

**T.C.
BOZOK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
MEKATRONİK MÜHENDİSLİĞİ ANABİLİM DALI**

Yüksek Lisans Tezi

**YAPILANDIRILABİLİR DONANIM ÜZERİNDEKİ
(FPGA) MİKROBİLGİSAYAR SİSTEMLERİ İÇİN
ANALOG HABERLEŞME ARAYÜZ TASARIMI**

Resul KÖKSAL

**Tez Danışmanı
Prof. Dr. Feyzullah TEMURTAŞ**

Yozgat 2014

**T.C.
BOZOK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
MEKATRONİK MÜHENDİSLİĞİ ANABİLİM DALI**

Yüksek Lisans Tezi

**YAPILANDIRILABİLİR DONANIM ÜZERİNDEKİ
(FPGA) MİKROBİLGİSAYAR SİSTEMLERİ İÇİN
ANALOG HABERLEŞME ARAYÜZ TASARIMI**

Resul KÖKSAL

**Tez Danışmanı
Prof. Dr. Feyzullah TEMURTAŞ**

Yozgat 2014

T.C.
BOZOK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

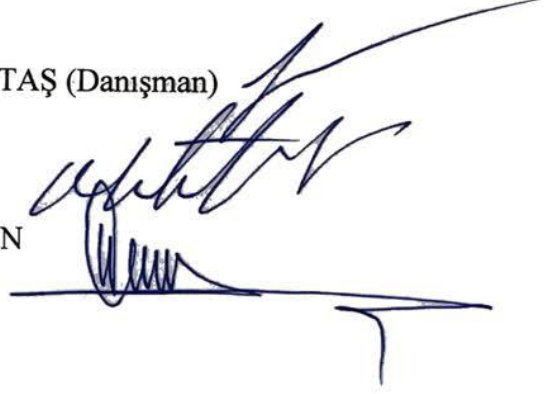
TEZ ONAYI

Enstitümüzün Mekatronik Mühendisliği Anabilim Dalı 7011010007 numaralı öğrencisi Resul KÖKSAL'ın hazırladığı “Yapılandırılabilir Donanım Üzerindeki (FPGA) Mikrobilgisayar Sistemleri İçin Analog Haberleşme Arayüz Tasarımı” başlıklı YÜKSEK LİSANS tezi ile ilgili TEZ SAVUNMA SINAVI, Lisansüstü Eğitim-Öğretim ve Sınav Yönetmeliği uyarınca 24/10/2014 Pazartesi günü saat 14:00’te yapılmış, tezin onayına OY BİRLİĞİYLE karar verilmiştir.

Başkan : Prof. Dr. Feyzullah TEMURTAŞ (Danışman)

Üye : Yrd. Doç. Dr. Mustafa YAZ

Üye : Yrd. Doç. Dr. Halit ÖZTEKİN



ONAY:

Bu tezin kabulü, Enstitü Yönetim Kurulu'nun 18./12./2014 tarih ve 39 sayılı kararı ile onaylanmıştır.



Doç. Dr. Halit ÖZTEKİN
Bozok Üniversitesi
Fen Bil.Enst.Müdürü

İÇİNDEKİLER

	<u>Sayfa</u>
ÖZET	vi
ABSTRACT	vii
TEŞEKKÜR	viii
TABLolar LİSTESİ	ix
ŞEKİLLER LİSTESİ	x
KISALTMALAR LİSTESİ	xiv
1. GİRİŞ	1
2. GENEL BİLGİLER VE TASARIM PLATFORMLARI	5
2.1. Yapılandırılabilir Donanımlar	5
2.1.1. Basit Programlanabilir Lojik Birim (SPLD).....	6
2.1.2. Karmaşık Programlanabilir Lojik Birim (SPLD).....	7
2.1.3. Alanda Programlanabilir Kapı Dizileri (FPGA).....	7
2.1.3.1. FPGA Temel Yapısı	8
2.2 VHDL	10
2.2.1. VHDL Kod Yapısı Temel Birimleri	11
2.2.1.1. Kütüphane Seçme (Library Declaration)	11
2.2.1.2. IEEE Kütüphanesinde Kullanılan Paketler	12
2.2.1.3. Data Tipleri	12
2.2.1.4. Aygıt Tanımlama (Entity)	14
2.2.1.5. Davranış Tanımlama (Architecture)	14
2.2.1.6. VHDL İle Oluşturulan Örnek Yapılar	15
2.3. Altera DE2 Geliştirme Kiti Ve Wolfson WM8731 Audio Kodek Chip...	20
2.3.1. Altera DE2 Geliştirme Ve Eğitim Seti	20
2.3.2. Wolfson Wm8731 Audio Kodek Chip	22
2.3.2.1. Wolfson Wm8731 Registerları	23
2.3.2.2. Wolfson Wm8731 Dijital Audio Arayüz	28
2.3.2.3. Wolfson Wm8731 Kontrol Arayüzü	30

2.4. QUARTUS II CAD Kullanımı	31
2.4.1 FPGA CAD Akış Diyagramı	31
2.4.2 Quartus II Programına Giriş	33
2.4.3. Quartus II Çevrimiçi Yardım/Destek	35
2.4.4. Yeni Bir Proje Oluşturma	35
2.4.5. Şematik Görüntü Kullanılarak Tasarıma Giriş	38
2.4.5.1. Blok Düzenleyici Kullanımı	39
2.4.5.2. Lojik Elemanların Kütüphaneden Çağrılması	41
2.4.5.3. Giriş - Çıkış Sembolleri Ekleme	42
2.4.5.4. Giriş - Çıkış Sembollerini İsimlendirme	42
2.4.5.5. Hat Ve Düğüm Bağlantısı	43
2.4.6. Derleyici Kullanımı	43
2.4.7. Hatalar	44
2.4.8. Tasarlanan Devrenini Simülasyonu	45
2.4.8.1. Waveform Editörü Kullanımı	45
2.4.8.2. Simülasyon Ayarları	49
2.5. BZK.SAU.FPGA Mikrobilgisayar Sistemi	51
3. I2C SERİ HABERLEŞME PROTOKOLÜ.....	55
3.1. Seri ve Paralel Haberleşme	55
3.2. I2C Seri Haberleşme Protokolü	56
3.2.1. I2C Haberleşme Protokolü Terimleri ve Açıklamaları	57
3.2.2. I2C Genel Karakteristiği	57
3.2.3. I2C Protokolünde Geçerli Data	58
3.2.4. I2C Protokolünde START ve STOP Durumları	58
3.2.5. I2C Protokolünde Data Transferi	59
3.2.6. I2C Protokolünde ACK (El Sıkışma)	60
3.2.7. I2C Protokolünde Adresleme Formatı	60
3.2.8. I2C Protokolünde 7-Bit Adresleme Formatı	62

4. YAPILANDIRILABİLİR DONANIM ÜZERİNDE SİSTEMLER VE MİKROBİLGİSAYAR SİSTEMLERİ İÇİN ANALOG HABERLEŞME ARAYÜZ TASARIMI.....	64
4.1. Frekans Bölme Ünitesi	68
4.2. Gerçekleştirilen I2C Kontrolör	71
4.2.1. I2C Protokolünün Yapılandırılması	73
4.2.2. Standart I2C Protokolünün Oluşturulması	79
4.3. Seri Bilginin Paralel Bilgiye Dönüştürülmesi	86
4.4. 8-Bit Paralel Bilginin Hafızaya Kayıt Edilmesi ve Kayıt Edilen Bilginin Hafızadan Okunması	90
4.5. Paralel Bilginin Seri Bilgiye Dönüştürülmesi	92
4.6. Tasarlanan Analog Haberleşme Arayüzünün BZK.SAU.FPGA Mikrobilgisayar Sistemine Uyarlanması	96
4.6.1. MikroBilgisayar Sistemleri İçin Kesme Sinyalinin Üretilmesi	97
4.6.2. Tasarlanan Analog Haberleşme Arayüzünün BZK.SAU.FPGA Mikrobilgisayar Sistemine Donanımsal Olarak Bağlanması	98
4.6.3. Tasarlanan Analog Haberleşme Arayüzünün BZK.SAU.FPGA Mikrobilgisayar Sistemine Yazılımsal Olarak Bağlanması	98
5. SONUÇLAR VE ÖNERİLER	101
KAYNAKLAR	106
ÖZGEÇMİŞ	109

**YAPILANDIRILABİLİR DONANIM ÜZERİNDEKİ (FPGA)
MİKROBİLGİSAYAR SİSTEMLERİ İÇİN ANALOG HABERLEŞME
ARAYÜZ TASARIMI**

Resul KÖKSAL

**Bozok Üniversitesi
Fen Bilimleri Enstitüsü
Mekatronik Mühendisliği Anabilim Dalı
Yüksek Lisans Tezi**

2014; Sayfa: 109

Tez Danışmanı: Prof. Dr. Feyzullah TEMURTAŞ

ÖZET

Bu çalışmada, yapılandırılabilir donanım tabanlı sistemler ve bu sistemler üzerinde gerçekleştirilen eğitimsel amaçlı mikrobilgisayar sistemleri için analog arayüz tasarımı yapılmıştır. Tasarlanan arayüz ile eğitimsel amaçlı mikrobilgisayar sistemlerinin analog sinyalleri işlemelerine katkı sağlanması hedeflenmiştir. Analog haberleşme arayüz modülü, modüler yapıda olup temel olarak altı modülden meydana gelmiştir. Birinci modül; sistemin çalışması için gerekli frekanslar üretilmiştir. İkinci modül; yapılandırılabilir donanım ile çevre birimlerinin haberleşebilmesi için standart bir i2c protokolü tasarlanarak WM8731 audio kodek entegrasyonu için gerekli register ayarları yapılmıştır. Üçüncü modül; analog girişten seri olarak alınan datalar 8-bit paralel sinyale dönüştürülmüştür. Dördüncü modül; 8-bit paralel veriler hafızaya kayıt edilmiştir. Beşinci modül; hafızadan alınan 8-bit paralel veriler seri sinyale dönüştürülerek analog çıkışa aktarılmıştır. Altıncı modül; tasarlanan analog haberleşme arayüzü BZK.SAU.FPGA mikrobilgisayar sistemine uyarlanmıştır.

Anahtar Kelimeler: BZK.SAU.FPGA, Eğitimsel Amaçlı Mikrobilgisayar, Altera DE2, VHDL, Şematik Tasarım, WM8731, I2C Seri Haberleşme Protokolü, Analog Sinyal İşleme.

THE ANALOG COMMUNICATION INTERFACE DESIGN FOR MICROCOMPUTER ON RECONFIGURABLE HARDWARE(FPGA)

Resul KÖKSAL

**Bozok University
Graduate School of Natural and Applied Sciences
Department of Mechatronic Engineering
Master of Science Thesis
2014; Page: 109**

Thesis Supervisor: Prof. Dr. Feyzullah TEMURTAŞ

ABSTRACT

In this study analog communication interface was designed for educational purpose microcomputer on reconfigurable hardware and reconfigurable hardware systems. Aim of designed analog communication interface is developed analog signal process on educational purpose microcomputer systems. Analog communication interface module has got six module arch. First module; generate frequency for microcomputer systems. Second module; standard i2c protocol designed for communication with peripheral device and made register set up for WM8731 audio codec chip control. Third module serial data which coming from dijital analog converters converted to 8-bit parallel data. Fourth module; 8-bit parallel data recorded to fifo memory. Fifth module; 8-bit parallel data which coming from memory converted to serial data and send to analog output. Sixth module; analog communication interface was adaptation for BZK.SAU.FPA microcomputer system.

Keywords: BZK.SAU.FPGA, Educational Purpose Mikrocomputer, Altera DE2, VHDL, Schematic Design, WM8731, I2C Serial Communication Protocol, Analog Signal Process.

TEŐEKKÜR

Öncelikle projeyi bitirmemi nasip eden Cenab-ı Hakka, beni gayrete getiren ve Őevklendiren deęerli hocam Prof. Dr. Feyzullah TEMURTAŐ'a, bana zaman ayıran ve fikirleri ile yol gsteren deęerli hocam Yrd. Doę. Dr. Halit ÖZTEKİN'e , tez, makale ve ęalıŐmalarından faydalandıęım tüm bilim insanlarına sonsuz teŐekkürlerimi sunarım.

Ayrıca haklarını hiçbir zaman ödeyemeyeceęim aileme ve bana kolaylık gsteren mesai arkadaşlarıma teŐekkür ederim.

TABLULAR LİSTESİ

	<u>Sayfa</u>
Tablo 2.1: Wm8731 Entegresi Register Haritası	23
Tablo 2.2: R0 ve R1 Register İçeriği	23
Tablo 2.3: R2 ve R3 Register İçeriği	24
Tablo 2.4: R4 Register İçeriği	25
Tablo 2.5: R5 Register İçeriği	25
Tablo 2.6: R6 Register İçeriği	26
Tablo 2.7: R7 Register İçeriği	27
Tablo 2.8: R8 Register İçeriği	27
Tablo 2.9: Lojik Fonksiyonun Doğruluk Tablosu	39
Tablo 2.10: BZK.SAU.FPGA Mikrobilgisayar Sistemi Özellikleri	52
Tablo 3.1: I2C Haberleşme Protokolü Terimleri Ve Açıklamaları	57
Tablo 3.2: İlk Bayttaki Bitlerin Açıklamaları	62
Tablo 4.1: Analog Haberleşme Modülü Pin Açıklamaları	66
Tablo 4.2: Kullanılan Register Adresleri ve Kullanım Amaçları	74
Tablo 4.3: Register Adresleri, Register İçerikleri ve Bu İçeriğin Anlamı	77
Tablo 4.4: FIFO Hafıza Pinleri ve Açıklaması	91
Tablo 4.5: Program Kodu	99

ŞEKİLLER LİSTESİ

	<u>Sayfa</u>
Şekil 2.1:	SPLD Genel Yapısı 6
Şekil 2.2:	CPLD Genel Yapısı 7
Şekil 2.3:	FPGA ve Mantık Hücresi Temel Yapısı 9
Şekil 2.4:	VHDL Temel Kod Yapısı 11
Şekil 2.5:	Kütüphane Tanımlama ve Kullanılan Kütüphaneler 12
Şekil 2.6:	VHDL İle İki Girişli VE Kapısı Tasarımı 15
Şekil 2.7:	VHDL İle DFF Flip Flop Tasarımı 16
Şekil 2.8:	VHDL İle Multiplexer Tasarımı 17
Şekil 2.9:	VHDL İle İstenen Fonksiyonu Yerine Getiren Aygıt Tasarımı 18
Şekil 2.10:	VHDL İle Dört Bitlik Register Tasarımı 19
Şekil 2.11:	Altera DE2 Geliştirme İle Kiti Video İşleme 20
Şekil 2.12:	Altera DE2 Geliştirme Kiti Klavye Ve Monitor İle Ork 21
Şekil 2.13:	Altera DE2 Geliştirme Kartı 21
Şekil 2.14:	Wm8731 Blog Diyagramı 22
Şekil 2.15:	Sola Dayalı Mod Data Akışı..... 29
Şekil 2.16:	I2S Mod Data Akışı 29
Şekil 2.17:	Sağa Dayalı Mod Data Akışı 30
Şekil 2.18:	DSP Mod Data Akışı 30
Şekil 2.19:	Tipik FPGA Programlama CAD Akış Diyagramı 32
Şekil 2.20:	Quartus II Ana Penceresi 34
Şekil 2.21:	File Menüsünün Görünümü 35
Şekil 2.22:	Tanıtım (Introduction) Penceresi 36
Şekil 2.23:	Proje Klasörü ve Proje İsmi Belirleme 36
Şekil 2.24:	Quartus II İstenilen Klasörün Oluşturulması 37
Şekil 2.25:	Tasarım Dosyalarının Eklendiği Pencere 37
Şekil 2.26:	Aygıt Ailesinin/Türünün Belirlenmesi 38
Şekil 2.27:	Lojik Fonksiyon 39
Şekil 2.28:	Tasarım Dosyasının Türünün Seçilmesi 40
Şekil 2.29:	Blok Editör Düzenleyici 40
Şekil 2.30:	Lojik Sembollerin Seçilmesi 41

Şekil 2.31:	Eklenen Kapı Sembolleri	42
Şekil 2.32:	Kapılar ve Pinlerin Düzenlenmesi	42
Şekil 2.33:	Tamamlanmış Devre	43
Şekil 2.34:	Derleme Raporu Özeti	44
Şekil 2.35:	Waveform Editörün Seçilmesi	45
Şekil 2.36:	Insert Node Or Bus Komutunun Çalıştırılması	46
Şekil 2.37:	Insert Node Or Bus İletişim Penceresi	46
Şekil 2.38:	Node Finder Penceresi	47
Şekil 2.39:	Simülasyon İçin Gerekli Nod(düğüm)lar	47
Şekil 2.40:	Giriş Sinyallerine Değer Atanması	48
Şekil 2.41:	Clock Penceresi	48
Şekil 2.42:	Giriş Sinyallerine Değer Atanmış Şekli	49
Şekil 2.43:	Simülasyon Modunun Belirlenmesi	50
Şekil 2.44:	Simülasyon Netlistinin Oluşturulması	50
Şekil 2.45:	Simülasyon Sonucu	51
Şekil 3.1:	Seri Ve Paralel Port Data Akışı	55
Şekil 3.2:	Örnek Bir I2C Haberleşme Sistemi	56
Şekil 3.3:	I2C Aygıtların Hatta Bağlanması	58
Şekil 3.4:	I2C Protokolünde Geçerli Data	58
Şekil 3.5:	I2C Protokolünde START ve STOP Durumları	59
Şekil 3.6:	I2C Protokolünde Data Transferi	59
Şekil 3.7:	I2C Protokolünde ACK Sinyali	60
Şekil 3.8:	I2C Protokolünde Tam Bir Data Transferi	61
Şekil 3.9:	I2C Protokolünde Data Transferinde Oluşabilecek Durumlar	61
Şekil 3.10:	START Durumundan Sonraki İlk Byte	63
Şekil 4.1:	Gerçekleştirilen Sistemin Blok Şeması	65
Şekil 4.2:	Tasarlanan Analog Haberleşme Arayüz Modülü	66
Şekil 4.3:	Analog Haberleşme Arayüz Modülü İç Yapısı	67
Şekil 4.4:	Tasarlanan Frekans Bölme Ünitesi Blok Diyagramı	68
Şekil 4.5:	Kullanılan PLL Fonksiyon	68
Şekil 4.6:	Tasarlanan Frekans Bölücü	69
Şekil 4.7:	Frekans Bölücü Çıkış Sinyali	69

Şekil 4.8:	Frekans Bölücü VHDL Kodları	70
Şekil 4.9:	12Mhz Sinyalden 100Khz, 10hz Ve 5hz Sinyal Üretilmesi	71
Şekil 4.10:	I2C Protokolü Blok Diyagramı	72
Şekil 4.11:	I2C Kontrolör Blok Şeması	72
Şekil 4.12:	Slave Adresi Üretmek İçin Kullanılan Şematik Devre	73
Şekil 4.13:	I2C Protokolüne Gönderilen Slave Adres Bilgisi	74
Şekil 4.14:	Register Adresi Üretmek İçin Tasarlanan 8*8 Multiplexer	75
Şekil 4.15:	8*8 Multiplexer Sürme Devresi	76
Şekil 4.16:	I2C Protokolüne Gönderilen Register Adresi Bilgisi	76
Şekil 4.17:	Register Data Üretmek İçin Tasarlanan 8*8 Multiplexer	78
Şekil 4.18:	I2C Protokolüne Gönderilen Register İçerik Bilgisi	79
Şekil 4.19:	Tasarlanan I2C Protokolü Çalışma Akış Diyagramı	80
Şekil 4.20:	I2C Protokolü SDA ve SCLK Sinyal Şekilleri	81
Şekil 4.21:	SCLK Sinyal Üretmek İçin Tasarlanan Devrenin Blok Şeması	82
Şekil 4.22:	SCLK ve SDA Sinyali Üretmek İçin Tasarlanan Devre	83
Şekil 4.23:	SDA Sinyali İçeriği	84
Şekil 4.24:	SDA Sinyal Üretmek İçin Tasarlanan Devrenin Blok Şeması	85
Şekil 4.25:	Start Şartının Sağlanması	85
Şekil 4.26:	SDA Slave Adres Ve Ack Sinyalleri	86
Şekil 4.27:	Seri Bilginin Paralele Dönüştürülmesi Blok Şeması	87
Şekil 4.28:	Seri Paralel Dönüştürücü Sinyal Şekilleri	88
Şekil 4.29:	Seri Paralel Dönüştürücü Devresi Ekran Görüntüsü	89
Şekil 4.30:	MegaWizard Plug-in Manager Sayfa 2	90
Şekil 4.31:	MegaWizard Plug-in Manager Sayfa 3	91
Şekil 4.32:	8-Bit 32 Kbyte FIFO Hafıza	91
Şekil 4.33:	Paralel Bilginin Seri Bilgiye Dönüştürülmesi Blok Şeması	92
Şekil 4.34:	Paralel Seri Dönüştürücü İçin Tasarlanan Devre	93
Şekil 4.35:	Paralel Seri Dönüştürücü Sinyal Şekilleri	94
Şekil 4.36:	74HC165 Entegresi İç Yapısı	95
Şekil 4.37:	Sistem Akış Şeması	96
Şekil 4.38:	Kesme Sinyali Üretme Devresi	97
Şekil 4.39:	Üretilen Kesme sinyali	97

Şekil 4.40:	Analog Arayüzden Gelen Sinyallerin BZK.SAU.FPGA Mikrobilgisayar Sistemine Uyarlanması	98
Şekil 5.1:	Analog Haberleşme Arayüzünden Gelen Dijital Verilerin BZK.SAU.FPGA Mikrobilgisayar Sisteminde Monitöre Yazdırılması	102
Şekil 5.2:	Analog Giriş Sinyali	103
Şekil 5.3:	Analog Çıkış Sinyali	103

KISALTMALAR LİSTESİ

ACK	: El Sıkışma, Onay (Acknowledge)
ADC	: Analog Dijital Dönüştürücü
ADCDAT	: Analog Dijital Dönüştürücüden Alınan Seri Bilgi Sinyali
ADCLRC	: Analog Dijital Dönüştürücü Sol Ve Sağ Kanal Saat Darbesi
ASIC	: Uygulamaya Özel Tümüleşik Devre (Application-Specific Integrated Circuit)
Bzk.Sau.FPGA	: Bozok ve Sakarya Üniversitesi Ortak Mikrobilgisayar Sistemi
CPLD	: Karmaşık Programlanabilir Lojik Birim
DAC	: Dijital Analog Dönüştürücü
DACDAT	: Dijital Analog Dönüştürücüye Gönderilen Seri Data
DACLRC	: Dijital Analog Dönüştürücü Sol Ve Sağ Kanal Clock Palsi
DoD	: Amerikan Savunma Sanayi (Department of Defense)
FPGA	: Sahada Programlanabilir Kapı Dizini (Field Programmable Gate Arrays)
HDL	: Donanım Tanımlama Dili (Hardware Description Language)
I/O	: Giriş / Çıkış
IEEE	: Elektrik Elektronik Mühendisleri Enstitüsü (Institute of Electrical and Electronics Engineers)
I2C	: Seri Haberleşme Protokolü (Inter-Integrated Circuit)
LSB	: En Değerli Bit Son Gelen Bit
MSB	: En Değerli Bit İlk Gelen Bit
NACK	: Red, Uyumsuzluk (Not Acknowledge)
SPLD	: Kolay Programlanabilir Lojik Birim
SDA	: Seri Data
SCL	: Seri Clock

1. GİRİŞ

Yapılan bilimsel arařtırmalarla, yeni bir bilginin sunumunda ve sunulan bilginin algılanma sürecinde etkili olan noktalar belirlenmiřtir. Buna gre bir konu hakkında en ideal đrenme, teorik bilgilerin pratik uygulamaya geirilmesi ile gerekleřmektedir. IEEE bilgisayar topluluđu ve bilgisayar bilimleri mfredatı dzenleme birliđi tarafından tavsiye edilen mfredat ierisinde yer alan ‘‘Tm đrenciler pratik alıřma ile teorik bilgiyi bir btn halinde đrenmek zorundadırlar’’ ifadesi derslerin iřlenmesinde pratik alıřmaların gerekliliđine vurgu yapmaktadır. Son yıllarda sıklıkla karřımıza ıkan yapılandırılabilir donanımlar zerinde geliřtirilen eđitsel uygulamalarda ki ama đrencilere teorik bilginin yanında pratik yapma imknının sađlamasıdır[1-3].

