, bu tümdevreler kullanılarak yapılan tasarımlarda, sadece üretici firmanın belirlediği sınırlar içinde çalışmak mümkündür. Şekil 4.3'de bu tümdevrelerden birinin blok diyagramı üretici firmanın verdiği katalogdan alınarak gösterilmiştir [42].



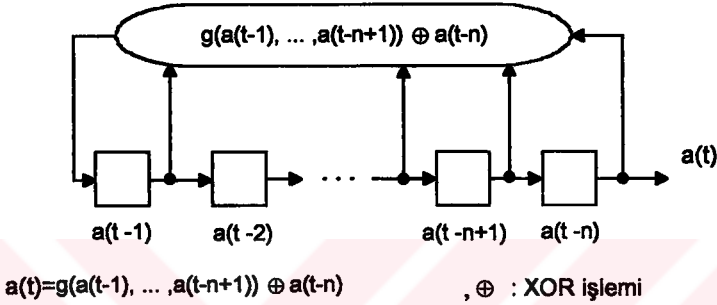
Şekil 4.3 Sayısal gürültü üreticinin (MM5837) lojik diyagramı.

Tümdevre yalancı rastgele dizi üretici temelli sayısal gürültü kaynağı olarak tanımlanmıştır. Ses frekansı uygulamalarında kullanılmak üzere geniş bantlı "beyaz" gürültü işareti üretecek şekilde tasarlanmıştır. Tekrarlama periyodu 1.1 ile 2.4 saniye arasında değişir. Tümdevre içindeki ötelemeli kaydedici, devreye güç uygulandığında sıfır olmayan rastgele bir değerden başlar.

4.1.1. Yalancı Rastgele İşaret Üreteçleri

Buraya kadar anlatılan elektronik devrelerde dikkati çeken özellik üretilen işaretin frekans ve genlik dağılımı gibi karakteristik özellikleri devrenin yapısında veya kullanılan devre elemanlarının yapısında saklı olmasıdır.

Bu nedenle, istenilen herhangi bir frekans veya genlik dağılımına sahip işaret üretimi olarak dışıdır.



Şekil 4.4 N-katlı geri beslemeli ötelemeli kaydedici.

Yalancı rastgele işaret üretici, periyodu kullanım süresinden çok uzun olan sayısal işaret üretici olarak tanımlanabilir. Bu tip üreticiler lojik devreler, bilgisayar yazılımı veya donanımı ile gerçekleştirilebilir. En çok kullanılan üreticiler tipi, geri beslemeli ötelemeli kaydedicilerdir (FSR=Feedback Shift Register). Şekil 4.4'de n katlı bir geri beslemeli ötelemeli kaydedicinin blok diyagramı görülmektedir [44]. Geri beslemeli ötelemeli kaydedici, n tane iki kararlı ve geri besleme fonksiyonundan meydana gelmiştir. Bu üreticilerde, geri besleme fonksiyonu g, lineer bir fonksiyon ise (sadece özel veya, XOR ile gerçekleştirilebiliyor ise) üreticiler, lineer geri beslemeli ötelemeli kaydedici (LFSR=Linear Feedback Shift Register) olarak adlandırılır. aksi halde lineer olmayan geri beslemeli ötelemeli kaydedici (NLFSR=Nonlinear Feedback Shift Register) olarak adlandırılır. Üreticinin periyodu, ötelemeli kaydedici sayısına ve geri besleme

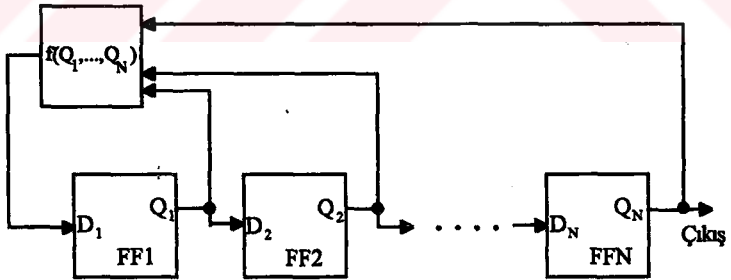
fonksiyonuna bağılı olarak değişir. Geri besleme fonksiyonları Dk(4.1)'de gösterilen biçimdedir [44].

$$a(t) = c_1 a(t-1) \oplus c_2 a(t-2) \oplus \dots \oplus c_{n-1} a(t-n+1) \oplus a(t-n), c_i \in \{0,1\} \quad (4.1)$$

Lineer geri beslemeli ötelemeli kaydedicinin geri besleme bağlantısı, geri besleme polinomu olarak adlandırılır ve Dk(4.2)'de gösterilen biçimdedir [44].

$$f(x) = 1 + c_1 x + c_2 x^2 + \dots + c_{n-1} x^{n-1} + x^n \quad (4.2)$$

Geri besleme polinomu, çıkış dizisinin periyodunu ve frekans dağılımını belirler. En uzun periyodu veren polinoma ilkel polinom denir. Periyodun uzunluğu aynı zamanda iyi istatistiğin sağlanması demektir. Garanti edilen en uzun periyod, n-katlı lineer geri beslemeli ötelemeli kaydedici için $2^n - 1$ saat periyodu süresidir.

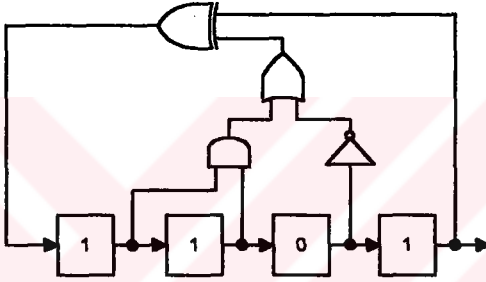


Şekil 4.5 Geri beslemeli ötelemeli kaydedici.

Periyodun sınırları, metal-oksit silikon yarıiletken büyük çapta tümleştirme (MOS LSI) teknolojisi kullanıldığında 2000 katlı iki kararlıdan oluşmuş ötelemeli kaydedicinin tek yongada bulunması halinde $2^{2000} - 1$ saat periyodu süresi

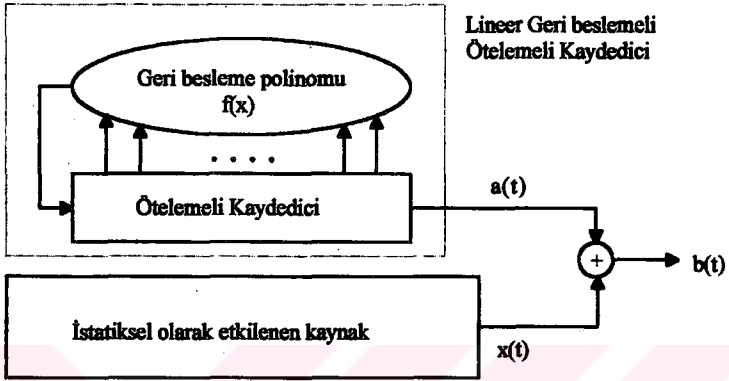
kadar artırılabilir. Şekil 4.5'de geri beslemeli ötelemeli kaydedicinin lojik diyagramı gösterilmiştir [2].

Geri besleme bağlantısı özel veya fonksiyonu ile gerçekleşmeyen bir lineer olmayan geri beslemeli ötelemeli kaydedicinin lojik diyagramı Şekil 4.6'da görülmektedir [44].

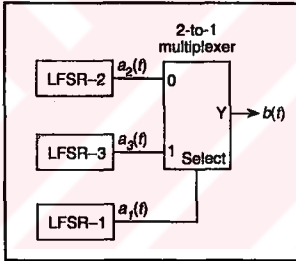


Şekil 4.6 Dört katlı De Bruijn ikili dizi üretici.

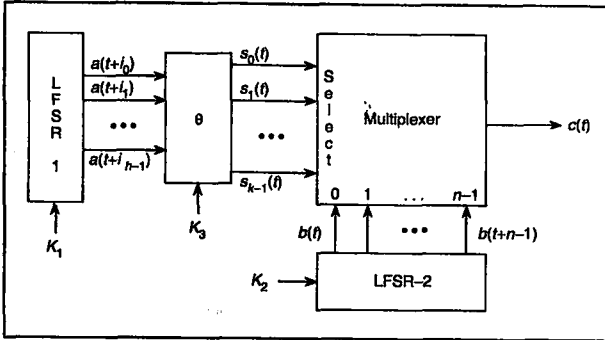
Yalancı rastgele sayısal işaret üreteçlerinin yaygın olarak kullanıldığı alanlardan en önemlisi, bilginin saklanması, iletilmesi sırasında korunmasına yönelik olarak şifrelenmesidir. Yukarıda belirtilen üreteçlerin ürettiği bit dizisi, lineer geri besleme yapılan türden olması nedeniyle, şifreleme amacına yönelik kullanılması yeterince güvenli olmamaktadır. Daha güvenli şifreleme yapmak için, lineer ve lineer olmayan sendrom hücumu gibi değişik lojik tasarımlar geliştirilmiştir. Bu tasarımlar, değişik araştırmacılar tarafından bulunmuş özel yapıdan meydana geldiği için araştırmacının verdiği ad ile anılmaktadır. Lineer sendrom hücum için kullanılan sistemin blok diyagramı Şekil 4.7'de ve araştırmalar sırasında karşılaşılan bazı tasarımlar Şekil 4.8'de görülmektedir [45,46,47,48,49,50,51].



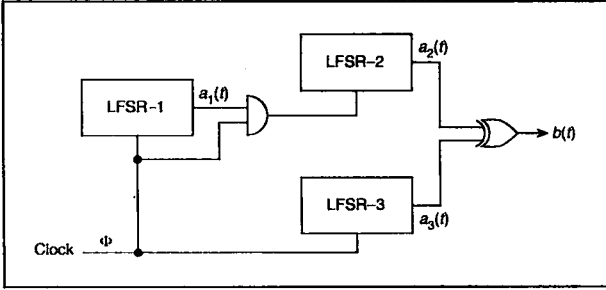
Şekil 4.7 Lineer sendrom atak türünden türetilenler için kullanılan model.



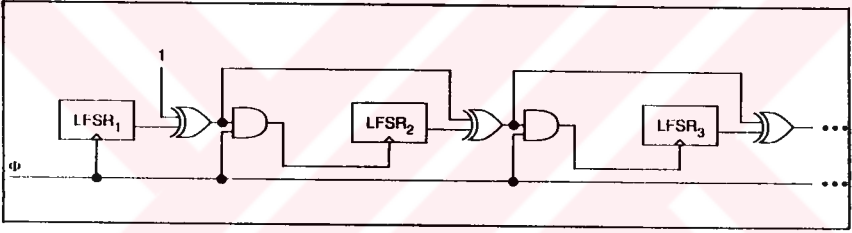
Şekil 4.8(a) Geffe türetilen.



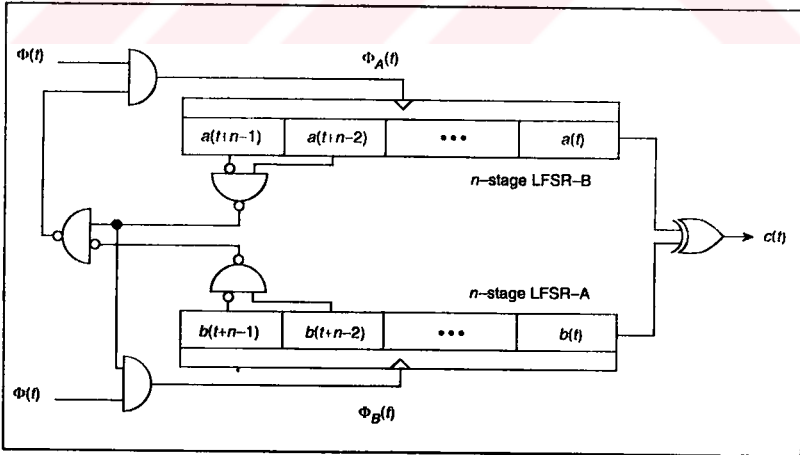
Şekil 4.8(b) Jennings türetilen.



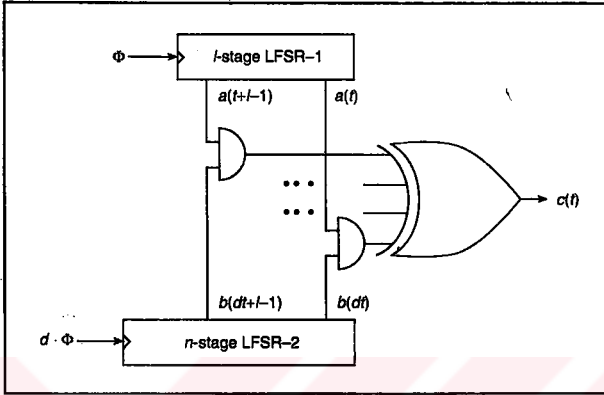
Şekil 4.8(c) Beth-Piper dur-ve-git üretici.



Şekil 4.8(d) Gollmann kaskad dur-ve-git üretici.



Şekil 4.8(e) İki yönlü adım denetim temelli yalancı rastgele ikili üretici.



Şekil 4.8(f) Massey-Rueppel değişen hızlı üretici.

