İSTANBUL TEKNİK ÜNİVERSİTESİ
ELEKTRİK - ELEKTRONİK FAKÜLTESİ

KRİPTOLOJİ UYGULAMALARINDA KULLANILACAK BİR MİKROİŞLEMCİNİN FPGA ÜZERİNDE GERÇEKLENMESİ

BİTİRME ÖDEVİ

BETÜL BAYRAKTAR

040080400

Bölümü : Elektronik ve Haberleşme Mühendisliği
Programı : Elektronik Mühendisliği

Danışmanı : Doç. Dr. Sıddıka Berna ÖRS YALÇIN

OCAK 2014

ÖNSÖZ

Bitirme çalışmam boyunca yardımlarını esirgemeyen ve değerli vaktini ayıran tez danışmanım Doç. Dr. Sıddıka Berna ÖRS YALÇIN' a, aynı zamanda her zaman yanımda olan ve eğitim hayatım boyunca her türlü özveriyi göstermiş olan canım aileme çok teşekkür ederim.

Bana insan olmanın gerekliliklerini ve karşılıksız sevgiyi öğreten, beraber büyüdüğüm GÖNÜLLÜLÜK KLÜBÜ'ndeki tüm gönüllü arkadaşlarıma, öğrencilerime, hep yanımda olan ev arkadaşlarıma ve ayrıca benim bu üniversiteyi kazanmamı sağlayan Ali YILDIZ hocama teşekkürü borç bilirim.

Ocak, 2014 Betül BAYRAKTAR

İÇİNDEKİLER

KISALTMALAR

ACI : Adjacent Channel Interference (Komşu Kanal Karışması)

AWGN : Additive White Gaussian Noise (Toplanır Beyaz Gürültü Kanalı)

BER : Bit Error Rate (Bit Hata Oranı)

BPF : Band Pass Filter (Bant Geçiren Süzgeç)

CP : Cyclic Prefix (Çevrimsel Önek)

CS : Cyclic Suffix (Çevrimsel Sontakı)

DAB : Digital Audio Broadcasting (Sayısal Ses Yayıncılığı)

DFT : Discrete Fourier Transform (Ayrık Fourier Dönüşümü)

DVB-T : Digital Video Broadcasting Terrestrial (Sayısal Karasal Video Yayıncılığı)

FDD : Frequency Division Duplexing (Frekans Bölmeli İkileme)

FFT : Fast Fourier Transform (Hızlı Fourier Dönüşümü)

FPGA : Field Programmable Gate Array (Sahada Programlanabilir Kapı Dizileri)

ICI : Inter Carrier Interference (Ara Taşıyıcı Girişimi)

IP : Internet Protocol (İnternet Protokolü)

ISI : Intersymbol Interference (Simgelerarası Girişim)

ISM : Industrial, Scientific and Medical (Endüstriyel, Bilimsel ve Tıbbi)

ITU : International Telecommunications Union (Uluslararası Telekomünikasyon Birliği)

ITU-T : International Telecommunications Union Telecommunication Standardization Sector (Uluslararası Telekomünikasyon Birliği Telekomünikasyon Standartlaştırma Birimi)

LTE : Long Term Evolution

MB-OFDM : Multi-band Orthogonal Frequency Division Multiplexing (Çoklu bant Dik Frekans Bölmeli Çoğullama)

MIMO : Multiple Input Multiple Output (Çoklu Giriş Çoklu Çıkış)

MOS : Mean Opinion Score (Ortalama Yargı Değeri)

OFDM : Orthogonal Frequency Division Multiplexing (Dik Frekans Bölmeli Çoğullama)

QAM **:** Quadrature Amplitude Modulation (Dördün Genlik Modülasyonu)

PESQ **:** Perceptual Evaluation of Speech Quality (Ses Kalitesinin Algısal Analizi)

PSK **:** Phase Shift Keying (Evre Kaydırmalı Kiplenim)

POLQA **:** Perceptual Objective Listening Quality Assessment (Algısal Nesnel Dinleme Kalitesi Değerlendirmesi)

RC **:** Raised Cosine (Yükseltilmiş Kosinüs)

SISO **:** Single Input Single Output (Tek Giriş Tek Çıkış)

SNR **:** Signal to Noise Ratio (İşaret Gürültü Oranı)

STO **:** Symbol Time Offset (Sembol Zaman Kaydırma)

TCP **:** Transmission Control Protocol (Gönderim Kontrol Protokolü)

USRP **:** Universal Software Radio Peripheral (Yazılım Tabanlı Radyo Kiti)

UWB **:** Ultra Wide-Band (Pek Geniş Bant)

VDSL **:**Very-high-bit-rate Digital Subscriber Line (Çok Yüksek Hızda Sayısal Abone Hattı)

W-CDMA **:**Wideband Code Division Multiple Access (Geniş Bant Kod Bölmeli Çoklu Erişim)

VoIP **:** Voice Over Inter Protocol (IP üzerinden Ses İletimi)

ZP **:** Zero Padding (Sıfır Dolgulaması)

ŞEKİL LİSTESİ

Sayfa No

ÖZET

Mikroişlemciler, günümüz elektronik teknolojisinde sağladıkları performans sayesinde geniş bir kullanım alanına yayılmıştır. Makine koduyla kodlanan işlemciler, temel aritmetik işlemlerinin yanısıra birçok işlemi de çok kısa sürelerde gerçekleştirmektedir.

Veri alışverişinin çok kolay olması ve bilgi güvenliğinin azalmasıyla kriptoloji alanı da önemini arttırmıştır. 2001 yılında kabul gören AES algoritması, kriptolojide en çok kullanılan şifreleme algoritmasıdır.

Bu çalışmada, kriptoloji uygulamalarında kullanılabilecek FPGA' de (Field Programmable Gate Array) tasarlanmış bir mikroişlemci gerçeklenmesi anlatılmaktadır. Bunun için 8-bit bir RISC işlemci kullanılmıştır. Bu mikroişlemci içerisinde S-Box ve Ters S-Box yapıları bellekten yazılan bir tablodan okunmaktadır. Bu tablo 16x16 'lık bir matristir ve hafızada çok fazla yer kaplamaktadır. Bu çalışmanın amacı, kullanılan hafızayı azaltmaktır. Bu işlem de S-Box ve Ters S-Box yapıları bloklarının kombinezonsal olarak gerçeklenmesiyle sağlanacaktır.

Tezde, temel olarak AES algoritması, mikroişlemci yapıları, S-Box devre yapısı ve bitirme çalışmasında yapılan işlemler anlatılacak ve bitirme çalışmasında kullanılan mikroişlemci üzerinde yeni tasarlanan modüller değiştirilerek hız ve alan karşılaştırması yapılacaktır.

SUMMARY

Microprocessors are widely used in todays electronical technology because of their good performances. Processors that are coded with machine code realize so many process in an very short time alongside basic aritmethic calculations.

Crytptology has an increasing importance because data exchange is really easy and security of information is decreased. AES algorithm that is astablished in 2001 is mostly used in criptology to cipher.

In this Project, design and implementation of a cryptographic ASIP on FPGA will be discussed and a 8-bits RISC processor is used . S-Box and Inverse S-Box modules are read from Look Up Table. Look Up Table is a matrice that has dimensions 16x16 and is very big for the memory. The purpose of this Project is reducing loaded memory in the processor. To do this, S-Box and Inverse S-Box modules will design as combinational logic circuits.

In this thesis, AES algorithm, microprocessor structure, S-Box circuit schema and processes that have done in the final Project will be discussed and new modules will be exchanged in processor. Then frequency and memory will be compared with respect to old microprocessor and new one.

1. GİRİŞ

Mikroişlemciler ilk bulundukları zamandan itibaren gelişen teknolojiyle birlikte, kişisel bilgisayarlardan (Personal Computer - PC) cep telefonlarına, hesap makinelerinden otomobillere, TV alıcılarına kadar birçok alanda kendine yer bulmuştur ref. İşlemcilerin keşfedilmesi 1940'lara denk gelmektedir. İlk bilgisayar işlemcisi, büyük elektron tüpleri ile yapılmıştır ref. Yarıiletken teknolojisinin gelişmesiyle daha küçük, daha hızlı ve daha az güç tüketen işlemciler yani mikroişlemciler ortaya çıkmıştır. Geçmişte elle yapılmaya çalışılan birçok işlem bugün mikroişlemcilerle çok hızlı ve çok kolay bir biçimde yapılabilmektedir. Hatta bugün insan eliyle yapılamayacak birçok işlem milisaniyelerden çok daha kısa zamanda gerçekleşmektedir. Bu yüzden, mikroişlemcilerin bulunuşu elektronik dünyası için çok önemli bir dönüm noktasıdır.

Mikroişlemciler, uygulama alanlarına göre özellikleri bakımından çeşitlilik göstermektedir. Temel iki tip mikroişlemciden biri Genel Amaçlı İşlemci (General Purpose Processor - GPP) ve diğeri Uygulamaya Özel Tümleşik Devre (Application Specific Integrated Circuit - ASIC)'dir. GPP, kullanım kolaylığı nedeniyle birçok uygulamada kullanılmaktadır ancak bazı uygulamalara entegresi çok iyi performans göstermemektedir ref. Bunlara bir örnek kriptografi uygulamalarıdır. Uygulamaya Özel Tümleşik Devre (Application Specific Integrated Circuit - ASIC) işlemciler ise, kullanım amacına göre çok daha avantajlıdır ancak bu işlemcilerin de yapımı çok maliyetlidir ref. Bu iki işlemcinin de dezavantajlarına çözüm olarak Uygulamaya Özel Komut Setli İşlemci (Application Specific Instruction Set Processor - ASIP) geliştirilmiştir ref. Bu tip işlemciler, kullanılacak uygulamaya yönelik tasarıma olanak sağlamaktadır. ASIP işlemciler, Alan Programlamalı Kapı Dizisi (Field Programmable Gate Array - FPGA) ile yapılandırılabilir. ASIP işlemcilerin, yazılım destekli donanım yapısında olması performans açısından diğer işlemcilere göre birçok avantaj sağlamaktadır.

Elektronik teknolojisinin gelişimiyle bilgisayar, cep telefonu, akıllı kartlar günlük hayatın vazgeçilmezleri olmuştur ref. Bu cihazlar hem veri depolamakta, hem verileri işleyerek iletişim kanalları yardımıyla başka elektronik cihazlara iletmektedir. Ancak bu durum ikinci şahıslarca verilere iletişimi çok kolay hale getirmekte ve güvenlik açısından tehdit yaratmaktadır.

Güvenlik, yaygın olarak kullanılan gömülü sistemlerde aranan bir özelliktir ref. Bu amaçla kriptoloji fonksiyonlarını içeren ve güvenlik uygulamalarını gerçekleştiren ASIP işlemciler performans açısından en uygun işlemcilerdir.

Bu bitirme çalışmasında, kriptoloji uygulamalarında kullanılabilen bir ASIP mikroişlemcinin FPGA üzerinde tasarlanması amaçlanmıştır. En çok kullanılan ve en yaygın şifreleme algoritması olan Gelişmiş Şifreleme Standardı (Advanced

Encryption Standard - AES) kullanılmıştır ref. Şifreleme işlemi yapılırken, AES algoritmasındaki işlemlerden olan bayt değiştirmede S-Box değerleri Başvuru Çizelgesi (Lookup Table - LUT)' den okunmak yerine FPGA' de kombinezonsal devre ile gerçeklenerek oluşturulmuştur.

Tezin 2. Bölümünde AES şifreleme algoritmasının tarihi, şifreleme işlemleri, ters şifreleme işlemleri, anahtar üretilmesi anlatılmıştır. 3. Bölümünde mikroişlemcilerin doğuşuna kısa bir bakış atılmış, işlemciyi oluşturan temel birimler ile işlemci mimarileri ele alınmıştır. 4. Bölümde ise Galois Field yapıları ve S-Box 'ın kombinezonsal olarak gerçeklenmesi anlatılmıştır. 5. Kısım ise sonuç ve tartışmadır.

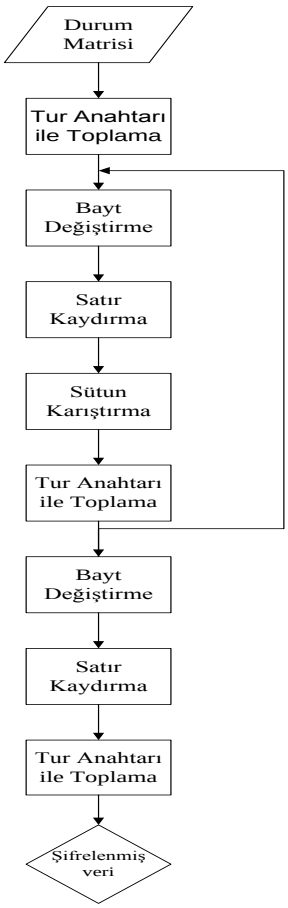
2. GELİŞMİŞ ŞİFRELEME STANDARDI

2.1. Giriş

Gelişmiş Şifreleme Standardı (Advanced Encryption Standard - AES), 2001 yılında NIST'in uzun halinin Türkçe'si (NIST'in uzun hali – NIST) tarafından şifreleme standardı olarak seçilmiştir ref. Bu standart, Belçikalı iki kriptografi uzmanı Joan Daemen ve Vincent Rijmen tarafından Rijndael algoritmasına göre geliştirilmiştir. Bu cümle yanlış Bu algoritma, daha uzun anahtar kullandığından Veri Şifreleme Standardı (uzun hal - DES)'e göre daha güvenlidir [1]. AES algoritmasında, girişe belirli işlemler uygulanır ve bu işlemler her turda farklı anahtar kullanılarak tekrarlanır. Tur sayısı ise anahtar uzunluğuna göre belirlenir.