Yapılandırılabilir donanımlar (FPGA- Field Programmable Gate Arrays) , bir eřit entegredir. FPGA’ leri standart entegrelerden farklı kılan ise imalat sonrası istenen fonksiyona gre donanım yapısının yani i konfigrasyonunun kullanıcı tarafından dizayn edilebilmesidir. Standart entegreler sadece yapıldıđı ama dođrultusunda kullanılan oyunculara FPGA ise her Őey olmaya msait oyun hamuruna benzetilebilir. Oyun hamuru ile bir masa, bir araba, bir ev veya bir kuř yapılabildiđi gibi FPGA ile de tasarımcının bilgi ve becerisiyle orantılı olarak deđiřik projeler geliřtirilebilir. Esnek ve yeniden yapılandırılabilir yapısından dolayı FPGA; akademik projelerin yanı sıra kk bteli projelerde sınırlı bir zaman ierisinde tasarım yaparak tasarımın sonularını gerek dnya Őartları altında grmek isteyen zel amalı tmdevre geliřtiren tasarımcılar iin son derece uygun ve ekonomik bir zmdr. FPGA’ ler dřk maliyetleri ve tasarım esnekliđi sebebiyle giderek artan bir kullanım alanına sahiptir. Eđitsel ve akademik projelerin yanı sıra savunma sanayi, sayısal iřaret iřleme, uzay sanayi, tıbbi grntleme ve otomotiv sektr FPGA’ nın uygulama alanlarından bazılarıdır [3-6].

Mikrobilgisayar sistemler adından da anlařılacađı zere modern bilgisayarların daha sade bir formudur. Temel olarak mikrobilgisayarlar; mikroiřlemci, harici hafıza ve giriř ıkıř birimlerinden meydana gelir. Yazılımsal olarak gerekleřtirilen mikrobilgisayar simlatrlerinin yanı sıra yeniden yapılandırılabilir donanımlar

(FPGA geliştirme ortamları) üzerinde eğitimsel amaçlı tasarlanmış gerçek dünya şartları altında çalışabilen mikrobilgisayar sistemleri de mevcuttur. FPGA geliştirme ortamında eğitimsel olarak tasarlanmış mikrobilgisayar sistemlerine yönelik yapılan çalışmalar geliştirme ortamının esnekliği sebebiyle son yıllarda oldukça artmıştır. FPGA geliştirme ortamı; kullanıcılara özgün tasarımlar geliştirebilme ve gerçek dünya şartları altında çalışma imkânı sunar. Bu tarz çalışmalarda temel amaç eğitimseldir ve öğrencilerin ‘Gömülü Sistemler’, ‘Bilgisayar Mimarisi’ ve ‘İşletim Sistemi’ gibi dersleri daha iyi ve daha kolay kavraması amaçlanır. Literatürde yapılan bu tür çalışmaların genel avantajları; esneklik ve uygun maliyetin yanı sıra öğrencilerin dersle ilgili soyut kavramları somutlaştırabilmeleri ve motivasyonu artırıcı etkiye sahip olmaları söylenebilir[7-13].

Analog sinyaller sürekli değişken bir akış halindedir. Görülen, işitilen, hissedilen tüm fiziksel büyüklükler birer analog sinyaldir. İnsanlar yaşadıkları ortamdaki fiziksel değişkenleri duyu organları ile hissederek bu değişkenleri zihinlerinde yorumlayabilirler. Bilgisayarlar ve diğer elektronik sistemler ise analog sinyalleri doğrudan işleyemezler, öncelikle gelen analog verileri sayısal değerlere dönüştüren arabirimlerin ADC (Analog Dijital Dönüştürücülerin) kullanılması gerekir. Analog sinyaller dijital sinyale dönüştürüldükten sonra bilgisayarlar ve diğer elektronik sistemler tarafından hafızada saklanabilir, uzak mesafelere aktarılabilir veya çeşitli ekranlarda gösterilebilir. Bazı durumlarda dijital değerleri yeniden analog değerlere dönüştürmek gerekir. Dijital değerleri analog değerlere dönüştüren devrelere DAC (Dijital Analog Dönüştürücü) denir[14].

Mikrobilgisayarlar ve çevre birimleri arasında haberleşme hem seri hem de paralel olarak yapılabilir. Paralel haberleşme de her bir bit için ayrı bir hat gerekir. Genel olarak paralel haberleşme hızlı olmasına karşın maliyeti yüksektir. Seri haberleşme; verilerin tek bir hat ya da devre üzerinden aynı anda tek bir bit şeklinde ardı sıra aktarılmasıdır. Seri haberleşme protokollerinin gelişmesi ve hızlanması seri haberleşmeyi daha popüler bir hale getirmiştir. Örneğin National Semiconductor firmasının 80’lerde ürettiği 24 pinli M58167B entegresi, 24 pinin 17 tanesini data alma ve data gönderme işlemi için kullanıyordu oysa seri haberleşme tekniklerinin gelişmesi ile 17 pin yerine sadece 2 pin kullanılarak aynı işlemler gerçekleştirilebilir.

Son yıllarda I2C seri haberleşme protokolünün kullanımı giderek artmıştır. Bu protokol, Seri Data (SDA) ve Seri Clock (SCL) olmak üzere iki hattan meydana gelir ve tüm iletişim bu iki hat üzerinden yapılır. Aynı hat üzerine pek çok cihaz bağlanabilir. I2C seri haberleşme protokolü ile yavaş (slow), hızlı (fast) ve yüksek hızlı (high speed) olmak üzere çeşitli hızlarda veri akışını sağlayabilir. Yavaş hızda standart veri iletişim hızı 100 kbit/s'ye kadar, hızlı modda veri iletişim hızı 400 kbit/s'ye kadar, yüksek hızlı iletişimde ise veri iletişim hızı 3,4 Mbit/s'ye kadar çıkabilmektedir[15-19].

Bu çalışmada yapılandırılabilir donanım üzerinde gerçekleştirilen sistemler ve eğitim amaçlı mikrobilgisayar sistemleri için I2C seri iletişim protokolü kullanılarak analog haberleşme arayüz tasarımı modüler olarak gerçekleştirilmiştir. Eğitsel amaçlı olarak gerçekleştirilen bu uygulama ile öğrencilerin dijital sistemlerde analog sinyallerin nasıl kontrol edildiğini daha kolay anlamaları hedeflenmiştir. Yapılan çalışmada temel amaç eğitimsel olduğu için, tasarımda kullanılan elemanların içyapısının incelenebilir, kullanıcı tarafından müdahale edilebilir ve geliştirilebilir olmasına özen gösterilmiştir. Eğitsel amaçlı tasarlanan analog haberleşme arayüzü derslerde müstakil kullanılabilceği gibi oluşturulan kesme (interrupt) devresi ile de açık kodlu tüm mikrobilgisayar sistemlerine uyarlanarak da kullanılabilir durumdadır. Mikrobilgisayar sistemlerine uyarlanabilirliği ile mikrobilgisayar sistemlerinin analog sistem kontrolündeki eksikliklerinin giderilmesi hedeflenmiştir.

Tasarımı yapılan arayüz son aşamada eğitsel olarak geliştirilen BZK.SAU.FPGA mikrobilgisayar sistemine entegre edilerek analog haberleşme arayüzünden alınan dijital veriler BZK.SAU.FPGA mikrobilgisayar sisteminin hafızasına kayıt edilmiştir. Daha sonra kayıt edilen dijital veriler BZK.SAU.FPGA mikrobilgisayar sisteminin hafızasından çağrılarak analog haberleşme arayüzüne gönderilmiş ve ses sinyaline dönüştürülmüştür. BZK.SAU.FPGA mikrobilgisayar sistemi açık kaynak kodlu olması ve modüler yapısı sayesinde kullanıcıların çalışan bir bilgisayar mimarisine müdahale etmesine olanak sağladığı için tercih edilmiştir. Analog haberleşme arayüz eklentisi ile BZK.SAU.FPGA mikrobilgisayar sisteminin analog sinyalleri işlemedeki eksikliklerinin giderilmesi amaçlanmıştır.

BZK.SAU.FPGA mikrobilgisayar sistemi 16 bit genişliğinde 64 Kbyte ana belleğe sahiptir. Belleğin önemli bir kısmı mikrobilgisayar sisteminin kendisi tarafından kullanılmaktadır. Bu uygulamada analog sinyal saniyede 8000 bin defa örneklenmiş ve dijitale dönüştürülen datalar ana belleğe kayıt edilmiştir.

Bu çalışmanın ikinci kısmında yapılandırılabilir donanımlar ve tasarım platformları hakkında bilgi verilmektedir. Üçüncü kısımda I2C haberleşme protokolü hakkında bilgi verildikten sonra dördüncü bölümde yapılandırılabilir donanım üzerinde sistemler ve mikrobilgisayar sistemleri için analog haberleşme arayüz tasarımı hakkında bilgi verilmektedir. Son bölümde ise sonuç ve önerilere yer verilmiştir.

2. GENEL BİLGİLER VE TASARIM PLATFORMLARI

Bu bölümde yapılandırılabilir donanımlar, VHDL (Very High Speed Integrated Circuit Hardware Description Language) donanım tanımlama dili ve tez çalışmasında üzerinde çalışılan Altera DE2 eğitim ve geliştirme kiti ile bu kit üzerinde bulunan Wolfson WM8731 entegresi hakkında bilgi verilecektir.

2.1. Yapılandırılabilir Donanımlar

Yapılandırılabilir donanımlar; imalat sonrası istenen fonksiyona göre donanım yapısının yani iç konfigürasyonunun kullanıcı tarafından dizayn edilebildiği elektronik elemanlar olarak tanımlanabilir.

Sayısal verileri işlemek için ağırlıklı olarak uygulamaya özgü entegre devreler (ASIC) kullanılır. Bu tip devreler özel bir fonksiyonu gerçekleştirmek amacıyla üretildiğinden, bu fonksiyonları etkin ve hızlı bir şekilde gerçekleştirirler. Ancak işlevleri sınırlıdır ve sadece ilgili oldukları uygulamaya yönelik üretilmişlerdir. ASIC devreler doğru ve hızlı sonuç vermesine rağmen çözüm ürettiği problemin çeşitli türevleri için kullanılamamaktadır. Yeni problemler için yeni donanımlara ve yeni ASIC yapılarına ihtiyaç duyulur. Bu da maliyet artışına ve zaman kaybına neden olmaktadır[20].

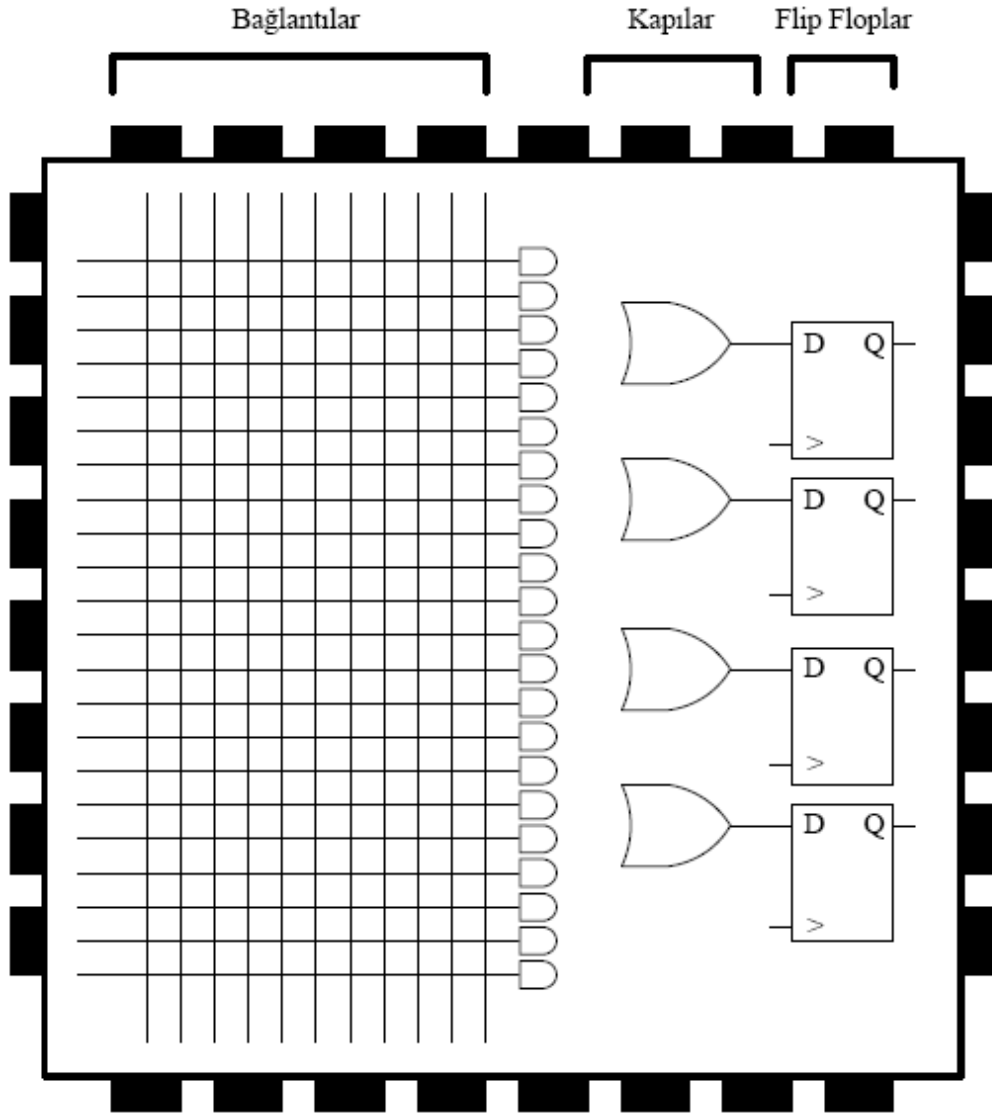
Yapılandırılabilir donanımlar ASIC devrelere göre daha yavaş olmasına karşın çok daha esnek bir yapıya sahiptir. Paralel işlem yapabilmeleri sayesinde aynı anda pek çok işlemi yapabilirler. Sistemde meydana gelecek yeni durum ve problemlere karşı donanım değişikliğine gidilmeden yazılımda yapılacak ufak müdahaleler ile sorun çözülebilir. Sürekli değişen ve gelişen piyasa şartları yapılandırılabilir donanımları daha popüler bir hale getirmiş ve kullanımları her geçen gün artmıştır.

Yapılandırılabilir donanımlar; lojik kapıların ve flip-flopların birbirlerine bağlanmasıyla meydana gelen devreler şeklinde tanımlanabilir. Temel olarak üç ana gruba ayrılır. Bunlar;

- SPLD (Basit Programlanabilir Lojik Birim)
- CPLD (Karmaşık Programlanabilir Lojik Birim)
- FPGA (Alanda Programlanabilir Kapı Dizileri)

2.1.1. Basit Programlanabilir Lojik Birim (SPLD)

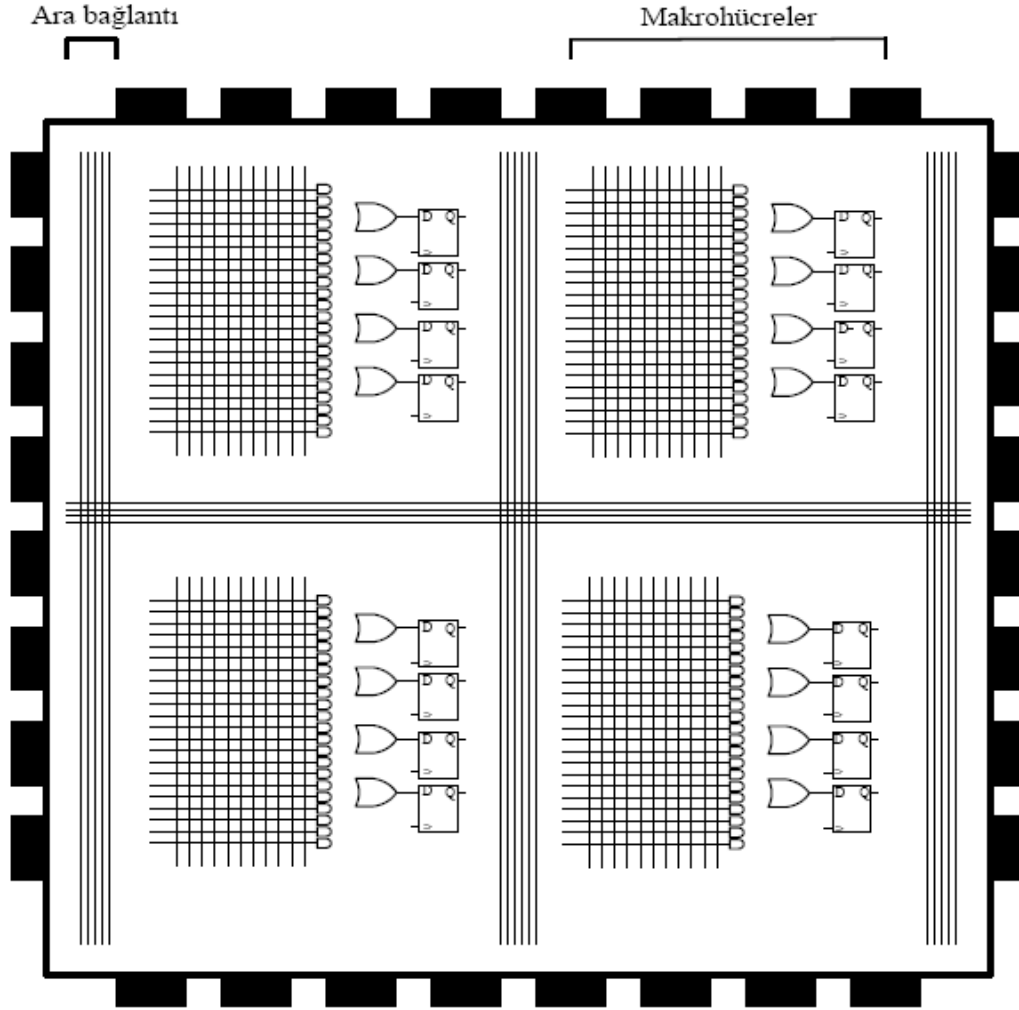
SPLD' ler düşük kapasiteli ve ucuz maliyetli yapılandırılabilir lojik ünitelerdendir. SPLD' lerdeki her hücrenin bir diğeri ile direk olarak bağlantısı vardır. Bu bağlantılar kullanıcı tarafından kontrol edilebilir. Şekil 2.1' de SPLD 'nin genel yapısı gösterilmiştir[5,20].



Şekil 2.1. SPLD Genel Yapısı[5].

2.1.2. Karmaşık Programlanabilir Lojik Birim (CPLD)

Karmaşıklıkları ve kapasiteleri bakımından SPLD ile FPGA'ler arasında yer alırlar. SPLD'lere göre anlamlı bir şekilde daha büyük tasarımları yapmaya izin verirken FPGA'lerden daha az lojik birim içerirler. Çoğunlukla pratik amaçlar için CPLD'ler tek çip içerisinde toplanmış çoklu SPLD'ler olarak tarif edilebilir. Bir CPLD'nin genel yapısı şekil 2.2.'de görülmektedir[5].



Şekil 2.2. CPLD Genel Yapısı[5].

2.1.3. Alanda Programlanabilir Kapı Dizileri (FPGA)

Alanda programlanabilir kapı dizileri yani FPGA , programlanabilir mantık blokları ve bu bloklar arasındaki programlanabilir ara bağlantılardan oluşan tümleşik devrelerdir. Tasarımcı devresini tasarlarken hem mantık bloklarını hem de ara bağlantılarını istediği gibi tasarlayabilmektedir. Alanda programlanabilir ismi

verilmesinin nedeni, mantık bloklarının ve ara bağlantıların imalat sürecinden sonra programlanabilmesidir[5].

Yapılarının daha basit olmasından dolayı SPLD, CPLD' den ve CPLD de FPGA' den daha hızlıdır. Yüksek kapasitelerinden dolayı da FPGA, CPLD' den ve CPLD de SPLD' den daha esnek, daha müdahale edilebilir bir yapıya sahiptir

FPGA bize mikroişlemciler ile yapamadığımız paralel işlem yapabilme özelliğini kazandırır. Bu sayede ASIC teknolojisinin hızına yaklaşılabilir. Sahip olduğu özellikler sayesinde FPGA'ler kendilerine birçok kullanım alanı bulmuşlardır. Havacılık ve savunma sanayi, otomotiv sanayi, Endüstriyel görüntüleme sistemleri, medikal sistemler ve güvenlik sistemleri gibi pek çok sistemde FPGA modülleri kullanılmaktadır[21].

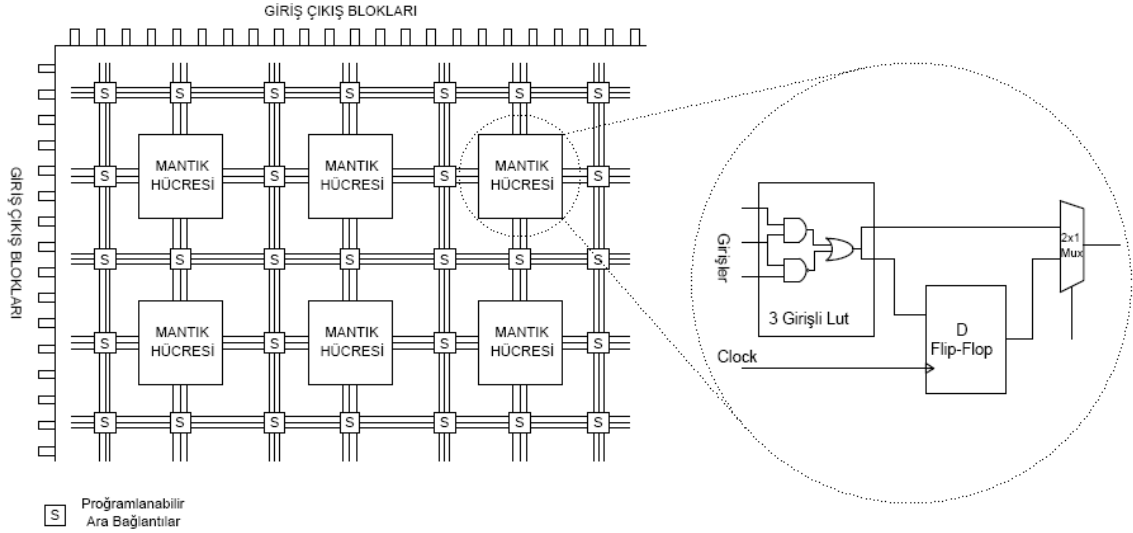
2.1.3.1. FPGA Temel Yapısı

FPGA temel olarak üç ana bölümden meydana gelir. Bunlar;

- *Mantık Hücreleri,
- *Programlanabilir Ara Bağlantılar,
- *Giriş/Çıkış(I/O) Bloklarıdır.

FPGA' in ana yapısını mantık hücreleri oluşturur. Bir mantık hücrelerini 1 adet Lookup Table (LUT), 1 adet D flip-flop ve 1 adet 2 giriş 1 çıkış multiplexer oluşturur. LUT' lar bir mantık işlemi yerine getiren küçük belleklerdir, n girişli bir LUT; 2^n kapasiteli bir belleği ifade işaret eder[22-23].

Mantık hücrelerinin ara bağlantıları matris şeklindeki veri yolları ve programlanabilir anahtarlarla sağlanır. Tasarımcı her bir mantık hücrelerinin uygulayacağı fonksiyonu ve programlanabilir anahtarların durumunu (açık/kapalı) belirleyerek bu mantık hücreleri arasındaki bağlantıları tanımlar. FPGA ve temel mantık hücresi temel yapısı şekil 2.3.'de verilmiştir[22].



Şekil 2.3. FPGA ve Mantık Hücresi Temel Yapısı.