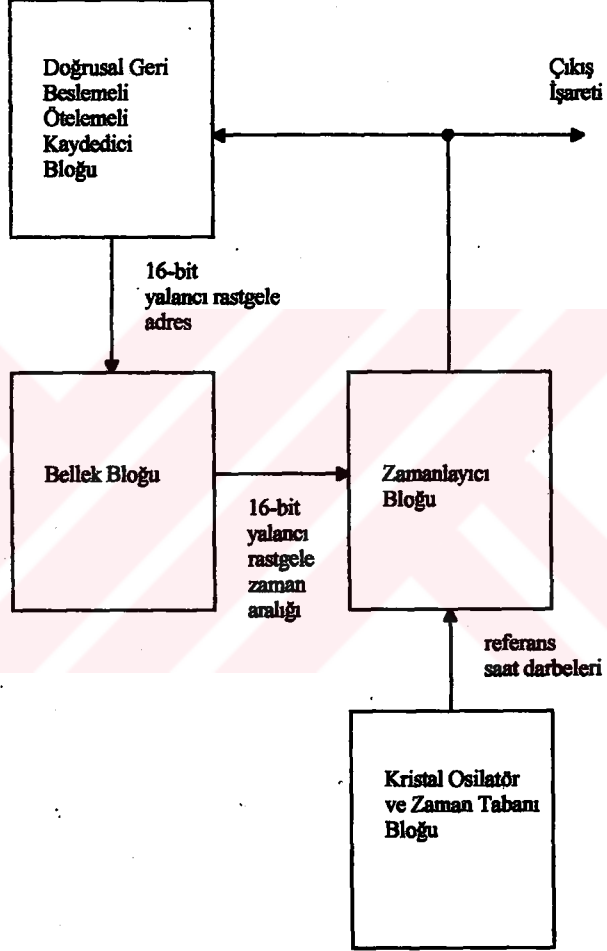
4.2. TASARLANARAK GERÇEKLEŞTİRİLEN YALANCI RASTGELE İŞARET ÜRETECİ

Bu kısımda, rastgele işaretlerin benzetimini yapmak üzere genel amaçlı bir yalancı rastgele üreteç tasarımı ve gerçekleşmesi üzerine çalışılmıştır. Bu amaçla olayın temel mekanizması incelenmiş ve fiziksel olarak gerçekleşen bu çalışmada, ayırık lojik tümdevreler kullanılarak bir elektronik düzen oluşturulmuştur. Bu üreteç genel amaçlı olarak, istatistiksel dağılımı bilinen bir işaretin, yazılım ile önceden belirlenen özelliklerde, çok uzun bir periyoda sahip olan yalancı rastgele sayısal işaretlerin üretimine olanak sağlar.

Tasarlanarak gerçekleştirilen yalancı rastgele üreteç sisteminin çalışma prensibi, içinde üretilecek işaretin istatistiksel dağılımını saklayan yazılım ile desteklenmiş programlanabilir bellek, sayıcı ve zamanlayıcı donanımına dayanır. Bu üreteç sistemi, rastgele değişen sayısal işaretleri, programlanabilir bellekteki verilere bağlı olarak belirlenen darbe arası süreleri, 100ns ile 10µs arasında değişen duyarlılıkla üretmeyi sağlar.

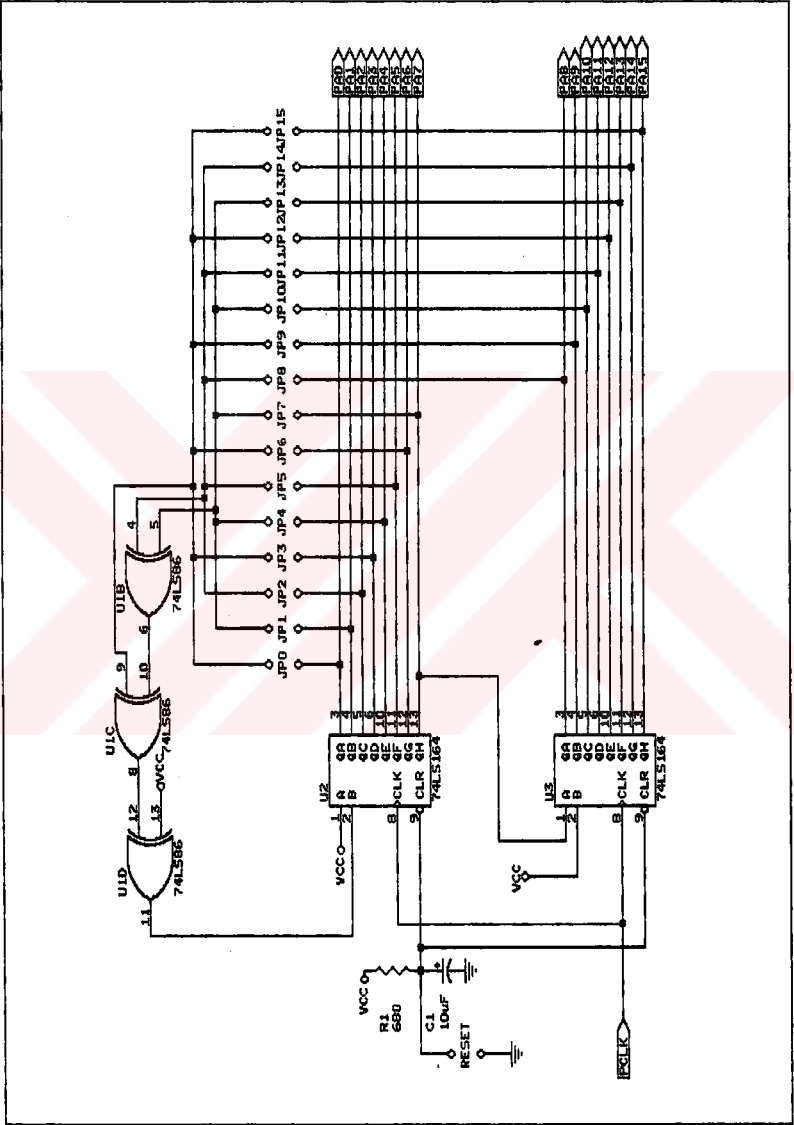
4.2.1. Üreteç Sisteminin Donanımı

Yalancı sayısal rastgele işaret üreticinin donanımı, darbe arası süreleri yazılım ile önceden belirlenerek programlanan sabit bellek bloğuna bağlı olarak değişen, bağımsız olarak çalışabilen bir lojik devredir. Şekil 4.9'da sistemin blok diyagramı görülmektedir. Ötelemeli kaydedici bloğunda, bellek bloğunda gerek duyulan, yazılım ile belirlenen geri besleme fonksiyonuna bağlı olarak yalancı rastgele veri adresleri üretilir. Üretilen veri adresleri yine yazılım ile belirlenen darbe arası süreleri saklayan bellek bloğuna uygulanır. Bu şekilde bellek



Şekil 4.9 Yalancı rastgele sayısal işaret üreticinin blok diyagramı.

bloğunda sıralı olarak saklanan darbe arası süreler, yalancı rastgele ikili kodlanmış ondalık sayısal veriler olarak zamanlayıcı bloğuna uygulanır. Zaman tabanı bloğunda, donanım ile belirlenen periyotlarda saat işareti üretilir. Zamanlayıcı bloğunda, zaman tabanı üreticinden alınan referans saat işareti ile, bellek bloğundan gelen sayısal veriyi ilk değer olarak yükleyip geriye doğru sayarak darbe arası süreler oluşturulur. Elde edilen bu çıkış işareti aynı zamanda ötelemeli kaydedici bloğuna uygulanarak daha sonra oluşturulacak darbe arası sürenin yalancı rastgele adresini belirler. Bu işlem dizisi, yazılım ile belirlenen periyod süresinde tekrarlanır. Periyod bitiminde başlangıç noktasına geri dönülerek aynı darbe arası süre dağılımı tekrarlanır. Ötelemeli kaydedici bloğu, 16-bit ötelemeli kaydedici ve geri besleme fonksiyonunu sağlayan özel veya (XOR) lojik devrelerinden meydana gelmiştir (Şekil 4.10). İki adet 8-bit genişliğindeki ötelemeli kaydedici tümdevresi, şekildeki gibi kaskad bağlanarak 16-bit genişliğinde ötelemeli kaydedici elde edilmiştir. Sistem sıfır başlangıçlı lineer geri beslemeli ötelemeli kaydedici olarak tasarlanmıştır. Üç noktalı geri besleme fonksiyonu 16 çıkışı kapsayacak biçimde kullanıcı tarafından Tablo 4.1'de gösterilen şekilde seçilebilir. Tabloda "Gr." ile belirtilen sütun üç noktalı geri beslemenin "A,B,C" olarak nokta grubunu gösterir. Bu gruplar dikkate alınarak geri besleme fonksiyonu seçilir. Aynı gruplar ile yapılacak seçimler hatalıdır ve sisteme zarar verebilir. Üç nokta A, B ve C gruplarından birer tane olarak seçilmelidir.

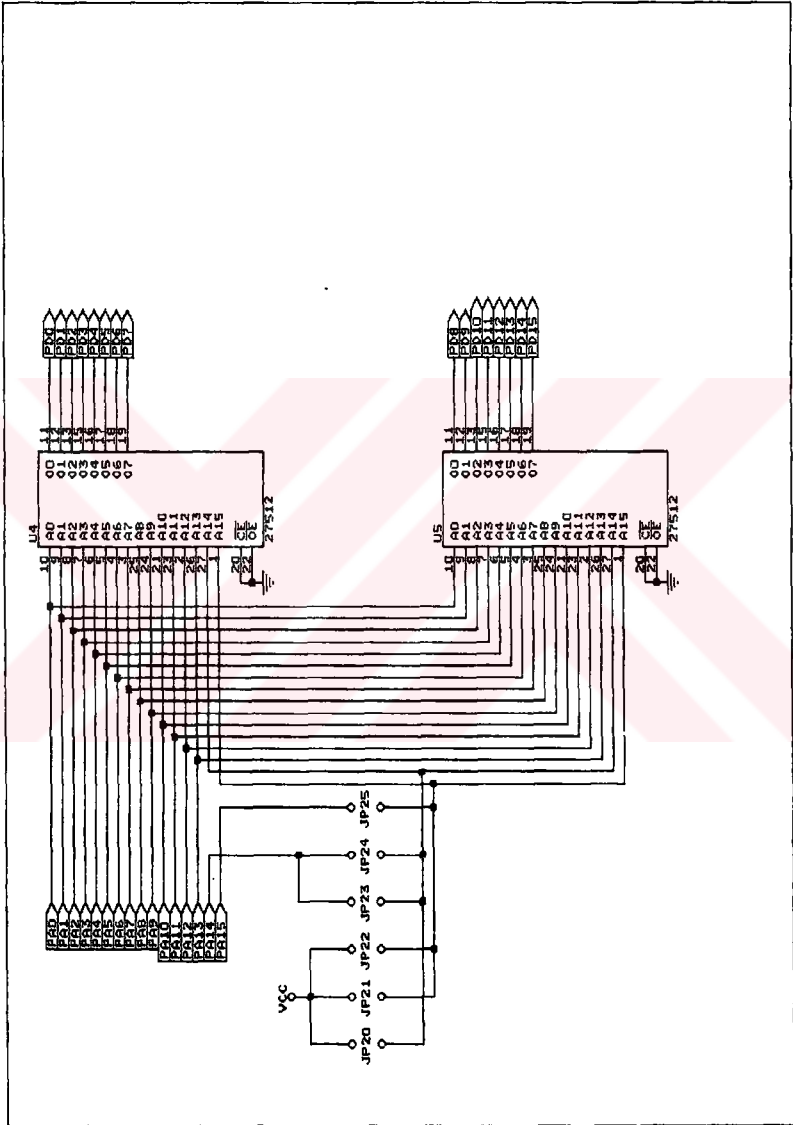


Şekil 4.10 Lineer geri beslemeli ötelemeli kaydedici bloğunun lojik devresi.

Tablo 4.1 Geri besleme fonksiyonunun seçimi.

		GERİ BESLEME FONKSİYONU				
Gr.		Q15@Q14@Q13	Q12@Q11@Q10	Q14@Q12@Q10	Q15@Q14@Q7
JP15	A	KAPALI	AÇIK	AÇIK	KAPALI	..
JP14	B	KAPALI	AÇIK	KAPALI	KAPALI	..
JP13	C	KAPALI	AÇIK	AÇIK	AÇIK	..
JP12	A	AÇIK	KAPALI	KAPALI	AÇIK	..
JP11	B	AÇIK	KAPALI	AÇIK	AÇIK	..
JP10	C	AÇIK	KAPALI	KAPALI	AÇIK	..
JP9	A	AÇIK	AÇIK	AÇIK	AÇIK	..
JP8	B	AÇIK	AÇIK	AÇIK	AÇIK	..
JP7	C	AÇIK	AÇIK	AÇIK	KAPALI	..
JP6	A	AÇIK	AÇIK	AÇIK	AÇIK	..
JP5	B	AÇIK	AÇIK	AÇIK	AÇIK	..
JP4	C	AÇIK	AÇIK	AÇIK	AÇIK	..
JP3	A	AÇIK	AÇIK	AÇIK	AÇIK	..
JP2	B	AÇIK	AÇIK	AÇIK	AÇIK	..
JP1	C	AÇIK	AÇIK	AÇIK	AÇIK	..
JP0	A	AÇIK	AÇIK	AÇIK	AÇIK	..

Üreteç sistemine güç verildiğinde R1, C1 elemanları kullanılarak başlangıçta, ötelemeli kaydedicilerin içeriği sıfırlanmaktadır. Ayrıca başlangıç uygunluğu sağlamak üzere sistem bir sıfırlama (RESET) girişine sahiptir. Bellek bloğu 16-bit genişliğinde veri saklayabilen ve büyüklüğü standart tümdevrelerden seçilebilen, elektriksel olarak programlanabilen ultraviyole ışık ile silinebilen türden sabit belleklerden (EPROM) meydana gelmiştir (Şekil 4.11). İki adet 8-bit EPROM tümdevresi adres yolları paralel, veri yolları seri bağlanarak 16-bit veri yolu genişliğine sahip bellek bloğu elde edilmiştir.



Şekil 4.11 Bellek bloğunun lojik devresi.