2.2. AES Yapısı ile Şifreleme

AES bir blok şifreleme algoritmasıdır. Giriş ve çıkış olarak 128 bitlik veri kullanır. Anahtar uzunluğu, 128, 196 veya 256 bit olabilir. Bu yapılar, anahtar uzunluğu ile paralel olarak AES-128, AES-196 ve AES-256 şeklinde gösterilir. Her bir anahtar uzunluğu için ayrı tur sayıları belirlenmiştir.AES-128 için 10, AES-196 için 12, AES-256 için 14 tur uygulanır.



bu resim yanlış baklava dilimi kararlar için kullanılır.

Şekil 2.1 AES Blok Yapısı

Şekil 2.1'de AES algoritmasına ait adımlar blok şemada gösterilmiştir. Şemada bazı işlemler her turda tekrarlanmaktadır ve her turda farklı anahtar kullanılır. Giriş verisi, şifrelenmeden önce, 128 bit olduğundan her bir elemanı 8 bit (1 bayt) olan 4x4'lük bir matris haline getirilir [2]. Bu matrise durum matrisi denir. Giriş bitleri, 0'dan 125'e kadar devam etmekte ve bu bitler 8'er 8'er baytlara ayrılarak her baytta da yine bitler 0'dan 7'ye numaralanmaktadır [2]. Aşağıdaki şekilde bitler üzerinde yapılan numaralandırma işlemi gösterilmiştir.

Input bit sequence	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	…
Byte number	0								1								2								…
Bit numbers in byte	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	…

Şekil 2.2 Bit ve Bayt Numaralandırılması [2]

Giriş i, çıkış o ve durum matrisi de $s_{[}$ r, $c_{]}$ ile ifade edilirse; bitlerin numaralandırılması aşağıdaki gibi olur.

i_0	i_4	i_8	i_{12}
i_1	i_5	i_9	i_{13}
i_2	i_6	i_{10}	i_{14}
i_3	i_7	i_{11}	i_{15}

$s_{0,0}$	$s_{0,1}$	$s_{0,2}$	$s_{0,3}$
$s_{1,0}$	$s_{1,1}$	$s_{1,2}$	$s_{1,3}$
$s_{2,0}$	$s_{2,1}$	$s_{2,2}$	$s_{2,3}$
$s_{3,0}$	$s_{3,1}$	$s_{3,2}$	$s_{3,3}$

o_0	o_4	o_8	o_{12}
o_1	o_5	o_9	o_{13}
o_2	o_6	o_{10}	o_{14}
o_3	o_7	o_{11}	o_{15}

Şekil 2.3 Giriş Serisi, Durum Matrisi ve Çıkış Serisi

Durum matrisinin eldesinin daha iyi anlaşılması için bir örnek veri girişi ile işlem yapılabilir. Onaltılık tabanda 4F 36 52 65 81 A7 2C 12 4B 90 43 04 6E 25 DA 14 giriş kabul edilsin. Bu veriye ait durum matrisi aşağıdaki şekilde olur.

4F	81	4B	6E
36	A7	90	25
52	2C	43	DA
65	12	04	14

Şekil 2.4 Örnek Bir Durum Matrisi

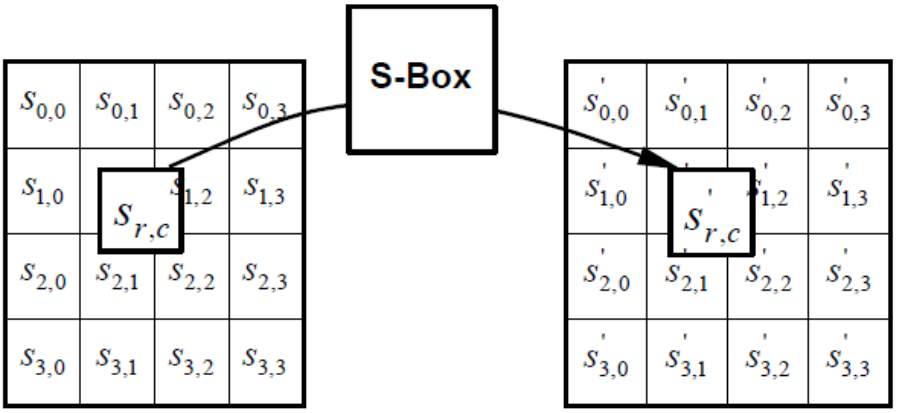
Durum matrisi elde edildikten sonra bu matrise, bayt değiştirme, satır kaydırma, sütun karıştırma, tur anahtarı ile toplama işlemleri uygulanır. Bu işlemlerin kaç defa uygulanacağı ise tur sayısı yani anahtar uzunluğuna bağlıdır. Anahtar uzunluğu ile tur sayısı arasındaki ilişki bu bölümün başında verilmiştir.

2.2.1 Bayt Değiştirme

Bayt değiştirme işlemi, durum matrisindeki herbir elemanın yani her $s_{[}r, c_{]}$ değerinin bir başkasıyla değişmesidir. Bu işlem, özel bir S-Box tablosu ile gerçekleştirilir. Her baytın, en yüksek ve en düşük anlamlı 4 bitine bakılır. Durum matrisindeki elemanlar S-Box tablosunda bu değerlere karşılık gelen değerlerle değiştirilir. Bayt değiştirme, Şifreleme işleminin tek nonlineer elemanı olarak ayrı bir öneme sahiptir. Şekil 2.5'te S-Box yapısı, Şekil 2.6'da ise bayt değiştirme işlemi gösterilmiştir.

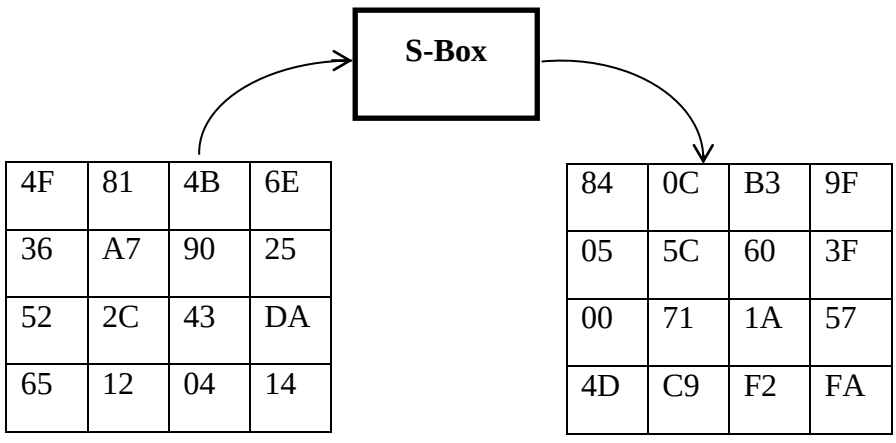
x \ y	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
d	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

Şekil 2.5 S-Box Tablosu (Onaltılık Tabanda) [2]



Şekil 2.6 Bayt Değiştirme İşlemi [2]

Şekil 2.4'deki durum matrisine bayt değiştirme işlemi uygulanırsa Şekil 2.7'deki matris elde edilir.



Şekil 2.7 Örnek Bayt Değiştirme

2.2.2 Satır Kaydırma

Satır kaydırma işleminde, ilk satır hariç her satır döngüsel kaydırma işlemine girer [2]. Basit şekilde, ilk satır sabit kalarak ikinci satır 1 birim, üçüncü satır 2 birim, üçüncü satır 2.4'deki örnek durum matrisi ise 2 birim sola ötelenir. Şekil şifreleme işlemine devam edilirse (Şekil 2.7'deki matriste satır kaydırma işlemi yapılırsa) Şekil 2.8 elde edilir.

$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix}$$

84	0C	B3	9F
05	5C	60	3F
00	71	1A	57
4D	C9	F2	FA

84	0C	B3	9F
5C	60	3F	05
1A	57	00	71
FA	4D	C9	F2

Şekil 2.8 Örnek Satır Kaydırma İşlemi

2.2.3 Sütun Karıştırma

Sütun karıştırma işlemi, durum matrisindeki her sütunun ayrı ayrı sabit bir matris ile çarpılmasıyla yapılmaktadır.

A =burası ne ?

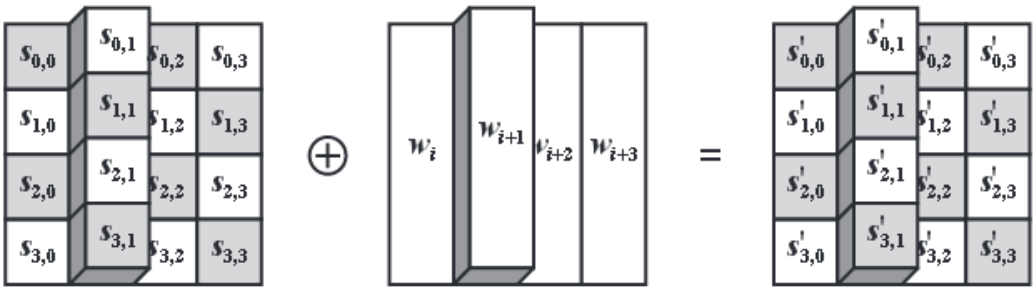
$$s' = A * s \qquad (2.1)\ [2]$$

(2.2) [2]

$$\begin{bmatrix} s'_{0,c} \\ s'_{1,c} \\ s'_{2,c} \\ s'_{3,c} \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} s_{0,c} \\ s_{1,c} \\ s_{2,c} \\ s_{3,c} \end{bmatrix}$$

2.2.4 Tur Anahtarı ile Toplama

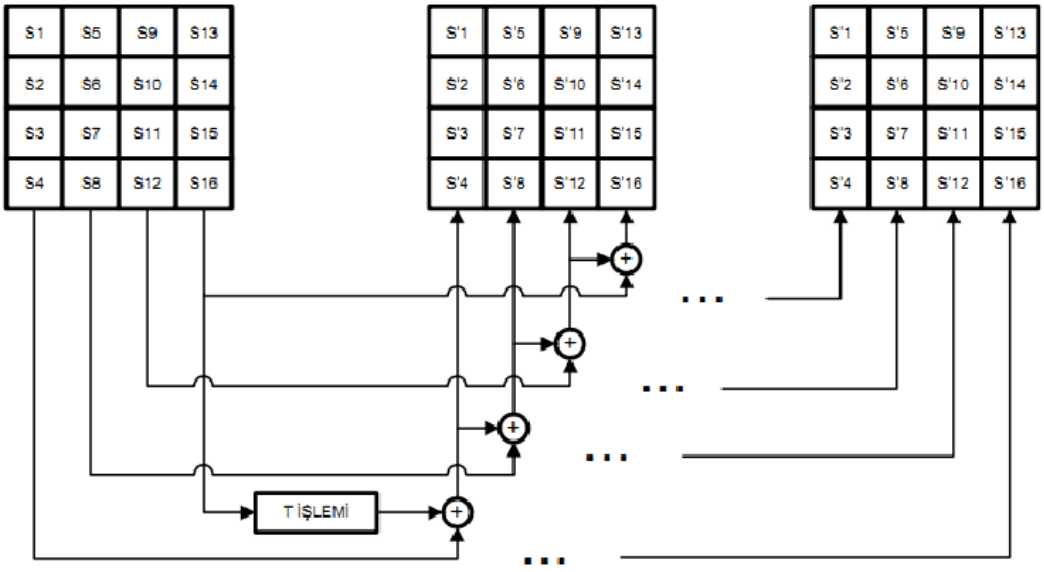
Durum matrisi ile her turda üretilen anahtar matrisinin toplanması, tur anahtarı ile toplama işlemidir. Buradaki toplama işlemi, xor işlemine karşılık gelmektedir.



Şekil 2.10 Tur Anahtarı ile Toplama İşlemi [3]

2.3 Tur Anahtarı ile Toplama İşlemi

AES şifreleme işleminde her turda farklı bir anahtar üretilir ve bu anahtar yalnızca o turdaki işlemlerde kullanılır. Bir sonraki tur için anahtar üretiminde o turdaki anahtar kullanılır. Aşağıdaki şekilde anahtar üretimi gösterilmiştir.