Şekil 2.3’de görüldüğü üzere FPGA temel olarak 3 ana bölüme ayrılır. Bunlar; giriş-çıkış blokları, mantık hücreleri ve programlanabilir ara bağlantılardır. Giriş-çıkış blokları aygıtın etrafını saracak biçimde yerleştirilmiştir. Bu blokların her biri FPGA paketinin dışındaki I/O pinlerine input, output ve çift yönlü (hem giriş hem çıkış) olarak erişimi sağlarlar. Yapıda, I/O bloklarının iç kısmında ise dikdörtgen şeklinde diziler halinde mantık hücreleri bulunur. Mantık hücreleri lojik işlemlerin gerçekleştirildiği kısımlardır. Ara bağlantılar ile de hem mantık hücreleri arasındaki bağlantılar programlanabilir hem de mantık hücresi ile giriş-çıkış blokları arasındaki bağlantılar kontrol edilebilir[5].

FPGA pinleri, ayrılmış pinler ve kullanıcı pinleri olmak üzere iki kategoriye ayrılır.

Ayrılmış Pinler: Bir FPGA’de tüm pinlerin %20 ila %30’u ayrılmış pinlerdir. Bu pinler, FPGA’de gerçekleştirdikleri özel fonksiyonlara göre 3’e ayrılır. Bunlar;

Güç Pinleri: FPGA için gerekli olan güç ve toprak (ground) sağlayan pinlerdir.

Konfigürasyon Pinleri: Programın FPGA’ye yüklenmesi için kullanılan pinlerdir.

Clock Pinleri: Clock sinyalleri için ayrılmış özel pinlerdir[22].

Kullanıcı Pinleri: Bu pinler kullanıcı tarafından konfigüre edilebilen standart I/O pinleridir. Giriş (input), çıkış (output) ve giriş/çıkış (inout) olarak 3 farklı şekilde kullanılabilir. Her bir I/O pini FPGA’de bir IO hücresine bağlıdır, bir grup IO pini 2,5 V ile çalışırken bir grup pin 3,3 V ile çalışabilmektedir[22].

2.2. VHDL

Bu kısımda tüm yapılandırılabilir aygıtlar için kullanılabilen donanım tanımlama (HDL) dilleri arasında en yaygın kullanım alanına sahip olan VHDL (Very High Speed Integrated Circuit Hardware Description Language) donanım tanımlama dili hakkında bilgi verilecektir.

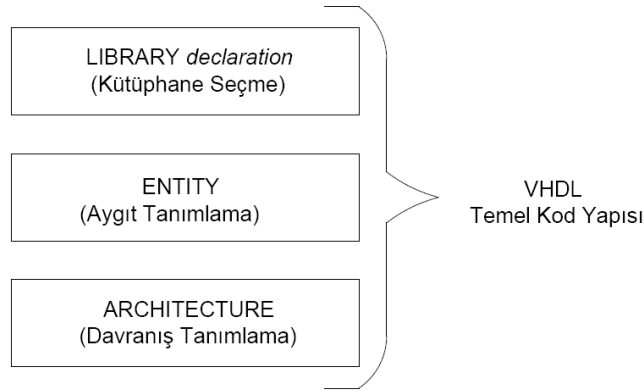
İhtiyaçlar belirlendikten sonra yapılandırılabilir aygıtların tasarım süreci başlar. Tasarım sözel olarak HDL (Hardware Description Language) veya FPGA üreticileri tarafından hazırlanan şematik editörler kullanılarak yapılabilir. Tasarım hangi yöntemle yapılırsa yapılsın son aşamada yani derleme işleminde tüm bağlantılar standart netlist (bağlantı) biçimine dönüştürülür.

VHDL ilk olarak 1970'lerin sonlarına doğru kullanılmaya başlandı ve 1981 yılında Amerikan Savunma Sanayi (Department of Defense, DoD) için ilk chipsetler üretildi. Bundan sonra da giderek gelişen ve yaygınlaşan bir dil oldu. İlk zamanlar farklı firmalar ürün geliştirmek için farklı donanım tanımlama dilleri kullandı. VHDL donanım tanımlama dili standartlaştırılması için IEEE (Institute of Electrical and Electronics Engineers)'ye transfer edilerek Aralık 1987 de IEEE tarafından onaylanarak IEEE 1076–1987 standardı haline getirildi. Beş yıl sonra dilin biçimine yeni özellikler eklenerek oluşturulan yeni versiyonu (IEEE Std 1076–1993) oluşturulmuştur. Yeni versiyon, saf standardın “genişletilmiş” versiyonu olup bu genişletme ile büyük çapta ihtiyaç duyulan veri tipleri ve alt programlar içeren paketleri (std_logic_1164, numeric_bit, numeric_std, ...) ya da IEEE 1076.6 'nın sentez altkütmesine benzeyen özel VHDL altkütmeleri eklenmiştir. Bu genişletilmiş versiyondan sonra VHDL'e analog ve karışık sinyal dil elemanlarının eklenerek VHDL'in superset'i olarak adlandırılan VHDL-AMS (analogue - mixed signal) isimli tanımlama dili oluşturulmuştur. VHDL, elektronik tasarım ve ürün geliştirmede gittikçe önemi artan bir dil konumuna gelmiştir günümüzde de elektronik ve dijital tasarımın kalbi konumundadır. Teknolojik ürünlerin, ASIC tipi entegrelerin ve yarı iletken teknolojisinin gelişmesinde önemli bir paya sahiptir. VHDL dilinin kullanılması ile yarı iletken malzemelerin boylarındaki küçülme doğru orantılıdır[24-27].

Bundan on yıl önce elektronik ürün tasarımında şematik tasarım ve VHDL beraber kullanılırken elektronik ürünlerin daha kompleks ve karmaşık bir yapıya dönüşmeleri sonucu şematik tasarım önemini kaybederken VHDL daha da popüler bir hale gelmiştir. VHDL' in yeni durum ve şartlara kolayca adapte edilebilen mimari yapısı dolayısıyla tasarımcıların ilk terchi konumuna gelmiştir[25].

2.2.1. VHDL Kod Yapısı Temel Birimleri

VHDL kodlar yazılırken dikkat edilecek hususlar; kodlar ';' ile sonlandırılır. Programda '-' açıklama yapmak için kullanılır. Derleyici '-' işaretini gördükten sonra o satırda işlem yapmaz. VHDL büyük küçük harf duyarlı değildir. VHDL kodlar şekil 2.4'de gösterildiği gibi en az üç temel bölümden meydana gelir.



Şekil 2.4. VHDL Temel Kod Yapısı[24].

2.2.1.1. Kütüphane Seçme (Library Declaration)

Programın ilk satırlarında yapılacak tasarıma göre mevcut kütüphallerden istenenler seçilerek programda tanımlı hale getirilir. Şekil 2.5 de genel kütüphane tanımlama formatı ve her programda kullanılan kütüphane ve paketleri gösterilmiştir. Uygulamalarda *std* ve *work* kütüphanesi başlangıçta otomatik olarak tanımlandığından sadece iee kütüphanesinin tanımlanması yeterlidir[24].

LIBRARY <i>kütüphane_adi</i> ; USE <i>kütüphane_adi.Paket_ismi.Paket_bölümü</i> ;	
LIBRARY <i>ieee</i> ; USE <i>ieee.std_logic._1164.all</i> ;	-- <i>ieee kütüphanesi</i> -- <i>std_logic paketinin 1164 bölümü</i>
LIBRARY <i>std</i> ; USE <i>std.standart.all</i> ;	-- <i>std kütüphanesi</i> -- <i>standart isimli paketin tamamı</i>
LIBRARY <i>work</i> ; USE <i>work.all</i> ;	-- <i>work kütüphanesi</i> -- <i>work isimli paketin tamamı</i>

Şekil 2.5. Kütüphane Tanımlama ve Kullanılan Kütüphaneler[24].

2.2.1.2. IEEE Kütüphanesinde Kullanılan Paketler

Std_logic_1164: Spesifik STD_LOGIC ve STD_ULONGIC çoklu değerli dijital tasarımlarda kullanılır[24].

Std_logic_arith: Spesifik SIGNED ve UNSIGNED data tipleri ile aritmetik ve karşılaştırma işlemleri tanımlanır. Paket ayrıca değişik türdeki dataların birbirine dönüştürülmesine de imkan tanır[24].

Std_logic_signed: SIGNED tipi STD_LOGIC_VECTOR datalardaki tüm işlemler için kullanılır[24].

Std_logic_unsigned: UNSIGNED tipi STD_LOGIC_VECTOR datalardaki tüm işlemler için kullanılır[24].

2.2.1.3. Data Tipleri

Bit ve Bit_Vector: Sadece sıfır ve bir değerlerini alabilirler. **Bit** ve **Bit_Vector** kullanımı ile ilgili örnekler aşağıda verilmiştir[24].

SIGNAL x:bit; -- *x, bit tipi bir data.*

SIGNAL y:bit_vector(3downto0); -- *y, bit_vector tipi 4 bitlik bir data*
(en değerli bit en solda).

SIGNAL x:bit_vector(0 to 5); --*z, bit_vector tipi 6 bitlik bir data*
(en değerli bit en sağda).

```

x<=1;           -- x'e 1 değeri atandı.
y<=1010;       -- y'ye 1010 değeri atandı (MSB=1).
z<=110010;    -- z'e 110010 değeri atandı (MSB=0).

```

Std_logic ve Std_logic_Vector: Bu tip datalar IEEE'de tanımlı 8 tür değeri alabilir. Bu değerler ve açıklamaları aşağıda verilmiştir[24].

```

'X' : Bilinmeyen değer.
'0' : Lojik0.
'1' : Lojik1.
'Z' : Yüksek empedans (tri state bufferlarda kullanılır).
'W' : Zayıf.
'L' : Düşük.
'H' : Yüksek.
'- ' : Önemsiz.

```

Boolean: Bu tip datalar sadece 'true' ve 'false' değerlerini alabilir.

Integer: - 2,147,483,467'den + 2,147,483,467' ye tüm tamsayı değerleri alabilir.

Natural: 0'dan + 2,147,483,467' ye kadar olan tamsayı değerleri alabilir.

Real: -1.0E38'den + 1.0E38'ye kadar tüm değerleri alabilir.

Bit, Bit_Vector, Std_logic, Std_logic_Vector ve **Integer** kullanımı ile ilgili örnekler aşağıda verilmiştir[24].

```

SIGNAL a:bit;           -- a, bit tipi bir bit data.
SIGNAL b:bit_vector(7 downto 0); -- b, bit_vector tipinde 8 bitlik bir data.
SIGNAL c:std_logic;    --c, std_logic tipinde bir bitlik bir data.
SIGNAL d:std_logic_vector(7 downto 0); -- d, std_logic_vector 8 bitlik bir data.
SIGNAL e:integer 0 to 255; -- e, 0-255 arası tamsayı değerleri.
a<=b(5);               -- DOĞRU
b(0)<=a;               -- DOĞRU
c<=d(5);               -- DOĞRU

```

```
d(0)<=c;           -- DOĞRU
a<=c;             -- YANLIŞ (data tipleri eşleşmiyor)
b<=d;             -- YANLIŞ (data tipleri eşleşmiyor)
e<=b;             -- YANLIŞ (data tipleri eşleşmiyor)
e<=d;             -- YANLIŞ (data tipleri eşleşmiyor)
```

2.2.1.4. Aygıt Tanımlama (Entity)

VHDL dilinde aygıt tanımlamaları ‘Entity’ içerisinde yapılır. Tasarlanan aygıtın kaç tane pini bulunacağı bu pinlerden hangilerinin giriş (in), hangilerinin çıkış (out) ve hangilerinin de giriş/çıkış (inout,buffer) olduğu bu kısımda belirtilir. Yine bu pinlerin skaler mi yoksa vektörel mi olacakları ve data tipleri de bu kısımda belirtilir. Genel kullanımı aşağıda gösterildiği gibidir[24].

```
ENTITY aygıt_ismi IS
  PORT (
    port_ismi:sinyal modu, sinyal türü;
    port_ismi:sinyal modu, sinyal türü;
    ...);
END aygıt_ismi;
```

2.2.1.5. Davranış Tanımlama (Architecture)

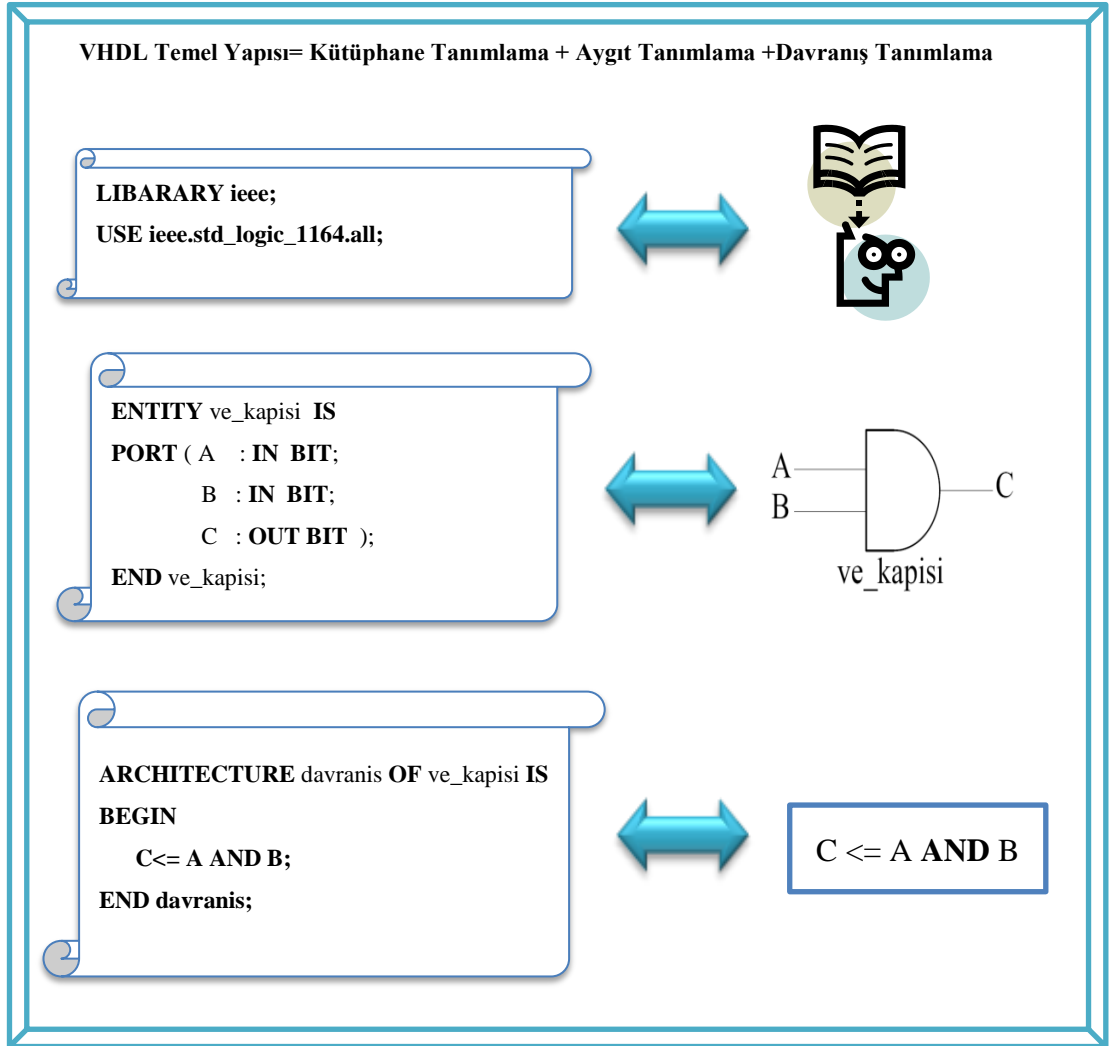
Tasarlanan aygıtın nasıl çalışması gerektiğinin belirtildiği kısımdır. Genel kullanım şekli aşağıda gösterildiği gibidir [24].

```
ARCHITECTURE yapı_ismi OF aygıt_ismi IS
  [tanımlamalar]
BEGIN
  [kod]
END yapı_ismi;
```

2.2.1.6. VHDL İle Oluşturulan Örnek Yapılar

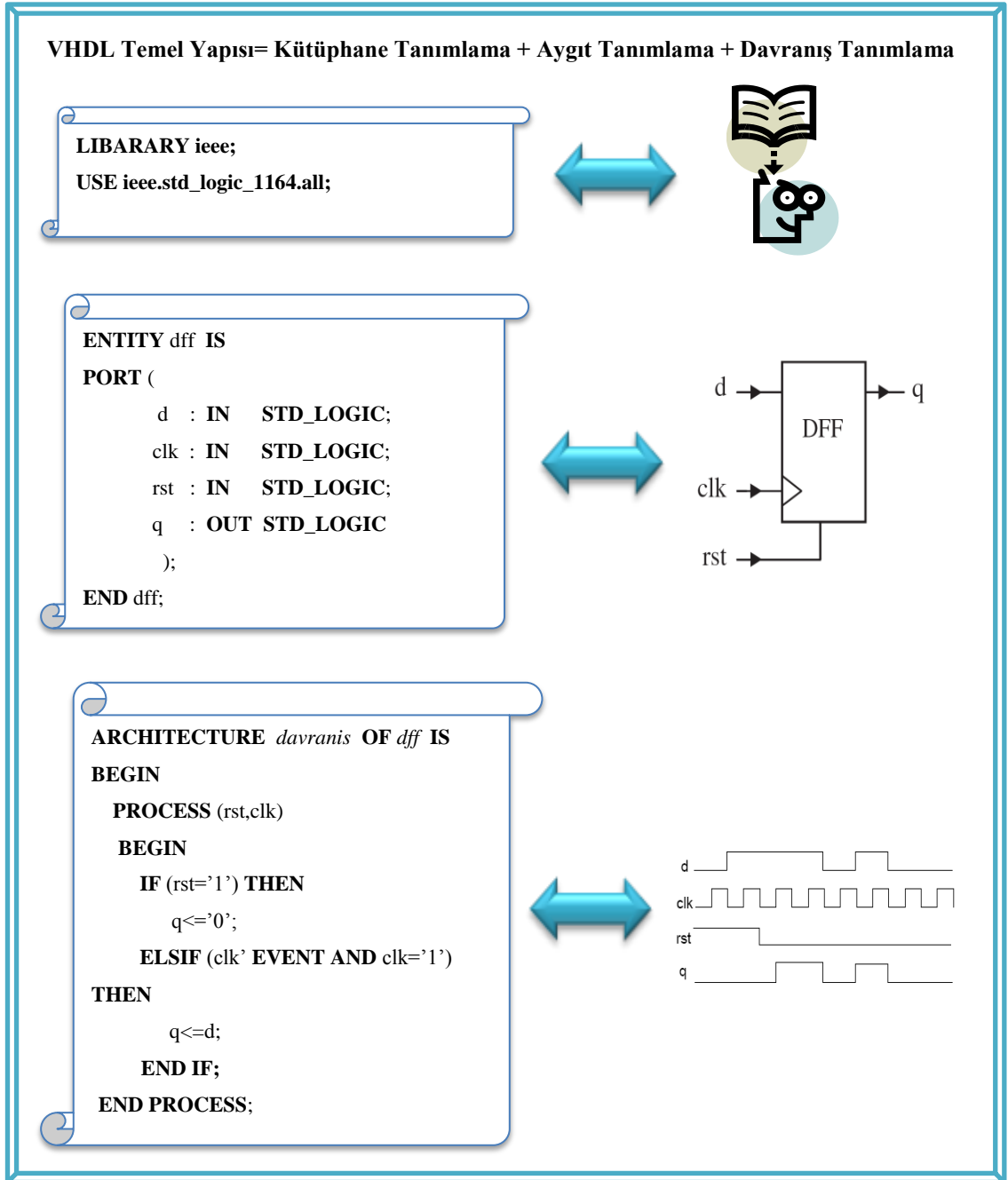
VHDL donanım tanımlama dilini kullanılarak oluşturduğumuz örnek tasarımlardan bazıları aşağıda verilmiştir.

Örnek 1: İki girişli ‘VE’ kapısının VHDL ile tasarlanması şekil 2.6’ da gösterilmiştir



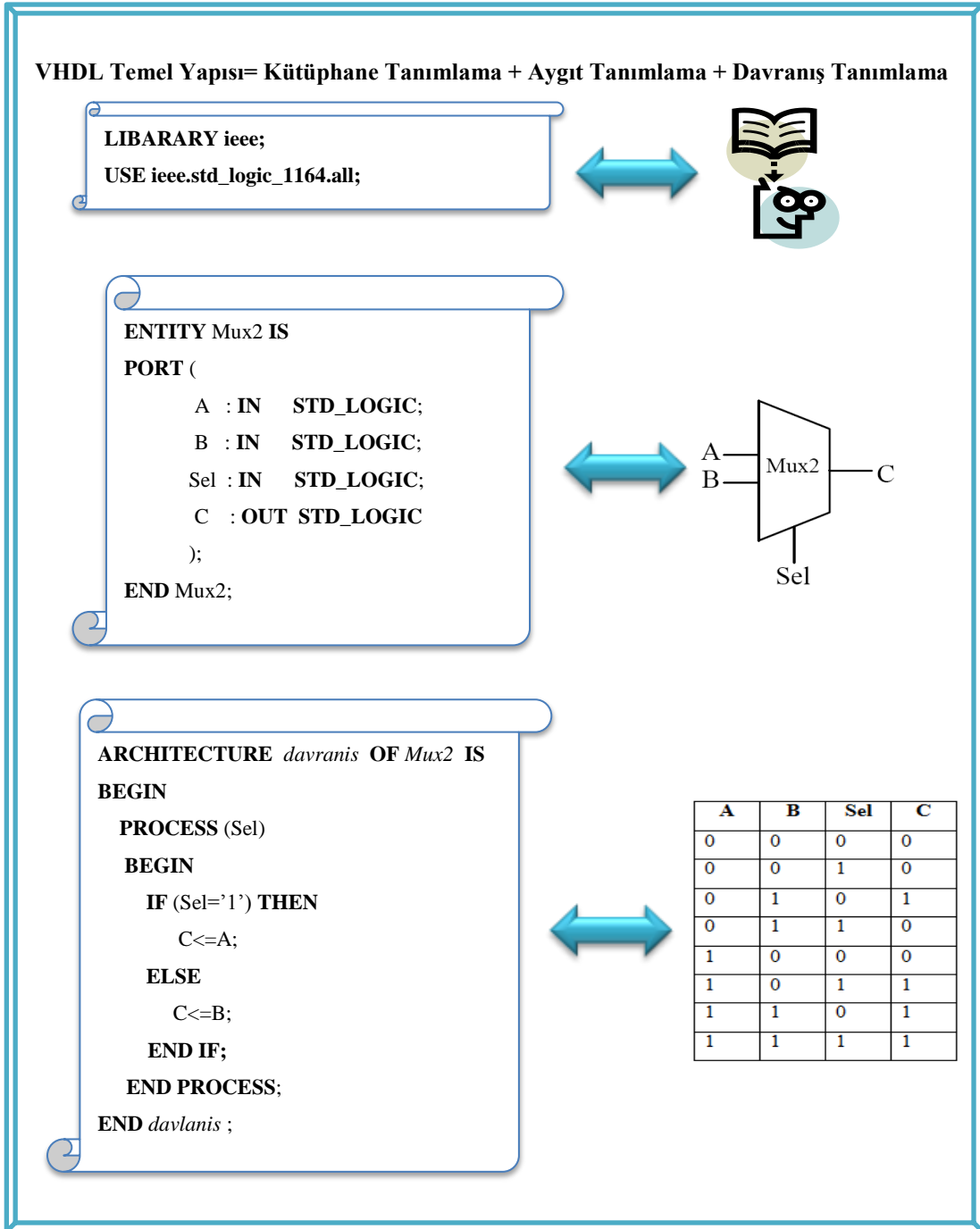
Şekil 2.6. VHDL İle İki Girişli VE Kapısı Tasarımı.

Örnek 2: VHDL ile DFF tipi flip flop tasarlanması şekil 2.7’de gösterilmiştir.