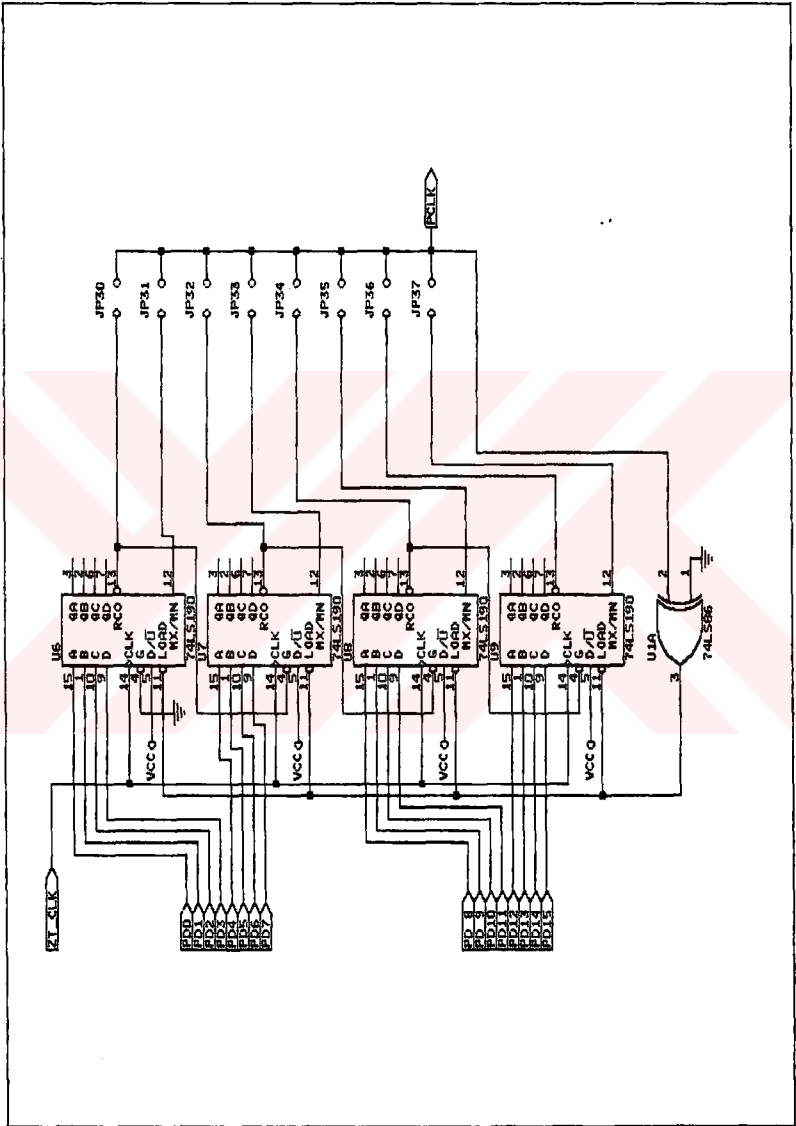
Üreteç sistemi isteğe bağlı olarak Tablo 4.2'de gösterilen biçimde seçilebilen 8, 16, 32 ve 64KB olmak üzere 4 değişik EPROM tümdevresi kullanabilir [52].

Tablo 4.2 Bellek tipinin seçimi.

BELLEK EPROM tipi		JP20	JP21	JP22	JP23	JP24	JP25
8KB	2764	KAPALI	KAPALI	AÇIK	AÇIK	AÇIK	AÇIK
16KB	27128	KAPALI	KAPALI	AÇIK	AÇIK	AÇIK	AÇIK
32KB	27256	AÇIK	AÇIK	KAPALI	KAPALI	AÇIK	AÇIK
64KB	27512	AÇIK	AÇIK	AÇIK	AÇIK	KAPALI	KAPALI

Zamanlayıcı bloğu kaskad bağlı 4 adet ikili kodlanmış ondalık programlanabilir sayıcı tümdevresinden meydana gelmiştir. Sayıcılar geriye sayıcı olarak tasarlanmıştır. Bellek bloğundan ilk yükleme değerini alarak bu değerden geriye doğru sayılır. Toplam sayım sonucu sifıra düştüğünde, son sayıcının (MX/MN) ucu lojik "1" olur. Bu işaret çıkış işareti olarak kullanılır. Aynı zamanda ötelemeli kaydediciye öteleme komutu veren saat işareti olarak uygulanmaktadır. Bu işareten faydalanılarak bir sonraki ön yükleme değerini zamanlayıcı bloğuna uygulama komutu (LOAD) üretilir.

Zamanlayıcının geriye sayma hızı, zaman tabanı bloğunda üretilen referans saat işaretinin periyodu ile belirlenir. Zamanlama işleminde kullanılacak zamanlayıcı büyüklüğü Tablo 4.3'de gösterilen şekilde seçilir.



Şekil 4.12 Zamanlayıcı bloğunun lojik devresi.

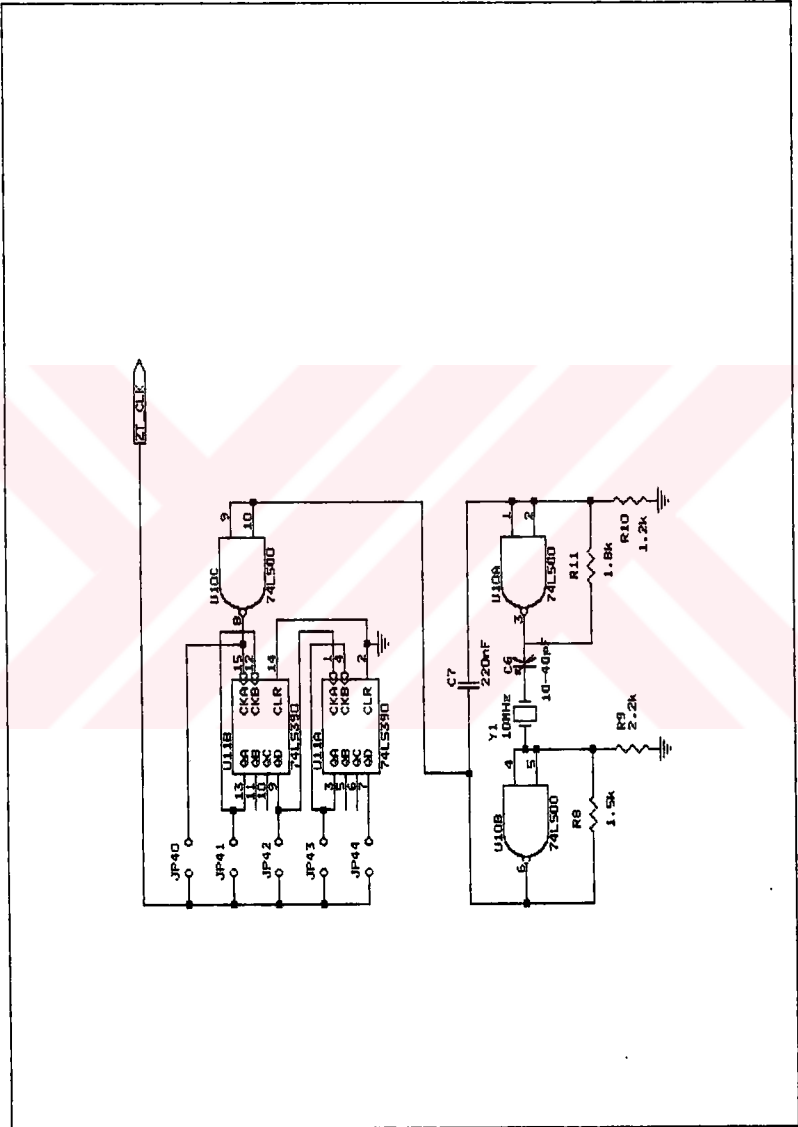
Tablo 4.3 Zamanlayıcı büyüklüğünün seçimi.

	JP30	JP31	JP32	JP33	JP34	JP35	JP36	JP37
4-bit	KAPALI	KAPALI	AÇIK	AÇIK	AÇIK	AÇIK	AÇIK	AÇIK
8-bit	AÇIK	AÇIK	KAPALI	KAPALI	AÇIK	AÇIK	AÇIK	AÇIK
12-bit	AÇIK	AÇIK	AÇIK	AÇIK	KAPALI	KAPALI	AÇIK	AÇIK
16-bit	AÇIK	AÇIK	AÇIK	AÇIK	AÇIK	AÇIK	KAPALI	KAPALI

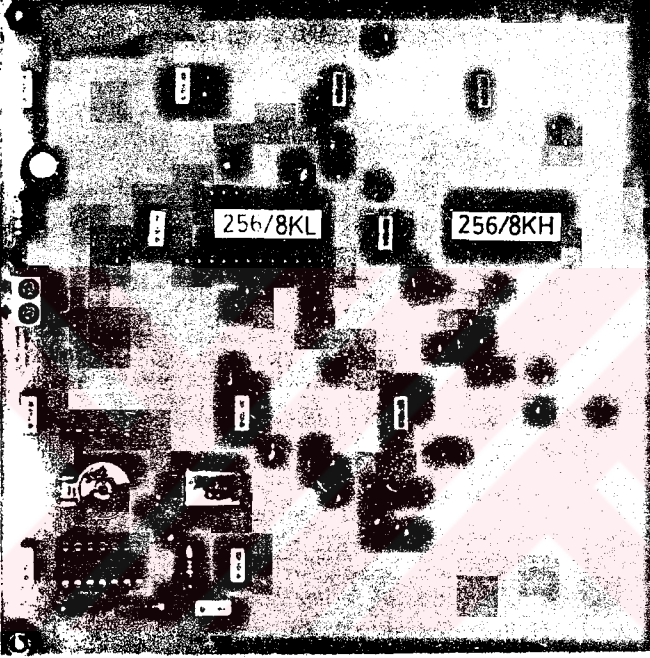
Saat ve zaman tabanı bloğunun ayrıntılı elektronik devre şeması Şekil 4.13'de görülmektedir. Saat üretici, U10A ve U10B vedeğil kapıları ile 10MHz kristalden oluşturulmuş bir kare dalga osilatördür. Osilatörün çıkışı U10C vedeğil kapısı ile güçlendirilerek zaman tabanı üreticine saat olarak verilir. Zaman tabanı üretici, kaskad bağlı 2 haneli BCD sayıcıdan oluşturulmuştur. Zaman tabanı seçimi Tablo 4.4'de gösterilen biçimde yapılabilir.

Tablo 4.4 Zaman tabanı seçim tablosu.

Zaman Tabanı	JP40	JP41	JP42	JP43	JP44
0.1 μ s	KAPALI	AÇIK	AÇIK	AÇIK	AÇIK
0.2 μ s	AÇIK	KAPALI	AÇIK	AÇIK	AÇIK
1 μ s	AÇIK	AÇIK	KAPALI	AÇIK	AÇIK
2 μ s	AÇIK	AÇIK	AÇIK	KAPALI	AÇIK
10 μ s	AÇIK	AÇIK	AÇIK	AÇIK	KAPALI



Şekil 4.13. Kristal osilatör ve zaman tabanlı tareticinin lojik devresi.

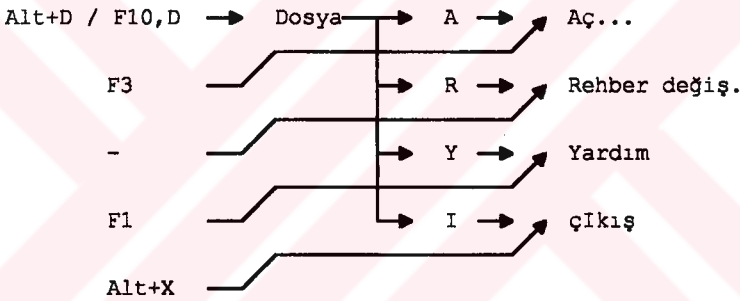


Şekil 4.14 Gerçekleştirilen kartın görünümü.

4.2.2. Üreteç Sisteminin Yazılımı

Tasarlanarak gerçekleştirilen yalancı rastgele sayısal işaret üretici sisteminin istenen özelliklerde çalışabilmesi için, gerekli olan ayarların kolay ve doğru yapılmasını sağlamak üzere, Pascal dilinde yazılmış bir tasarım ve analiz yazılımı gerçekleştirilmiştir.

Yazılım fonksiyon tuşları veya fare ile seçilebilen ana işlem ve bunların altında yer alan alt seçeneklere sahiptir (Şekil 4.15).



Şekil 4.15 Yazılımın ana işlem ve alt seçeneklerine erişim biçimi.

İlk seçenek disk üzerindeki yazısal dosyaları ve yardım dosyasını görüntüleyen "Dosya" adlı seçenektir. Bu seçeneğin altında: disk üzerindeki yazısal dosyaları açmak için "Aç", disk üzerinde istenen alt rehbera geçmek için "Rehber deęiş...", içinde tasarım ve analiz programı ile ilgili açıklayıcı bilgiler bulunan yardım dosyasını ekrana getiren "Yardım" ve programdan çıkmak için "çıkış" alt seçenekleri bulunur (Şekil 4.16).

Dosya Pencere dOnanım olasılık Fonk. Koştur

Aç...	F3
Rehber deęiş...	
Yardım	F1
Çıkış	Alt-X

Alt-X Çıkış F1 Yardım F3 Aç F5 Boyut Alt-F3 Kapat

Şekil 4.16 "Dosya" ana işlem seçeneęi ve alt seçeklerinin görünümü.

İkinci seçenek ekranda oluşturulan pencerelerin boyutlarını, seçilmesini, yerleşim düzenini ve kapatma denetimini sağlayan "Pencere" adlı seçenektir. Bu seçeneęin altında: açık bir pencerenin boyutunun ayarlanmasını ve ekranda taşınmasını sağlayan "boyut/taşı", boyutunun bir önceki boyuta ayarlanmasını veya ekranı kaplamasını sağlayan "Büyük/küçük", bir sonraki pencerenin seçilmesini sağlayan "Sonraki", aktif pencerenin kapatılmasını sağlayan "Kapat", açık bulunan pencerelerin ekranı eşit bölerek görünmesini sağlayan "Döşe" ve arka arkaya sıralanarak "bAsamakla" alt seçenekleri bulunur (Şekil 4.17).

Dosya Pencere donanım olasılık fonk. koştur

Boyut/Açış	Ctrl-F5
Büyük/küçük	F5
Sonraki	16
kapat	Alt-F3
Döşe	
bAsamakla	

Alt-X Çıkış 11 Yardım F3 Aç F5 Boyut Alt-F3 kapat

Şekil 4-17 "Pencere" ana işlem seçeneği ve alt seçeneklerinin görünümü.

Üçüncü seçenek, üreteç sisteminin donanımında kullanılan bağlantı dizilerinin konumlarını ve tasarımla ilgili giriş parametrelerinin verildiği "dOnanım" adlı seçenektir. Bu seçeneğin altında: yukarıda belirtilen giriş işlemlerinin kolay ve basit bir şekilde yapılmasını sağlayan "Kur" adlı alt seçenek bulunur (Şekil 4.18). Bu alt seçenek tarafından oluşturulan etkileşimli ekran görüntüsü Şekil 4.19'da görülmektedir. Bu seçenek tablosunda seçenek dizileri arasında ileri hareket TAB tuşu, geri hareket ise SHIFT+TAB ile yapılır. Seçenek dizisi içinde hareket ise yön tuşları ile yapılır.