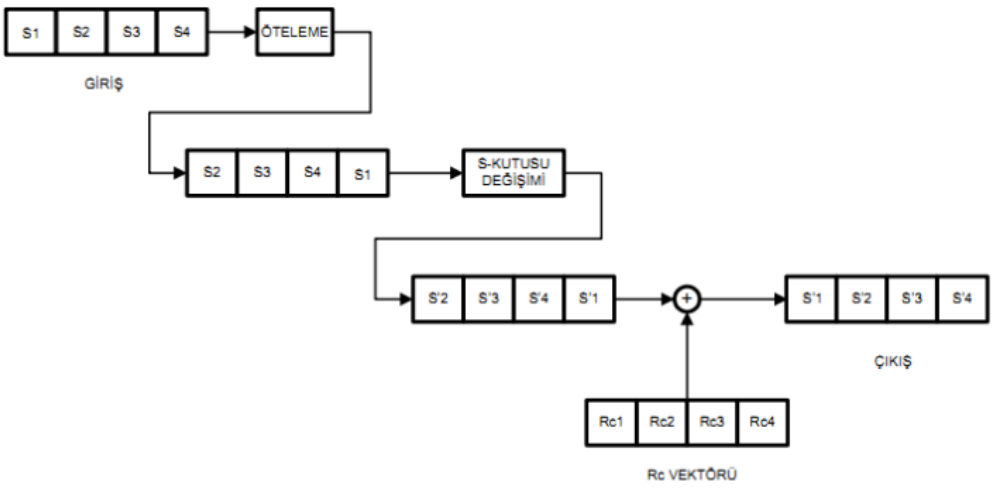
Şekil 2.11 Tur Anahtarı Üretimi [4]

Tur anahtarı üretiminde ilk sütun oluşturulurken ilk matrisin son sütunu bir T değişkenler kullanılırken equation yaz. Veya italik işleminden geçirilir ve daha sonra ilk sütunla toplanır yani xor'lanır. Sonrasında ilk matrisin ikinci sütunuyla yeni matriste oluşturan ilk sütunla toplanır ve bir sonraki anahtarın ikinci sütununu oluşturur. Bu işlem, diğer sütunlar için de devam eder.

Yukarıda bahsedilen T işlemi, öteleme, S-Box'dan geçirme ve Rc vektörü ile toplama (xor) işlemlerini içinde bulunduran bir bloktur. Rc vektörü tur sayısına göre değişiklik göstermektedir. Aşağıda öncelikle Rc vektörü tablosu ve daha sonra T işlemi blok yapısı gösterilecektir.

Tablo 2.1 Rc(x) Vektörleri [4]

Tur Sayısı	Rc Değeri
1	01 00 00 00
2	02 00 00 00
3	04 00 00 00
4	08 00 00 00
5	10 00 00 00
6	20 00 00 00
7	40 00 00 00
8	80 00 00 00
9	1B 00 00 00
10	36 00 00 00



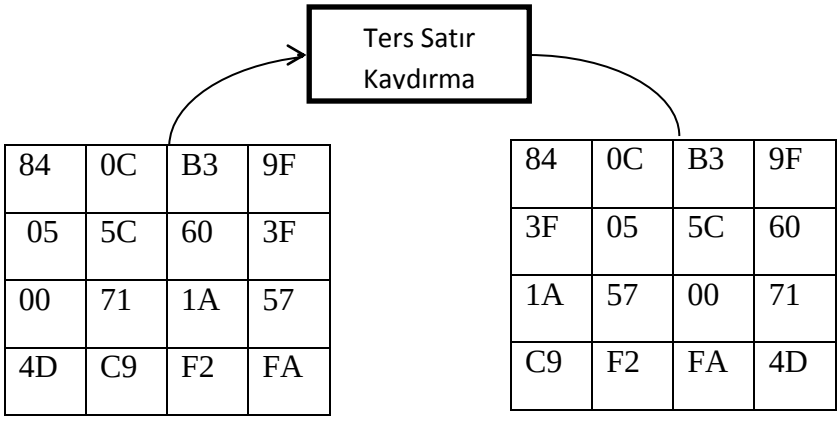
Şekil 2.12 T İşlemi Blok Yapısı [4]

2.4 Şifre Çözme

AES yapısı içi şifre çözme işlemi, şifreleme işlemindeki adımların tam terslerini içermektedir. Alt ana modül olarak ters satır kaydırma, ters bayt değiştirme, ters sütun karıştırma ve şifre çözme için kullanılan tur anahtarı ile toplama yer almaktadır.

2.4.1 Ters Satır Kaydırma

Ters satır kaydırma işlemi, şifrelemedekinin tersine, sağa doğru gerçekleştirilir. Birinci satır aynı kalırken ikinci satır 1 sağa, üçüncü 2 ve dördüncü 3 sağa ötelenir. Aşağıda ters satır kaydırmaya örnek verilmiştir.



Şekil 2.13 Örnek Ters Satır Kaydırma İşlemi

2.4.2 Ters Bayt Değiştirme

Şifreleme işlemindeki bayt değiştirme işlemini sağlayan S-Box tablosunun ters simetriği olan ters S-Box tablosu da bulunmaktadır. Bu noktada karışıklık olmaması açısından şunu belirtmek gerekir ki, S-Box tablosunun girişlerini çıkış, çıkışlarını giriş olarak değiştirip oluşturmak yanlış bir yaklaşımdır çünkü S-Box tablosunun oluşturulması Galois Field yapısına (Galois Field konusu Bölüm 4'de detaylı şekilde anlatılacaktır.) dayanır ve matematik alt modüllerden oluşur. Bu işlemlerin tersi yapıldığında oluşan tablo ters S-Box tablosudur.

		y															
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
x	0	52	09	6A	D5	30	36	A5	38	BF	40	A3	9E	81	F3	D7	FB
	1	7C	E3	39	82	9B	2F	FF	87	34	8E	43	44	C4	DE	E9	CB
	2	54	7B	94	32	A6	C2	23	3D	EE	4C	95	0B	42	FA	C3	4E
	3	08	2E	A1	66	28	D9	24	B2	76	5B	A2	49	6D	8B	D1	25
	4	72	F8	F6	64	86	68	98	16	D4	A4	5C	CC	5D	65	B6	92
	5	6C	70	48	50	FD	ED	B9	DA	5E	15	46	57	A7	8D	9D	84
	6	90	D8	AB	00	8C	BC	D3	0A	F7	E4	58	05	B8	B3	45	06
	7	D0	2C	1E	8F	CA	3F	0F	02	C1	AF	BD	03	01	13	8A	6B
	8	3A	91	11	41	4F	67	DC	EA	97	F2	CF	CE	F0	B4	E6	73
	9	96	AC	74	22	E7	AD	35	85	E2	F9	37	E8	1C	75	DF	6E
	A	47	F1	1A	71	1D	29	C5	89	6F	B7	62	0E	AA	18	BE	1B
	B	FC	56	3E	4B	C6	D2	79	20	9A	DB	C0	FE	78	CD	5A	F4
	C	1F	DD	A8	33	88	07	C7	31	B1	12	10	59	27	80	EC	5F
	D	60	51	7F	A9	19	B5	4A	0D	2D	E5	7A	9F	93	C9	9C	EF
	E	A0	E0	3B	4D	AE	2A	F5	B0	C8	EB	BB	3C	83	53	99	61
	F	17	2B	04	7E	BA	77	D6	26	E1	69	14	63	55	21	0C	7D

Şekil 2.14 Ters S-Box Tablosu [3]

2.4.3 Ters Sütun Karıştırma

Şifreleme işlemindeki gibi ters sütun karıştırmada da her sütun birebir matris çarpımına tabii tutulur. Ancak buradaki matris şifrelemedekinden farklıdır.

$$\begin{bmatrix} s'_{0,c} \\ s'_{1,c} \\ s'_{2,c} \\ s'_{3,c} \end{bmatrix} = \begin{bmatrix} 0e & 0b & 0d & 09 \\ 09 & 0e & 0b & 0d \\ 0d & 09 & 0e & 0b \\ 0b & 0d & 09 & 0e \end{bmatrix} \begin{bmatrix} s_{0,c} \\ s_{1,c} \\ s_{2,c} \\ s_{3,c} \end{bmatrix} \quad (2.3)\ [2]$$

2.4.4 Şifre Çözmede Tur Anahtarı ile Toplama

Şifrelemede yapılan tur anahtarı ile toplama işlemi, şifre çözmedekiyle aynıdır. Yine üretilen anahtar ile her turda, anahtar matrisi ile durum matrisi üzerinde toplama işlemi yani xor işlemi gerçekleşmektedir.

3. MİKROİŞLEMCİ YAPILARI

3.1 Giriş

Mikroişlemci, makine koduyla kodlanmış, merkezi işlem biriminin (uzun hal CPU) fonksiyonlarını çözümleyip gerçekleyen programlanabilir sayısal tümdevredir ref. Genel bilgisayar sistemlerinin, gömülü sistemlerin ve mobil cihazların ana işlemcisi olarak mikroişlemciler kullanılmaktadır.

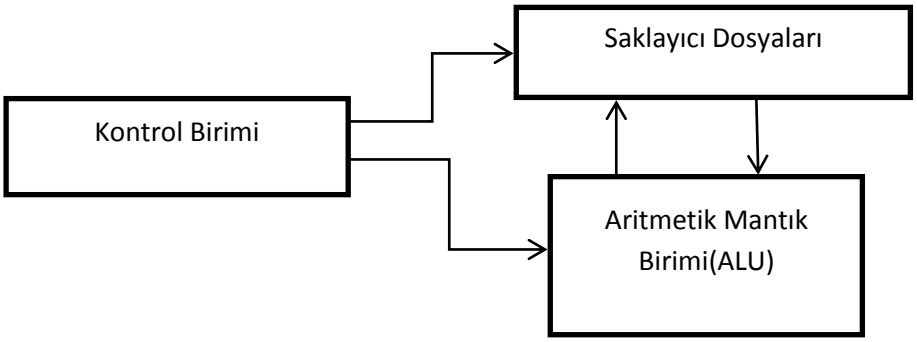
Mikroişlemcilerin bulunmasından önce, elektronik ana işlem birimleri sadece birkaç tranzistörden oluşuyordu. Bu durum, şirketleri önceden programlanabilen gömülü sistemlere yöneltti ref. 1970'de Garrett AirResearch, Birleşik Devletler ordusuna Tomcat adında bir F-14 uçağına ana uçuş kontrol bilgisayarı yapmıştır. Ancak bu tasarım, çok küçük ve güvenilir olmasına rağmen ordunun sistemi çok ileri yapıda görmesinden dolayı 1997'de yayınlandı. 1971'de Texas Instruments(TI), 4-bit TMS 1000'i; Intel ise 4-bit 4004 çipini üretti. Bu ilk mikroişlemciler, hesap makinesinin işlevlerini gerçekleyen tümdevrelerdi. Daha sonra 1972'de Intel ilk 8-bit işlemciyi, 1975'de National ilk tek çip 16-bit mikroişlemciyi, 1982'de ise AT&T Bell Labs ilk tek çip 32 bit mikroişlemci piyasaya sürülmüştür.

Mikroişlemciler, çok amaçlı işlemleri küçük boyutta tümdevrelerde sağlayabilmekte ve yüksek performans seviyelerine düşük bütçelerde erişebilmektedir. İlk mikroişlemcilerden itibaren mikroişlemci teknolojisi, hem kazandırdıklarıyla hem gelişme hızıyla elektronik dünyasında atılan önemli adımlardan olmuştur. Bunu sağlayan en dikkat çekici etken, işlem yapabilme ve kontrol etme işlevlerinin yazılımla sağlanabilmesidir. Golden Moore'un 1965 yılında ortaya attığı Moore yasasında "Her sene mikroişlemci içerisindeki tranzistör sayısının iki katına çıkacağı" ifadesinin doğruluğu mikroişlemcilerin geçirdiği ilerlemeyi oldukça güzel özetlemiştir.

Mikroişlemciler, birçok elektronik sistemin temel yapıtaşlarındandır. Bunlara gömülü sistemlerde akıllı evler, görsel bileşenler (DVD oynatıcılar vs.), oyuncaklar gösterilebilirken haberleşme, işaret işleme, robotik gibi birçok uygulama alanı daha ilave edilebilir.

3.2 Temel Mikroişlemci Elemanları

Temel mikroişlemci yapısında aritmetik mantık birimi, kontrol birimi, saklayıcı dosyaları ve veri yolu yapıları bulunmaktadır ref. Aşağıda temel mikroişlemci yapısındaki elemanlar gösterilmiştir.

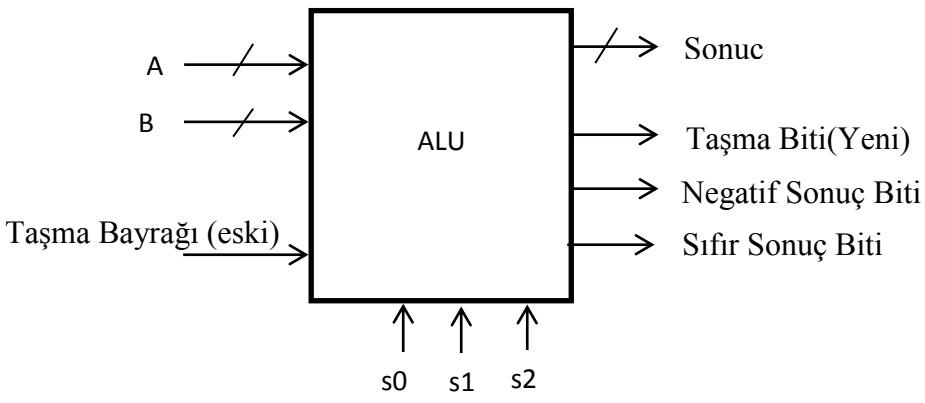


Şekil 3.1 Temel Mikroişlemcinin Başlıca Elemanları

3.2.1 Aritmetik Mantık Birimi

Aritmetik Mantık Birimi (İngilizce uzun hal - ALU), basitçe aritmetik ve lojik işlemleri gerçekleyen sayısal devredir ref. ALU mikroişlemciden gelen buyrukları gerçekleştirmek üzere donanımsal olarak gerçekleştirilen birimdir.