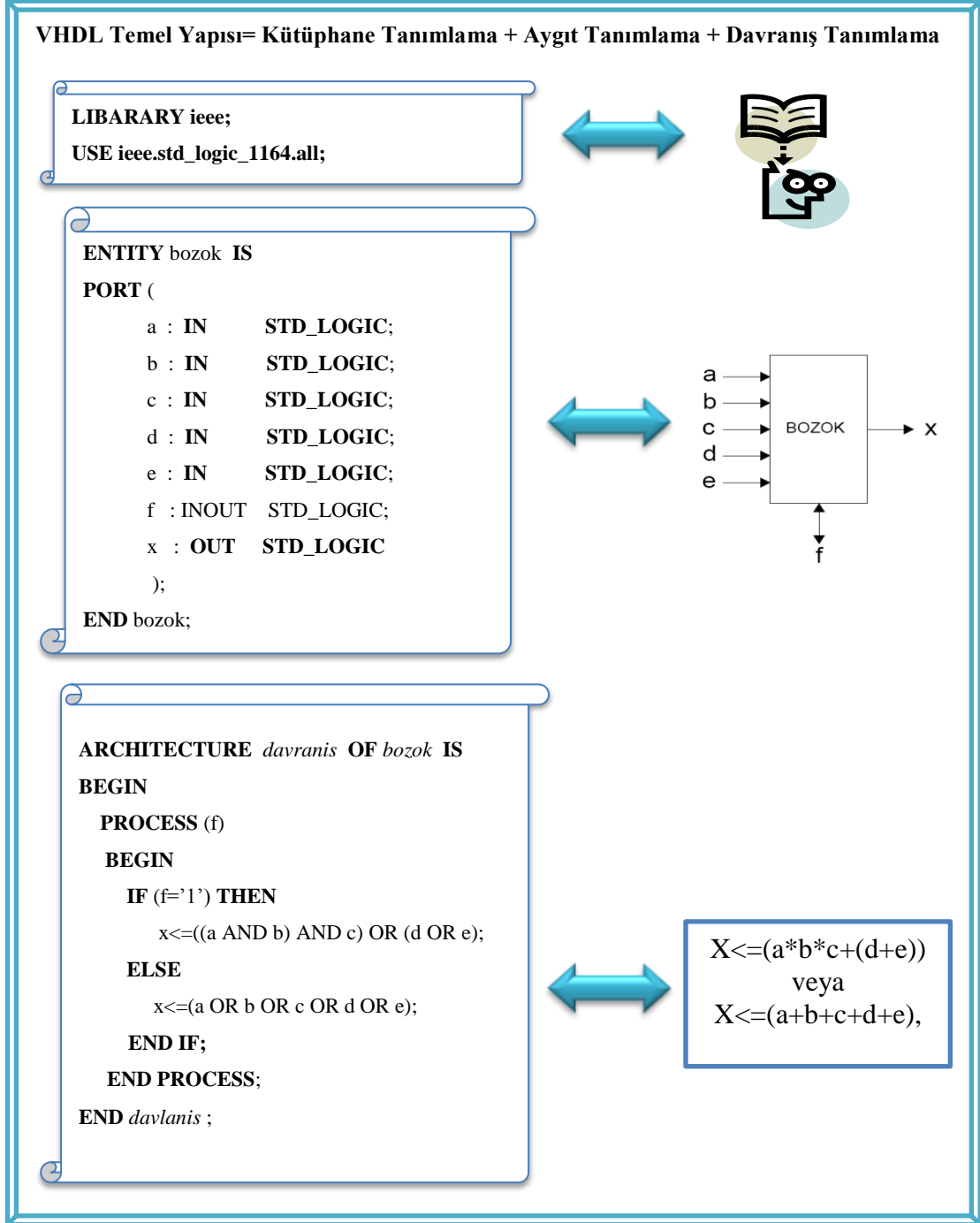
Şekil 2.7. VHDL İle DFF Flip Flop Tasarımı.

Örnek 3: İki girişli Multiplexerin VHDL ile tasarlanması şekil 2.8’de gösterilmiştir.



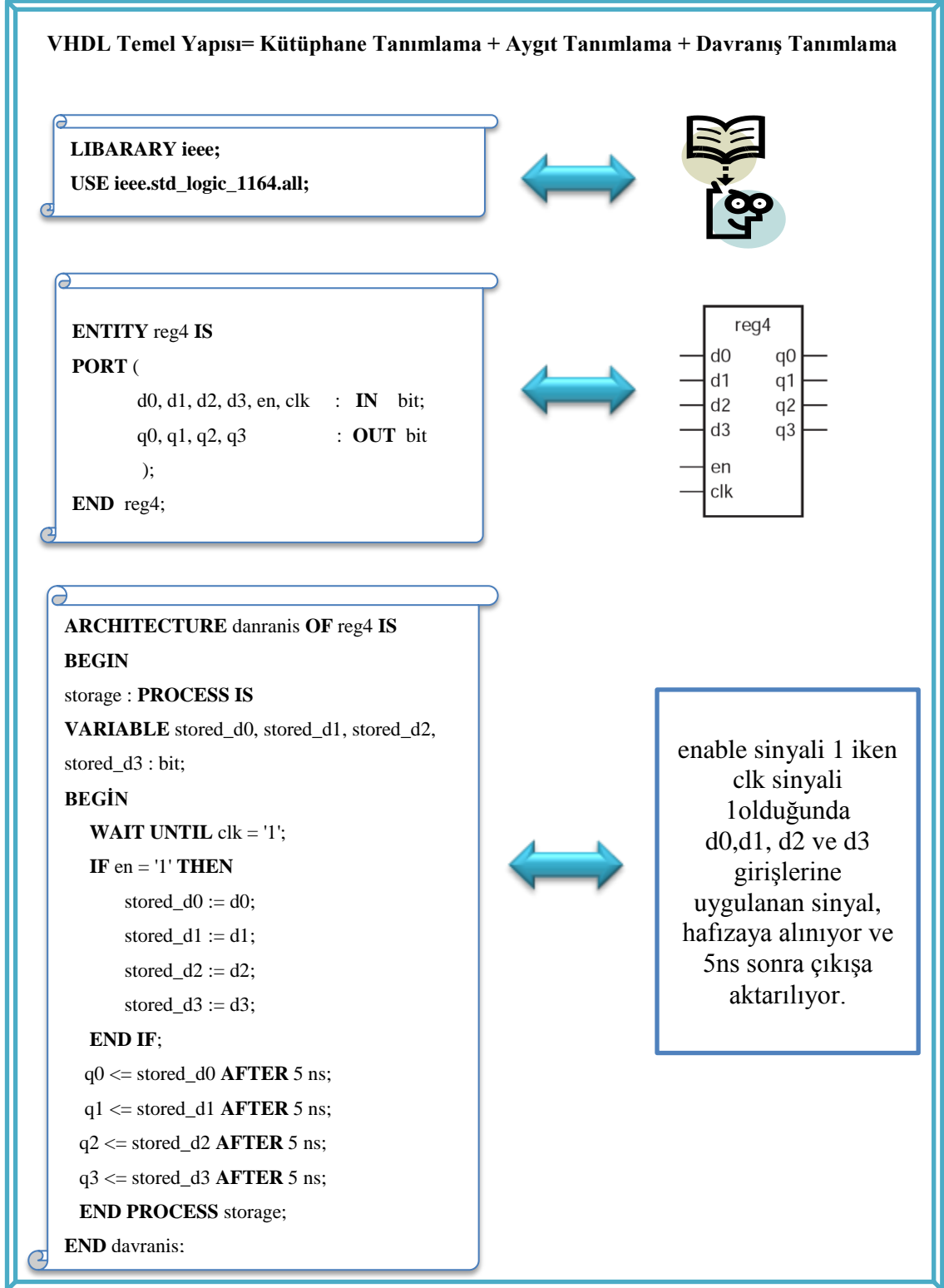
Şekil 2.8. VHDL İle Multiplexer Tasarımı.

Örnek 4: VHDL ile toplama ve çarpma işlemi yapabilen, istenen fonksiyonu çalıştıran BOZOK isiminde bir aygıt tasarımı şekil 2.9’ da gösterilmiştir.



Şekil 2.9. VHDL İle İstlenen Fonksiyonu Yerine Getiren Aygıt Tasarımı.

Örnek 5: VHDL ile 'reg4' isimli dört bitlik register tasarımı şekil. 2.10' da gösterilmiştir



Şekil 2.10. VHDL İle Dört Bitlik Register Tasarımı[28].

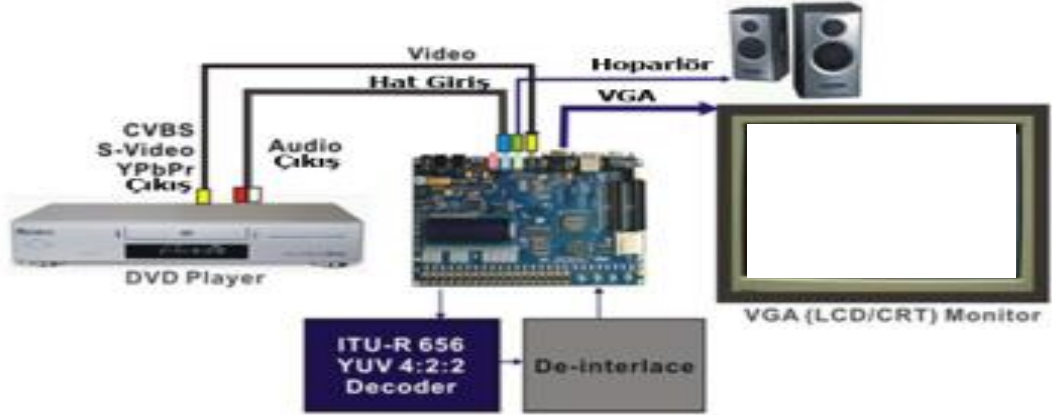
2.3. Altera De2 Geliştirme Kiti Ve Wolfson Wm8731 Entegresi

Bu bölümde Altera DE2 geliştirme ve eğitim seti ile bu set üzerinde bulunan Wolfson WM8731 entegresi ve özellikleri hakkında bilgi verilecektir.

2.3.1. Altera DE2 Geliştirme Ve Eğitim Seti

Altera Firmasının ürettiği Altera DE2 eğitim seti; dijital sistemleri, bilgisayar organizasyonları ve FPGA konularını geliştirmek için ideal bir ortam oluşturmaktadır. Bu set çok zengin alternatifler sunarak hem öğrencilere hem de profesyonel düzeyde çalışma yapan kişilere hitap etmektedir[29].

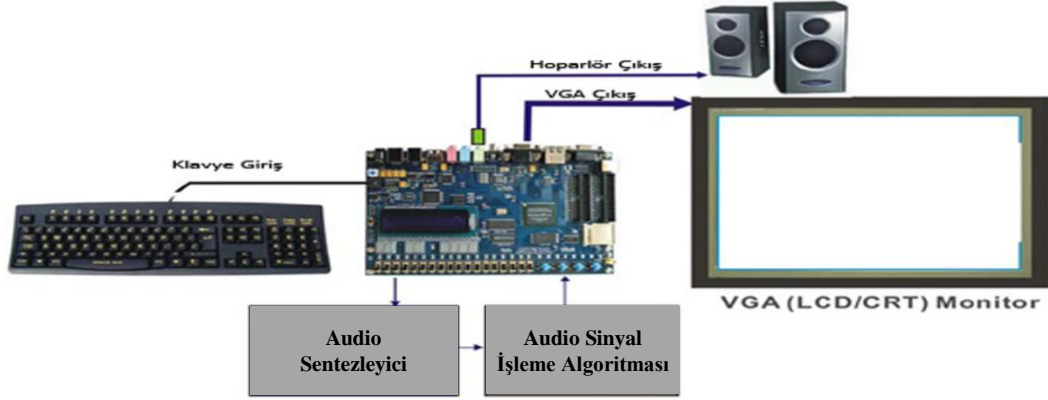
Altera DE2 geliştirme ve eğitim seti, standart giriş/çıkış ara yüzleri için yazılım desteği ve çeşitli bileşenlere erişim için kontrol paneli imkânına sahiptir. DE2 kitiyle ses ve video uygulamaları, display, LCD panel ve monitör uygulamaları, seri ve paralel port uygulamaları, USB ve ethernet bağlantısı vb. gibi daha birçok uygulama yapılabilir. Örneğin VGA çıkışı ve ses kodeği kullanılarak bir DVD oynatıcıdan video ve ses girişleri oynatılabilir. Video girişine bağlanan bir kamera ile gerçek zamanlı görüntü işleme yapılarak VGA monitöre görüntü aktarılabilir. Altera DE2 geliştirme kiti ile video işleme örneği şekil 2.11’de gösterilmiştir[30].



Şekil 2.11. Altera DE2 Geliştirme İle Kiti Video İşleme[30].

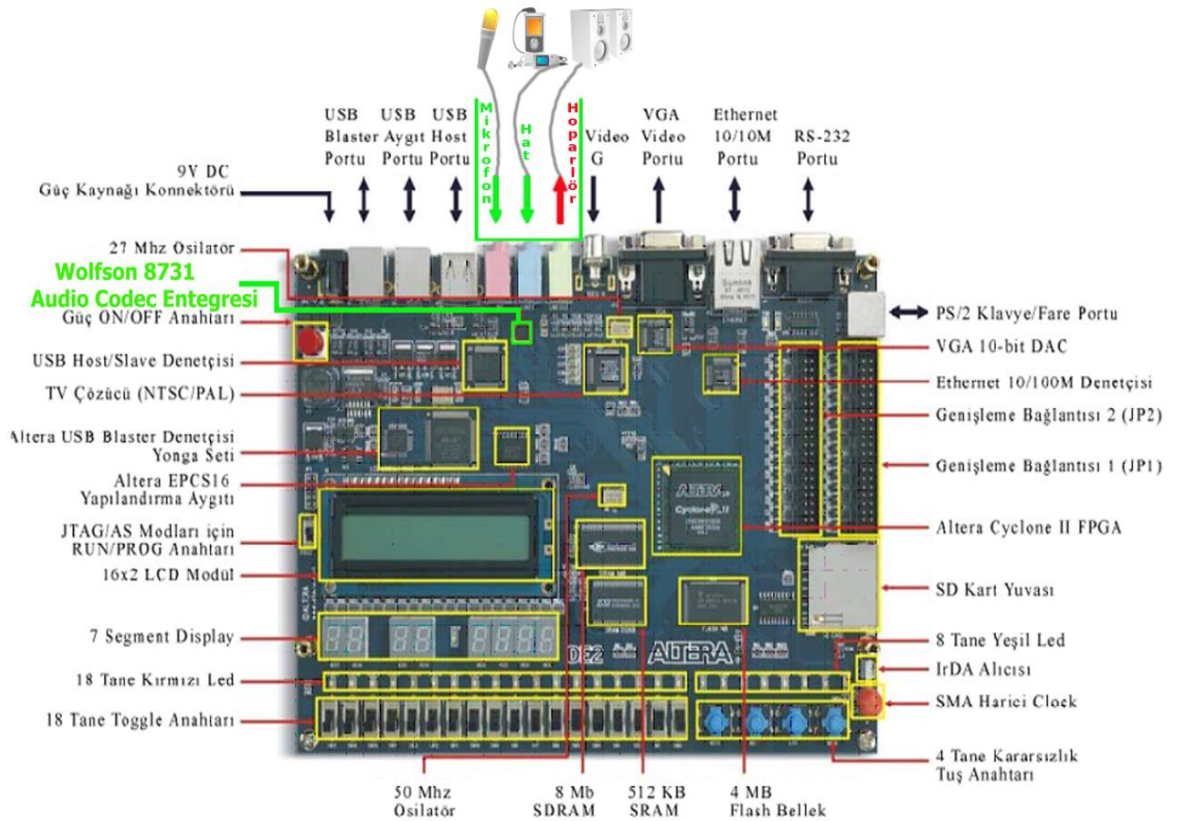
Altera DE2 geliştirme ve eğitim seti, farklı bir uygulama olarak host ve aygıt uygulamaları için tam bir USB çözümü sağlar. DE2 kitiyle karaoke makine uygulaması oluşturmak için mikrofon giriş, hat giriş ve çıkış portları kullanılabilir. Hızlı Ethernet kontrolcüsü kullanılarak veri paketleri gönderilip alınabilir. Ayrıca DE2 kiti SD kart erişimi ve profesyonel ses performansı için ihtiyaç duyulan

donanım ve yazılımı sağlar. DE2 kitini kullanarak PS/2 klavye ve bir hoparlör kullanarak birçok sesli elektronik klavye yapılabilir. PS/2 klavye ile yapılmış org şeması şekil 2.12’ de verilmiştir[30].



Şekil 2.12. Altera DE2 Geliştirme Kiti Klavye Ve Monitor İle Ork[30].

Altera DE2 geliştirme kartının sahip olduğu özellikler şekil 2.13’de gösterilmiştir.



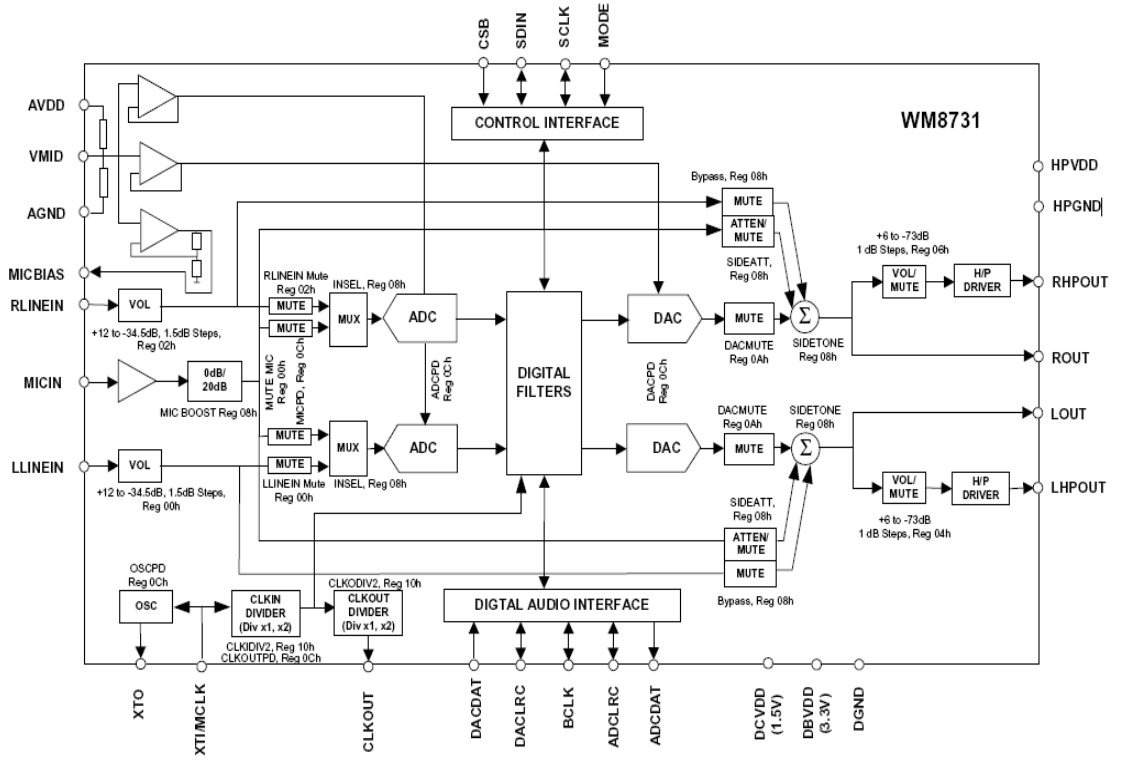
Şekil 2.13. Altera DE2 Geliştirme Kartı[30].

2.3.2. Wolfson Wm8731 Entegresi

Wm8731 entegresi özellikle taşınabilir mp3 player, pda, akıllı telefon, tablet, ses kayıt ediciler, media oynatıcılar, cd çalar gibi cihazlar için tasarlanmış küçük güç tüketimli audio kodek entegresidir.

Wm8731 entegresi kendi içerisinde analog dijital dönüştürücü (ADC) ve dijital analog dönüştürücü (DAC) 'a sahiptir. ADC çıkış sinyal genişliği 16-32 bit arasında ve örnekleme frekansı 8kHz-96 kHz arasında kullanıcı tarafından set edilebilmektedir. 28 pine sahip olan bu aygıt diğer aygıtlar ile seri olarak haberleşebilmektedir. Wm8731 entegresi mono mikروفon girişine, stereo hat girişine ve logaritmik değişen hoparlör çıkışına sahiptir.

Wm8731 entegresi register haritası, kontrol arayüzü ve dijital audio arayüzü olmak üzere üç bölümde incelenecektir. Bu bölümler oluşturulurken üretici firmanın ürün bilgi sayfalarından faydalanılmıştır[31]. Wm8731 entegresine ait blok diyagram şekil 2.14' de verilmiştir.



Şekil 2.14. Wm8731 Blok Diyagramı[31].

2.3.2.1. Wolfson Wm8731 Registerları

Wm8731 Entegresi 11 adet 16 bitlik registera sahiptir. Sol kısımdaki 7 bit, register adresini, sonraki 9 bit ise register içeriğini ifade eder. Tablo 2.1’ de Wm8731 entegresi register haritası verilmiştir.

Tablo 2.1. Wm8731 Entegresi Register Haritası.

Register	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
R0(00hex)	0	0	0	0	0	0	0	LRIN BOTH	LIN MUTE	0	0	LINVOL				
R1(02hex)	0	0	0	0	0	0	1	RLIN BOTH	RIN MUTE	0	0	RINVOL				
R2(04hex)	0	0	0	0	0	1	0	LRHP BOTH	LZCEN	LHPVOL						
R3(06hex)	0	0	0	0	0	1	1	RLHP BOTH	RZCEN	RHPVOL						
R4(08hex)	0	0	0	0	1	0	0	0	SIDEATT		SIDE TONE	DACSEL	BY PASS	INSEL	MUTE MIC	MIC BOOST
R5(0Ahex)	0	0	0	0	1	0	1	0	0	0	0	HPOR	DACMU	DEEMPH		ADC HPD
R6(0Chex)	0	0	0	0	1	1	0	0	PWR OFF	CLK OUTPD	OSCPD	OUTPD	DACPD	ADCPD	MICPD	LINEINPD
R7(0Ehex)	0	0	0	0	1	1	1	0	BCLK INV	MS	LR SWAP	LRP	IWL		FORMAT	
R8(10hex)	0	0	0	1	0	0	0	0	CLKO DIV2	CLKI DIV2	SR				BOSR	USB/ NORM
R9(12hex)	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	ACTIVE
R15(1Ehex)	0	0	0	1	1	1	1	0	RESET							

R0 ve R1 registerleri sağ ve sol kanal hat girişlerini kontrol eder. Girişler yüksek empedans ve düşük kapasitans özelliği gösterir. Bu şekilde harici kaynaklardan yüksek kalite sinyal alınır. R0 ve R1 register özellikleri tablo 2.2’ de verilmiştir.

Tablo 2.2. R0 ve R1 Register İçeriği.

Register Adres	Bit	Etiket	Varsayılan Değer	Açıklama
0000000 Sol Kanal Hat Giriş	4:0	Linvol[4:0]	10111 (0db)	Sol Kanal Hat Girişi Ses Kontrol 11111= 12db..Her Adımda 1.5db Değişim..00000= -34.5db
	7	Linmute	1	Sol Kanal Ses Giriş 1:Ses Açık 0:Sessiz
	8	Lrinboth	0	Sağ Ve Sol Kanal Kullanımı 1:Eşzamanlı Kullanım Açık 0:Eşzamanlı Kullanım Kapalı

0000001 Sağ Kanal Hat Giriş	4:0	Rinvol[4:0]	10111 (0db)	Sağ Kanal Hat Girişi Ses Kontrol 11111= 12db..Her Adımda 1.5db Değişim..00000= -34.5db
	7	Rinmute	1	Sağ Kanal Ses Giriş 1:Ses Açık 0:Sessiz
	8	Rlinboth	0	Sağ Ve Sol Kanal Kullanımı 1:Eşzamanlı Kullanım Açık 0:Eşzamanlı Kullanım Kapalı

R2 ve R3 registerları sağ kanal (RHPOUT) ve sol kanal (LHPOUT) olmak üzere stereo çıkışı kontrol eder. Hoparlör çıkışı yüksek ses seviye ayarları ve özel ses fonksiyonlarına sahiptir. R2 ve R3 register özellikleri tablo 2.3' de verilmiştir.

Tablo 2.3. R2 ve R3 Register İçeriği.