Dosya Pencere dOnanım olasılık Fonk. Koştur

Kur

Alt-X Çıkış F1 Yardım F3 Aç F5 Boyut Alt-F3 Kapat

Şekil 4-18 "Donanım" ana işlem seçeneği ve alt seçeneklerinin görüntüsü.

Ekranın sol kenarında "GeriBes." başlığı altında dikey doğrultuda, üreteç donanımında yer alan, lineer geri beslemeli ötelemeli kaydedici bloğunun, geri besleme fonksiyonunu belirleyen bağlantı dizisinin görüntüsü, bir seçenek dizisi olarak yer alır (bkz. Şekil 4.10). Bu seçenek dizisinde bağlantı olan elemanlar "X" işaretli, bağlantı olmayan elemanlar boş olarak görülür. Seçenek tablosundaki kutucukları işaretleme ve boşaltma işlemi "boşluk" tuşu veya farenin sağ tuşuna basılarak yapılır. Seçenek tablosunda hareket işlemi tuş takımındaki hareket tuşları veya doğrudan fare ile sağlanır. İşaretleme birden fazla ve istenen biçimde yapılabilir. Fakat rastgele yapılacak işaretlemeler, istenen sonucu sağlamayan geri

besleme fonksiyonlarını belirtebilir (bkz. Tablo 4.1). Bu nedenle yardımcı olmak amacı ile yardım yazısal dosyası içinde geri besleme şekline bağlı olarak elde edilebilecek periyod uzunlukları bir tablo şeklinde verilmiştir.

Dosya Pencere dOnanım olasılık Fonk. Koştur
[■] Donanımın Tanıtımı

GeriBes.	EPROM Tipi	Zaman Tabanı
[] A Q0	(*) 8K, 2764	(*) R 0.1us
[] B Q1	() 16k, 27128	() S 0.2us
[] C Q2	() 32K, 27256	() U 1.0us
[] D Q3	() 64k, 27312	() V 2.0us
[] E Q4		() Y 10.0us
[] F Q5	Bellek Boyutu	
[] G Q6	8192	
[] H Q7		
[] I Q8	Parça Sayısı	
[] J Q9	128	
[X] K Q10		
[X] L Q11	Ortalama Değer	
[X] M Q12	128	
[] N Q13		
[] O Q14		
[] P Q15		

Tamam İptal

Alt-X Çıkış F1 Yardım F3 Aç F5 Boyut Alt-F3 Kapat

Şekil 4-19 "Donanımın tanıtımı" adlı etkileşimli seçenek tablosunun görüntüsü.

Bu seçenek dizisinin sağında, üretç donanımında kullanılan bellek tipinin seçimini sağlayan "EPROM Tipi" seçeneği bulunur (bkz. Şekil 4.11, Tablo 4.2). Burada seçilen EPROM tipine bağlı olarak, ötelemeli kaydedicinin hesaplamada kullanılan bit sayısı, genişliği belirlenir. Dolaylı olarak üretilebilecek en fazla yalancı rastgele adres sayısı da belirlenmiş olur.

Bu seçeneğin altında yer alan "Bellek Boyutu" seçeneği ile kullanılan EPROM tümdevresinin tamamı yerine daha az

bir bölümünün kullanılma olanağı sağlanmıştır. Bu giriş seçeneğinin ilk değeri 8KB'lık EPROM tipi için 8192 olarak verilmiştir. Diğer EPROM tipleri seçilecek olursa, bu giriş değerinin uygun bir değer ile değiştirilmesi zorunludur.

"Parça Sayısı" seçeneğinde, olasılık fonksiyonunun kaç ayrık parçadan oluşturularak hesaplanacağı girilir. Ayrık parça gerçek fiziksel olayda darbe arası sürenin kanal aralığına karşılık düşer. Poisson dağılım fonksiyonu için yapılacak giriş en fazla 1024 ayrık parça olabilir.

"Ortalama Değer" seçeneğinde ise, olasılık fonksiyonunun parça sayısına göre, orta noktasında yer alan darbe arası sürenin değeri girilir. Bu şekilde darbe arası sürelerin, zaman ekseninde istenen noktaya kaydırılması sağlanır.

"Zaman Tabanı" seçenek dizisinde, üreteç donanımında yer alan zaman tabanı bloğunda bulunan, bağlantı dizisinin anlamlı konumu verilerek tasarımda yapılan hesaplar ile fiziksel işaret arasındaki uygunluk sağlanır (bkz Şekil 4.13, Tablo 4.4). Bu seçenek dizisindeki seçeneklerden sadece birisi seçilebilir.

Dördüncü seçenek, bir önceki donanım seçeneği ile girilen parametrelere bağlı olarak olasılık fonksiyonunu hesaplayan ve yazıcıda yazdıran "olasılık Fonk." adlı seçenektir. Bu seçeneğin altında: olasılık fonksiyonunu Poisson dağılımına göre gerçek değerlerini hesaplayıp disk üzerinde OF_REEL.TXT adlı yazısal dosyada saklayan "Poisson Dağılımı", olasılık fonksiyonunu kullanıcı tarafından dışarıda hesaplanarak oluşturulmuş OF_REELX.TXT adlı yazısal dosyadan okuyan "OF_REELX.TXT Dağılımı" ve hesaplanan olasılık fonksiyonunu OF_REEL.TXT adlı yazısal

Dosya Pencere dOnanım olasılık Fonk. Kořtur

Poisson dađılımlı
OF_REELX.TXT Dađılımlı
OF_REEL.TXT Yazdır

Alt-X Çıkış F1 Yardım F3 Aç F5 Boyut Alt-F3 Kapat .

Őekil 4.20 "Olasılık fonk." ana işlem seçeneđi ve alt seçeneklerinin görünümü.

dosyayı okuyarak yazıcıda yazdıran "OF_REEL.TXT Yazdır" alt seçenekleri bulunur (Őekil 4.20).

Beşinci ve son seçenek, önceki iki seçenek çalıştırıldıktan sonra donanım seçeneđinde girilen giriş parametrelerine ve disk üzerinde oluşan olasılık fonksiyonuna bađlı olarak üreteç sisteminde kullanılan belleklerin içeriđini disk üzerinde oluşturan "Tasarım" adlı bir alt seçeneđi olan "Kořtur" adlı seçenektir (Őekil 4.21). Bu seçenek çalıştırıldıktan sonra, tasarım işlemi sırasında ara işlemler sonucu disk üzerinde oluşturulan yazısal dosyalara "Dosya" seçeneđi ile bakılarak tasarım işleminin denetimi ve analizi yapılabilir.

Dosya Pencere donanım olasılık Fonk. Koştur

Tasarım

Alt-X Çıkış F1 Yardım F3 Aç F5 Boyut Alt-F3 Kapat

Şekil 4.21 "Koştur" ana işlem seçeneği ve alt seçeneklerinin görünümü.

Yazılımın çalışma biçimi Şekil 4.22'deki akış diyagramında ana hatlarıyla gösterilmiştir. Yazılım çalıştırıldıktan sonra ilk işlem olarak "donanım" seçeneği seçilir ve donanım yazılıma tanıtılır. Sonrasında "olasılık fonksiyonu" seçeneği seçilerek tasarlanacak işaretin olasılık dağılımı yazılıma girilir. Bu iki işlemden sonra son olarak "Koştur" seçeneğine girilerek program çalıştırılır. Yazılım tasarımıyla ilgili çıkışları, yazısal dosya olarak disk üzerinde oluşturur. Bu çıkışlar tasarımı ile yazısal olarak veya istatistiksel dağılımları grafiğe çevirebilen hazır yazılımlar ile grafik olarak görülebilir ve incelenebilir.



Şekil 4.22 Üreteç tasarım programının çalışma akış diyagramı.

Üreteç tasarımını gerçekleştirme işlemi, yazılımda "donanımın tanıtımı" tablosunda yapılan işlemlerin donanıma uygulanması ve yazılımın disk üzerinde oluşturduğu bellek ile ilgili bilgilerin bellek tümdevrelerine aktarılarak donanımdaki yerine takılması ile sağlanır. Devreye güç verildiğinde üreteç sistemi tasarlanan özelliklere uygun çıkış işareti verecektir.

4.5.1. Üreteç Sistemi Yazılımının Kaynak Listesi

Bu kısımda tasarım yazılımının ayrıntılı açıklaması, yazılımın kaynak listesi içindeki açıklamalardan yararlanılarak yapılmıştır. Üreteç sistemi yazılımı, "Turbo Pascal" ve bunun içinde yer alan "Turbo Vision" yazılım paketleri kullanılarak yazılmış ve derlenmiştir [53,54,55].

```
{*****}
{* BU PROGRAM BAĞIMSIZ ÇALIŞABİLEN YALANCI RASTGELE      *}
{* SAYISAL İŞARET ÜRETECİNİN DONANIMINI DÜZENLEMEK İÇİN *}
{* KULLANILAN TASARIM YAZILIMIDIR                       *}
{* Yazılımı (ve donanımı) gerçekleştiren: Tuncay UZUN    *}
{*****}
```

```
 {$X+}
```

```
program YR_SAYI_URETEC_TASARIM;
```

```
 {$M 16384,16384,655360}
```

```
uses crt,printer,
```

```
    Objects, Drivers, Views, Menus, Dialogs, App,
```

```
    Dos, Memory, StdDlg, MsgBox;
```

```
const
```

```
    nu=16;           { Lineer Geri beslemeli ötelemeli }
                    { Kaydedicinin en fazla bit sayısı. }
    segs=4;          { Dağılım fonk. en fazla durum }
    elms=8192;       { sayısı 8*8k=64k }
    fseg=1024;       { Dağılım fonk. en fazla parça sayısı. }
```

```
const
  cmFileOpen      = 100;
  cmChangeDir     = 101;
  hlChangeDir     = cmChangeDir, {eski bilgiler}
  cmYard          = 102;
  cmDonanKur      = 103;
  cmOlasYaz       = 104;
  cmTasarla       = 105;
  cmOlasFonk      = 106;
  cmOlasPoisson   = 107;
```