ALU modern işlemcilere uyması açısından ikiye tümleyen sayı gösterim biçimini kullanmalıdır. Bu gösterim, işaretli ve işaretsiz sayıların gösterimini basitleştirdiğinden işlem kolaylığı sağlar. Ayrıca Aritmetik Mantık Birimi, kombinezonsal bir devre olduğundan saat işaretine gerek duymaz. ALU işlemleri, genellikle toplayıcı ve kaydırıcı devrelerle gerçeklenir. İşlem süresi, bu devrelerdeki kapı ve yolların oluşturduğu gecikme ile hesaplanır. Aşağıda 8-bit bir ALU yapısına örnek verilmiştir.



Şekil 3.2 Örnek ALU Bloğu bu ne

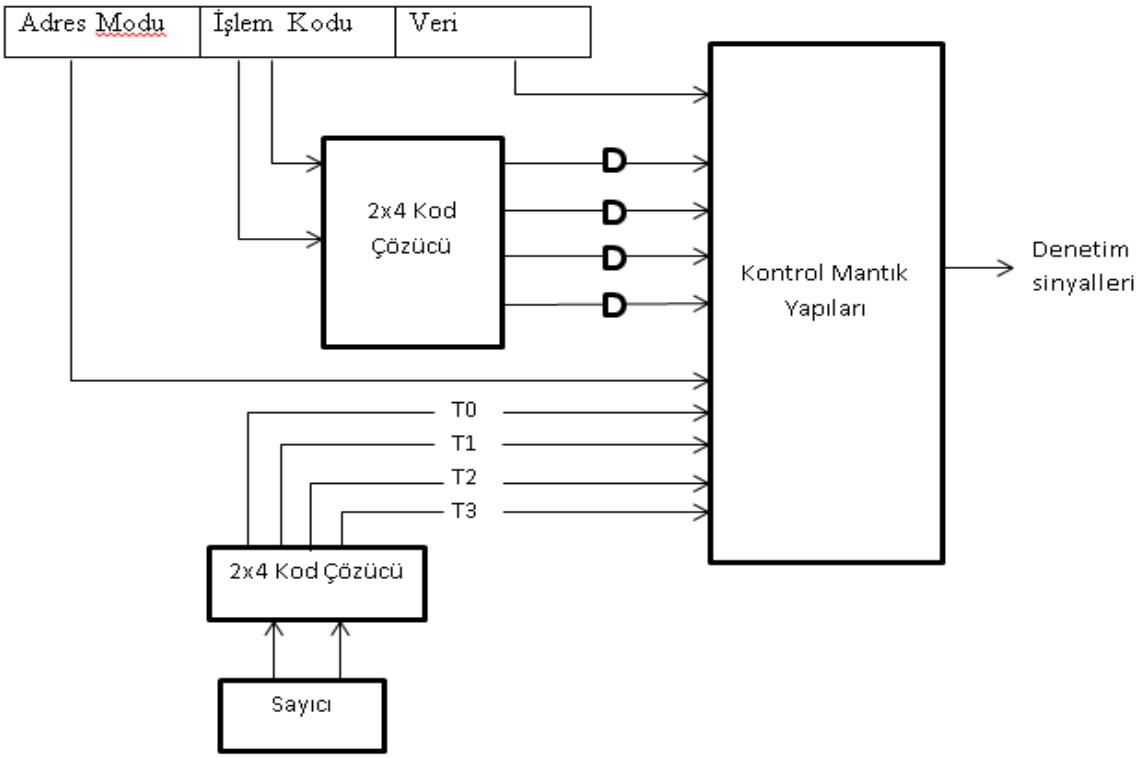
Şekil 3.2'deki ALU bloğunda, girişlerde saklayıcılar ile alınan veriler seçim girişleri ile belirlenen işlemlerden geçerek o seçim girişlerine ait çıkış bulunur. Çıkış değeri ise ALU ile tekrar saklayıcılara yazılır. Bu örnekte girişler spesifik olarak 8 bit alınmıştır. Kullanılan girişlere göre bayraklar güncellenir.

3.2.2 Kontrol Birimi

İşlemci içerisinde kontrolü sağlayan birim, saklayıcılara ne yazılacağını hangi veri yoluna yazılacağını yine bu birim tarafından üretilen denetim sinyalleri ile sağlamaktadır ref. Ayrıca işlemci içerisindeki donanımların senkronizasyonunu da sağlamaktadır.

Kontrol biriminde, öncelikle denetim sinyalleri üretilir ve bu sinyallere göre donanım birimlerinde yetkilendirme yapılarak işlemler gerçekleşir. Bu sırada yapılacak işlemlerin farklı zamanlarda gerçekleşmesi gerekebilir. Bu yüzden kontrol birimi saat darbelerini de yönlendirir.

Aşağıda örnek bir kontrol birimi yapısı gösterilmiştir. Mikroişlemcide oluşan komuttaki işlem komutu kod çözücüden geçerek işlemin belirlenmesi sağlanır. Sayıcıyı da giriş kabul eden kontrol birimi, buna göre çıkışın ne olacağını belirler.



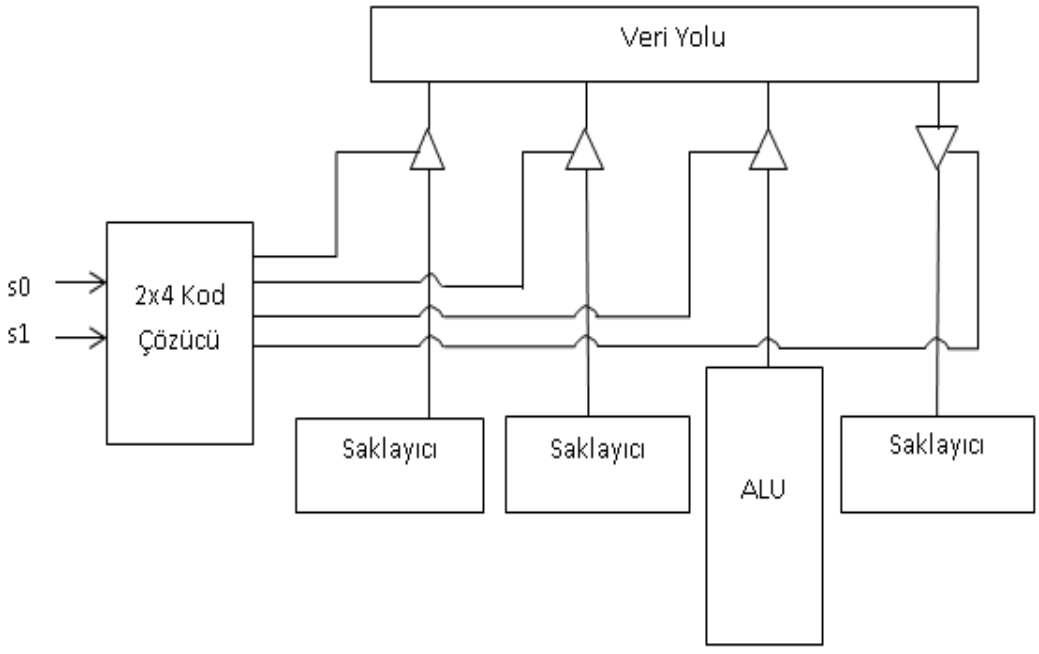
Şekil 3.3 Kontrol Birimi

3.2.3 Saklayıcı Dosyaları

Saklayıcı dosyası içinde iç yapı saklayıcıları bulundurur. Genel amaçlı iç yapı saklayıcıları, veri ve adres saklayıcıları olarak ikiye ayrılır. Veri saklayıcıları çoğunlukla olarak akümülatörler olarak adlandırılır. Akümülatörler, Aritmetik mantık Biriminin çıktılarını saklarlar. Adres saklayıcıları ise verilerin hafızadaki yerlerini tutarlar. Mikroişlemci içerisinde, ara işlemler gerçekleşirken saklayıcılara yazma veya saklayıcılardan okuma sağlanır.

3.2.4 Veri Yolu Yapıları

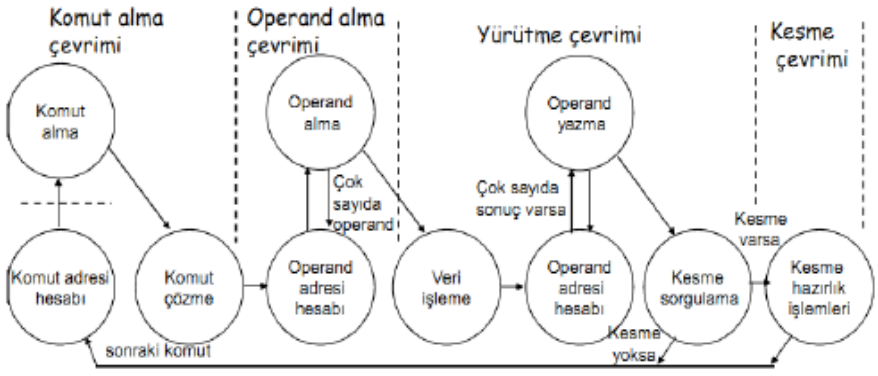
Mikroişlemcilerde, birçok donanım yapısı ve saklayıcı yapıları bulunmaktadır. Bu birimlerin birbirleriyle veri iletişimini sağlaması için geliştirilen sistem ortak veri yollarıdır. Ortak veri yolları ile, hızlı ve etkin biçimde veri alış verişi sağlanır. Saklayıcılar bu veri yolları üzerinde veri iletimini sağlamakta ve denetim sinyalleri sayesinde veri yoluna erişecek saklayıcılar belirlenmektedir. Aşağıda örnek bir veri yolu yapısı gösterilmiştir.



Şekil 3.4 Üç Durumlu Tamponlar ile Tasarlanan Veri Yolu Yapısı

3.2.5 Mikroişlemcide Komutların Gerçeklenmesi

Mikroişlemcide, komutların gerçeklenmesi belli bir sistemde adım adım yapılmaktadır ref. Bu adımlar üç tanedir ve daha önceden de belirtilen kontrol birimi tarafından denetleme yapılmaktadır. Komutlar, program belleğinden veya ortak bellekten alınarak uygun sıra ile belirlenen işlemlerden geçer. Komutların gerçeklenme sürecinde, ilk olarak program belleğinden alınan komutun çözülmesidir. Bu işlem genellikle Fetch&Decode Türkçesi ne olarak bilinir. Yapılacak her program program sayıcıda bir adreste yüklüdür. Öncelikle program sayıcıdan bu adres alınır ve adresteki komut bellekten okunarak kod çözücü yardımıyla çözülür. İkinci adımda, dolaylı adresleme kipinde verilmiş olan komutlar için etkin adres hesaplanır ve bu etkin adrese ait komut okunur. Üçüncü adımda ise, komutu gerçeklemek üzere kontrol birimi tarafından denetim sinyalleri üretilir. Aşağıdaki şekilde bir buyruğun gerçeklenmesi süresinde mikroişlemcide oluşan durumlar gösterilmiştir.



Şekil 3.5 Mikroişlemcide Komut Çalışma Döngüsü [5]

3.3 Mikroişlemci Mimarileri

Mikroişlemcilerin temel işlevi, verilerin işlenmesi, birimler arası iletişimi ve bellekte saklanmasıdır. İşlemcinin içerdiği komut kümesi, bellek yapısı ve saklayıcı özellikleri mikroişlemcinin mimarisini de belirlemektedir. İşlemci mimarileri, komut kümesi-yazılım (Instruction Set Architecture) ve donanım sistem mimarisi (Hardware System Architecture) özelliklerine göre farklılık göstermektedir.

İşlemci mimarileri, komut seti yapılarına göre ikiye ayrılmaktadır; Karmaşık Komut Kümeli Mikroişlemciler (CISC) ve Azaltılmış Komut Kümeli Mikroişlemciler (RISC). Donanım sistem mimarileri olarak Von Neumannn ve Harvard mimarisi yer almaktadır.

3.3.1 CISC

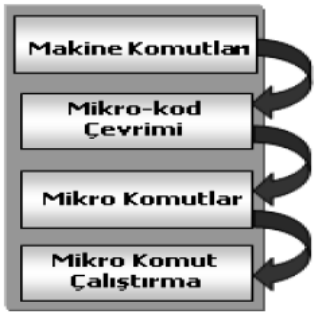
CISC işlemcilerde, her işlemi gerçekleştirmek için komutlar tasarlanmıştır. Yapılacak işlem sayısı arttıkça komut sayısı da artmakta ve bu durum da işlemcinin karmaşıklığını arttırmaktadır. Ancak CISC işlemcilerin tasarım amacı, daha az bellek kullanarak daha hızlı donanım yapısı elde etmektir[6].