Register Adres	Bit	Etiket	Varsayılan Değer	Açıklama
0000010 Sol Hoparlör hat Çıkış	6:0	Lhpvol[6:0]	1111001 (0db)	Sol Hoparlör Hat Çıkış Ses Kontrol 1111111= 6db..Her Adımda 1db Değişim 0110000= -73db 0101111 Den 000000'a= Sesiz
	7	Lzcen	0	Sol Kanal Düşen Kenar Algılama 1:Aktif 0:Pasif
	8	Lrhpboth	0	Sol Kanaldan Sağa Doğru Eşzamanlı Kullanım 1:Eşzamanlı Kullanım Açık 0:Eşzamanlı Kullanım Kapalı
0000011 Sağ Hoparlör Hat Çıkış	4:0	Rhpvol[6:0]	1111001 (0db)	Sağ Hoparlör Hat Çıkış Ses Kontrol 1111111=6db..Her Adımda 1db Değişim 0110000=-73db 0101111 Den 0000000'a=Ses
	7	Rzcen	0	Sağ Kanal Düşen Kenar Algılama 1:Aktif 0:Pasif
	8	Rlhboth	0	Sağ Kanaldan Sola Doğru Eşzamanlı Kullanım 1:Eşzamanlı Kullanım Açık 0:Eşzamanlı Kullanım Kapalı

R4 registeri analog veri yolunu yani hat ve mikrofon girişini kontrol eder. ADC girişi olarak hem hat girişi hem de mikrofon seçilebilir. Mikrofon girişi yüksek empedans

ve düşük kapasitansı ile değişik özellikte mikrofonlardan gelen ses sinyallerini hassas bir şekilde alabilmektedir. R4 registeri ve özelliği tablo 2.4' de verilmiştir.

Tablo 2.4. R4 Register İçeriği.

Register Adres	Bit	Etiket	Varsayılan Değer	Açıklama
0000100 Analog Audio Sinyal Kontrol	0	Micboost	0	Mikrofon Giriş Seviyesi 1=Mikrofon Ses Seviyesini Kuvvetlendir 0=Mikrofon Ses Seviyesi Normal
	1	Mutemic	1	ADC Giriş İçin Kaynak Seçimi 1:Hat Giriş Aktif 0:Hat Giriş Pasif
	2	Insel	0	Adc İçin Kaynak Seçimi 1=Mikrofon Seçildi 0=Hat Giriş Seçildi
	3	Bypass	1	Mikrofonu Bypass Et 1=Aktif 0=Pasif
	4	Dacsel	0	Dijital Analog Dönüştürücü Çıkışı 1=Aktif 0=Pasif
	5	Sidetone	0	Mikrofon Girişi 1=Aktif 0=Pasif
	6:7	Sideatt [1:0]	00	Sidetone Seçilen Girişi Zayıflat 11= -15db Azalt, 10= -12db Azalt, 01= -9db Azalt, 00= -6db Azalt.

R5 registeri dijital audio işlemleri set eder. Analog Dijital Dönüştürücü (ADC); yüksek geçiren bir filtreye sahiptir ayrıca tasarımcıya yazılımla kontrol imkânı vermektedir. Filtre tipi ve özellikleri seçilen örnekleme frekansına bağlı olarak otomatik olarak değişmektedir. Tablo 2.5' de R5 registeri özelliği verilmiştir.

Tablo 2.5. R5 Register İçeriği.

Register Adres	Bit	Etiket	Varsayılan Değer	Açıklama
0000101 Dijital Audio Sinyal Kontrol	0	ADCHPD	0	ADC Yüksek Geçiren Filtre 1= Filtre Pasif 0=Filtre Aktif
	2:1	DEEMP [1:0]	00	Yüksek Geçiren Filtre Frekansı 11 = 48kHz 10 = 44.1kHz

				01 = 32kHz 00 = Filtre Pasif
	4	HPOR	0	ADC Yüksek Geçiren Filtre Pasif İken 1:offset değerin depola 0:ofset değerini 0 yap

R6 Registeri performanstan ziyade güç tasarrufunun önemli olduğu yerlerde kullanılır. Wm8731 Entegresinde kullanılan enerjiyi minimum seviyede tutar. Tablo 2.6' da R6 registeri ve fonksiyonları verilmiştir.

Tablo 2.6. R6 Register İçeriği.

Register Adres	Bit	Etiket	Varsayılan Değer	Açıklama
0000110 Güç Kontrol	0	LINEINPD	1	Hat Giriş Güç Azalt 1= Aktif, 0= Pasif.
	1	MICPD	1	Mikrofon Girişi Güç Azalt 1= Aktif, 0= Pasif.
	2	ADCPD	1	ADC Güç Azalt 1= Aktif, 0= Pasif.
	3	DACPD	1	DAC Güç Azalt 1= Aktif, 0= Pasif.
	4	OUTPD	1	ADC Güç Azalt 1= Aktif, 0= Pasif.
	5	OSCPD	0	Osilatör Güç Azalt 1= Aktif, 0= Pasif.
	7	POWEROFF	1	Güç Kapatma 1= Aktif, 0= Pasif.

R7 registeri dijital audio arayüzünde data formatı, data genişliği ve aygıtın master veya slave olarak çalışması gibi gerekli ayarlar yapılır. Yapılan uygulamada data akış formatı olarak I2S, data genişliği olarak 16 bit, çalışma modu olarak slave mod ve DACLRC düşük iken sağ kanal bilgisi gelecek şekilde set edildi. Tablo 2.7' de R7 registeri ve fonksiyonları verilmiştir.

Tablo 2.7. R7 Register İçeriği.

Register Adres	Bit	Etiket	Varsayılan Değer	Açıklama
0000111 Dijital Audio Arayüzü	1:0	FORMAT [1:0]	10	Audio Data Formatı Seçilir 11= DSP Mod, 10= I2S Mod - Sola Dayalı, 01= MSB - Sola Dayalı, 00= MSB – Sağa Dayalı.
	3:2	IWL [1:0]	10	Giriş Data Genişliği 11= 32 Bit 10= 24 Bit, 01= 20 Bit, 00= 16 Bit.
	4	LRP	0	Dijital Analog Konvertör Sağ Sol Clk DACLRC 1= Dijital Analog Konvertöre DACLRC yüksek iken sağ kanal dataları gelir 0= Dijital Analog Konvertöre DACLRC düşük iken sağ kanal dataları gelir
	6	MS	0	Master Saleve Mod 1= Master Mod, 0= Slave Mod.
	7	BCLKINV	0	Bit Clok Frekansını Tersle 1=Tersleme Aktif, 0= Tersleme Pasif.

Wm8731 entegresi ADC ve DAC için mod ve normal mod, 2 farklı motta clok frekansı üretmektedir. Eğer istenirse ADC ve DAC için farklı örnekleme frekansları seçilebilir. R8 Registeri ile spesifik master clock frekansı ve örnekleme frekans ayarları yapılır. Yapılan uygulamada usb mod seçildi ve örnekleme frekansı 8khz yapıldı. Tablo 2.8’ de R87 registeri ve fonksiyonları verilmiştir.

Tablo 2.8. R8 Register İçeriği.

Register Adres	Bit	Etiket	Varsayılan Değer	Açıklama
0001000 Örnekleme Frekans Kontrol	0	Usb/ Normal	0	Mod Seçimi 1= Usb Mod, 0= Normal Mod.
	1	Bosr	0	Örnekleme Oranı 0= Usb Mod 250fs Normal Mod 256fs 1= Usb Mod 272fs

				Normal Mod 384fs
	5:2	Sr[3:0]	0000	Örnekleme Frekansı 0000= 48khz, 0011= 8khz, 0110=32khz, 0111= 96khz,

R9 registeri sadece dijital audio arayüzünü aktif veya pasif yapmak için kullanılırken R10 registeri tüm registerları resetlemek için kullanılır.

2.3.2.2. Wolfson Wm8731 Dijital Audio Arayüz

'0000111 hex' adresindeki R7 registeri data akış formatı olarak tasarımcıya 4 adet audio arayüz mod imkanı sunmaktadır. Bunlar;

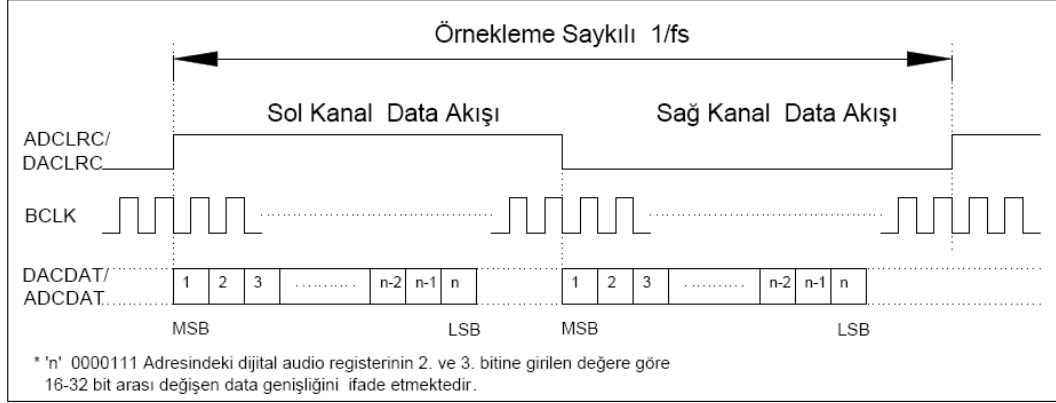
- * Sağa dayalı (Right justified) mod,
- * Sola dayalı (Left justified) mod,
- * I2S mod,
- * DSP (Dijital Sinyal İşleme) mod.

Bu dört motta da en kıymetli bit (MSB) ilk gelen bittir ve data genişliği 16-32 bit aralığında değişebilmektedir.

'0000111 hex' adresindeki dijital audio arayüz bilgiyi dahili analog dijital dönüştürücünün dijital filtresinden alır ve aldığı bilgiyi ADCDAT çıkışına gönderir. ADCDAT çıkışında sol ve sağ kanal bilgileri beraber olarak gelir. ADCLRC palsi ADCDAT üzerindeki sol ve sağ kanal bilgilerinin birbirinden ayrılması için kullanılır. ADCDAT ve ADCLRC sinyalleri BCLK ile senkron olarak hareket ederler ve BCLK'nın her birden sıfıra geçişinde ADCDAT üzerinden bir bit bilgi alınır. BCLK'nın giriş veya çıkış olması aygıtın master veya slave olarak çalışmasına bağlı olarak değişir.

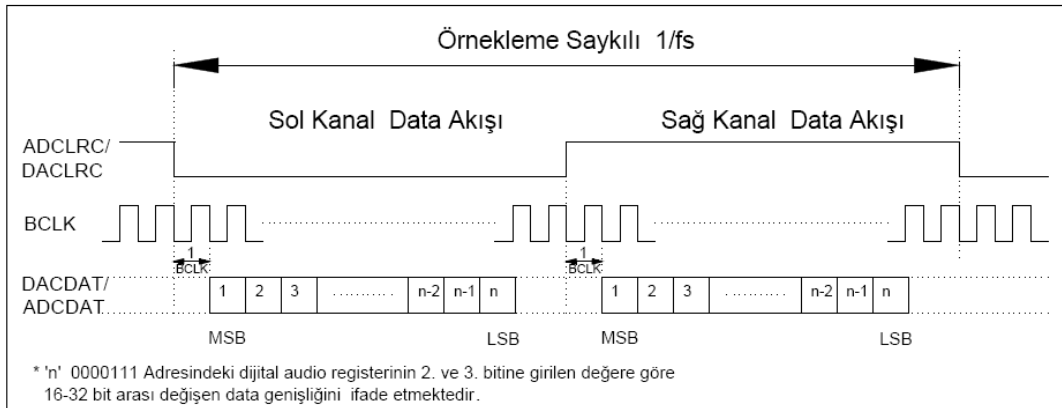
Dijital audio arayüz dahili dijital analog dönüştürücü için bilgiyi DACDAT girişinden alır. DACDAT girişinde de sağ ve sol kanal bilgileri beraber gelir. DACLRC palsi DACDAT üzerindeki sol ve sağ kanal bilgilerinin birbirinden ayrılması için kullanılır. DACDAT ve DACLRC sinyalleri BCLK ile senkron olarak hareket ederler ve BCLK'nın her birden sıfıra geçişinde DACDAT 'a bir bit bilgi gönderilir. DACDAT daima giriş olarak tanımlanırken, BCLK ve DACLRC aygıtın

master veya slave olmasına göre giriş veya çıkış olabilir. Sola dayalı mod seçildiğinde bilgi akışı şekil 2.15’de gösterildiği gibi gerçekleşir.



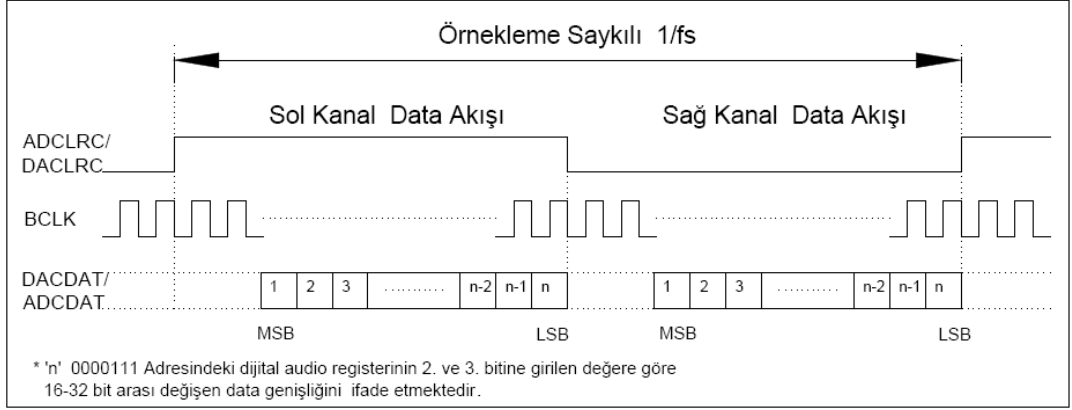
Şekil 2.15. Sola Dayalı Mod Data Akışı.

Dijital audio arayüzünden I2S mod seçildiğinde bilgi akışı şekil 2.16’de gösterildiği gibi gerçekleşir. Dikkat edilirse data akışı ADCLRC / DACLRC sinyali durum değiştirdikten sonra hemen başlamayıp ikinci BCLK pulsü ile başlamaktadır. Yine bu registerdan set edilen data genişliği kadar bit geçince data akışı ADCLRC / DACLRC değerleri değişene kadar kesilmektedir.



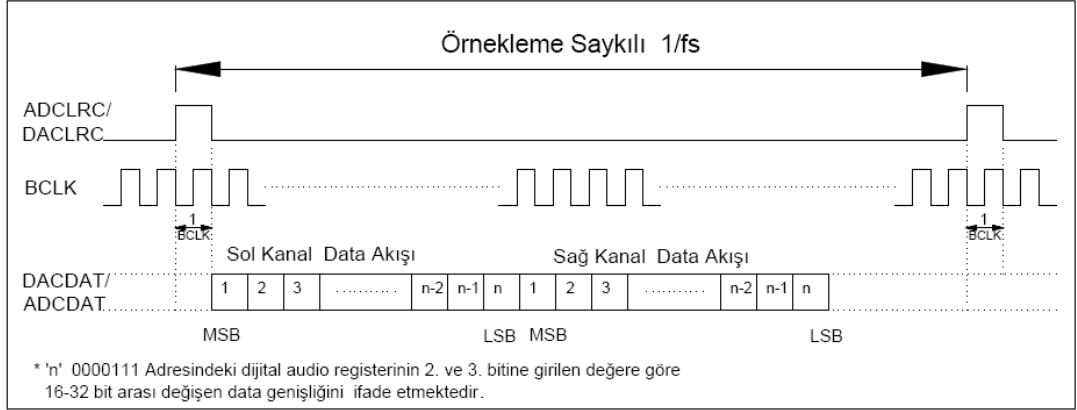
Şekil 2.16. I2S Mod Data Akışı.

Dijital audio arayüzünden Sağa dayalı mod seçildiğinde bilgi akışı şekil 2.17’de gösterildiği gibi gerçekleşir



Şekil 2.17. Sağa Dayalı Mod Data Akışı.

DSP mod seçildiğinde bilgi akışı şekil 2.18’da gösterildiği gibi gerçekleşir



Şekil 2.18. DSP Mod Data Akışı.

Tüm modlarda ADCLRC / DACLRC sinyali BCLK sinyalinin düşen kenarında değişmektedir.

2.3.2.3. Wolfson Wm8731 Kontrol Arayüzü

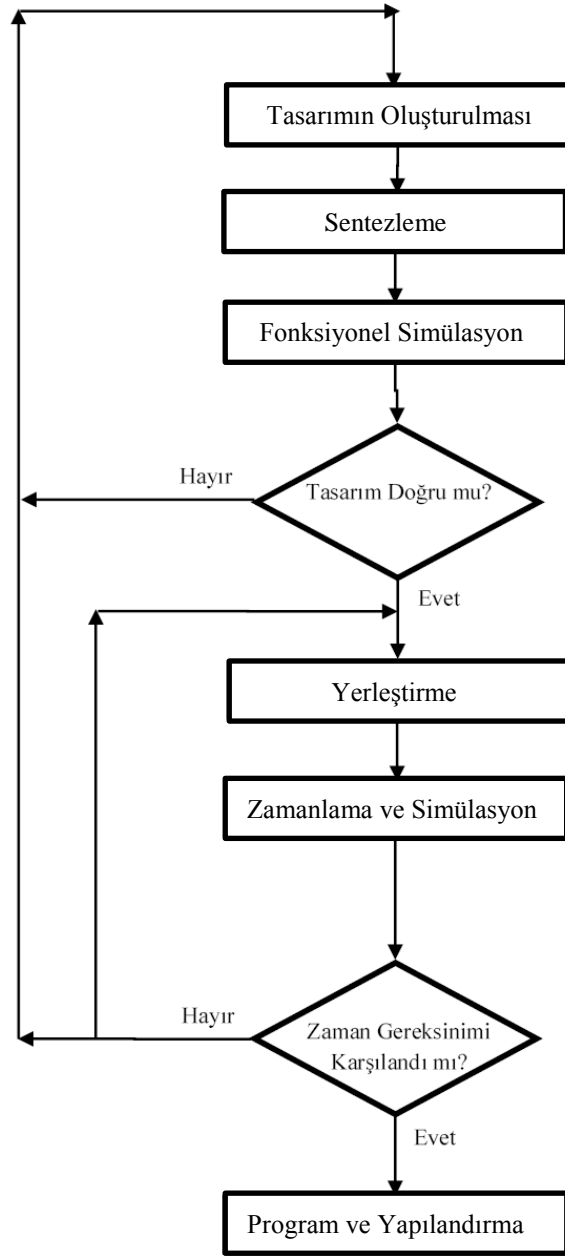
Yazılım kontrol arayüzü seri haberleşme protokollerinden hem SPI hemde I2C protokolünü desteklemektedir. SPI protokolü daha hızlı çalışabilmesine karşın herbir slave aygıt için yeni bir iletken bağlantı gerektirmektedir. I2C protokolü ise daha yavaş olmasına karşın slave cihaz adetine bakılmaksızın iki kablo ile tüm aygıtlarla haberleşebilmektedir. Wm8731 entegresinin CSB pini mod seçimi için kullanılır. Eğer CSB sıfır ise I2C seçilmiş olurken CSB bir ise SPI mod seçilmiş olur. Bu entegrede varsayılan mod I2C moddur. I2C protokolü ile ilgili geniş bilgi üçüncü bölümde verilecektir.

2.4. QUARTUS II CAD KULLANIM PROGRAMI

Bu kısımda Altera firması tarafından geliştirilen QUARTUS II programı hakkında bilgi verilecektir. Bu bölümde kullanılan ekran görüntüleri ve şekiller Altera firması tarafından geliştirilen QUARTUS II programı sürüm 12.0' a ait olup kullanıcı ayarlarından kaynaklı ufak değişiklikler olabilir[32-33]. Vector Waveform editörüdürü hakkında bilgi verilirken Quartus 9.0 versiyonu kullanılmıştır.

2.4.1. FPGA CAD Akış Diyagramı

Bilgisayar destekli lojik devre tasarım programları; lojik elemanların ve FPGA lerin tasarım ve geliştirilmesini kolaylaştıran yazılımlardır. Tipik bir FPGA CAD programının akış şeması şekil 2.19'de verilmiştir[29].



Şekil 2.19. Tipik FPGA Programlama CAD Akış Diyagramı[33].

Tasarımın Oluşturulması; İstenen devre donanım tanımlama yöntemlerinden her hangi biri kullanılarak oluşturulur. Bu bölümde şematik yöntem ile donanım tanımlaması yapılmıştır. Bu yöntem ile elektronik bileşenlerin davranışsal ve yapısal tanımlaması kolayca yapılarak esnek ve hızlı tasarım yapılabilir.

Sentezleme; Devre için gerekli olan mantıksal öğeler ve bu öğeler arasındaki bağlantılar oluşturulmaktadır.

Fonksiyonel Simülasyon; Sentezlenen devre test edilir.

Yerleştirme; Sentezlenen devreyi FPGA içerisine yerleştirir ve gerekli bağlantıları oluşturur.

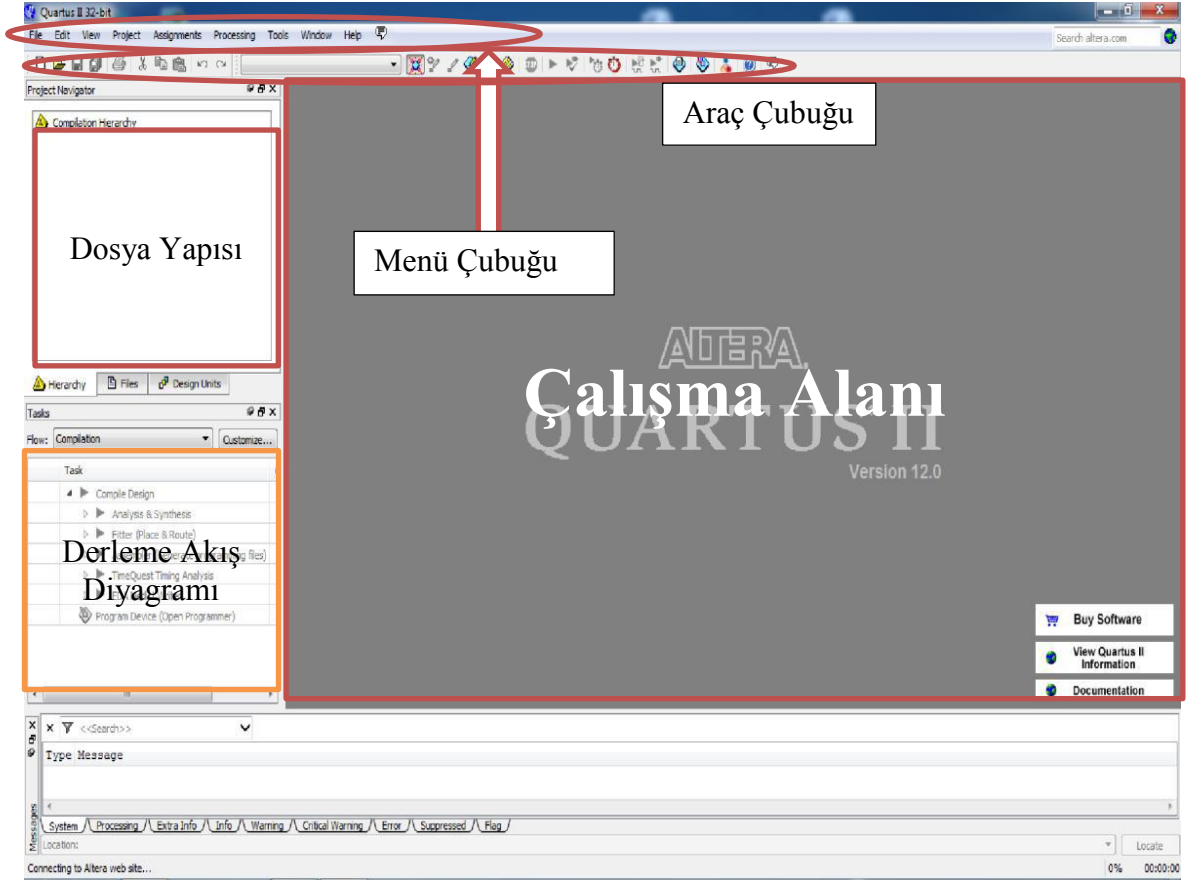
Zamanlama ve Simülasyon; Sentezlenen ve yerleştirilen devre zamansal olarak test edilir.

Program ve Yapılandırma; Usb bağlantısı ile program FPGA içerisine yüklenerek, aygıtın içerisi program doğrultusunda yapılandırılır.