type

```
{* Dosyaları gösteren kısımda *}
{* kullanılan yapıların tanıtımı *}
```

```
{ TLineCollection }
```

```
PLineCollection = ^TLineCollection;
TLineCollection = object(TCollection)
  procedure FreeItem(P: Pointer); virtual;
end;
```

```
{ TGoster }
```

```
PGoster = ^TGoster;
TGoster = object(TScroller)
  FileLines : PCollection;
  IsValid    : Boolean;
  constructor Init(var Bounds: TRect; AHScrollBar, AVScrollBar: PScrollBar,
    var FileName: PathStr);
  destructor Done; virtual;
  procedure Draw; virtual;
```



```
function Valid(Command: Word): Boolean; virtual;  
end;
```

```
{ TFileWindow }  
PFileWindow = ^TFileWindow;  
TFileWindow = object(TWindow)  
    constructor Init(var FileName: PathStr);  
end;
```

```
{ TGosterUyg }  
PGosterUyg = ^TGosterUyg;  
TGosterUyg = object(TApplication)  
procedure HandleEvent(var Event: TEvent); virtual;  
procedure InitMenuBar; virtual;  
    procedure InitStatusLine; virtual;  
    procedure OutOfMemory; virtual;  
    procedure DonanKur;  
    procedure Yard;  
end;
```

```
{*Donanımın Tanıtımı tablosunda *}  
{* kullanılan yolların tanıtımı *}
```

```
DialogData = record  
    GeriBesFonk      : word;  
    EPROMtipi       : word;  
    ZamanTaban      : word;  
    BellekBoyutu    : string[128];  
    ParcaSay        : string[128];  
    OrtDeg          : string[128];  
end;
```

```
PDemoDialog = ^TDemoDialog;  
TDemoDialog = object(TDialog)  
end;
```

```
var
```

```
TDialogData: DialogData;  
{* Yalancı rastgele sayısal işaret *}  
{* üretici tasarım kısmında *}  
{* kullanılan yapıların tanıtımı *}
```

```
type
```

```
epr = array [0..elems-1] of word;
```

```
var
```

```
pf,ef : text;  
x,xm,l,i,j,ii,T,n,s : word;  
sayi,k,gbn,OrtT : word;  
peprdat : array [0..segs-1] of ^epr;  
padr : array [0..segs-1] of pointer;
```

```
function kuv(say,ust:integer):integer;
```

```
begin
```

```
kuv:=round(exp(ust*ln(say)));
```

```
end; {func kuv}
```

```
procedure KurSart;
```

```
{*****}  
{* Donanımın tanıtımı seçenek tablosunun kabul edilen ilk *}  
{* değerlerinin ana programdan hesaplanarak alınması. *}  
{*****}
```

```
var
```

```
Bellek: word;
```

```
begin
  s:=(TDialogData.EPROMtipi+1)*8192;
  val(TDialogData.BellekBoyutu,Bellek,i);
  if (Bellek>0) and (Bellek<=s)
    then s:=Bellek;
  dec(s);
  i:=0;
  repeat
    inc(i);
    x:=kurv(2,i);
  until x>=s;
  n:=i;
  val(TDialogData.ParcaSay,xm,i);
  val(TDialogData.ParcaSay,ii,i);
  if (ii>0) and (ii<1024)
    then xm:=ii;
  dec(xm);
  l:=round(xm/2);
  val(TDialogData.OrtDeg,ii,i);
  if (ii>0) and (ii<65535)
    then OrtT:=ii;
  case TdialogData.ZamanTaban of
    0 : i:=100;
    1 : i:=50;
    2 : i:=10;
    3 : i:=5;
    4 : i:=1;
  end;{case}
  OrtT:=round(OrtT*i div 10-l);
end;{proc KurSart}
```

```
procedure OlasPoisson;
{*****}
{* 0 ile xm arasındaki sayılar için poisson olasılık      *}
{* fonksiyonunun aldığı değerlerin hesaplanması      *}
{* "OF_REEL.TXT" adlı dosya olarak diskte saklanması.  *}
{*****}
var
  fakt,tx : extended;
  p       : extended;
begin
  gotoxy(3,24);
  textbackground(red);write('Lütfen Bekleyiniz...');
  KurSart;
  assign(pf,'OF_REEL.TXT');
  rewrite(pf);
  fakt:=1; x:=0;
  p:=exp(-1)*exp(x*ln(1))/fakt;
  tx:=p;
  writeln(pf,p);
  for x:=1 to xm do
  begin
    fakt:=fakt*x;
    p:=exp(-1)*exp(x*ln(1))/fakt;
    tx:=tx+p;
    writeln(pf,p);
  end;
  close(pf);
  gotoxy(3,24);
  textbackground(green);write('İşlem Tamamlandı....');
  textbackground(black);
end;{OlasPoisson}
```

```
procedure OlasFonk;
{*****}
{* 0 ile xm arasındaki sayılar için . . . . . olasılık      *}
{* fonksiyonunun aldığı değerlerin "OF_REELX.TXT" adlı     *}
{* yazısal dosyadan okunarak "OF_REEL.TXT" adlı yazısal    *}
{* dosya olarak diskte saklanması.                          *}
{*****}
var
  tx   : extended;
  p    : extended;
begin
  gotoxy(3,24);
  textbackground(red);write('Lütfen Bekleyiniz...');
  KurSart;
  assign(ef,'OF_REELX.TXT');
  reset(ef);
  assign(pf,'OF_REEL.TXT');
  rewrite(pf);
  x:=0; tx:=0;
  while not(eof(ef)) and (x<=xm) do
  begin
    readln(ef,p);
    inc(x); tx:=tx+p;
    writeln(pf,p);
  end;
  close(pf);
  close(ef);
  gotoxy(3,24);
  textbackground(green);write('İşlem Tamamlandı....');
  textbackground(black);
end; {proc OlasFonk}
```

```
procedure OlasFonkYaz;
{*****}
{* Olasılık fonksiyonunun reel değerlerinin 4 sütun      *}
{* halinde yazıcıya gönderilmesi.                        *}
{*****}
var
  p      : extended;
begin
  assign(pf,'OF_REEL.TXT');
  reset(pf);
  writeln(lst,chr(15));
  x:=0;
  while x<xm do
  begin
    i:=0;
    while (i<4) and (x+i<=xm) do
    begin
      readln(pf,p);
      write(lst,' p(',x+i:3,')=' ,p, ', ');
      inc(i);
    end;{while i}
    writeln(lst);
    x:=x+i;
  end;{while x}
  close(pf);
end;{proc OlasFonkYaz}
```

```
procedure YalanRastAdr;
```

```
{*****}  
{* Yalancı rastgele (Pseudorandom) bellek adresi olarak      *}  
{* kullanılacak sayı dizisinin "n" bitlik lineer geri        *}  
{* beslemeli ötelemeli kaydedici (LFSR) yöntemi ile          *}  
{* hesaplanması ve "YR_ADR.TXT" adlı yazısal dosya olarak    *}  
{* diskte saklanması.                                         *}  
{*****}
```

```
var
```

```
Q      : array[0..nu-1]of boolean;
```

```
Qe     : boolean;
```

```
function zero:boolean; {ötelemeli kaydedici başlangıç testi}
```

```
{* ötelemeli kaydedicinin içeriğinin sıfır olup      *}
```

```
{* olmadığını kontrol eden mantıksal sonuçlu fonksiyon. *}
```

```
begin
```

```
for j:=0 to n-1 do if Q[j] then zero:=false;
```

```
end{zero};
```

```
begin
```

```
for i:=0 to n-1 do Q[i]:=false; {RESET SR}
```

```
assign(cf,'YR_ADR.TXT');
```

```
rewrite(cf);
```

```
i:=0;
```

```
while ( (i<=s) and not( (i>0) and (zero) ) ) do
```

```
begin
```

```
sayi:=0;
```

```
for j:=0 to n-1 do
```

```
begin
```

```
if j=0 then k:=1
```

```
else k:=k*2;
```

```
if Q[j] then
    sayi:=sayi+k;
end;
{rasgele sayıların ekrana ve diske yazılması}
writeln(cf,sayi);
{Q(n-1) eski durumunun saklanması}
Qe:=Q[n-1];
{bilgilerin sağa doğru kaydırılması}
for j:=n downto 1 do Q[j]:=Q[j-1];
{Q(n-1)'den Q(0)'a geri besleme yapılması}
gbn:=TDialogData.GeriBesFonk*2;
for j:=1 to n-1 do
    if (gbn and kuv(2,j))>0
        then Qe:=Qe xor Q[j];
Q[0]:=not(Qe);
inc(i);
end; {while}
close(cf);
T:=i;
end; {proc YalanRastAdr}
```

```
procedure OlasFonkTam;
{*****}
{* Üretilcek işaretin olasılık fonksiyonunun diskten      *}
{* okunarak EPROM'un içinde bulunacak bilgilerin        *}
{* "OF_TAM.TXT" adlı dosya olarak diske yazılması.      *}
{*****}
var
    KT      : real;
    preel   : array [0..fseg-1] of extended;
```



```
begin
  assign(pf,'OF_REEL.TXT');
  reset(pf);
  ii:=0;
  while not(eof(pf)) do
    begin
      readln(pf,preel[ii]);
      inc(ii);
    end;
  dec(ii);

  KT:=T-1/s;
  repeat
    i:=0;
    KT:=KT+1/s;
    for j:=0 to ii do
      i:=i+round(KT*preel[j]);
    until i>=T;
  assign(ef,'OF_TAM.TXT');
  rewrite(ef);
  for j:=0 to ii do
    writeln(ef,round(KT*preel[j]));
  close(ef);
end; {OlasFonkTam}
```

```
procedure OlasFonkDag;
{*****}
{* Hesaplanan olasılık fonksiyonuna göre dağılımın *}
{* düzenlenerek "OF_DAG.TXT" adlı dosyaya sıralı olarak *}
{* yazılması. *}
{*****}
```

```
begin
  assign(pf,'OF_TAM.TXT');
  reset(pf);
  assign(ef,'OF_DAG.TXT');
  rewrite(ef);
  i:=0;
  while not(eof(pf)) do
  begin
    readln(pf,j);
    if j<>0 then
      for k:=1 to j do
        writeln(ef,i+OrtT);
    inc(i);
  end; {while pf}
  cclose(pf);
  close(ef);
end; {OlasFonkDag}
```

```
procedure DagDegRast;
{*****}
{* Olasılık fonksiyonunun değerlerinin diskten okunarak      *}
{* EPROM'un yalancı rastgele adreslerinde bulunacak olan    *}
{* bilgilerin "OFD_TAM.TXT" adlı dosya olarak diske          *}
{* yazılması.                                                *}
{*****}
begin
  assign(pf,'OF_DAG.TXT');
  assign(ef,'YR_ADR.TXT');
  reset(pf);
  reset(ef);
```

```
for ii:=0 to segs-1 do
begin
  new(peprdat[ii]);
  mark(padr[ii]);
  for i:=0 to elms-1 do
    peprdat[ii]^i:=65535;
end; {for i,ii}
while not(eof(ef)) do
begin
  readln(pf,sayi); {sıralı darbe arası süre dağılımı}
  readln(ef,j); {yalancı rastgele adres}
  ii:=j div elms;
  i:=j-ii*elms;
  release(padr[ii]);
  peprdat[ii]^i:=sayi;
end; {while ef}
close(pf);
close(ef);
assign(ef,'OFD_TAM.TXT');
rewrite(ef);
j:=0;
while j<=s do
begin
  ii:=j div elms;
  i:=j-ii*elms;
  release(padr[ii]);
  writeln(ef,peprdat[ii]^i);
  inc(j);
end; {while s}
close(ef);
end; {DagDegRast}
```

```
procedure DagdegBCD;
{*****}
{* Olasılık fonksiyonunun dağılım değerlerinin diskten   *}
{* okunup ondalık paketlenmiş ikili (BCD) biçiminde,     *}
{* "EPRALT.BIN" ve "EPRUST.BIN" adlı iki ayrı dosya      *}
{* olarak diske yazılması.                                *}
{*****}
var
  el,eh  : file of byte;
  bits,d1,d2,d3,d4 : byte;
begin
  assign(ef,'OF_DAG.TXT');
  reset(ef);
  assign(pf,'OFD_TAM.TXT');
  reset(pf);
  assign(el,'EPRALT.BIN');
  rewrite(el);
  assign(eh,'EPRUST.BIN');
  rewrite(eh);
  while not(eof(ef)) do
  begin
    readln(pf,sayi);
    if sayi <> 65535
    then
    begin
      readln(ef,sayi);
      d4:=sayi div 1000;
      d3:=(sayi-d4*1000) div 100;
      d2:=(sayi-d4*1000-d3*100) div 10;
      d1:=sayi-d4*1000-d3*100-d2*10;
      d1:=d1+d2*16;
```

```
write(e1,d1);
d3:=d3+d4*16;
write(eh,d3);
end
else
begin
d1:=255;
write(e1,d1);
write(eh,d1);
end;{if}
end;{while ef}
close(ef);
close(pf);
close(e1);
close(eh);
end;{DagDegBCD}
```

procedure Tasarla;

```
{*****}
{* Alt programların uygun sıra ile çağrılarak tasarımın *}
{* gerçekleştirilmesi. *}
{*****}
```

```
begin
gotoxy(3,24);
textbackground(red);write('Lütfen Bekleyiniz...');
KurSart;
YalanRastAdr;
OlasFonkTam;
OlasFonkDag;
DagDegRast;
DagDegBCD;
```

```
gotoxy(3,24);
    textbackground(green);write("İşlem Tamamlandı....");
textbackground(black);
end; {proc Tasarla}
```

```
{* Yazısal Dosyaların diskten okunmasında *}
{* ve ekran şekillendirme için kullanılan *}
{* altprogramların tanımlamaları *}
```

```
{ TLineCollection }
procedure TLineCollection.FreeItem(P: Pointer);
begin
    DisposeStr(P);
end;

{ TGoster }
```

```
if IOResult <> 0 then
begin
  MessageBox('Dosya açılmıyor!' + Filename +!', nil, mfError + mfOkButton);
  IsValid := False;
end
else
begin
  MaxWidth := 0;
  while not Eof(FileToView) and not LowMemory do
  begin
    Readln(FileToView, Line);
    if Length(Line) > MaxWidth then MaxWidth := Length(Line);
    FileLines^.Insert(NewStr(Line));
  end;
  Close(FileToView);
end;
{SI+}
SetLimit(MaxWidth, FileLines^.Count);
end;

destructor TGoster.Done;

begin
  Dispose(FileLines, Done);
  TScroller.Done;
end;

procedure TGoster.Draw;
var
  B: TDrawBuffer;
  C: Byte;
```

```
I: Integer;
S: String;
P: PString;
begin
  C := GetColor(1);
  for I := 0 to Size.Y - 1 do
    begin
      MoveChar(B, ' ', C, Size.X);
      if Delta.Y + I < FileLines^.Count then
        begin
          P := FileLines^.At(Delta.Y + I);
          if P <> nil then S := Copy(P^, Delta.X + 1, Size.X)
          else S := "";
          MoveStr(B, S, C);
        end;
      WriteLine(0, I, Size.X, 1, B);
    end;
  end;

function TGoster.Valid(Command: Word): Boolean;
begin
  Valid := IsValid;
end;

{ TFileWindow }
constructor TFileWindow.Init(var FileName: PathStr);
const
  WinNumber: Integer = 1;
var
  R: TRect;
```



```
begin
  Desktop^.GetExtent(R);
  TWindow.Init(R, Filename, WinNumber);
  Options := Options or ofTileable;
  Inc(WinNumber);
  GetExtent(R);
  R.Grow(-1, -1);
  Insert(New(PGoster, Init(R,
    StandardScrollBar(sbHorizontal + sbHandleKeyboard),
    StandardScrollBar(sbVertical + sbHandleKeyboard), Filename)));
end;

{ TGosterUyg }
procedure TGosterUyg.HandleEvent(var Event: TEvent);

procedure FileOpen;
var
  D: PFileDialog;
  FileName: PathStr;
  W: PWindow;
begin
  D := PFileDialog(ValidView(New(PFileDialog, Init('*.*TXT', 'Dosya Aç',
    '~A~dt', fdOpenButton, 100))));
  if D <> nil then
  begin
    if Desktop^.ExecView(D) <> cmCancel then
    begin
      D^.GetFileName(FileName);
      W := PWindow(ValidView(New(PFileWindow, Init(FileName))));
      if W <> nil then Desktop^.Insert(W);
    end;
  end;
end;
```

```
    Dispose(D, Done);  
end;  
end;
```

```
procedure ChangeDir;
```

```
var
```

```
    D: PChDirDialog;
```

```
begin
```

```
    D:=PChDirDialog(ValidView(New(PChDirDialog,Init(0,  
hlChangeDir))));
```

```
    if D <> nil then
```

```
        begin
```

```
            Desktop^.ExecView(D);
```

```
            .Dispose(D, Done);
```

```
        end;
```

```
    end;
```

```
procedure Tile;
```

```
var
```

```
    R: TRect;
```

```
begin
```

```
    Desktop^.GetExtent(R);
```

```
    Desktop^.Tile(R);
```

```
end;
```

```
procedure Cascade;
```

```
var
```

```
    R: TRect;
```

```
begin
```

```
    Desktop^.GetExtent(R);
```

```
    Desktop^.Cascade(R);
```

end;

begin

TApplication.HandleEvent(Event);

case Event.What of

evCommand:

begin

case Event.Command of

cmFileOpen: FileOpen;

cmChangeDir: ChangeDir;

cmCascade: Cascade;

cmTile: Tile;

cmYard: Yard;

cmOlasFonk: OlasFonk;

cmOlasPoisson: OlasPoisson;

cmOlasYaz: OlasFonkYaz;

cmDonanKur: DonanKur;

cmTasarla: Tasarla;

else

Exit;

end;

ClearEvent(Event);

end;

end;

end;

procedure TGosterUyg.InitMenuBar;

{*****}

{* Ana seçenek ekranın üst satırında bulunan seçeneklerin *}

{* görüntü ve tuş denetiminin düzenlenmesi. *}

{*****}

var

R: TRect;

begin

GetExtent(R);

R.B.Y := R.A.Y+1;

MenuBar := New(PMenuBar, Init(R, NewMenu(

NewSubMenu('~D~osya', 100, NewMenu(

NewItem('~A~ç...', 'F3', kbF3, cmFileOpen, hcNoContext,

NewItem('~R~ehber deęiş...', ", kbNoKey, cmChangeDir, hcNoContext,

NewItem('~Y~ardım', 'F1', kbF1, cmYard, hcNoContext,

NewItem('ç~I~kış', 'Alt-X', kbAltX, cmQuit, hcNoContext, nil))));