CISC mimarisi, bellek kullanımı azalttığından çok büyük yapıdaki sistemlerde tercih edilmişlerdir. 1960 yıllarından sonra gelişmeye başlayan mimarinin en bilinen örnekleri Intel 8086, IBM 360 ve Motorola 68030'dür.

CISC mimarisinde komutlar kademeli olarak gerçekleştirilir. Bir komut bitmeden diğerine geçilmez yani aynı anda sadece bir komut çalışmaktadır[6]. Bu kademeler aşağıdaki sırayla gerçekleşmektedir.

1. Program sayıcısının gösterdiği adresten ilgili komutun alınması
2. Alınan komutun ve mikro-kod oluşumu
3. ALU' da komut işlemlerinin yapılması
4. İşlemden sonra oluşan çıkış verisinin ilgili saklayıcıya yüklenmesi

Aşağıda CISC mimarisinde mikrokod çevrimi gösterilmiştir.



Şekil 3.6 CISC Bir Komutun Mikrokod Çevrimi [6]

3.3.2 RISC

CISC mimarili işlemciler, komut seti açısından çok büyük ve karmaşıklığı çoktur. Her işleme tanımlı komut olmasına rağmen bütün bu komutlar çoğu zaman kullanılmaz. Bu yüzden 1970'lerden sonra basitleşme açısından komut kümelerinin azaltılması amaçlanmış ve RISC (Reduced Instruction Set Computer) mimarisi böylece ortaya çıkmıştır.

RISC işlemcilerin temel özellikleri aşağıda verilmiştir [7]:

- Az sayıda buyruk ve adresleme kipi
- Yalnızca LOAD ve STORE komutları ile bellek erişimi
- Donanımsal kontrol yapısı
- Komutların kademeli yerine tek seferde yapılması
- Sabit uzunluklu ve kolay çözülebilen komut yapısı

RISC işlemcilerde, daha az komut olması ve daha basit olması açısından CISC işlemcilere göre daha hızlı çalışır. RISC işlemcinin daha hızlı olmasındaki en büyük etken Kanal Komut İşleme Tekniği(Pipeline) kullanılmasıdır[6].

3 bits	3 bits	3 bits	7 bits
opcode	reg A	reg B	Signed immediate (-64 to 63)

Şekil 3.7 16 Bit'lik Örnek Komut Yapısı

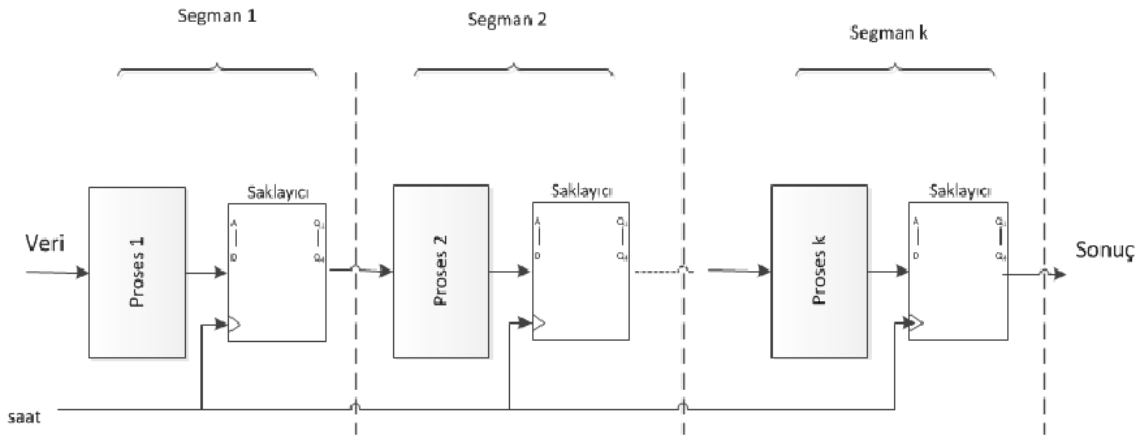
Şekil 3.7'da görüldüğü gibi 16 bitlik bir komutun, ilk 3 bitte opcode bilgisi yer almaktadır. Yani işlemci bu üç bite bakarak yapacağı işlemi belirler. Sonraki üçer bitlik kısımda ise saklayıcılar bulunmaktadır. İşlem yapılması istenen sayı ise son 7 bitlik kısımda yer almaktadır.

3.3.3 İş hattı (Pipeline)

Bilgisayar sistemlerinde iş hattı, birinin çıkışı diğerinin girişi olacak şekilde düzenlenmiş seri işlem kümesine denir. Ana işlemler iş hattında paralel olarak gerçekleştirilir. Bu yüzden tampon depolama alanı işlemler arasında paylaştırılır.

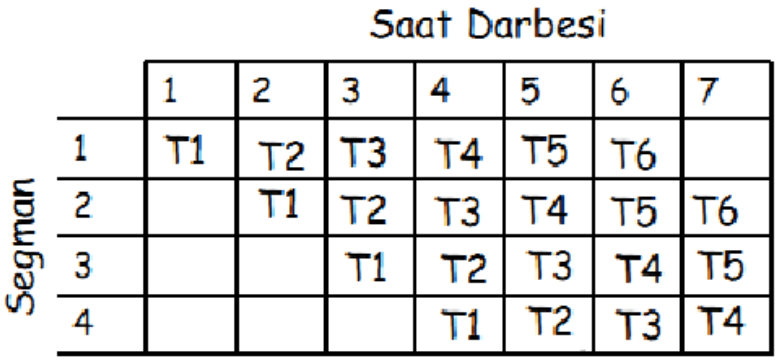
Büyük işlemler, sırayla yapıldığında çok fazla zaman gerektirmektedir. Ancak işlemler küçük alt parçalara ayrılıp her bir işlem aynı anda yürütüldüğünde bu süre çok daha kısa olmaktadır. İş hattı, çok büyük işlemlerin daha çabuk yapılmasını sağlamakta ve işlemcinin hızını arttırmaktadır.

İş hattının alt katmanlarına segman denir. Veri girişi ile işlem başlar. Her segman çıkışında saklayıcılar bulunmaktadır. Bu saklayıcılar, alt işlemlerin sonucunu tutar. Saklayıcılarda tutulan sonuçlar daha sonra bir sonraki segmanın girişini oluşturur ve süreç bütün segmanlar bitene kadar devam eder. İş hattı yapısı şekil 3.8' da gösterilmektedir.



Şekil 3.8 İş Hattı Yapısı [5]

Her segmanda zaman-işlem durumunu gösteren uzay-zaman diyagramı bulunur. Uzay-zaman diyagramları, hangi segmanda hangi işlerin yürütüleceğini gösterir [5]. Şekil 3.7' de 4 segman içeren bir iş hattının uzay-zaman yapısı gösterilmiştir.

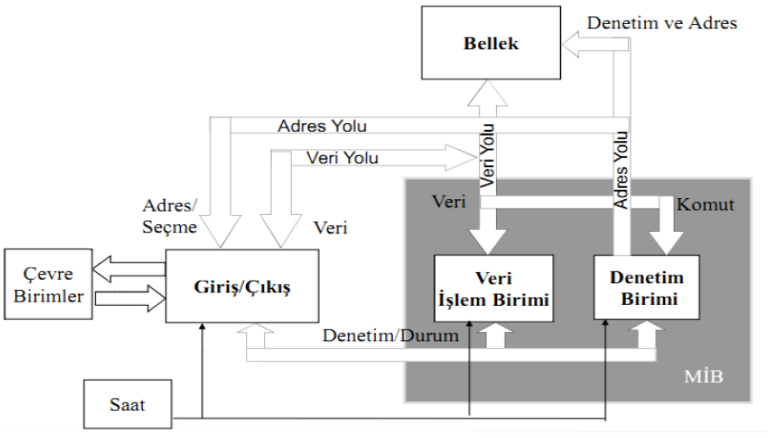


Şekil 3.9 Örnek Uzay-Zaman Grafiği [5]

Örnekte verilen uzay-zaman grafiğinde segman sayısı 4' tür. T ile simgelenen durumlar yapılacak işlemlerdir. İlk saat darbesi ile birinci segmanda T1 işlemi yürütülür. İkinci saat darbesinde birinci segmanda T2 işlemi yürütülürken ikinci segmanda T1 işlemi yapılmaya başlanır. Aynı süreç diğer segmanlar için de olmaktadır. Görüldüğü üzere iş hattı sayesinde ana işlem 4 saat darbesinde tamamlanır.

3.3.4 Von Neumann

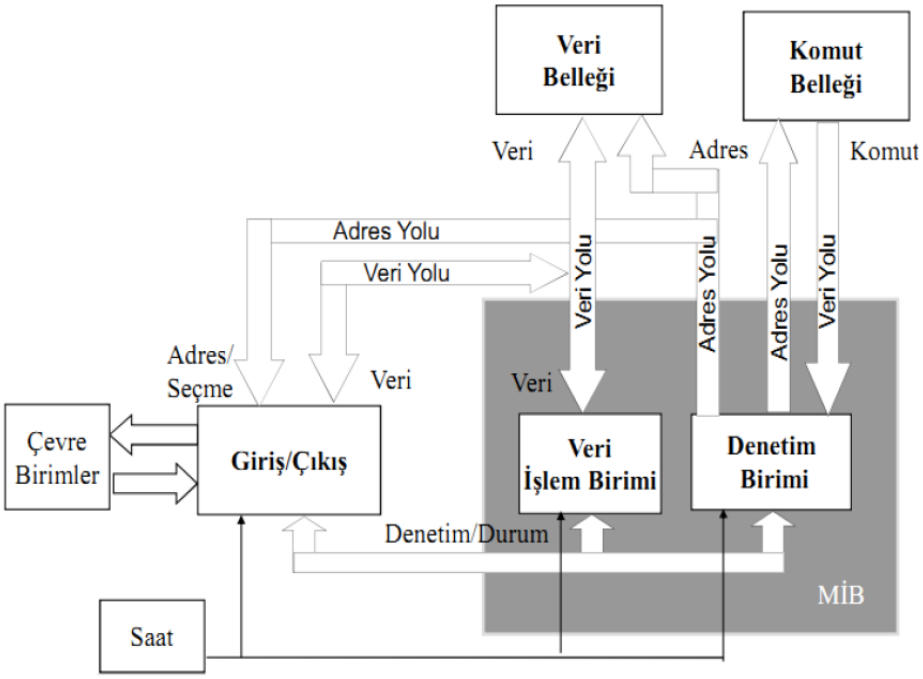
Von Neumann mimarisi, komut ve verilerin aynı bellekte yer aldığı donanım yapısıdır. Aynı veri yollarını kullanılarak komut ve verilere erişim sağlanır. Bu yüzden belleğe bağlı bir adres yolu ve bir veri yolu ile okuma/yazma işlemleri yapılır. Komut ve veriler farklı saat darbelerinde okunur. Örnek bir blok diyagram Şekil 3.10'da gösterilmiştir.



Şekil 3.10 Von Neumann Mimarisi [5]

3.3.5 Harvard Mimarisi

Komut ve veri ayrı belleklerde tutulur ve veri yolları da ayrıdır. Veri yollarının ayrı olması komut ve verinin aynı anda iletilmesini sağlamaktadır. Harvard mimarisi, yüksek performans gerektiren sistemlerde kullanılmaktadır. Sayısal işaret işlemede ve güvenliğin çok önemli olduğu mikrodenetleyicilerde tercih edilmektedirler[6]. Örnek bir Harvard mimari yapısı aşağıdaki örnekte gösterilmiştir.