2.4.2. Quartus II Programına Giriş

Quartus II programında tasarlanan lojik devre ya da alt devrelerin her birine proje denir. Aynı anda sadece bir proje üzerinde çalışılabilir ve çalışılan proje ile ilgili tüm bilgiler bir klasörde tutulur. Programda çalışmaya başlarken yapılacak ilk iş bir klasör oluşturmaktır. Burada tasarım dosyalarını saklamak için, örnek1 klasörü kullanılmıştır. Klasörün ismi ve konumu önemli olmadığı için kullanıcı her hangi bir klasör kullanabilir.

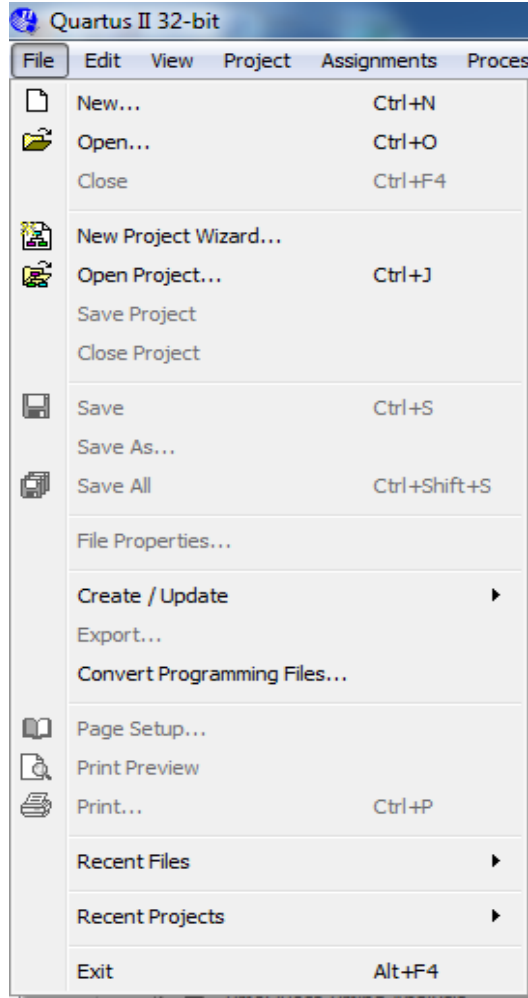
Masaüstünden veya programlar menüsünden Quartus II yazılımı çalıştırıldığında ekrana şekil 2.20' de gösterilen arayüz gelecektir. En üstte projenin isminin yazıldığı başlık çubuğu yer alırken onun altında menü çubuğu ve araç çubukları yer almaktadır. Sayfanın sol kısmında çalışılan projenin dosya yapısı ve alt üniteleri ile derleme işlemi akış diyagramı yer alır. Gri renkli bölüm ise çalışma alanının göstermektedir.



Şekil 2.20. Quartus II Ana Penceresi.

Örneğin, şekil 2.20’de File menüsü farenin sol tuşu ile seçilecek olursa ekrana şekil 2.21’ de gösterilen açılır menü gelecektir. Diğer tüm bilgisayar programlarında olduğu gibi bu menüde dosya açma, kapatma ve kaydetme gibi standart dosya işlemleri yapılabilir.

Tüm menülerde sık kullanılan komutlar ikon şeklinde araç çubuğuna eklenmiştir. Mevcut araç çubuğu listesini görmek için *Tools/ Customize /Customize Toolbar* seçeneği seçilir.



Şekil 2.21. File Menüsünün Görünümü.

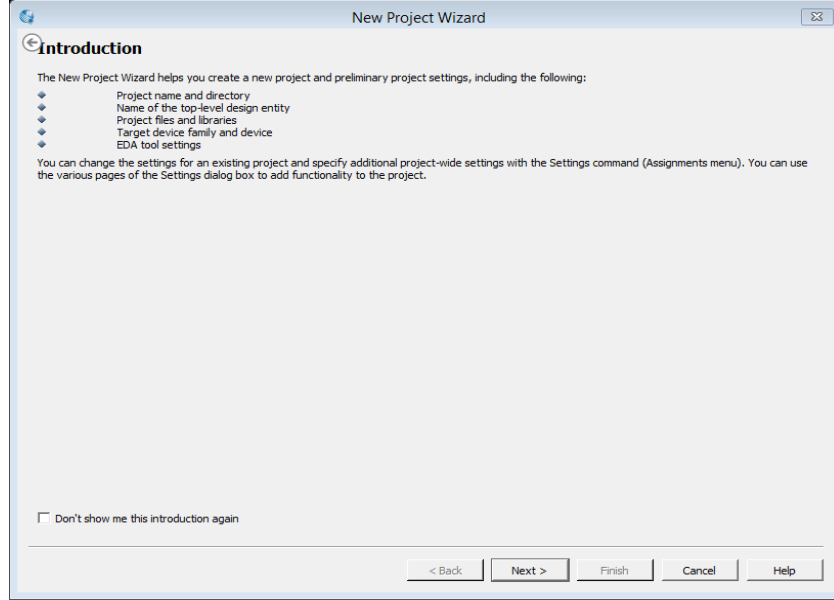
2.4.3. Quartus II Çevrimiçi Yardım/Destek

Quartus II, tasarım sürecinde ortaya çıkabilecek sorunları çözümlen kapsamlı çevrim içi dokümantasyon bulundurmaktadır. Bu dokümantasyona Help penceresindeki menüden de ulaşılabilir.

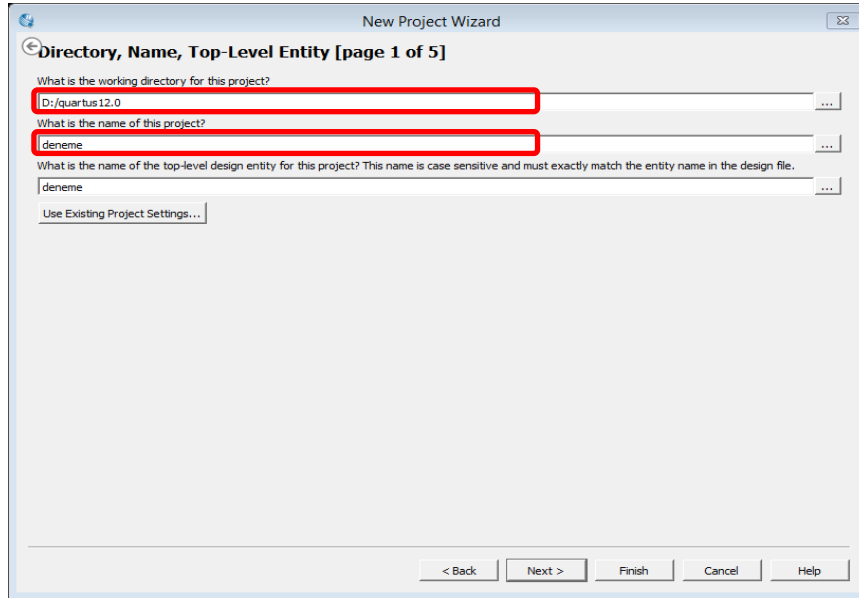
2.4.4. Yeni Bir Proje Oluşturma

Yeni proje oluşturulurken Quartus II sihirbaz desteği, tasarımcının işini kolaylaştırır. Sihirbaz yardımı penceresine ulaşmak için, *File / New Project Wizard* seçeneğini seçilir. İlk olarak karşımıza çıkacak olan şekil 2.22’de gösterin tanıtım (Introduction) penceresini geçmek için *next* seçeneği seçilir. Bir sonraki adımda proje kalsörü ve proje adı belirlenir. Şekil 2.23’de gösterilen pencerede çalışma klasörünü

d:\quartus12.0 ve proje ismini de *deneme* olarak ayarlayalım, Next (İleri) seçeneğini seçelim.



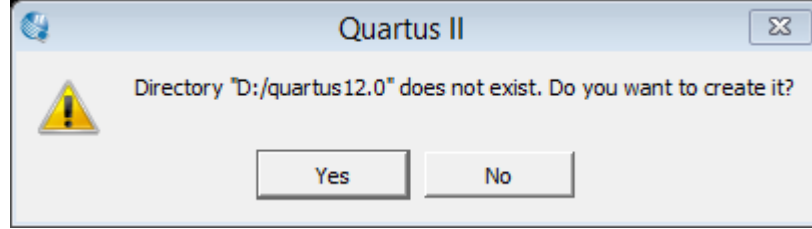
Şekil 2.22. Tanıtım (Introduction) Penceresi.



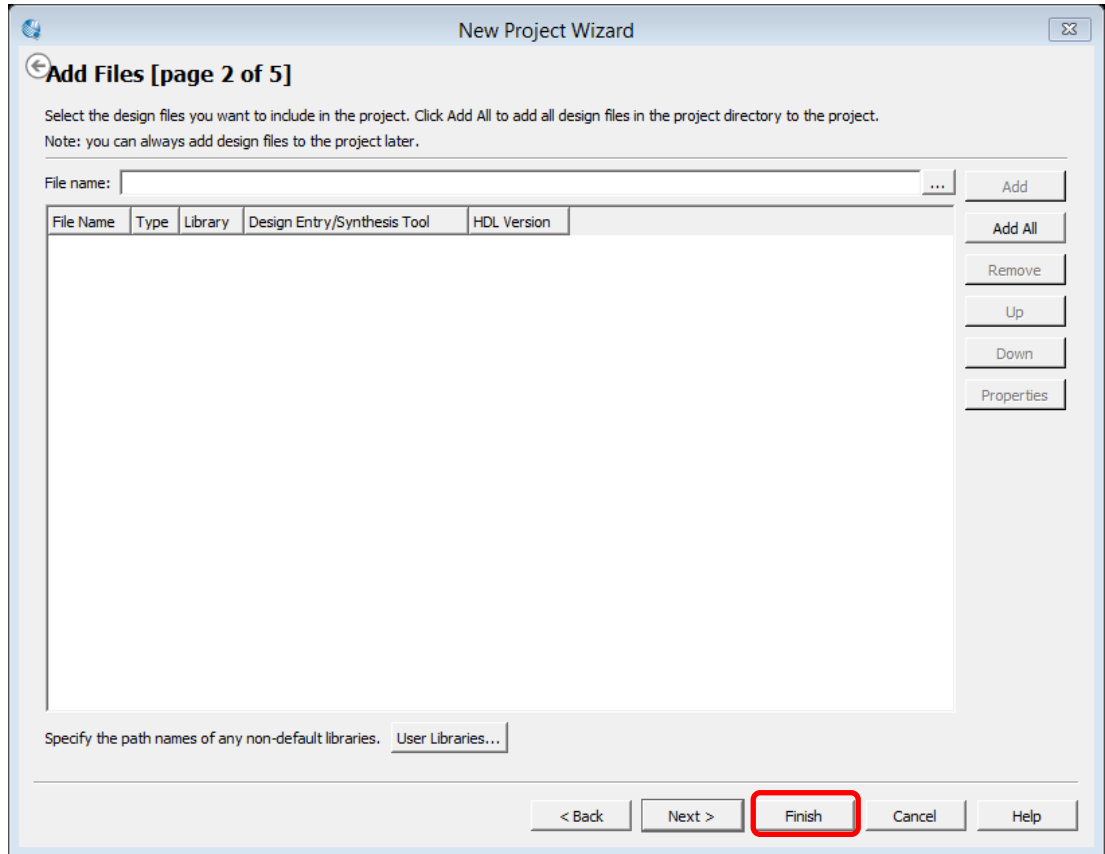
Şekil 2.23. Proje Klasörü ve Proje İsmini Belirleme.

Quartus II, d:\quartus12.0 klasörü henüz oluşturulmadıysa, istenilen dizinin oluşturulup oluşturulmadığını Şekil 2.24'de görüntülenen açılan kutu ile sorar. Yes (Evet) seçeneğini seçilir. Ekranı şekil 2.25'de verilen görüntü gelecektir. Bu

pencerede, tasarımcı (varsa) projede kullanacağı daha önce oluşturulmuş dosyaları ekleyebilir. Daha önce oluşturulmuş dosya yoksa Next (İleri) seçeneğini seçilir.



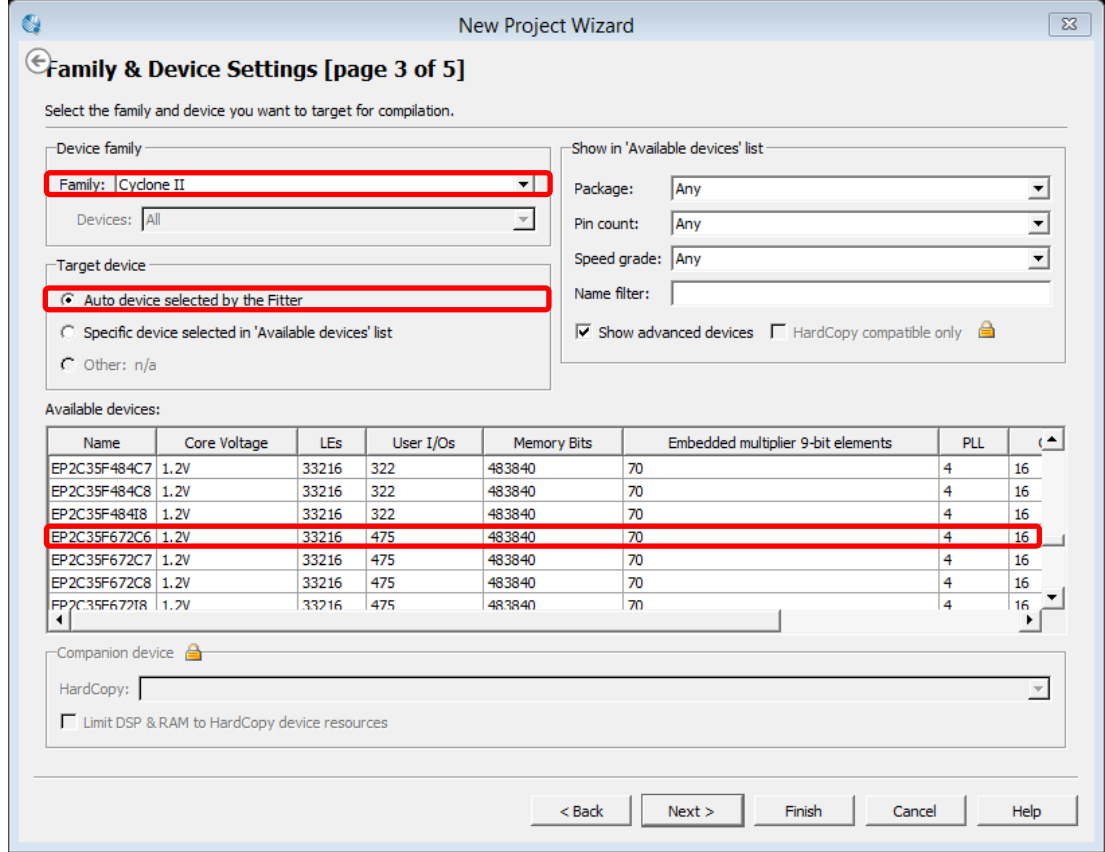
Şekil 2.24. Quartus II İstenilen Klasörün Oluşturulması.



Şekil 2.25. Tasarım Dosyalarının Eklendiği Pencere.

Şekil 2.26’de, tasarlanan devrenin uygulanacağı aygıtın seçildiği pencere görülmektedir. Tasarımcı yaptığı programı yapılandırılabilir donanıma yüklenmeyecekse aygıtın seçimi önemli değildir. Eğer tasarımcı programı derledikten sonra yapılandırılabilir donanıma yükleyecekse uygun aygıtı seçmelidir. Burada Altera DE2-70 board(kart) üzerinde kullanılan FPGA türlerinden Cyclone II aygıt ailesini seçelim. Available Devices (Mevcut Araçlar) listesinden

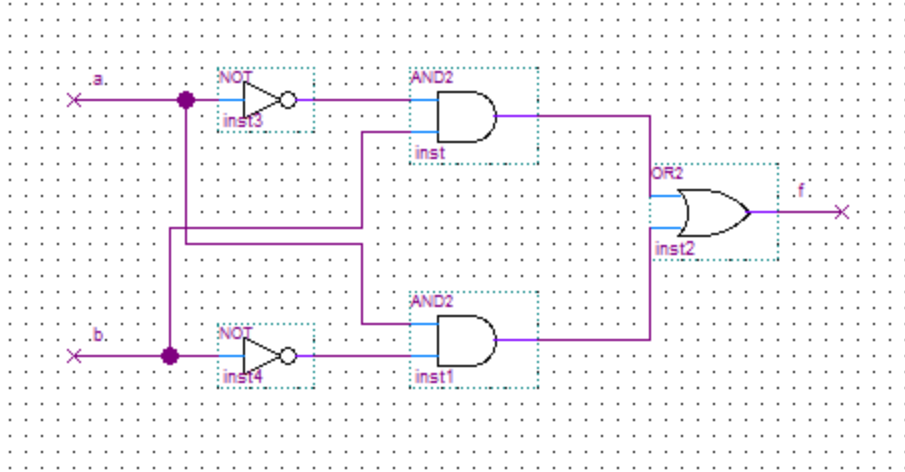
EP2C35F672C6 isimli aygıt seçilmiştir. Şekil 2.20’de gösterilen Quartus II ekranına dönmek için ‘Finish (Bitir) butonu seçilir.



Şekil 2.26. Aygıt Ailesinin/Türünün Belirlenmesi.

2.4.5. Şematik Tasarıma Giriş

Burada şematik proje oluşturulması hakkında örnek bir devre üzerinde açıklama yapılacaktır. Örneğin $f=a'b+ab'$ lojiksel fonksiyonu için lojik devreyi çizelim. 'f' için devre şeması şekil 2.27’de türetilmiştir. 'f' için doğruluk tablosu, tablo 2.9’ da gösterildiği gibidir.



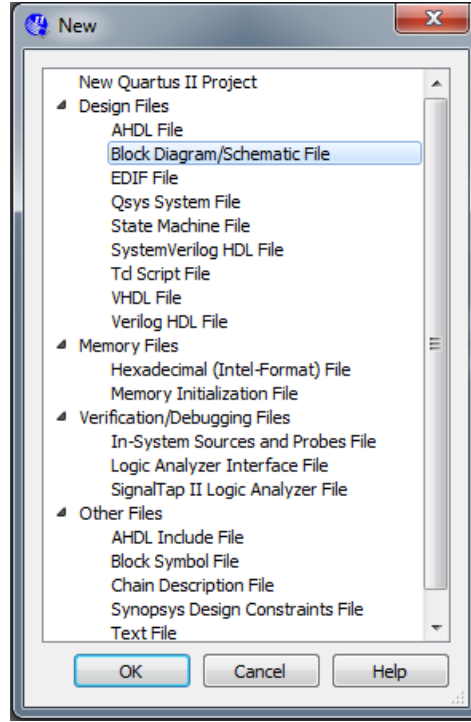
Şekil 2.27. Lojik Fonksiyon.

Tablo 2.9. Lojik Fonksiyonun Doğruluk Tablosu.

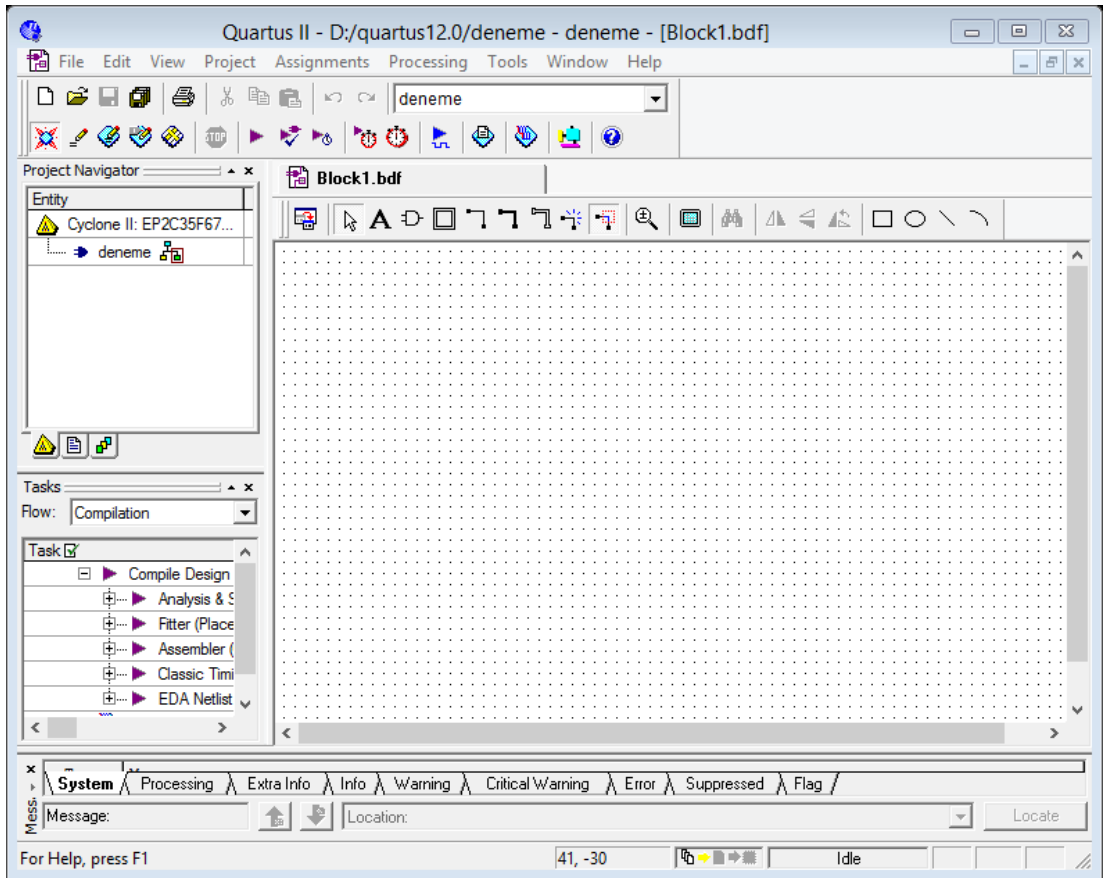
a	b	f
0	0	0
0	1	1
1	0	1
1	1	0

2.4.5.1. Blok Düzenleyici Kullanımı

Menü çubuğundan *File/New* seçeneğini seçelim. Şekil 2.28’ da gösterildiği gibi bir pencere açılacaktır. *Block diagram/Schematic File* seçeneği seçildiğinde ekrana şekil 2.29’da gösterilen blok editör düzenleyici gelecektir.



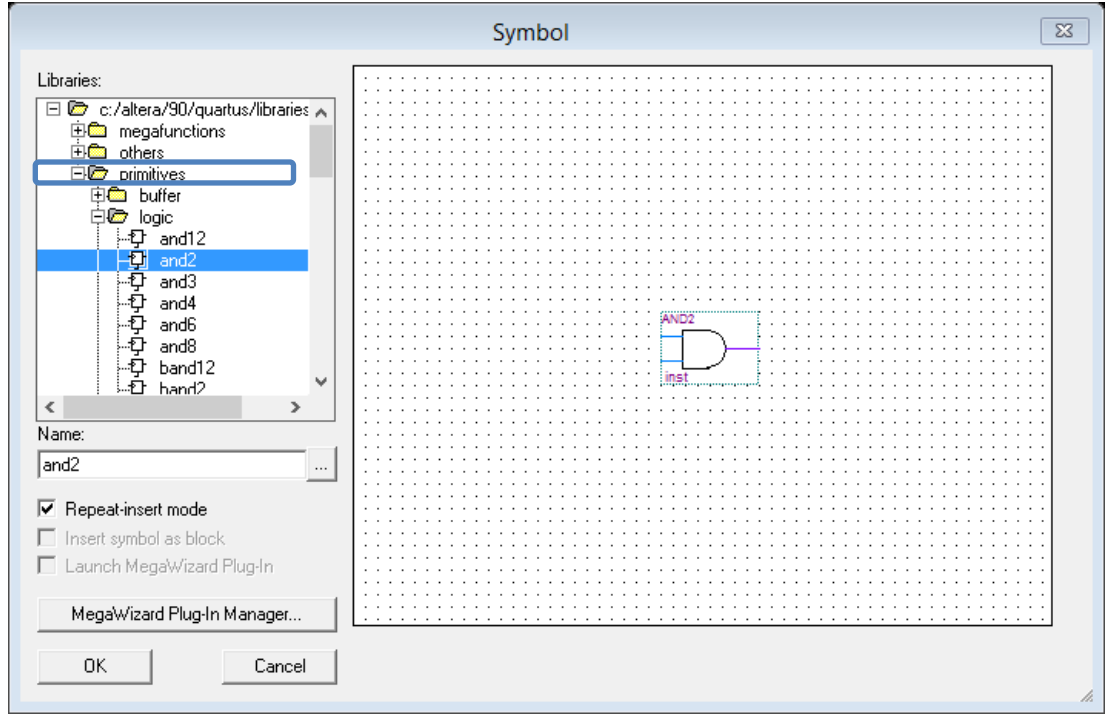
Şekil 2.28. Tasarım Dosyasının Türünün Seçilmesi.