NewSubMenu('~P~encere', hcNoContext, NewMenu(

NewItem('b~O~yut/taşı', 'Ctrl-F5', kbCtrlF5, cmResize, hcNoContext,

NewItem('~B~üyük/küçük', 'F5', kbF5, cmZoom, hcNoContext,

NewItem('~S~onraki', 'F6', kbF6, cmNext, hcNoContext,

NewItem('~K~apat', 'Alt-F3', kbAltF3, cmClose, hcNoContext,

NewItem('~D~öşe', ", kbNoKey, cmTile, hcNoContext,

NewItem('b~A~samakla', ", kbNoKey, cmCascade, hcNoContext, nil))));

NewSubMenu('d~O~nanım', hcNoContext, NewMenu(

NewItem('~K~ur', ", kbNoKey, cmDonanKur, hcNoContext, nil)),

NewSubMenu('olasılık ~F~onk.', hcNoContext, NewMenu(

NewItem('~P~oisson dağılımı', ", kbNoKey, cmOlasPoisson, hcNoContext,

NewItem('OF_REELX.TXT ~D~ağılımı', ", kbNoKey, cmOlasFonk,

hcNoContext,

NewItem('OF_REEL.TXT ~Y~azdır', ", kbNoKey, cmOlasYaz, hcNoContext,

nil))));

NewSubMenu('~K~oştur', hcNoContext, NewMenu(

NewItem('~T~asarım', ", kbNoKey, cmTasarla, hcNoContext, nil)),

nil))));

end;

```
procedure TGosterUyg.InitStatusLine;
{*****}
{* Ana seçenek ekranın altında bulunan durum satırının *}
{* görüntü ve tuş denetiminin düzenlenmesi. *}
{*****}
var
  R: TRect;
begin
  GetExtent(R);
  R.A.Y := R.B.Y - 1;
  StatusLine := New(PStatusLine, Init(R,
    NewStatusDef(0, $FFFF,
      NewStatusKey(" kbF10, cmMenu,
      NewStatusKey('~Alt-X~ Çıkış', kbAltX, cmQuit,
      NewStatusKey('~F1~ Yardım', kbF1, cmYard,
      NewStatusKey('~F3~ Aç', kbF3, cmFileOpen,
      NewStatusKey('~F5~ Boyut', kbF5, cmZoom,
      NewStatusKey('~Alt-F3~ Kapat', kbAltF3, cmClose, nil)))))),
    nil));
end;
```

```
procedure TGosterUyg.OutOfMemory;
var
  D: PDialog;
  R: TRect;
  C: Word;
begin
  MessageBox('Bellek alanı işlemi tamamlamak için yeterli değil.',
    nil, mfError + mfOkButton);
end;
```

```
procedure TGosterUyg.DonanKur;
{*****}
{* Donanım ana seçeneğinin altında yer alan "kur" alt   *}
{* seçeniginde bulunan "donanın tanıtımı" seçenek     *}
{* tablosunun görüntü ve tuş denetimini düzenlenmesi.  *}
{*****}
var
```

```
Bruce: PView;
```

```
Dialog: PDemoDialog;
```

```
R: TRect;
```

```
C: Word;
```

```
begin
```

```
R.Assign(0, 0, 79, 23);
```

```
Dialog := New(PDemoDialog, Init(R, 'Donanımın Tanıtımı '));
```

```
with Dialog^ do
```

```
begin
```

```
R.Assign(3, 3, 14, 19);
```

```
Bruce := New(PCheckBoxes, Init(R,
```

```
NewSItem('~A~ Q0',
```

```
NewSItem('~B~ Q1',
```

```
NewSItem('~C~ Q2',
```

```
NewSItem('~D~ Q3',
```

```
NewSItem('~E~ Q4',
```

```
NewSItem('~F~ Q5',
```

```
NewSItem('~G~ Q6',
```

```
NewSItem('~H~ Q7',
```

```
NewSItem('~I~ Q8',
```

```
NewSItem('~J~ Q9',
```

```
NewSItem('~K~ Q10',
```

```
NewSItem('~L~ Q11',
```

```
NewSItem('~M~ Q12',  
NewSItem('~N~ Q13',  
NewSItem('~O~ Q14',  
NewSItem('~P~ Q15',  
nil)))))))))
```

```
));
```

```
Insert(Bruce);
```

```
R.Assign(3, 2, 14, 3);
```

```
Insert(New(PLabel, Init(R, 'GeriBes.', Bruce)));
```

```
R.Assign(18, 3, 34, 7);
```

```
Bruce := New(PRadioButtons, Init(R,  
NewSItem('~8~K, 2764',  
NewSItem('~16~K, 27128',  
NewSItem('~32~K, 27256',  
NewSItem('~64~K, 27512',  
nil))))
```

```
));
```

```
Insert(Bruce);
```

```
R.Assign(18, 2, 34, 3);
```

```
Insert(New(PLabel, Init(R, 'EPROM Tipi', Bruce)));
```

```
R.Assign(38, 3, 52, 8);
```

```
Bruce := New(PRadioButtons, Init(R,  
NewSItem('~R~ 0.1us',  
NewSItem('~S~ 0.2us',  
NewSItem('~U~ 1.0us',  
NewSItem('~V~ 2.0us',  
NewSItem('~Y~ 10.0us',  
nil))))))
```

```
));
```

```
Insert(Bruce);  
R.Assign(38, 2, 52, 3);  
Insert(New(PLabel, Init(R, 'Zaman Tabanı', Bruce)));
```

```
R.Assign(18, 9, 34, 10);  
Bruce := New(PInputLine, Init(R, 128));  
Insert(Bruce);  
R.Assign(18, 8, 34, 9);  
Insert(New(PLabel, Init(R, 'Bellek Boyutu', Bruce)));
```

```
R.Assign(18, 12, 34, 13);  
Bruce := New(PInputLine, Init(R, 128));
```

```
Insert(Bruce);  
R.Assign(18, 11, 34, 12);  
Insert(New(PLabel, Init(R, 'Parça Sayısı', Bruce)));
```

```
R.Assign(18, 15, 34, 16);  
Bruce := New(PInputLine, Init(R, 128));  
Insert(Bruce);  
R.Assign(18, 14, 34, 15);  
Insert(New(PLabel, Init(R, 'Ortalama Değer', Bruce)));
```

```
R.Assign(22, 18, 32, 20);  
Insert(New(PButton, Init(R, '~T~amam', cmOK, bfDefault)));  
R.Assign(35, 18, 45, 20);  
Insert(New(PButton, Init(R, 'iPta!', cmCancel, bfNormal)));  
end;  
Dialog^.SetData(TDialogData);  
C := Desktop.ExecMenu(Dialog);  
if C <> cmCancel then Dialog^.GetData(TDialogData);
```

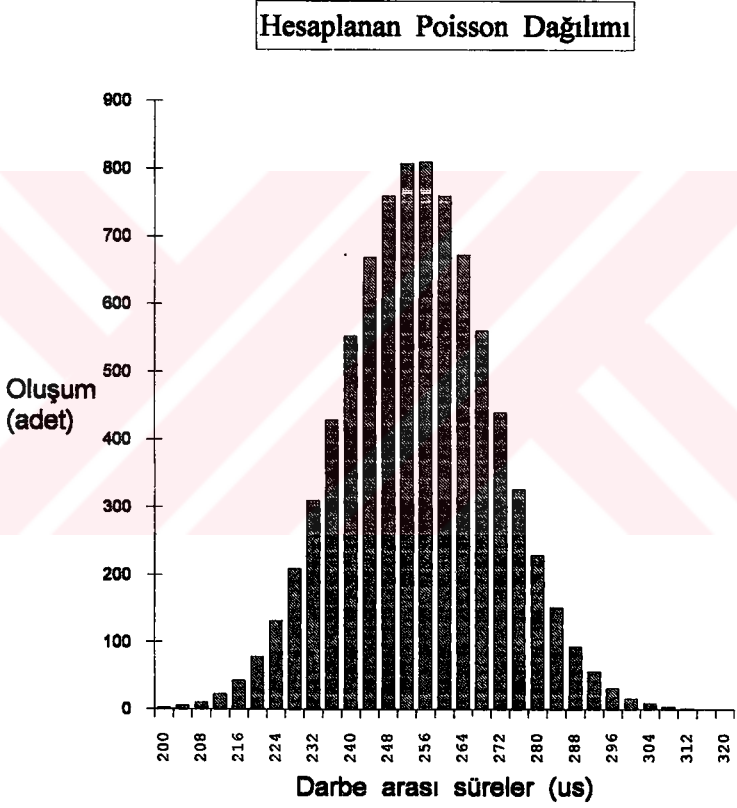


```
Dispose(Dialog, Done);
end;
procedure TGosterUyg.Yard;
var
  FileName: PathStr;
  W: PWindow;
begin
  FileName:='PRNGDEST.HLP';
  W := PWindow(ValidView(New(PFileWindow,Init(FileName))));
  if W <> nil then Desktop^.Insert(W);
end;
{*****}
{*****      ANA PROGRAM      *****}
{*****}
var
  GosterUyg: TGosterUyg;
begin
  with TDialogData do
  begin
    GeriBesFonk := 7168;
    EPROMtipi := 0;
    ZamanTaban := 0;
    BellekBoyutu := '8192';
    ParcaSay :='128';
    OrtDeg :='128';
  end;
  KurSart;
  GosterUyg.Init;
  GosterUyg.Run;
  GosterUyg.Done;
end.
```

T.C. YÜKSEKÖĞRETİM
DOKÜMANTASYON MERKEZİ

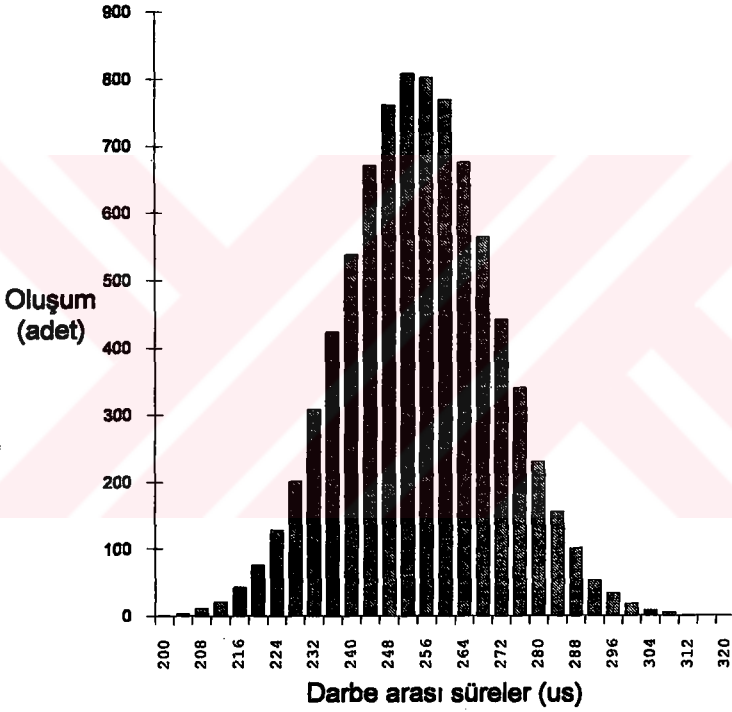
Üretilmesi tasarlanan 512 parça ayırıcılık ile 8192 örnekten oluşan, ortalama darbe arası süresi 256µs, zaman tabanı 1µs olan Poisson dağılımına sahip yalancı rastgele işaret, tasarım yazılımı ile hesaplanır (Şekil 5.1 1.işlem). Yazılımın çalışması sonucu elde edilen EPROM bilgileri bir EPROM programlayıcı kullanılarak, tasarım yazılımında seçilen uygun EPROM tümdevresine aktarılır (Şekil 5.1 2.işlem). Tümdevreler üreteç sistemindeki soketlerine takılır (Şekil 5.1 3.işlem). Üreteç sistemindeki bağlantı dizileri tasarım yazılımda belirlenen şekilde şartlandırılır (Şekil 5.1 4.işlem). Böylece üreteç sisteminin tasarlanan özelliklerdeki işareti bağımsız olarak üretmesi sağlanır.

Üçüncü bölümde anlatılan ölçüm sistemini oluşturmak üzere, bilgisayar kartı bağlantı dizileri uygun şekilde şartlandırılarak üretilen işareti ölçecek biçime getirilir (Şekil 5.1 5.işlem). Üreteç sisteminin çıkış işareti ölçüm kartına bağlanır (Şekil 5.1 6.işlem). Ölçüm yazılımı çalıştırılır (Şekil 5.1 7.işlem). Yazılım giriş parametrelerine bağlı olarak aldığı ölçümleri, virgül ile ayrılmış değerler (CSV) biçiminde yazısal dosya olarak disk üzerinde saklar. Elde edilen bu değerler, matematik ve istatistik işlemler yapabilen bir program ile okunarak şekillendirildikten sonra değerlendirilebilir (Şekil 5.1 8.işlem) [57,58]. Burada histogramlar bir tablolama programı kullanılarak 32 kanal olarak çıkarılmıştır [56,57]. Histogramların yazıcı çıktıları yine aynı tablolama programı ile düzenlenerek elde edilmiştir (Şekil 5.1 9.işlem). Üretilmesi hesaplanan işaretin histogramı Şekil 5.2'de , ölçülen işaretin histogramı ise Şekil 5.3'de gösterilmiştir.



Şekil 5.2 Hesaplanan Poisson dağılımın histogramı.