Şekil 3.11 Örnek Harvard Mimarisi Blok Diyagramı

4. S-BOX'IN KOMBİNEZONSAL OLARAK GERÇEKLENMESİ

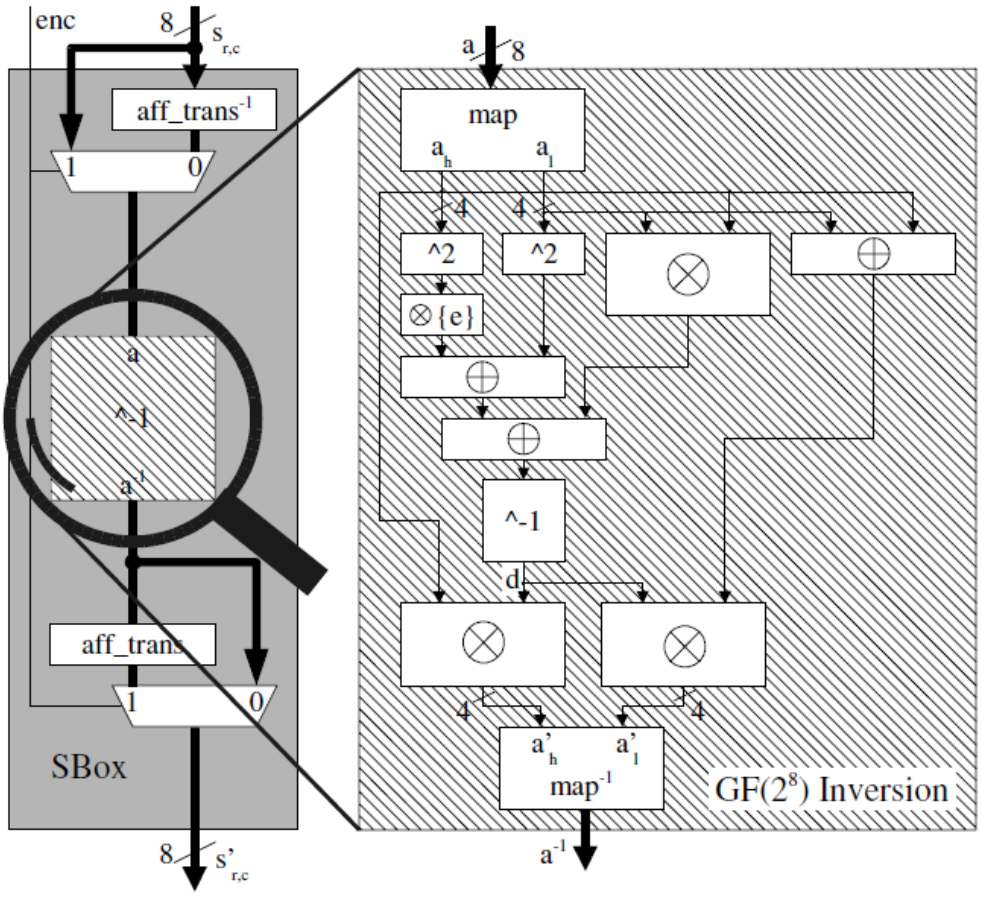
4.1. Giriş

Rijndael algoritması, Gelişmiş Şifreleme Standardı olarak seçildiğinden beri birçok şifre çözme yöntemi ile karşı karşıya kalmış ancak hiçbiri başarılı olamamıştır [9]. Bu durumdaki en büyük rol, tek nonlineer operasyon olarak S-Box'a aittir. S-Box, temel olarak 2 bölümden oluşur: Çarpmaya göre ters alma ve afin dönüşüm.

Bu bölümde, S-Box'ın kombinezonsal olarak gerçeklenmesindeki adımlar anlatılacaktır.

4.2.AES S-Box Yapısı

S-Box, temel olarak 2 bölümden oluşur: Çarpmaya göre ters alma ve afin dönüşüm. Şifreleme işleminde kullanılan S-Box gerçeklenirken ilk olarak çarpmaya göre ters alınır ve daha sonra afin dönüşüm işlemi uygulanır. Şifre çözme işleminde ise önce ters afin dönüşüm işlemi uygulanır, daha sonra yine çarpmaya göre ters alma işlemi gerçekleştirilir. Aşağıdaki şekilde S-Box'ın yapısı gösterilmektedir.



Şekil 3.1 AES S-Box Blok Diyagramı [11]

map : İzomorf Haritalama

map^{-1} : Ters İzomorf Haritalama

^2 : Kare Alma

$\otimes$: $GF(2^4)$ italik veya equation'de Çarpma

^-1 : $GF(2^4)$'de Ters Bulma

$\oplus$:$GF(2^4)$'de Toplama İşlemi (XOR)

{e} = {1110}

Aşağıda önce çarpmaya göre ters alma işlemi ve alt modülleri, daha sonra ise afin dönüşüm ve ters afin dönüşüm anlatılacaktır.

4.2.Çarpmaya Göre Ters Alma

Çarpmaya göre ters alma işleminde, sonlu alanlar aritmetiği kullanılır. Bu yüzden öncelikle, sonlu alanlar anlatılacak daha sonra çarpmaya göre ters alma modülüne geçilecektir.

4.2.1. Sonlu Alanlar

n pozitif tamsayı ve p asal sayı iken sonlu alan kuvveti p^n olan aritmetiktir. Genellikle $GF(p^n)$ olarak gösterilir. GF, Fransız matematikçi Evarist Galois'in yaratıcısı olduğu Galois Fields yapısının kısaltılmış halidir ref. Tasarlanan işlemci ve dolayısıyla giriş de 8-bit olduğundan, p değeri 2'ye, n değeri de 8'e eşittir. Bu bitlerin hepsi $GF(2^8)$'de bir polinomun katsayılarını oluşturur. En yüksek anlamlı bit, en yüksek dereceli terimin katsayısını, en düşük anlamlı bit ise en düşük dereceli terimin katsayısını yani sabit terimi oluşturacak şekilde yerleşir ve bütün bitler bu şekilde konumlandırılır. Katsayılar 0 veya 1, polinomun derecesi de 7'dir. Örneğin; {10111001} sayısı, $GF(2^8)$' de $x^7 + x^5 + x^4 + x^3 + 1$ şeklindeki polinoma eşdeğerdir. Burada işlemler yapılırken, belirli polinomlara göre indirgeme yapılır. Bu polinomlara **"indirgenemez polinomlar (irreducible polynomials)"** neden bold denir. Basitçe bu polinomlara göre mod alındığı düşünülebilir. Ek olarak, Galois Field yapısında toplama işlemi XOR işlemine denktir.

Çarpmaya göre ters alınırken, birçok alt modül bulunmakta ve bu modüller oldukça kompleks matematik işlemler içermektedir. Bu nedenle $GF(2^8)$ 'den, $GF(2^4)$ gibi düşük dereceli alan yapılarına geçiş yapılır.

Aşağıda bazı indirgenemez polinomlar verilmiştir:

$GF(2^8) \rightarrow x^2 + x + \lambda$

$GF(2^4) \rightarrow x^2 + x + \varphi$

$GF(2^2) \rightarrow x^2 + x + 1$

Burada $\varphi = \{10\}_2$ ve $\lambda = \{1100\}_2$ değerlerine eşittir[10].

4.2.2 İzomorfik ve Ters İzomorfik Haritalama (map ve map^{-1})

$GF(2^8)$' de ters alma işlemi çok karmaşık olduğu için öncelikle $GF(2^4)$'e geçiş yapılır ve işlemlere devam edilip tekrar $GF(2^4)$'den $GF(2^8)$ yapısına geçilir. $GF(2^8)$, $GF((2^4)^2)$ şeklinde yazılabildiği için herhangi bir $\alpha \in GF(2^8)$ iken,

$$a_h x + a_l = map(a), \quad a_h, a_l \in GF(2^4) \quad , \quad a \in GF(2^8) \qquad [11]$$

$$a_A = a_1 \oplus a_7 \, , a_B = a_5 \oplus a_7 \, , a_C = a_4 \oplus a_6$$

$$a_{l0} = a_C \oplus a_0 \oplus a_5, \; a_{l1} = a_1 \oplus a_2 \; , \; a_{l2} = a_A \, , \; a_{l3} = a_2 \oplus a_4$$

$$a_{h0} = a_C \oplus a_5, \; a_{h1} = a_A \oplus a_C, \; a_{h2} = a_B \oplus a_2 \oplus a_3 \; , \; a_{h3} = a_B$$

Yukarıda verilen eşitlikler izomorfik haritalama yani map fonksiyonunu gerçeklemektedir.

$$a = map^{-1}(a_h x + a_l), \quad a_h, a_l \in GF(2^4) \quad , \quad a \in GF(2^8) \qquad [11]$$

$$a_A = a_{l1} \oplus a_{h3} \, , \qquad a_B = a_{ho} \oplus a_{h1}$$

$$a_0 = a_{l0} \oplus a_{h0} \, , \; a_1 = a_B \oplus a_{h3} \; , \; a_2 = a_A \oplus a_B \; , \; a_3 = a_B \oplus a_{l1} \oplus a_{h2}$$

$$a_4 = a_A \oplus a_B \oplus a_{l3} \; , \; a_5 = a_B \oplus a_{l2}, \quad a_6 = a_A \oplus a_{l2} \oplus a_{l3} \oplus a_{h0}$$

$$a_7 = a_B \oplus a_{l2} \oplus a_{h3}$$

Yukarıdaki eşitliklerde ise ters izomorfik haritalama işlemi yapılmaktadır. Denklemlerde bulunan a 8 bit, a_h, a_l 4 bitlik sayılardır.

4.2.3 GF(2^4)'de Çarpma

$GF(2^4)$'de *a(x)* ve *b(x)* polinomlarının çarpımı yapıldığında, bu iki polinom da 4.derece olduğu için çarpımları 8.dereceden olacaktır. Yine $GF(2^4)$ yapısında kalınabilmesi için indirgenemez polinomlar (irreducible polynomals) kullanılır.

$$m_4(x) = x^4 + x + 1 \qquad [11]$$

İki polinomun çarpım ifadesi yazılıp yukarıdaki indirgenemez polinoma göre mod alınırsa; çarpma işlemi şu eşitliklerle sağlanır:

$$q(x) = a(x) \oplus b(x) = a(x).b(x) \ mod \ \ m_4(x), \qquad a(x), b(x), q(x) \in GF\left(2^4\right) \ [11]$$

$$a_A = \ a_0 \oplus \ a_3, \qquad a_B = \ a_2 \oplus \ a_3$$

$$q_0 = \ a_0 b_0 \oplus a_3 b_1 \oplus a_2 b_2 \ \oplus \ a_1 b_3$$

$$q_1 = \ a_1 b_0 \oplus a_A b_1 \oplus a_B b_2 \ \oplus (a_1 \oplus b_3) b_3$$

$$q_2 = \ a_2 b_0 \oplus a_1 b_1 \oplus a_A b_2 \ \oplus a_B b_3$$

$$q_3 = \ a_3 b_0 \oplus a_2 b_1 \oplus a_1 b_2 \ \oplus a_A b_3$$

4.2.4 Kare Alma

Kare alma, çarpma işleminin özel bir versiyonudur.

$$q(x) = a(x)^2 \ \ mod \ \ m_4(x), \ a(x), q(x) \in GF\left(2^4\right) \qquad [11]$$

$q_0 = a_0 \ \ \oplus \ \ a_2 \ , \qquad q_1 = a_2$

$q_2 = a_1 \ \ \oplus \ \ a_3 \ , \qquad q_3 = a_3$

4.2.5 GF(2^4)'de Ters Bulma

a, GF(2^4)'de bir eleman olsun. Bu elemanın çarpmaya göre tersi bulunurken temel mantık, $a \otimes a^{-1} = \{1\}$eşitliğinin sağlanmasından gelir. Ters alma işlemi yapılırken genellikle Genişletilmiş Öklid Algoritması (Extended Euclidean Algorithm - EEA) kullanılır. Ancak maalesef bu algoritma donanım için uygun değildir.

$$a(x).a^{-1} \ mod \ m_4(x) = 1$$

$$q(x) = \ a(x)^{-1} \ mod \ m_4(x), \ \ q(x), a(x) \ \in GF(2^4) \qquad [11]$$

$$a_A = \ a_1 \oplus \ a_2 \oplus \ a_3 \oplus \ a_1 a_2 a_3$$

$$q_0 = \ a_A \oplus \ a_0 \oplus \ a_0 a_2 \oplus \ a_1 a_2 \oplus a_0 a_1 a_2$$

$$q_1 = \ a_0 a_1 \oplus \ a_0 a_2 \oplus a_1 a_2 \oplus a_3 \oplus \ a_1 a_3 \oplus \ a_0 a_1 a_3$$

$$q_1 = \ a_0 a_1 \oplus a_2 \oplus a_0 a_2 \oplus a_3 \oplus \ a_0 a_3 \oplus \ a_0 a_2 a_3$$

$$q_3 = \ a_A \oplus \ a_0 a_3 \oplus a_1 a_3 \oplus \ a_2 a_3$$

4.2.6 Afin Dönüşüm

X uzayından Y uzayına afin dönüşüm, $x \rightarrow Mx + b$ şeklindedir. Burada M matris ve b vektördür.

$$M = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \end{bmatrix} \quad , \quad b = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} \qquad [12]$$

$$q = aff\ trans(a) \qquad [11]$$

$a_A = a_0 \oplus a_1$, $a_B = a_2 \oplus a_3$

$a_C = a_4 \oplus a_5$, $a_D = a_6 \oplus a_7$

$$q_0 = a_0 \oplus a_C \oplus a_D$$

$$q_1 = a_5 \oplus a_A \oplus a_D$$

$$q_2 = a_2 \oplus a_A \oplus a_D$$

$$q_3 = a_7 \oplus a_A \oplus a_B$$

$$q_4 = a_4 \oplus a_A \oplus a_B$$

$$q_5 = a_1 \oplus a_B \oplus a_C$$

$$q_6 = a_6 \oplus a_B \oplus a_C$$

$$q_7 = a_3 \oplus a_C \oplus a_D$$

Ters afin dönüşüm ise aşağıdaki eşitliklerle gerçekleştirilir.