Şekil 2.29. Blok Editör Düzenleyici.

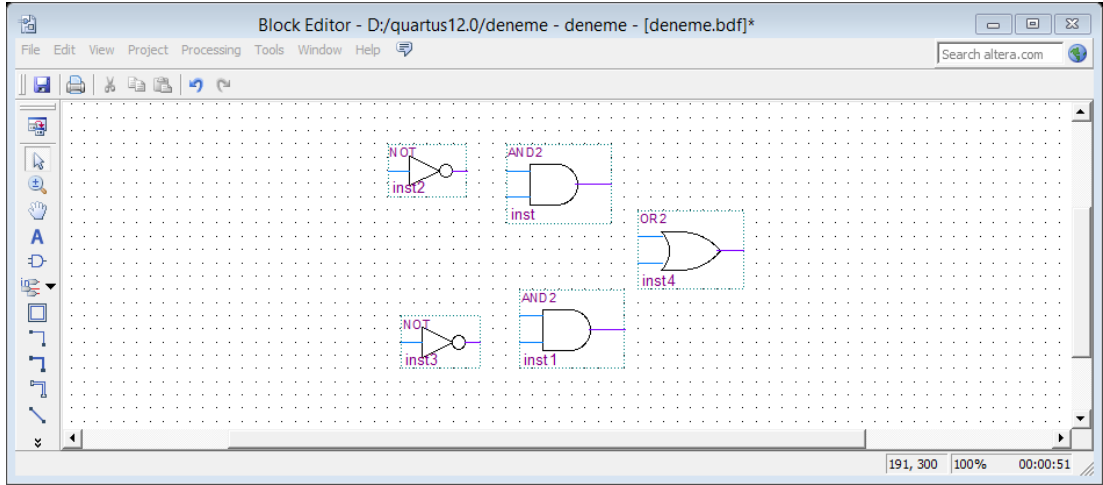
2.4.5.2. Lojik Elemanların Kütüphaneden Çağırılması

Quartus II programında lojik elemanlara primitives kütüphanesini kullanarak erişebiliriz.. Bu kütüphaneye block editörü penceresi içinde Şekil 2.30’de gösterilen pencerenin açılması için boş alanda çift tıklanarak erişilebilir (Diğer yöntem ise araç çubuğu üzerinde AND kapı sembolü seçilerek bu pencerenin açılmasıdır).



Şekil 2.30. Lojik Sembollerin Seçilmesi.

Şekilde görüldüğü gibi, sırası ile *primitives* ve sonra *logic* klasörüne çift tıklayarak lojik elemanların bulunduğu klasöre erişebiliriz. Buradan sırası ile *and2*, *or2* ve *not* lojik elemanlarını seçerek çalışma alanına şekil 2.31’de gösterildiği gibi yerleştirelim.



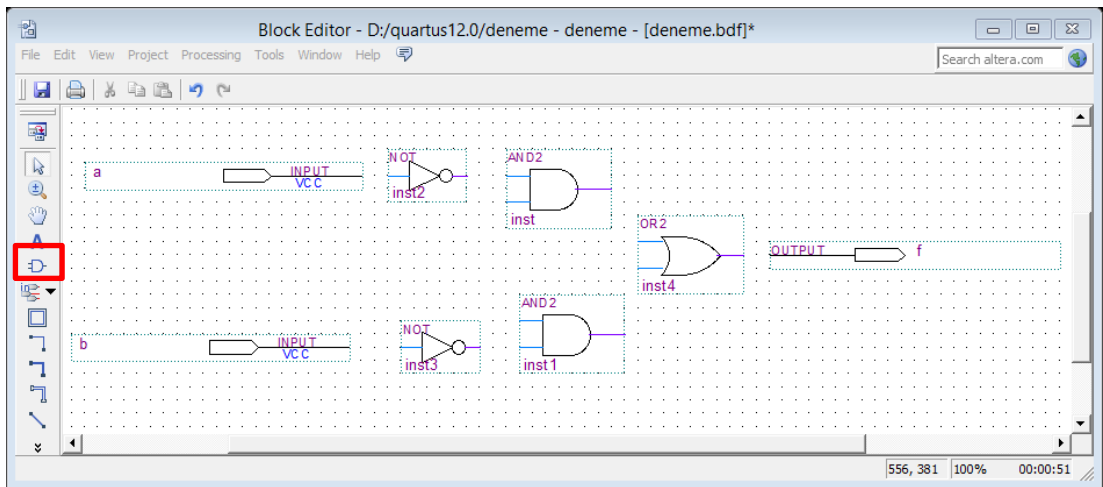
Şekil 2.31. Eklenen Kapı Sembolleri.

2.4.5.3. Giriş - Çıkış Sembolleri Ekleme

Lojik kapı sembollerinin kaydedilmesi için, devrenin giriş çıkış portlarında giriş – çıkış sembollerinin eklenmesi gereklidir. Bu pinler primitives kütüphanesinden eklenebileceği gibi şekil 2.32’de dikdörtgen içine alınan ikon seçilerek de eklenebilir.


2.4.5.4. Giriş - Çıkış Sembollerini İsimlendirme

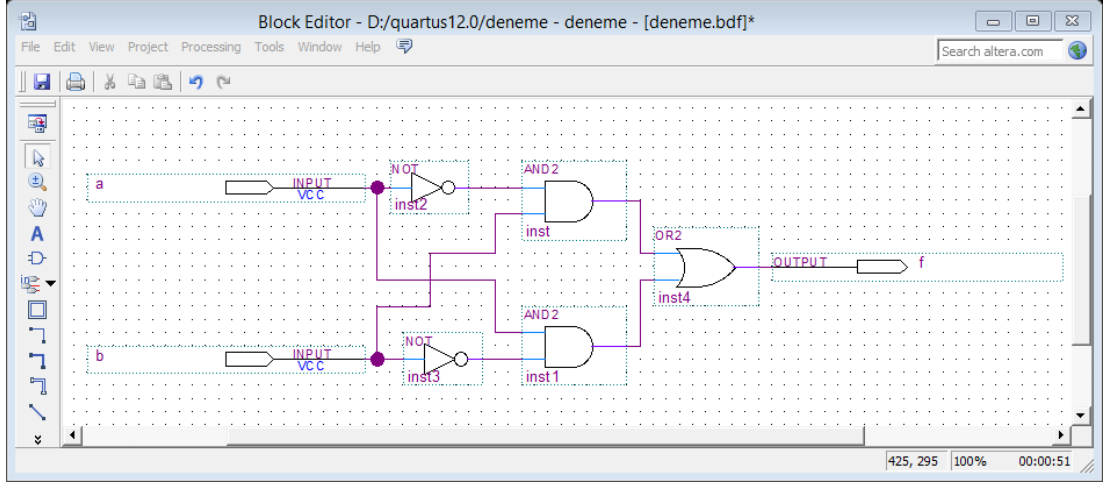
Pin sembolü üzerinde sol üst köşesinde gösterilen pin adı seçilerek fare yardımıyla iki sefer tıklayalım. Yeni pin ismi girilerek enter tuşuna basılır. Giriş pinlerini sırasıyla ‘a’ ve ‘b’ olarak isimlendirelim. Son olarak da, çıkış pinini ‘f’ olarak isimlendirelim.



Şekil 2.32. Kapılar ve Pinlerin Düzenlenmesi.

2.4.5.5. Hat Ve Düğüm Bağlantısı

Sonraki adım, devrede sembolleri birbirine bağlamak için hatlar çizilecektir. Araç kutusundan  ikonuna tıklanarak şekil 2.33’de gösterildiği gibi gerekli bağlantılar oluşturulur.




Şekil 2.33. Tamamlanmış Devre.

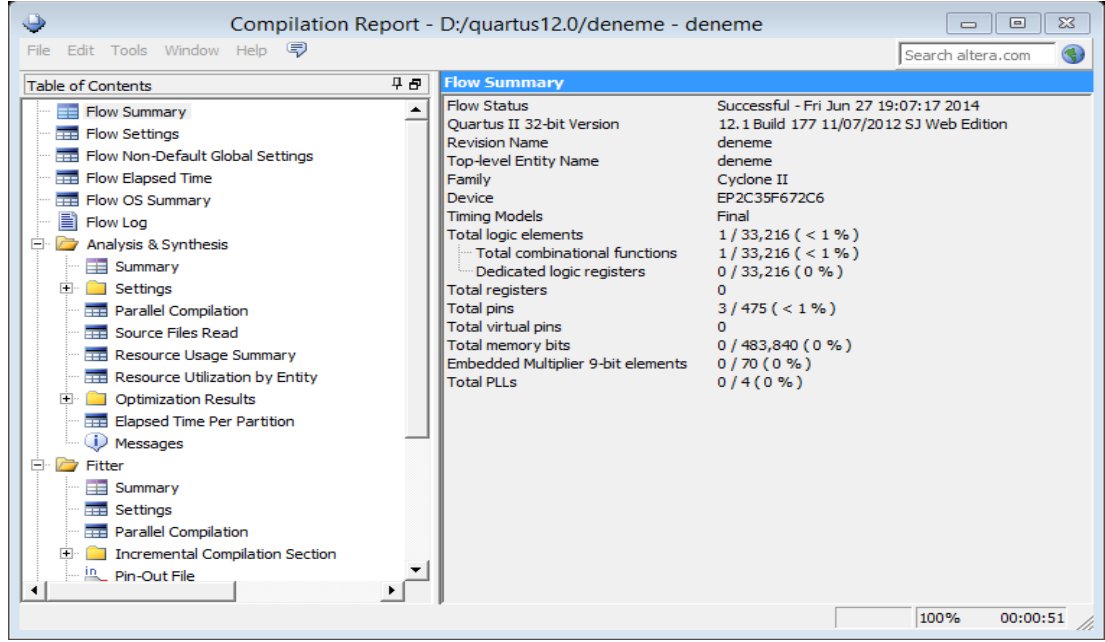
Ekranda görüntülenen devre bölümünü artırmak veya azaltmak için, araç çubuğunda büyütece benzeyen simgeyi kullanın. *File/Save As* seçimini kullanarak devreyi *deneme* ismi ile kaydedin. Dosyanın *deneme.bdf* ismi ile kaydedildiğine dikkat ediniz.

Örnek devremiz oldukça sade olmakla birlikte, karmaşık devreler çizmeye gerek kalmadan devredeki bütün hatları çizmek de kolaydır. Bununla birlikte, daha büyük devrelerde bağlantılı olması gereken bazı nodlar(düğüm) uzak düştüğü durumlarda bu nod(düğüm)lar arasına hat çizmek kullanışlı değildir. Buna benzer durumlarda, hat çizimi yerine nod(düğüm) etiketleme yolu ile bağlanır. Daha ayrıntılı açıklama için Help bölümü incelenmelidir.

2.4.6. Derleyici Kullanımı

Yapılan tasarım yapılandırılabilir donanıma yüklenmeden önce derlenmeli ve program aygıtın anlayabileceği netlistlere dönüştürülmelidir. Quartus II modülleri Compiler(Derleyici) olarak isimlendirilen uygulama programı tarafından kontrol edilir. Derleyici bir seferde bir modül çalıştırmak için kullanılabilir, veya sırayla birden fazla modül çalıştırılabilir. Quartus II kullanıcı arayüzünde derleyiciye erişim

için birden fazla yöntem bulunmaktadır. Derleyiciye erişmenin yaygın yöntemlerinden biri, Processing / Start seçimini kullanmaktır. Diğeri ise araç çubuğunda da mor üçgene benzer  ikonu tıklamaktır. Derleme işlemi bittiğinde şekil 2.34’ de gösterilen derleme raporu oluşturulur. Rapor özeti Cyclone II FPGA içinde küçük tasarımımızın sadece üç pin ve bir lojik eleman kullandığını gösterir



Şekil 2.34. Derleme Raporu Özeti.

2.4.7. Hatalar

Quartus II derleme sürecinde mesajları, mesaj penceresinde görüntüler. Bu pencere Şekil 2.30’da gösterildi gibi Quartus II ekranının alt kısmındadır. Şematik tasarım doğru olarak yapılırsa, mesaj olarak ‘Hata bulunamadı ve derleme süreci başarıyla sonuçlandı’ ifadesiyle karşılaşılabilmektedir[5].

Hata yapıldığında ne olacağını görmek için, ‘b’ girişini alttaki AND kapısında bağlayan hattı kaldırın ve değiştirilmiş şematiği derleyin. Bu durumda, derleme başarılı değildir ve iki hata mesajı görüntülenir. İlk olarak ‘Söz konusu AND kapısının kaynağının bulunamadığı’ tasarımcıya iletilir. İkinci olarak ‘Hata bulundu’ uyarısı tasarımcıya iletilir. Büyük bir devre söz konusu iken, her hangi bir hatanın konumunu bulmak zor olabilir. Quartus II yazılımı, hata mesajı üzerine iki sefer tıklandığında kullanıcıya yardımcı olur ki ilgili konum(bizim örneğimizde

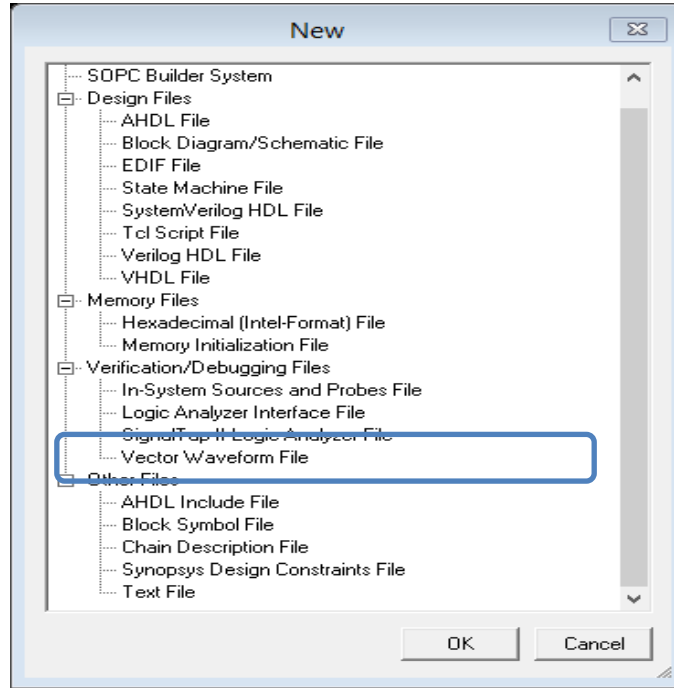
AND kapısı için) vurgulanır. Kaldırılmış hattı yeniden bağlayın ve doğrulanmış devreyi yeniden derleyin[5].

2.4.8. Tasarlana Devrenin Simülasyonu

Quartus II yazılımında tasarlanan devrenin davranışını simüle etmeye yardımcı benzetim araçları bulunmaktadır. Devre simüle edilmeden önce, giriş sinyallerini göstermek için test vectors olarak isimlendirilen dalga şekillerini oluşturmak gereklidir. Test vectors çizmek için Quartus II Waveform Editörünü kullanacağız.

2.4.8.1. Waveform Editörü Kullanımı

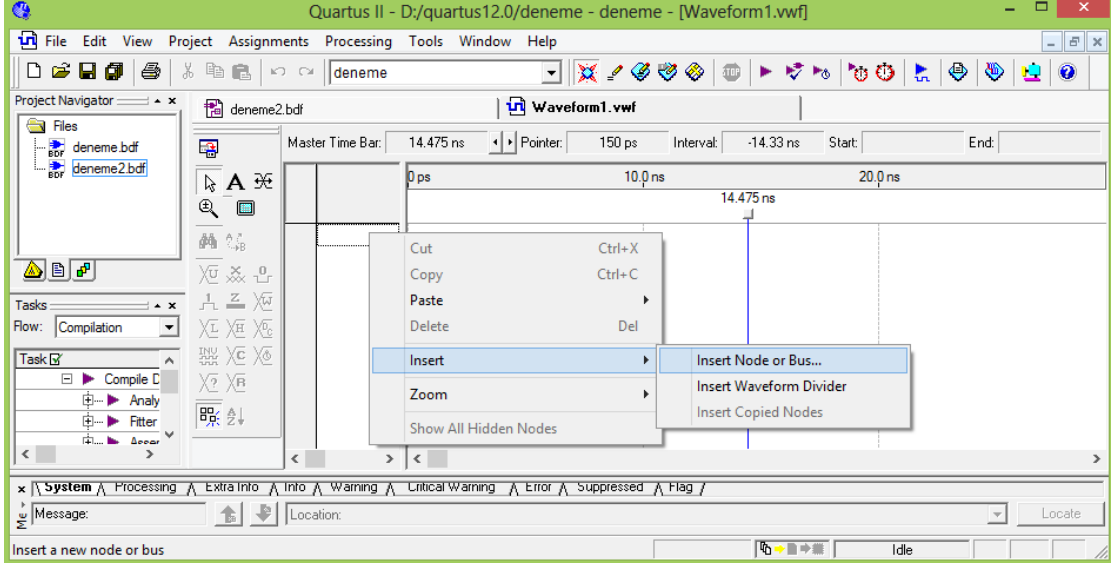
File / New seçimi ile şekil 2.35’de gösterilen pencereden Vector Waveform File seçin, OK düğmesi ile onaylayın.



Şekil 2.35. Waveform Editörün Seçilmesi.

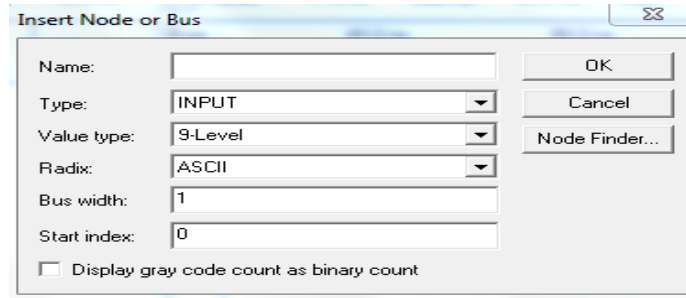
Waveform Editör Penceresi şekil 2.35’ da gösterildiği gibidir. Simüle edilecek devrenin giriş ve çıkış nod(düğüm)ları kapsama dâhil etmek istiyoruz. Bu işlem, Node Finder uygulaması kullanılarak yapılır. Node Finder uygulamasını çalıştırmak için şekil 2.36’da gösterildiği gibi çalışma alanında farenin sağ tuşuna tıklayarak açılan menüden ‘insert’ komutunu oradan da ‘Insert Node or Bus’ komutu seçilebilir

veya *Edit / Insert Node or Bus* seçimi uygulanmalıdır. Şekil 2.37’de gösterilen pencere açılacaktır.



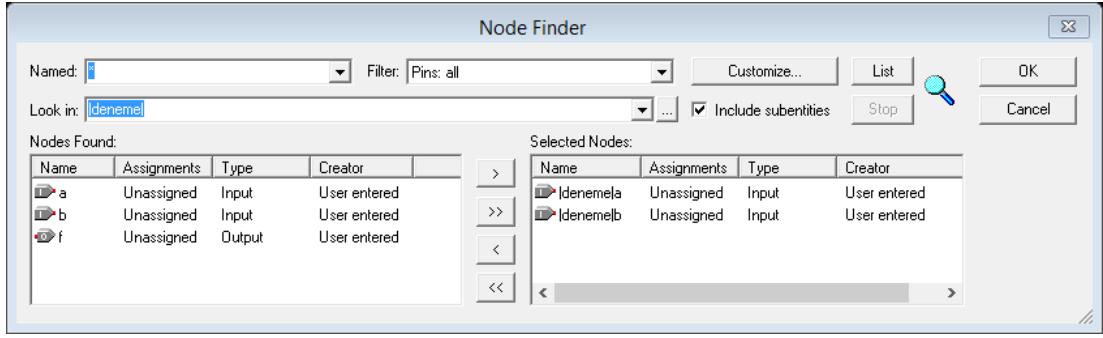
Şekil 2.36. Insert Node Or Bus Komutunun Çalıştırılması.

Şekil 2.37’ de *Insert Node Or Bus* penceresi görülmektedir. Ekleme istediğimiz düğüm noktalarını tek tek isimini ve tipini belirterek ekleyebileceğimiz gibi sağ kısımdaki ‘Node Finder’ ikonuna tıklayarak daha pratik bir şekilde ekleyebiliriz.



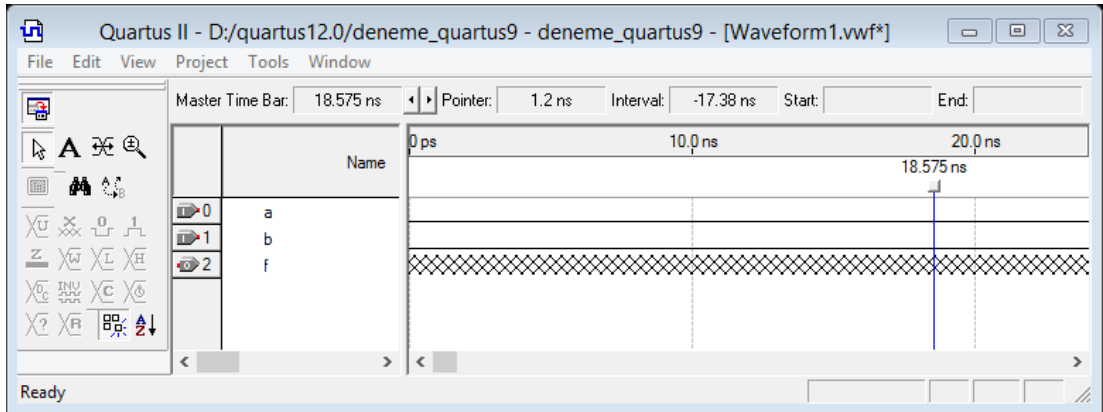
Şekil 2.37. Insert Node Or Bus İletişim Penceresi.

Şekil 2.39’ da Node Finder penceresi görülmektedir. Node Finder uygulamasında ne çeşit nod(düğüm)ların bulunduğunu göstermek için kullanılan bir filtre bulunmaktadır. Giriş ve çıkış pinleriyle ilgilendiğimiz için, filtreyi Pins :all durumuna göre ayarlayın. Giriş ve çıkış nod(düğüm)larını bulmak için List düğmesi seçilmelidir.



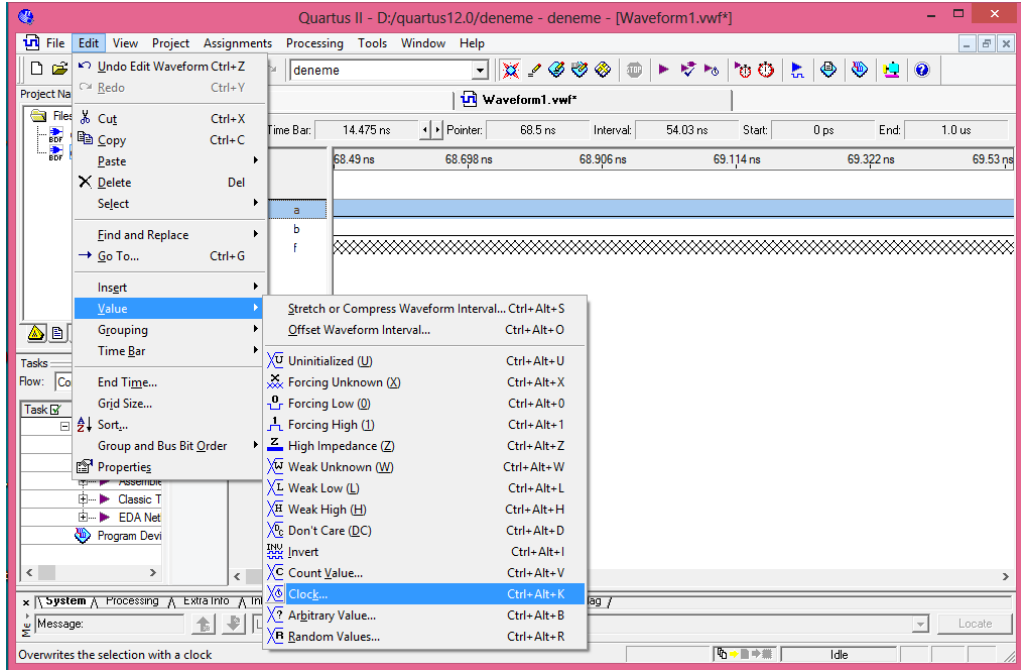
Şekil 2.38. Node Finder Penceresi.