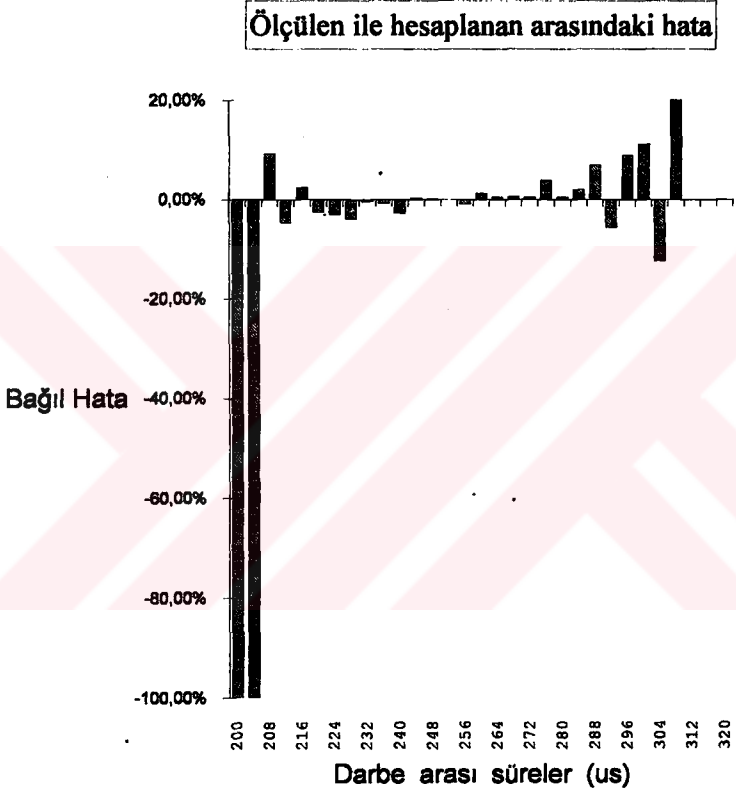
Ölçülen Poisson Dağılımı



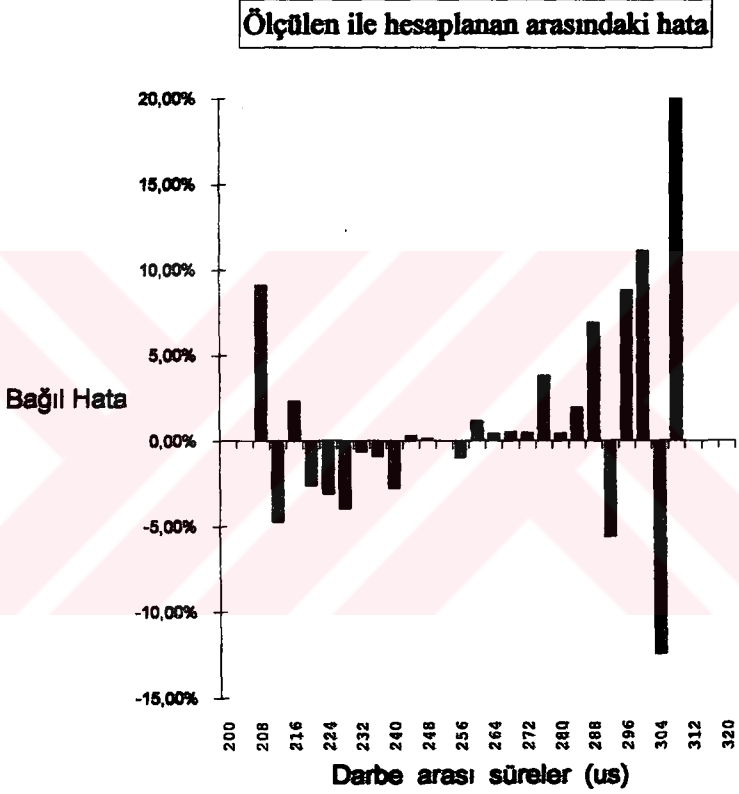
Şekil 5.3 Ölçülen işaretin histogramı.

Darbe arası	Ölçülen	Hesaplanan	mutlak hatı	bağıl hata
200	1	2	-1	-100.00%
204	3	6	-3	-100.00%
208	11	10	1	9.09%
212	21	22	-1	-4.76%
216	43	42	1	2.33%
220	76	78	-2	-2.63%
224	128	132	-4	-3.13%
228	201	209	-8	-3.98%
232	308	310	-2	-0.65%
236	424	428	-4	-0.94%
240	538	553	-15	-2.79%
244	671	689	2	0.30%
248	781	760	1	0.13%
252	808	808	0	0.00%
256	802	810	-8	-1.00%
260	769	760	9	1.17%
264	676	673	3	0.44%
268	584	561	3	0.53%
272	442	440	2	0.45%
276	340	327	13	3.82%
280	230	229	1	0.43%
284	155	152	3	1.94%
288	101	94	7	6.93%
292	53	56	-3	-5.68%
296	34	31	3	8.82%
300	18	16	2	11.11%
304	8	9	-1	-12.50%
308	5	4	1	20.00%
312	1	1	0	0.00%
316	0	0	0	
320	0	0	0	

Tablo 5.1 Darbe arası sürelerle göre ölçülen ve hesaplanan histogramların arasındaki, mutlak ve bağıl hatanın sayısal değerleri.



Şekil 5.4 Bağıl hata grafiği.



Şekil 5.5 Bağıl hata grafiğinin orta bölümünün ayrıntılı görünümü.

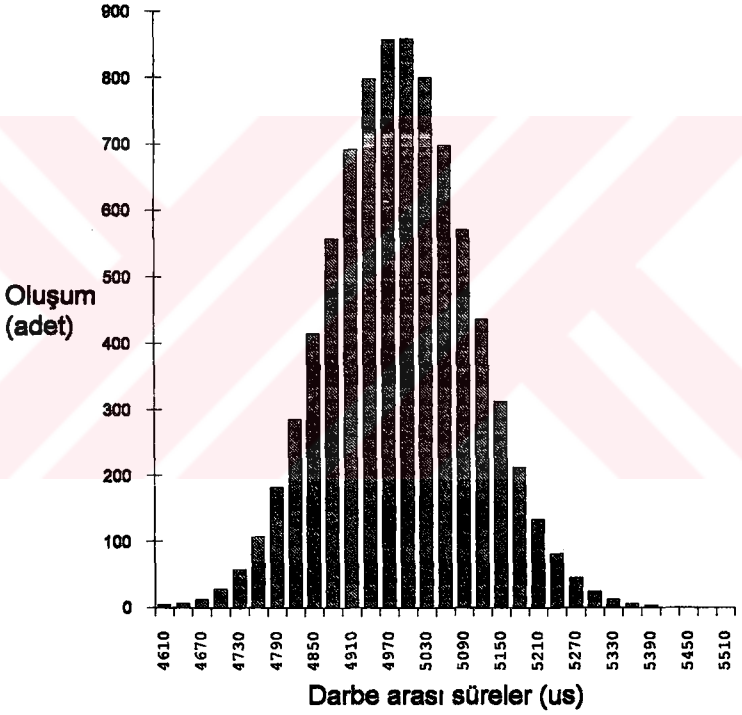
Bu iki histogram karşılaştırıldığında ölçülen işaretin histogramının hesaplanan Poisson dağılımına uygun olduğu görülmektedir. Hesaplanan ve ölçülen işaretlerin histogramlarını oluşturan sayısal değerler ve bu iki histogram arasındaki mutlak ve bağıl hata değerleri hesaplanarak Tablo 5.1'de verilmiştir. Ölçülen işaretin histogramının hesaplanan işaretin histogramına göre bağıl hatasını gösteren grafik Şekil 5.4'de verilmiştir. Bağıl hata değişiminin daha iyi görülebilmesi için grafiğin orta kısmı Şekil 5.5'de gösterilmiştir.

Burada histogramın ortalama değerinden uzaklaştıkça bağıl hatanın arttığı görülmektedir. Bağıl hata ortalama değer etrafında ± 1 'i geçmediği, uzaklaştıkça ± 20 'ye kadar çıktığı görülmektedir.

Bağıl hatanın nereden kaynaklandığının daha iyi yorumlanabilmesi için, ölçüm sisteminden daha doğru yanıt alınabilen, darbe arası sürelerin ortalama değeri 5ms seçilerek yapılan işlemler tekrarlanmıştır.

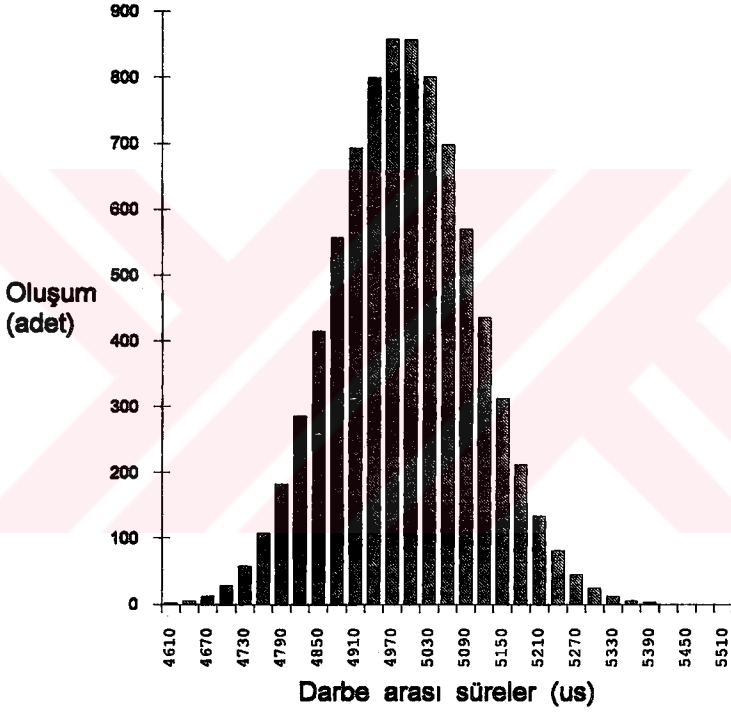
Tasarımda 256 parça ayırıcılık ile 8192 örnekten oluşan poisson dağılımı hesaplanmıştır. Hesaplanan işaretin histogramı Şekil 5.6'da ve ölçülen işaretin histogramı ise Şekil 5.7'de verilmiştir. Hesaplanan ve ölçülen değerlerin histogramlarının birbirine uygun olduğu görülmektedir. Bu iki histogramın ve hataların sayısal değerleri ise Tablo 5.2'de görülmektedir. Bu tablodan yararlanılarak oluşturulan bağıl hata grafiği ise Şekil 5.8'de gösterilmiştir. Bu grafiğin orta bölümü ayrıntılı olarak Şekil 5.9'da verilmiştir.

Hesaplanan Poisson Dağılımı



Şekil 5.6 Hesaplanan Poisson dağılımın histogramı.

Ölçülen Poisson Dağılımı

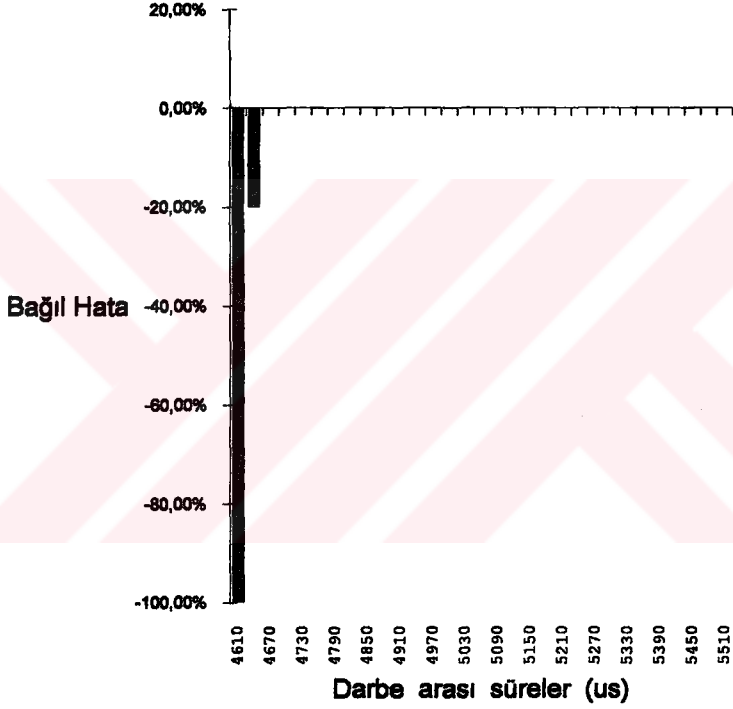


Şekil 5.7 Ölçülen işaretin histogramı.

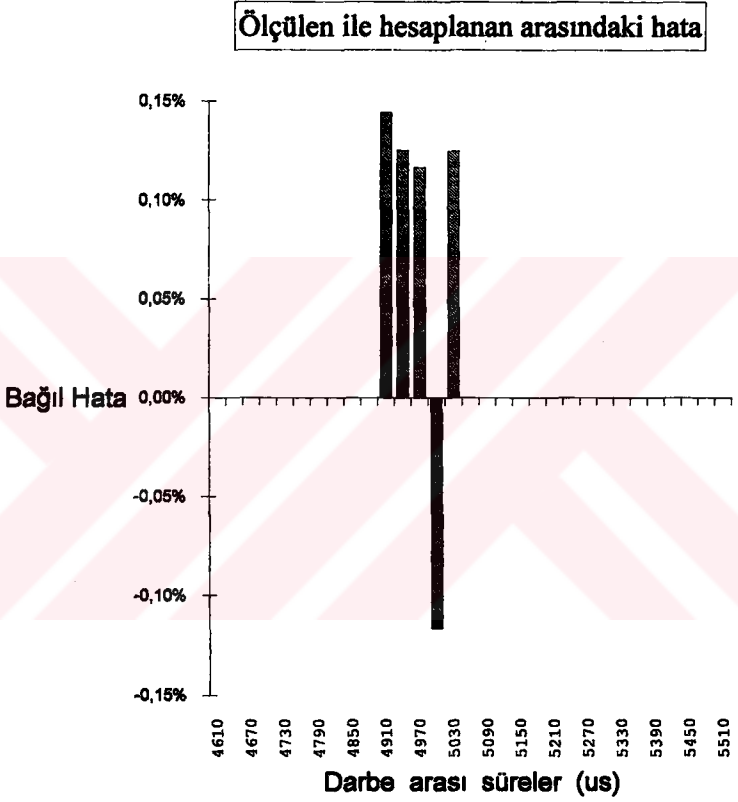
Darbe arası	Ölçülen	Hesaplanan	mutlak hata	bağıl hata
4610	2	4	-2	-100,00%
4640	5	6	-1	-20,00%
4670	12	12	0	0,00%
4700	28	28	0	0,00%
4730	57	57	0	0,00%
4760	107	107	0	0,00%
4790	182	182	0	0,00%
4820	286	286	0	0,00%
4850	415	415	0	0,00%
4880	557	557	0	0,00%
4910	693	692	1	0,14%
4940	799	798	1	0,13%
4970	858	857	1	0,12%
5000	857	858	-1	-0,12%
5030	801	800	1	0,12%
5060	698	698	0	0,00%
5090	570	570	0	0,00%
5120	436	436	0	0,00%
5150	313	313	0	0,00%
5180	212	212	0	0,00%
5210	134	134	0	0,00%
5240	81	81	0	0,00%
5270	45	45	0	0,00%
5300	24	24	0	0,00%
5330	12	12	0	0,00%
5360	5	5	0	0,00%
5390	3	3	0	0,00%
5420	0	0	0	
5450	0	0	0	
5480	0	0	0	
5510	0	0	0	

Tablo 5.2 Darbe arası sürelerle göre, ölçülen ve hesaplanan histogramların arasındaki, mutlak ve bağıl hatanın sayısal değerleri.