$$q = aff\ trans^{-1}(a) \qquad [11]$$

$a_A = a_0 \oplus a_5$, $a_B = a_1 \oplus a_4$

$a_C = a_2 \oplus a_7$, $a_D = a_3 \oplus a_6$

$$q_0 = a_5 \oplus a_C$$

$$q_1 = a_0 \oplus a_D$$

$$q_2 = a_7 \oplus a_B$$

$$q_3 = a_2 \oplus a_A$$

$$q_4 = a_1 \oplus a_D$$

$$q_5 = a_4 \oplus a_C$$

$$q_6 = a_3 \oplus a_A$$

$$q_7 = a_6 \oplus a_B$$

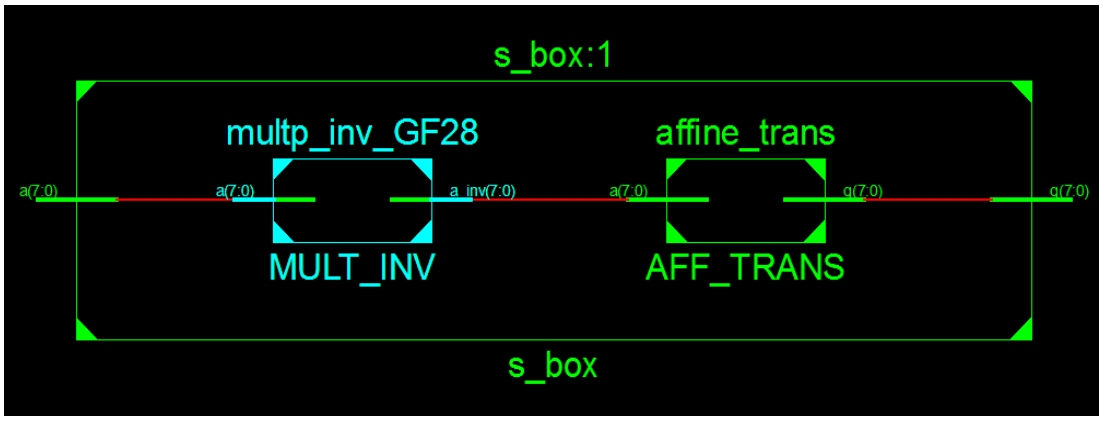
5. Mikroişlemcinin VHDL' de Gerçeklenmesi

Bu bölümde, mikroişlemci içerisindeki bloklar, Yüksek Hızlı Tümleşik Devre Donanımı Tanıma Dili (Very High speed integrated circuit Hardware Description Language - VHDL) kullanılarak tasarlanacaktır. FPGA üzerinde gerçeklenecek sayısal donanımların tasarlanmasında kullanılan donanım tanımlama dili olan VHDL, kombinezonsal veya senkron ardışıl devrelerin gerçekleştirilmesinde kullanılır. Bu devrelerin hem simülasyonu yapılabilmekte hem de doğruluklarının anlaşılması için test edilebilmektedirler. Bu bölümde önce bir önceki bölümde anlatılan S-Box' ın VHDL' de tasarımı anlatılacak daha sonra ise önceden tasarlanan mikroişlemci yapısı kısaca gösterilerek

S-Box modülün bu mikroişlemciye entegrasyonu ile tasarım tamamlanacaktır.

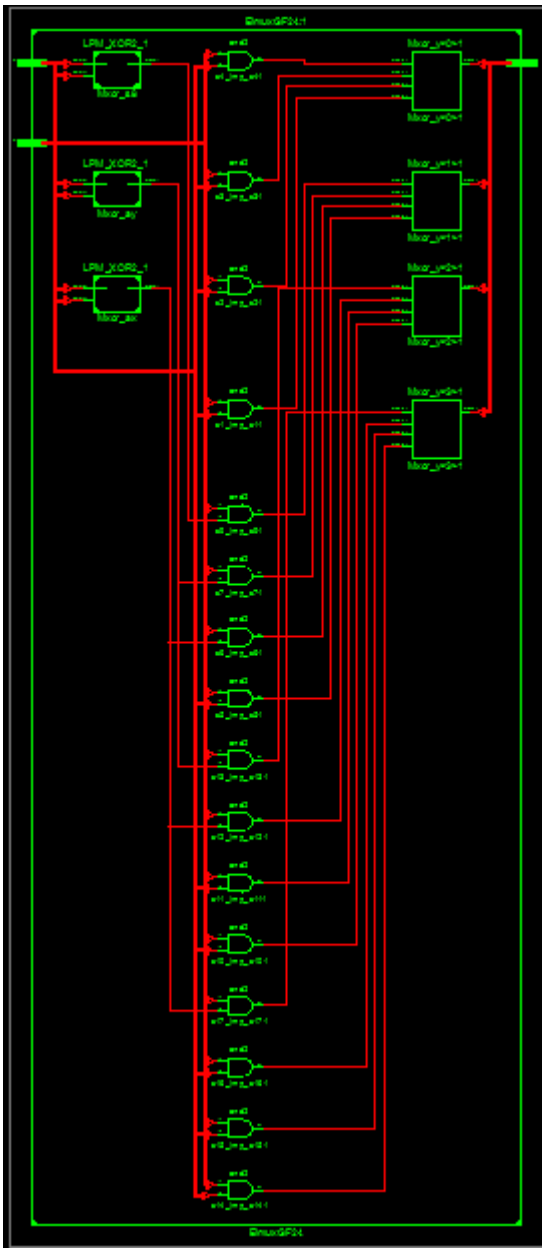
5.1 S-Box' ın Tasarlanması

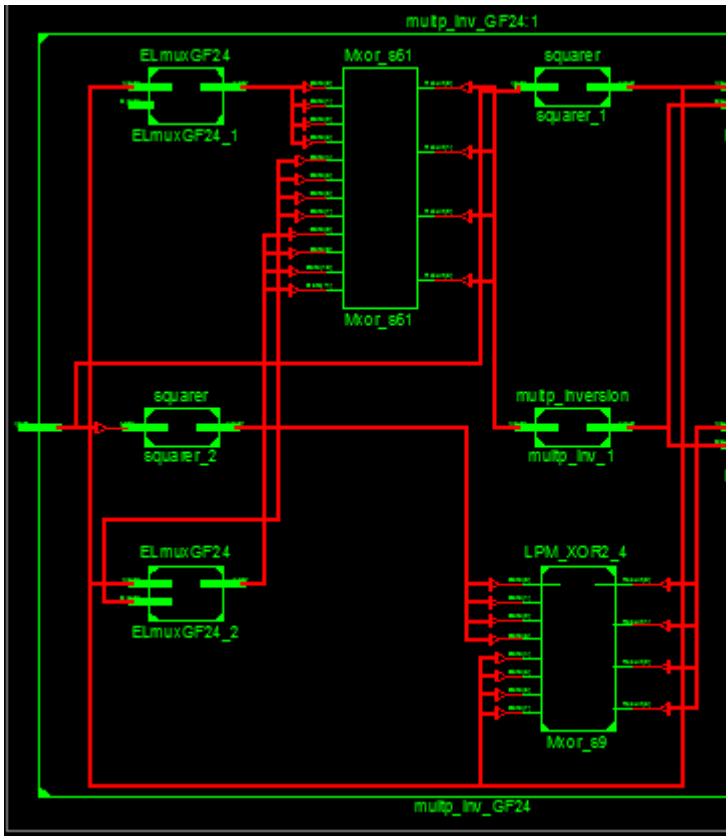
4.bölümde anlatılan S-Box yapısı VHDL dili ile tasarlanmıştır. Tasarıma önce S-Box blok diyagramında bulunan alt modüllerden başlanmıştır. Daha sonra ise bu modüller bir araya getirilerek S-Box yapısı elde edilmiştir.



Şekil 5.1 S-Box Modülü resimlerin arka planını beyaza çevirebilirsen iyi olur. Basılmış halinde bu renkler görünmez. Xilinx de bir seçenek bu.

Yukarıdaki şekilde görüldüğü üzere S-Box modülü daha önceden anlatıldığı gibi 2 bölümden oluşmaktadır. Çarpmaya göre ters alma (multp_inv_GF28 modülü) ve afin dönüşüm (AFF_trans). Çarpmaya göre ters alma işlemi büyük bir blok olduğundan birçok alt modülden oluşmaktadır. Bu alt modüllerin şematiği aşağıdaki şekilde gösterilmiştir.





Şekil 5.2 GF(2^8)' de Çarpmaya Göre Ters Alma Modülü

ELmuxGF24: GF(2^4)' de Çarpma

squarer: Kare Alma

multıp_inversion: GF(2^4)' de Çarpmaya Göre Ters Alma

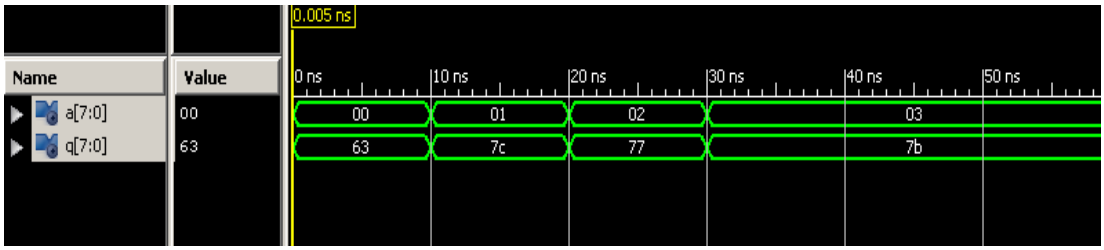
Çarpmaya göre ters alma modülünde bulunan {e} ile çarpma aslında GF(2^4)' de çarpma işlemine denktir. Sadece girişlerden biri sabit olan {e}= {1110} sayısıdır. Gerçekte modül içerisinde izomorf haritalama ve ters izomorf haritalama da vardır. Ancak bu modüller XOR işlemleri içerdiğinden program sentezleme yaparken devreyi optimize ettiği için yukarıdaki şekilde ayrı olarak gözükmemektedir. Bu modüller Mxor yapılarının içinde bulunmaktadırlar.

Yandaki şekilde ELmuxGF24 yani

$GF(2^4)$' de çarpma işlemi gösterilmiştir. Burada da görüldüğü gibi çarpma İşleminde birçok işlem bulunmakta ve bu işlemleri XOR ve NOT kapıları oluşturmaktadır. Çarpma işlemine ait eşitlikler bölüm 4'de verilmiştir. Çarpmaya göre ter alma modülündeki gibi bazı XOR işlemlerini program optimize etmektedir.

Şekil 5.3 $GF(2^4)$' de Çarpma Modülü

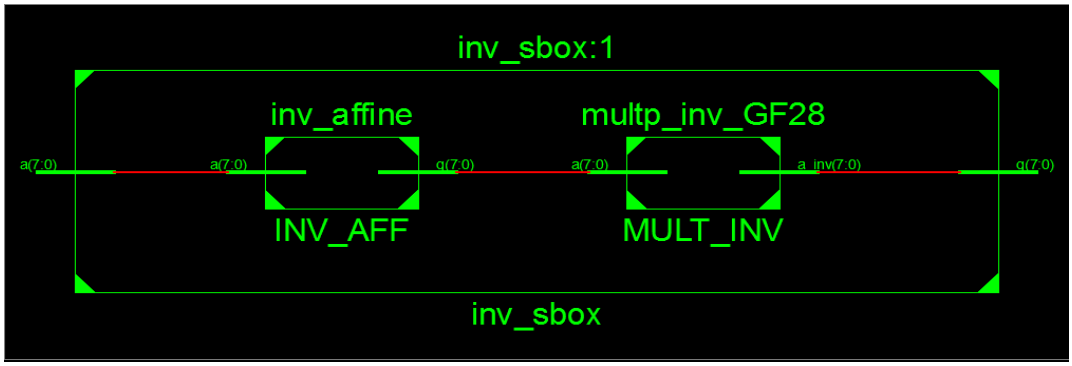
S-Box VHDL ile gerçeklendikten ve benzetimi yapıldıktan sonra kombinezonsal devrenin doğru çalışıp çalışmadığının test edilmesi gerekmektedir. Bu yüzden test amacıyla girişlere bazı değerler verilip çıkışların tablodaki çıkış değerleriyle uyuşup uyuşmadığına bakılmıştır. Aşağıdaki şekilde S-Box' a ait test sonuçları bulunmaktadır.

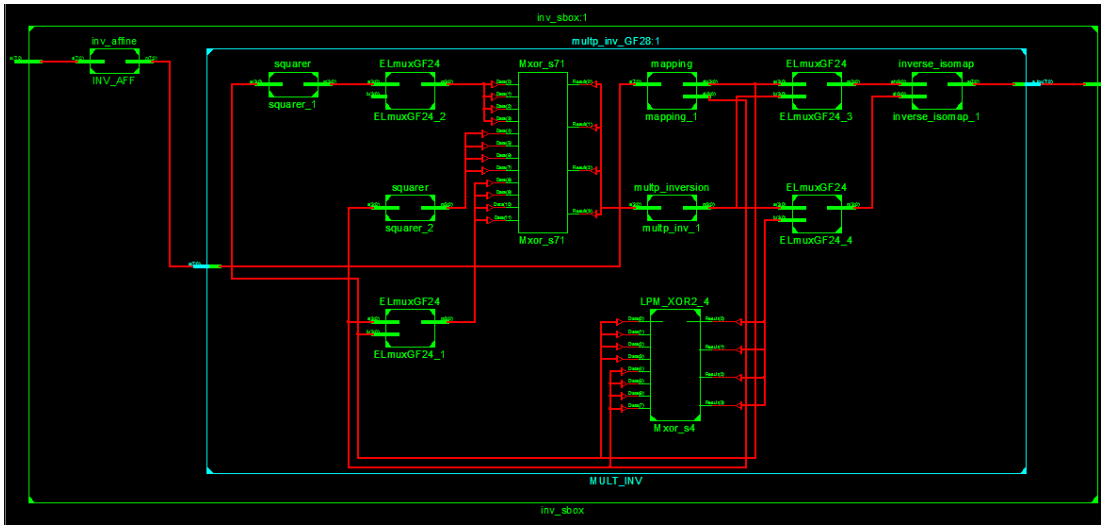


Şekil 5.4 Örnek S-Box Giriş ve Çıkışları

Yukarıdaki şekilde girişlere sırayla “00”, “01”, “02”,”03” verilmiştir ve çıkış değerleri sırayla “63”, “7c”, ”77”, ”7b” olmuştur. S-Box tablosuna tekrar bakılırsa çıkış değerlerinin doğru olduğu ve devrenin doğru çalıştığı görülecektir.