Ölçülen ile hesaplanan arasındaki hata



Şekil 5.8 Bağıl hata grafiği.



Şekil 5.9 Bağıl hata grafiğinin orta bölümünün ayrıntılı görünümü.

Bağıl hata grafikleri ile birlikte tablodaki değerler incelendiğinde bağıl hatanın ortalama değer etrafında ± 0.1 düzeyinde olduğu, bir çok eksen değerinde ise hatanın olmadığı görülmektedir. Bu sonuç daha önce elde edilen bağıl hata grafiği karşılaştırıldığında çok daha iyi bir sonuç elde edilmiştir. Bu sonuca dayanılarak tasarlanan işaretin istenen yeterlikte üretilebildiği, ölçüm sisteminin ise düşük yanıt süresinde istenen doğruluk ve ayırtıcılıkta ölçüm yapabildiği söylenebilir.

6. SONUÇ

Rastgele olaylar ile ilgili ölçüm ve üretim yöntemleri çeşitli olmakla birlikte bunların özgün çalışmalar sonucu üretilen yöntemler olması ilgi çekicidir. Bu ortamdan kuvvet alınarak, yapılan çalışmalarda konu ile araştırmalardan da faydalanılarak, özgün yazılım ve donanım yöntemi oluşturularak, problemin çözümünde önemli ölçüde yol alındığı görülmektedir.

Tasarlanarak fiziksel olarak gerçekleştirilen, kişisel bilgisayar kartı ve bu kartın çalışması için yazılmış olan yazılımdan oluşan ölçüm sistemi ile yine tasarlanarak fiziksel olarak gerçekleştirilen, üreteç tasarım yazılımı ve donanımından oluşan üreteç sisteminin çıkış işareti ölçülmüş ve alınan ölçümler istatistiksel ve matematik analiz ve grafik yeteğine sahip program paketleri kullanılarak incelenmesi ve değerlendirilmesi sonucunda istenen üretim ve ölçümün gerçekleştiği görülmüştür.

Her iki sisteminde elektronik devresinin tasarımı, analizi ve baskılı devresinin tasarımı bütünüyle elektronik devrelerin çizim ve analizini yapan hazır yazılım paketleri kullanılarak bilgisayar destekli tasarım yapılmıştır [59, 60,61]. Bu şekilde bir elektronik sistemin kişisel bilgisayar ile tasarımı için gerekli olan yazılım ve donanımın uygun şekilde kullanımı sağlanmıştır.

Nükleer elektronik alanında yapılan çalışmalar, bu konuda daha önce yapılan çalışmalar hakkında çok bilgi elde edilememesi ve bu tip çalışmaların yapılabilmesi için gerekli olan yazılı döküman ve laboratuvar olanaklarının kısıtlı olması nedeniyle çok zor olmaktadır. Bütün bu zorluklar kişisel gayret ve maddi olanaklarla aşılmaya çalışılmıştır. Bunun sonucunda ulaşılan nokta, ilgilenilen

konu üzerinde yapılan çalışmalar ile karşılaştırıldığında doğal olarak teknolojik olarak üstünlük sağlanamaması dışında, problemin çözümüne karşı geliştirilen özgün düşünce ve çalışma yöntemi açısından olumlu sonuçlar elde edilmiştir. Ayrıca yapılan çalışmalar temel düzeyden ele alınarak geliştirildiği için bu konuda çalışacak olanlara referans olacak şekilde sunulmuştur.



REFERANSLAR

- [1] P. W. Nicholson, "Nuclear Electronics", John Wiley and Sons, 1974.
- [2] H.Taub, D.Schilling, "Digital Integrated Electronics", Mc-Graw Hill, 1981.
- [3] Glenn F.Knoll, "Radiation Detection and Measurement", John Wiley and Sons, 1979.
- [4] P. L. Meyer, "Introductory Probability and Statistical Applications", Addison-Wesley Pub. Co., London, 1970.
- [5] G. P. Wadsworth, J.G.Bryan, "Applications of Probability and Random Variables", McGraw-Hill , 1974.
- [6] A. Papoulis, "Probability Random Variables and Stochastic Processes", McGraw-Hill 1965.
- [7] O. Sasaki et al., "A Precision Pulse Generator for TDC calibration", IEEE Trans. Nucl. Sci., Vol.36, No 1, February 1989, 416.
- [8] H. Ohgaki et al., "A Ratio-To-Digital Converter for Charge Division Position Sensing", IEEE Trans. Nucl. Sci. Vol NS-34, No 1, February 1987, 110.
- [9] M. Matoba et al., "Realization of the Concept of the Ratio-To-Digital Converter (RDC)", IEEE Trans. Nucl. Instr. and Meth., 224(1984), 173.
- [10] G. P. Westphal, IEEE Trans. Nucl. Instr. and Meth., 134(1979), 387.
- [11] B. E. Fischer, IEEE Trans. Nucl. Instr. and Meth., 141(1977), 173.
- [12] M. Atac, F. Bedeschi, IEEE Trans. Nucl.Sci. NS-29, 1982, 396.
- [13] C. W. Sobczynski et al., "25ps Resolution, 12-Bit, 64 Channel Fastbus Time-To-Digital Converter", IEEE Trans. Nucl.Sci. Vol 36, February 1989, 426.
- [14] J. Kostamovaara, R.Myllyla, "A Time-To-Amplitude Converter With Constant Fraction Timing Discriminators For Short Time Interval Measurement", Nucl. Inst. and Meth. A239 1985, 568.
- [15] F. Bourgeois et al., IEEE Trans. Nucl. Sic. Vol. NS-25, 1985.
- [16] F. Bourgeois et al., "High-Density CAMAC And FASTBUS Charge Integrating ADC For Use At The LEP OPAL Detector ", IEEE Trans. Nucl. Sci. Vol. NS-34, No 1, February 1987, 240.

- [17] O. Sasaki et al., "1.2GHz GaAs Shift Register IC for Dead-Time-Less TDC application", IEEE Trans. Nucl. Sci. Vol 36, No 1, February 1989, 512.
- [18] M.M. Morii et al., "Low noise Fast Bipolar Transistors for Dedector preamplifiers", IEEE Trans. Nucl. Sci.
R. A. Boie et al., IEEE Trans. Nucl. Sci. NS-28, 1981, 603.
- [19] S. Eiseman et al., IEEE Trans. Nucl. Sci. NS-30, 1983, 149.
W. Farr et al., Nucl. Instru. and Meth. 190, 35(1981).
- [20] Lecroy Models 1878/1879, "96 channel pipeline TDC".
- [21] O. Sasaki et al., "A High Resolution TDC in TKO BOX System", IEEE Trans. Nucl. Sci. Vol 35, No 1, February 1988, 342.
- [22] Y. Arai et al., "TMC - A CMOS Time-To-Digital Converter VLSI", IEEE Trans. Nucl. Sci. Vol 36, No 1, February 1989, 528.
- [23] Y. Arai and T. Ohsugi, "An Idea of Deadtimeless Read-out System by using Time Memory Cell", Proceedings of the 1986 Summer Study on the Physics of the Superconducting Super Collider, p455. KEK Preprint 86-64(1986).
- [24] Y. Arai and T. Ohsugi, "TMC: A Low-Power Time to Digital Converter LSI", IEEE NS Symposium, Oct. 1987. KEK Preprint 87-113(1987).
- [25] Y. Arai and T. Baba, "A CMOS Time to Digital Converter VLSI for High-Energy Physics", 1988 Symposium on VLSI Circ., Aug. 1988/Tokyo, IEEE CAT. No.88, TH 0227-9, p121 .
- [26] IBM Technical Reference, IBM, 1983.
- [27] IBM PC & AT Technical Reference, IBM, 1985.
- [28] James R. Drummond, "Three Bus Interface Designs for the PC", BYTE "Inside the IBM PCs" Volume 12, Number 12, McGraw-Hill, 1987.
- [29] Russell Rector, George Alexy "The 8086 Book", Osborne/McGraw-Hill 1980.
- [30] "Microprocessor and Peripheral Handbook", Vol. 1 Microprocessor, Intel, 1987.
- [31] "Microprocessor and Peripheral Handbook", Vol. 2 Peripheral, Intel, 1987.
- [32] John Uffenbeck, "Microcomputers and Microprocessors: The 8080, 8085, and Z-80 Programming, Interfacing, and Troubleshooting", Prentice-Hall, 1985.
- [33] "Turbo C Reference Guide 2.0", Borland, 1988.
- [34] Herbert Schild, "Turbo C/C++ The Complete Reference", Osborne McGraw-Hill 1990.

- [35] Herbert Schild, "C Made Easy", Osborne McGraw-Hill, 1985.
- [36] Herbert Schild, "Advance Turbo C", Osborne McGraw-Hill, 1987.
- [37] Leo J. Scanlon, "IBM PC & XT Assembly Language", BRADY, 1983.
- [38] Peter NORTON, John SOCHA, "Peter Norton's Assembly Language Book for the IBM PC", BRADY, 1989.
- [39] "Turbo Assembler Reference Guide 2.0", Borland, 1990.
- [40] "Turbo Assembler User's Guide 2.0", Borland, 1990.
- [41] Ian Hickman, "Making a right white noise", Electronics World + Wireless Word, March 1992.
- [42] MOS Databook, National Semiconductor, 1980.
- [43] LS/S/TTL Databook, National Semiconductor, 1989.
- [44] K. Zeng, C.H. Yang, D-Y. Wei and T.R.N. Rao, "Pseudorandom Bit Generators in Stream-Chipher Cryptography", IEEE Computer Proceedings, February 1991.
- [45] K.C. Zeng, M.Q. Huang, "On the linear Syndrome Method in Cryptanalysis", Proc. Crypto '88, Springer-Verlag Lecture Notes in Computer Science, No 403, New York, 1988, pp. 469-478.
- [46] K.C. Zeng, C.H. Yang, and T.R.N. Rao, "An Improved Linear Syndrome Algorithm in Cryptanalysis with Applications", Proc. Crypto '90, Springer-Verlag Lecture Notes in Computer Science, New York, 1990.
- [47] P.R. Geffe, "How to Protect Data with Ciphers that are Realy Hard to Break", Electronics, Vol. 46, No 1, Jan. 1973, pp 99-101.
- [48] S.M. Jennings, "Multiplexed Sequences:: Some Properties of the Minimum Polynomial", Proc. Workshop on Cryptography, No 149, Springer-Verlag Lecture Notes in Computer Science, New York, 1982, pp. 189-206.
- [49] T. Beth and F.C. Piper, "The Stop-and-Go Generator", Proc. Eurocrypt '84, Springer-Verlag Lecture Notes in Computer Science, New York, No 209, 1984, pp. 88-92.
- [50] W.G. Chambers and D. Gollmann, "Lock-in Effect in Cascades of Clock-Controlled Shift-Registers", Proc. Eurocrypt '88, Springer-Verlag Lecture Notes in Computer Science, New York, No 330, 1988, pp. 331-343.

- [51] J.L. Massey and R.A. Rueppel, "Linear Ciphers and Random -sequence Generators with Multiple Clocks", Proc. Eurocrypt '84, Springer-Verlag Lecture Notes in Computer Science, New York, No 209, 1984, pp. 74-87.
- [52] "IC Memory Products Databook", Hitachi, 1986.
- [53] Robert SEDGEWICK, "Algorithms", Addison-Wesley, 1988.
- [54] Stephen K.O'BRIEN, "Turbo Pascal 6.0 The Complete Reference", McGraw-Hill, 1991.
- [55] "Turbo Pascal Turbo Vision Guide Ver. 6.0", Borland, 1990.
- [56] Tom SHELDON, "Windows 3.1 The Complete Reference", McGraw-Hill, 1992.
- [57] "Microsoft EXCEL 4.0 Function Reference", Microsoft, 1992.
- [58] "MathCAD 3.1 User's Guide", MATHSOFT, 1992.
- [59] "P Spice User's Guide", MicroSim Corp., 1989.
- [60] "OrCAD Schematic Design Tools User's Guide", OrCAD, 1989.
- [61] "OrCAD PCB II User's Guide", OrCAD, 1989.

T.C. YÜKSEKÖĞRETİM KURULU
DOKÜMANTASYON MERKEZİ

ÖZGEÇMİŞ

Doğum tarihi	20.08.1963
Doğum yeri	Şile (İstanbul)
Eğitim	Bartın Cumhuriyet İlkokulu (1973-1974), Üsküdar Halide Edip Adıvar Ortaokulu (1976-1977), Kartal Endüstri Meslek Lisesi, Elektronik Bölümü (1979-1980), Yıldız Üniversitesi, Elektronik ve Haberleşme Mühendisliği (1984-1985), Yıldız Üniversitesi, F.B.E. Elektronik ve Haberleşme Yüksek Mühendisliği (1986-1987), Yıldız Teknik Üniversitesi, F.B.E. Elektronik ve Haberleşme Yüksek Mühendisliği Anabilim Dalında Doktor Mühendis (Ocak 1994).
İş Deneyimleri	Mayıs 1986'dan bugüne kadar, Yıldız Teknik Üniversitesi, Elektrik-Elektronik Fakültesi, Elektronik ve Haberleşme Mühendisliği Bölümü Devreler ve Sistemler Anabilim Dalında Araştırma Görevlisi olarak çalışmaktadır.
Yabancı Dil	İngilizce
Çalışma Alanları	Kişisel bilgisayar donanımı ve yazılımı, bilgisayarlı ölçme ve kontrol sistemleri, mikroislemciler, mikrodenetleyiciler, programların denetleyiciler, programlama dilleri konularında çalışmaları bulunmaktadır.