S-Box’ la aynı şekilde ters S-Box da VHDL ile tasarlanmıştır. S-Box’ tan tek farkı önce ters afin dönüşüm bloğu sonra çarpmaya göre ters alma bloğu gelmesidir. Bu yüzden ek olarak ters afin dönüşüm bloğu tasarlanmış ve daha sonra iki modül arka arkaya birbirine bağlanmıştır. Aşağıdaki şekilde Ters S-Box modülünün ana yapısı ve genişletilmiş şematiği gösterilmiştir.

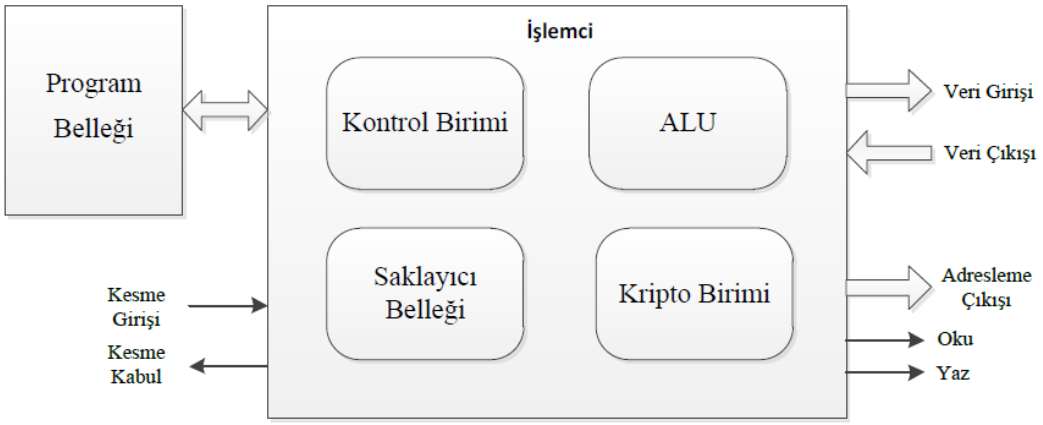




Şekil 5.4 Ters S-Box Modülü

5.2 S-Box Modülünün Mikroişlemciye Entegresi

Bu bölümde, daha önceden VHDL' de tasarlanmış bir mikroişlemci yapısına yeni tasarlanan kombinezonsal S-Box yapıları entegre edilecektir. Kullanılan mikroişemci 8-bit RISC işlemcisidir. Bu işlemcide aritmetik birim, mantık birimi, kaydırma birimi ve kripto birimi bulunmaktadır. Kripto biriminde S-Box modülleri tablo şeklinde Başvuru Çizelgesinden (Look Up Table) alınmıştır. Önce bölümde tasarlanan S-Box blok yapıları işlemcideki kripto biriminde bulunan S-Box yapılarıyla değiştirilmiştir. Aşağıdaki şekilde kullanılan mikroişlemcinin genel yapısı bulunmaktadır.



Şekil 5.5 Mikroişlemcinin Genel Görünümü

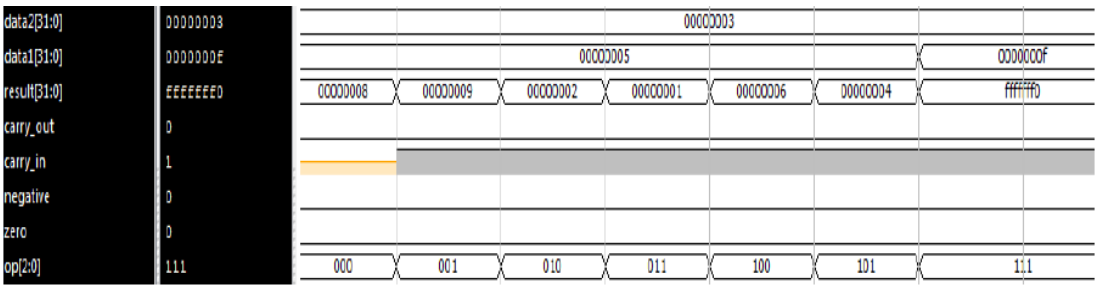
Yeni tasarlanan modüller mikroişlemciye entegre edildikten sonra işlemcide bulunan her birim için test yapılarak işlemcinin çalıştığı doğrulanmıştır.

Aşağıdaki tabloda aritmetik birimi için tanımlanmış komutlar bulunmaktadır.

Makine Kodu	Sembolik İfade	İşlem
D0ddmmnn	ADD rN, rM, rd	rd = rN + rM
D1ddmmnn	ADDC rN, rM, rd	rd = rN + rM + c
D2ddmmnn	SUB rN, rM, rd	rd = rN - rM
D3ddmmnn	SUBC rN, rM, rd	rd = rN - rM - c
D40000nn	INC rN	rN = rN + 1
D50000nn	DEC rN	rN = rN - 1
D600mmnn	CMP rN, rM	rN ↔ rM
D70000nn	COM rN	rN = rN'

Tablo 5.1 İşlemcide Bulunan Aritmetik Birim Komutları

Yukarıda bulunan komutlar test edilerek giriş çıkış sinyallerine bakılmıştır. Aşağıdaki şekilde sinyallerin durumu bulunmaktadır.



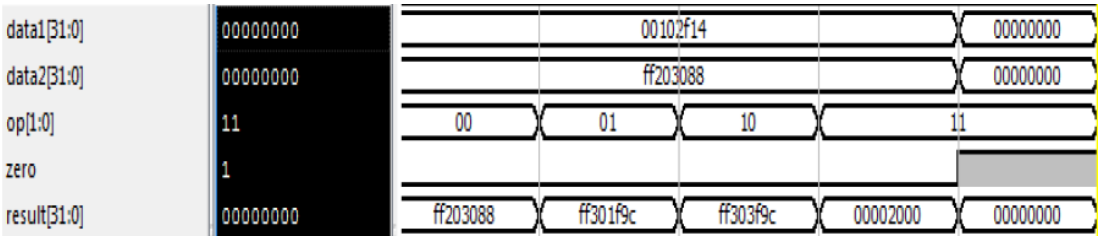
Şekil 5.6 Aritmetik Birim Simulasyon Çıktısı bu resim sana ait mi? Öyle olmalı

Mantık birimi tasarımında ise aşağıdaki komutlar tanımlıdır.

Makine Kodu	Sembolik İfade	İşlem
C0dd00nn	Mov rN, rd	rN = rd
C1ddmmnn	Xor rN, rM, rd	rd = rN xor rM
C2ddmmnn	Or rN, rM, rd	rd = rN or rM
C3ddmmnn	And rN, rM, rd	rd = rN and rM

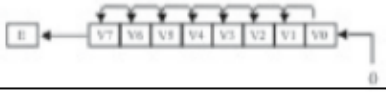
Tablo 5.2 Mantık Birimi Komutları

Mantık birimi komutları test sonucunda aşağıdaki şekilde çıkış vermiştir.

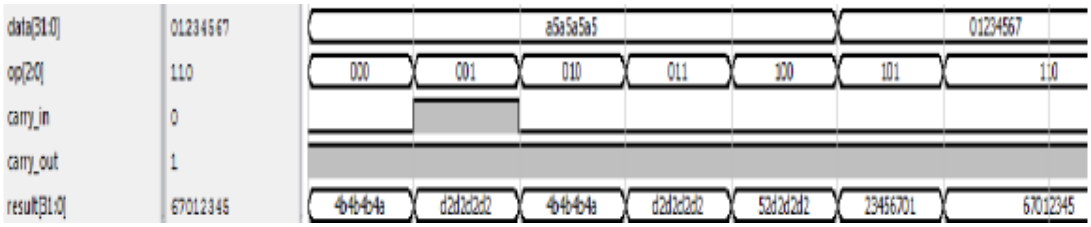


Şekil 5.7 Mantık Birimi Simülasyon Sonuçları bu resim sana ait mi? Öyle olmalı

Mikroişlemcide bulunan kaydırma birimi komutları aşağıda verilen tablodaki gibidir. Yapılan testin sonuçları ise şekil 5.8’ de bulunmaktadır.

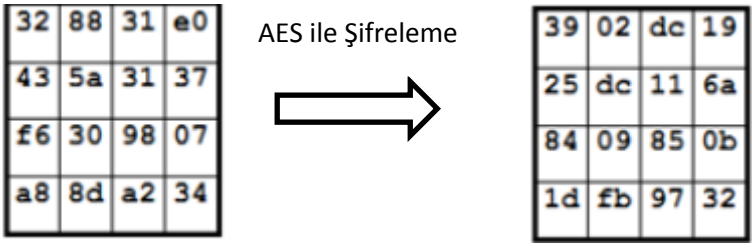
Makine Kodu	Sembolik İfade	İşlem
F00000nn	rotL rN	
E10000nn	rotR rN	
E20000nn	Asl rN	
E30000nn	Asr rN	
E40000nn	Lsr rN	
E50000nn	RotLB	Sola Bir Bayt Öteleme
E60000nn	RotRB	Sağa Bir Bayt Öteleme

Tablo 5.3 Kaydırma Birimi Komutları

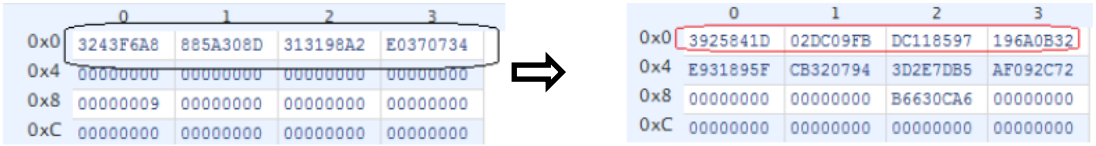


Şekil 5.8 Kaydırma Birimi Simülasyonu bu resim sana ait mi? Öyle olmalı

AES şifreleme işlemi için algoritmada bulunan işlemler de komut dizisine yazılmıştır. İşlemcideki AES komut ile gerçekleştirilen testin sonuçları aşağıdadır.



Şekil 5.9 Örnek Şifreleme



Şekil 5.10 Şifreleme Öncesi ve Sonrası Bellek Görünümü

6. SONUÇLAR ve TARTIŞMA

Bitirme çalışması için öncelikle AES algoritması hakkında araştırma yapılmıştır. Daha sonra S-Box ve Ters S-Box' ın kombinezonsal devre şeklinde nasıl gerçekleneceği konusu araştırılmış ve bu yapı elde edilirken kullanılan Galois Field yapıları üzerine literatür çalışması yapılmıştır. Daha sonra VHDL' de S-Box ve Ters S-Box devreleri tasarlanmış ve simülasyonları yapılarak sonuçların, S-Box ve ters S-Box tablolarındaki değerlere denk geldiği görülmüştür.

İkinci adım olarak, 8 bitlik bir RISC işlemci üzerinde işlem yapılmıştır. Bu işlemci Uygulamaya Özel Komut Setli İşlemci yapısındadır ve kripto birimi içermektedir. Yalnız S-Box ve Ters S-Box yapıları Başvuru Çizelgesinden (Look Up Table, LUT) alınmıştır. Bu çalışmada ise kombinezonsal olarak tasarlanan S-Box ve Ters S-Box modülleri bu işlemcidekilerle değiştirilmiştir.

Son adımda, mikroişlemci test edilerek çalıştığı doğrulanmıştır. Bitirme çalışmasında amaç,işlemci üzerinde bulunan kripto biriminde S-Box ve Ters S-Box işlemlerini Başvuru Çizelgesinden almadan kombinezonsal olarak gerçeklemektir. Böylece hafızada çok yer kaplayan bu tabloların yerine bellekte daha az yer kaplayan kombinezonsal devre ile kullanılan hafıza oranı aza inecektir. Kombinezonsal olarak gerçeklenen S-Box devrelerinin hafızada getirdiği avantajın yanında, hız açısından da performansta oluşan düşme nedeniyle bir dezavantaj getireceği öngörülmüştür. Mikroişlemcinin ilk hali ve son hali çalıştırılıp performans değerlendirmesi yapıldığında ise; son halde, mikroişlemcinin hızının daha düşük olduğu ancak bellekte kapladığı alanın daha az olduğu görülmüştür. Sonuçlar, yapılan öngörünün doğruluğunu ispatlamıştır. Hız performanslarının karşılaştırılması simülasyon programı üzerinden direk gözlenmiş, hafıza performansları ise kullanılan saklayıcı, LUT vb. sayıları üzerinden değerlendirilmiştir.

AES kodunun tamamının çalıştığını göstermen lazım.

.