Node Finder, pencerenin sol tarafında ‘f’, ‘a’ ve ‘b’ nod(düğüm)larını görüntüler. ‘a’ nod(düğüm)unu seçin ve ‘>’ işareti ile şeklin sağ tarafındaki Selected Nodes bölümüne ekleyin. Aynı işlemi ‘f’ ve ‘b’ için de yapın. Node Finder penceresini kapatmak için OK düğmesini onaylayın, ve sonra ekrana gelecek olan şekil 2.38’de gösterilen pencerede OK düğmesini onaylayın. Böylece, Şekil 2.39’da gösterildiği gibi simülasyon için gerekli düğüm noktaları görüntülenir.



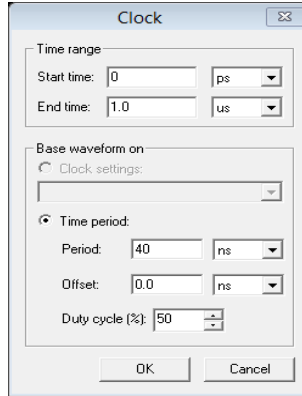
Şekil 2.39. Simülasyon İçin Gerekli Nod(düğüm)lar.

Simülasyon süresince giriş sinyalleri için kullanılacak lojik değerler tasarımcı tarafından belirlenirken ‘f’ çıkışındaki lojik değerler ise simülatör tarafından otomatik olarak üretilir. Giriş pinlerine değer vermek için şekil 2.40’de gösterildiği gibi değer verilecek pinin fare ile seçiminden sonra *Edit / Value / Clock* yolu ile şekil 2.41’deki clock penceresi açılır.

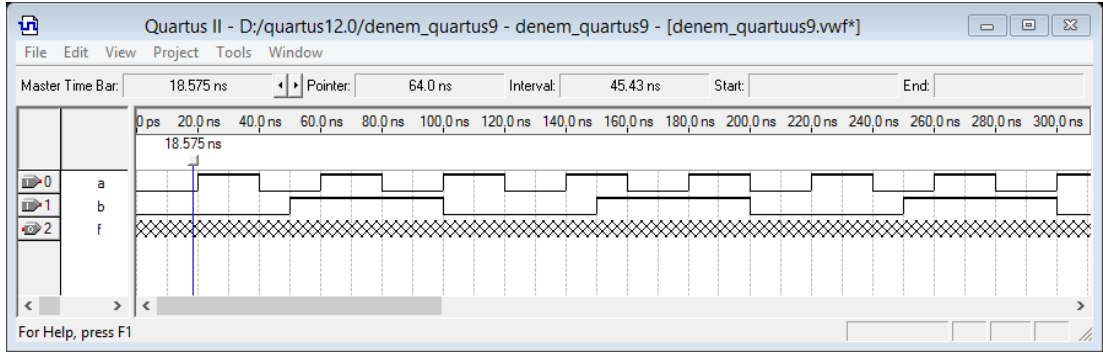


Şekil 2.40. Giriş Sinyallerine Değer Atanması.

Clock pencereden sırası ile 'a' giriş periyodunu 40 ns ve 'b' giriş periyodunu 100 ns ayarlarsak şekil 2.42' yi elde ederiz. Kullanıcı isterse sol kısımda bulunan Waveform editör penceresi ikonlarını kullanarak giriş sinyallerine 0,1,Z gibi değerleride istediği zaman aralığında verebilir.



Şekil 2.41. Clock Penceresi

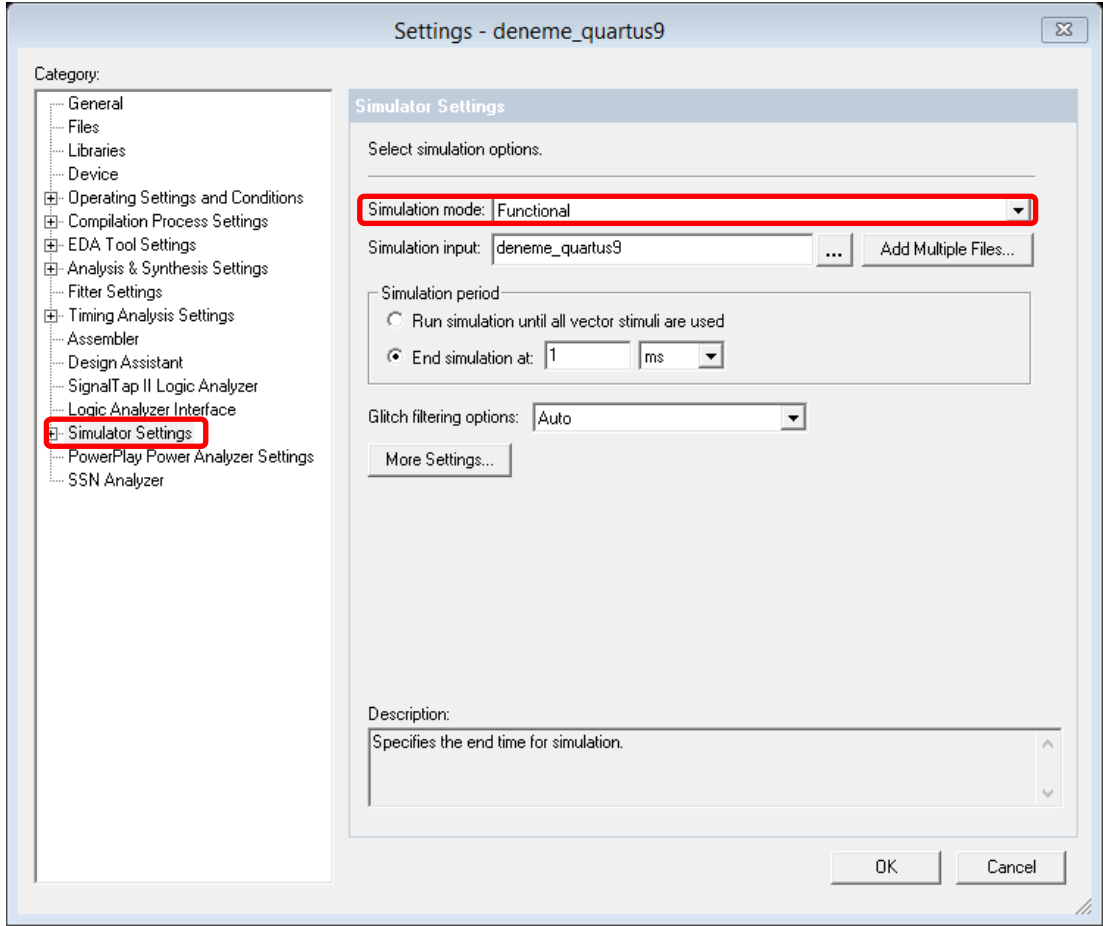


Şekil 2.42. Giriş Sinyallerine Değer Atanmış Şekli.

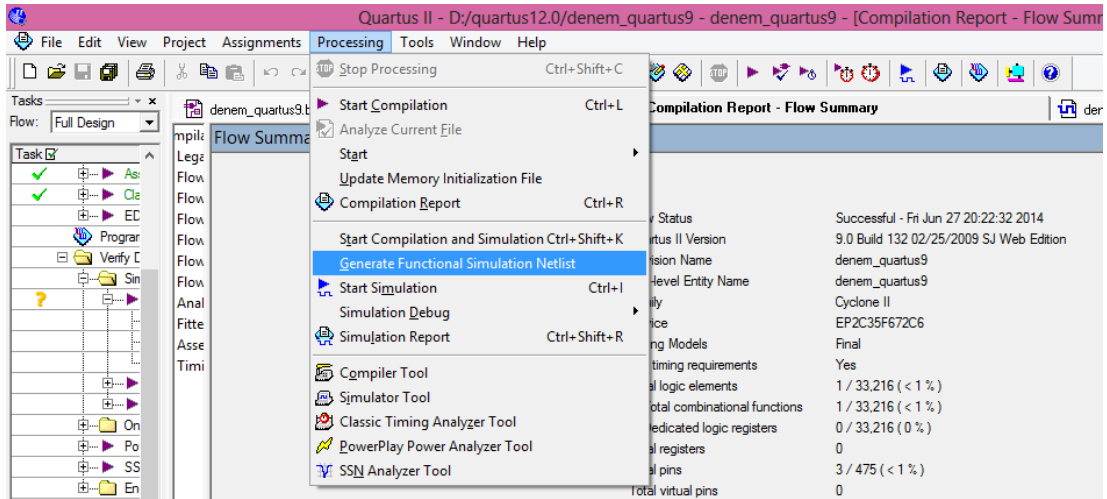
2.4.8.2. Simülasyon Ayarları

Bir devre iki şekilde simule edilebilir. En kolay yöntem, lojik elemanların ve ara bağlantı hatlarının mükemmel olduğunu, böylece devre içinde sinyallerin yayılmasında hiçbir gecikme olmadan yayıldığını varsaymaktır. Bu durum fonksiyonel simülasyon olarak isimlendirilir. Daha karmaşık bir alternatif ise, zamanlama simülasyonu ile sonuçlanan tüm yayılım gecikmelerini hesaba katmaktır. Genellikle, fonksiyonel simülasyon tasarlanan bir devrenin fonksiyonel doğruluğunu doğrulamak için kullanılır. Bu, daha az zaman alır çünkü simülasyon devreyi tanımlayan lojik ifadeler kullanılarak kolay bir şekilde yürütülebilir. Bu dokümanda yalnız fonksiyonel simülasyon kullanılacaktır.


Fonksiyonel simülasyonu yürütmek için, Settings penceresini açmak için *Assignments / Settings* seçimini uygulayın. Pencerenin sol tarafında, şekil 2.43'deki pencereyi görüntülemek için Simulator seçeneğini ve simülasyon modunu da 'Functional' olarak seçin. Simülatörün çalışmasını tamamlamak için şekil 2.44'de gösterildiği gibi, *Processing / Generate Functional Simulation Netlist* yolu takip edilir.

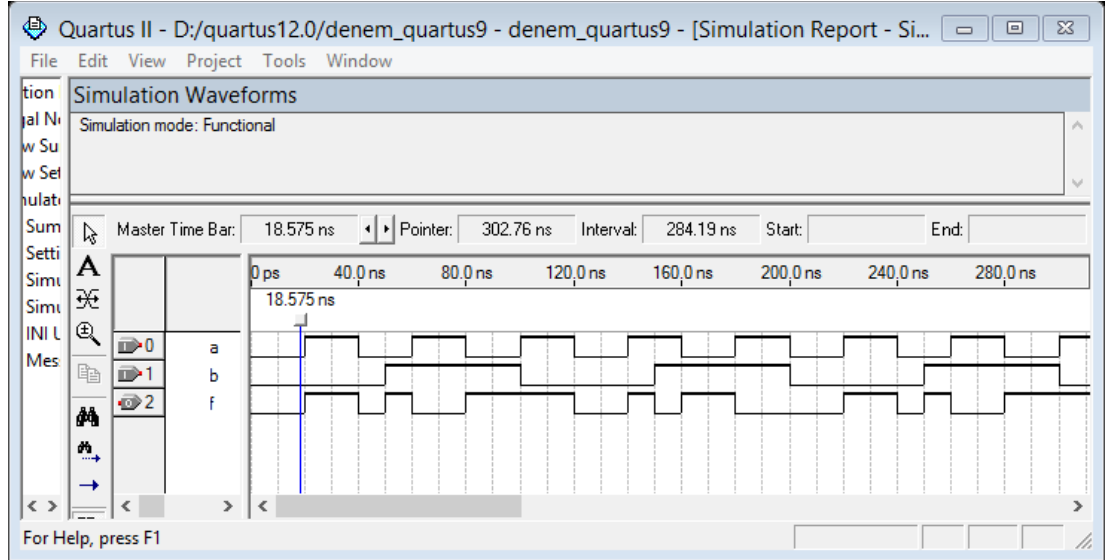


Şekil 2.43. Simülasyon Modunun Belirlenmesi.



Şekil 2.44. Simülasyon Netlistinin Oluşturulması.

Quartus II simülatorü test girişlerini alır ve deneme.vwf dosyasında tanımlı çıkışları üretir. Bir simülasyon çalışması, *Processing / Start Simulation* seçimi ile veya araç çubuğu üzerinde altındaki kare dalga ile beraber mavi üçgene benzeyen kısayol simgesi () kullanılarak başlatılabilir. Similasyon çalıştırıldığında şekil 2.45 elde edilir.



Şekil 2.45. Simülasyon Sonucu.

Simülasyonun bitiminde, Quartus II yazılımı, simülasyonun başarıyla tamamlandığını bildirir ve bir simülasyon raporu görüntüler.

2.5. BZK.SAU.FPGA Mikrobilgisayar Sistemi

Bu bölümde Bozok Üniversitesi (BZK) ve Sakarya Üniversitesi (SAU) tarafından FPGA ortamında ortaklaşa geliştirilen BZK.SAU.FPGA mikrobilgisayar sistemi hakkında kısaca bilgi verilecektir.

BZK.SAU.FPGA mikro bilgisayar sistemi 2009 yılında yüksek lisans tezi kapsamında H. ÖZTEKİN tarafından 'Gömülü Sistemler', 'Bilgisayar Mimarisi' ve 'İşletim Sistemi' gibi derslerin daha iyi anlaşılabilmesi için eğitimsel amaçlı olarak tasarlanan BZK.SAU mikrobilgisayar simülatorünün 2010 yılında FPGA ortamına aktarılması ile gerçekleştirilmiştir[3].

Öğrencilerin sistemin çalışmasına zarar vermeden sisteme müdahil olmasını sağlamak için sisteme modülerlik özelliği kazandırılmıştır. Modülerlik özelliği

sayesinde öğrenciler kendi tasarladıkları yapıyı sisteme monte edebilmekte ve sistemin çalışmasını sanal ortam yerine gerçek dünya şartları altında izleyebilmektedir. Bu yöntem ile öğrencilerin sistemin karmaşasından kurtularak yeni sistemler geliştirme konusunda özgüven kazandıkları yapılan istatistiki çalışmalarla kanıtlanmıştır[3].

BZK.SAU.FPGA mikrobilgisayar sistemine kullanıcıların uzaktan erişim imkanının yanında, klavye, mouse, monitör, flash bellek kontrolü, paralel ve seri port haberleşmesi gibi özellikler kazandırılarak eğitim yönü kuvvetlendirilmiştir. Bu tez çalışması ile de sisteme analog haberleşme modülünün eklenmesi ile BZK.SAU.FPGA mikrobilgisayar sistemi eğitsel açıda daha da güçlendirilmiş ve daha kullanışlı hale getirilmiştir[3,5,34-37].

BZK.SAU.FPGA mikrobilgisayar sisteminde Altera firmasına ait FPGA geliştirme ortamlarından olan DE2 modelinin Cyclone II ailesine ait E2C35F672C6 ile Cyclone III ailesine ait EP3C120F780C7 aygıtları kullanılmıştır. Tasarımın büyük bir kısmı DE2 modeli üzerinde yoğunlaşmış olup Cyclone III modeli sadece 64 KB kapasiteli ana bellek tasarımı için kullanılmıştır[3].

Tasarımı yapılan Mikro Bilgisayar Mimarisinin sahip olduğu özellikler belirlenirken projenin eğitim amaçlı olması nedeniyle temel seviyede olmasına özen gösterilmiştir. BZK.SAU.FPGA Mikrobilgisayar sisteminin özellikleri Tablo 3.1’de verilmiştir.

Tablo 2.10. BZK.SAU.FPGA Mikrobilgisayar Sistemi Özellikleri[3].

Özellik	Açıklama
Sistem Adı	BZK.SAU.FPGA
Sistemin İnşa Edildiği Donanım	FPGA (Altera DE2-70)
Çıkış Birimi	VGA Monitör (640×480)
Ekran Alanı	40 sütun×24 satır(320×384)
Giriş Birimi	PS/2 klavye
Sistem Tanımlama Dili	Şematik(Donanımsal)
İşlemci Mimarisi	Von-Neumann(SISD mimarisi)
İşlemci Tipi	16-bit
Adres Yolu	16-bit
Veri Yolu	16-bit

Özellik	Açıklama
Sistem Kaydedicileri	10 adet(Giriş ve çıkış kaydedicileri 8-bit diğerleri 16-bit genişliğinde)
Ana Bellek	64 KB – 16 bit
İkincil Bellek	Flash Bellek(4 MB) – 8 bit
Bellek Kelime Yerleşim Düzeni	Big-Endian
Komut Mimarisi	CISC
Komut Seti	Fonksiyonel, Kontrol, Transfer, Giriş-Çıkış ve Yığın Komutları(59 komut)
Komut Yapısı	16 bit(15-12. bitler adresleme modu, 11-0. bitler opcode alanı)
Komut İşleme Metodu	None Pipeline
Adresleme Mod Çeşidi	6(İvedi, direkt, dolaylı, indeks, göreceli ve doğal)
Kontrol Birim Yapısı	Donanımsal
ALU Birimi	16-bit(Sadece tamsayılar)
Sayı Sistemi	2'ye Tümleme
İşletim Sistemi	Tek Kullanıcı-Tek Görev (BZK.SAU.OS)
Dosya Sistemi	FAT-16
Assembly Dili	BZK.SAU.ASM Assembly dili

BZK.SAU.FPGA mikrobilgisayar sisteminde kullanılan komutlar hem tam bir bilgisayarda olması gereken temel komutlardan oluşmakta hem de eğitimsel açıdan komutların işleyişini öğrencilerin anlayabileceği şekilde tasarlanmıştır.

Kullanılan komutların işlemcideki süreçleri birbirinden farklı olduğundan bilgisayar mimarisi CISC yapıdadır. Mimaride kullanılan kaydedici birim sayısı 10 adet olup, tam özellikli bir bilgisayar mimarisinde olması gereken 16-bit genişliğinde kaydediciler seçilmiştir. Eğitimsel amaçlı bilgisayar mimarilerinde yaygın olarak kullanılan 6 adet temel adresleme modu çeşidi kullanılmaktadır. Komutların kontrolünde programlama bilgisini gerektirecek mikro program yapısı yerine eğitimsel açıdan öğrenme sürecini optimum seviyeye çekecek olan donanımsal kontrol yapısının kullanılması tercih edilmiştir[3].

Bilgisayar Mimarisinde bulunan aritmetik ve lojik birimi mevcut haliyle 16-bit tamsayılar üzerinde işlem yapabilmektedir. Negatif sayılarını gösterimi için literatürde sıkça kullanılan 2'ye tümleme mantığı kullanılmıştır[3].

BZK.SAU.FPGA mikrobilgisayar sistemi projenin temel amacı olan eğitimsel yönü nedeniyle basit seviyede oluşturulmuştur. Bundan dolayı tasarlanan işletim sistemi tek kullanıcı ve tek görev yapısında bir işletim sistemidir[3].

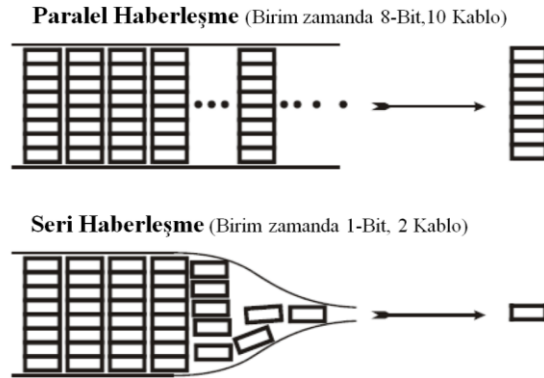
3. I2C SERİ HABERLEŞME PROTOKOLÜ

Bu bölümde Philips firması tarafından geliştirilen ve elektronik aygıtların birbirleri ile haberleşmesini sağlayan I2C (Inter-Integrated Circuit) seri haberleşme protokolü hakkında bilgi verilecektir.

3.1. Seri ve Paralel Haberleşme

Seri haberleşme; dijital bilginin tek bir hat ya da devre üzerinden aynı anda tek bir bit şeklinde ardı sıra aktarılmasıdır. Seri haberleşmede transfer edilen bilginin her biti bir yerden başka bir yere iletmek için data hattı ve clock hattı olmak üzere iki kabloya ihtiyaç duymaktadır [5].

Paralel haberleşme; alıcı ve verici sistemleri arasındaki bütün dijital bilgilerin aynı anda iletilmesi esasına dayanmaktadır. Paralel portlar birim zamanda 1 byte (8-bit) veri gönderebilen veya alabilen kişisel bilgisayar ara yüzleridir. Paralel haberleşme; karşı tarafa gönderilecek kaç tane bit varsa o kadar sayıda iletim hattı üzerinden yapılır. Bilgisayar sistemlerinde, ana kartların içerisindeki mesafeler kısa olduğu için maliyet çok fazla olmayacağından iletim yöntemi olarak paralel haberleşme seçilmektedir. Paralel haberleşmede dijital bilgiler gönderilecek bit sayısı kadar hat üzerinden iletildiği için hızlı bir haberleşme gerçekleşmektedir. Şekil 3.1’de seri ve paralel port data akışı gösterilmiştir[18].

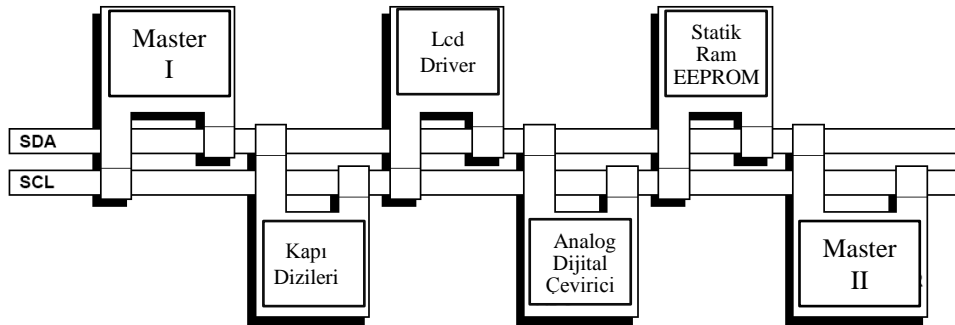


Şekil 3.1. Seri Ve Paralel Port Data Akışı[5].

3.2. I2C Seri Haberleşme Protokolü

Dijital sistem ve aygıt geliştiriciler için önemli parametrelerden biriside tasarladıkları sistem ve aygıtın fiziksel boyutlarıdır. İdeal bir tasarım minimum ebatlar ve maksimum verim olarak özetlenebilir. Dijital sistem ve aygıtların fiziksel büyüklükleri milyonlarca transistörün entegre içerisine yerleştirilmesi ve entegrelerin baskı devreye bağlandıkları pin sayıları ile doğru orantılıdır. Örneğin National Semiconductor firmasının 80'lerde ürettiği 24 pinli M58167B entegresi, 24 pinin 17 tanesini data alma ve data gönderme işlemi için kullanıyordu. Oysa seri haberleşme tekniklerinin gelişmesi ile 17 pin yerine sadece 2 pin kullanılabilir. 17 pin yerine 2 pin kullanmak aygıtın fiziksel olarak küçülmesinin de önünü açar[19].

I2C haberleşme protokolü, Seri Data (SDA) ve Seri Saat Darbesi (SCL) olmak üzere iki hattan meydana gelir. Aygıtlar arasındaki iletişim bu iki hat üzerinden yapılır. Sistem üzerindeki her aygıt yalnız bir adresle tanınır (FPGA, Mikrodenetleyici, LCD sürücü, Hafıza, Klavye) ve özelliğine göre Gönderici (Transmitter), Alıcı (Receiver) veya hem Gönderici hem de Alıcı olarak çalışır. Örneğin hatta ki eleman bir LCD sürücü ise sadece yöneticiden gelen bilgileri alırken hattaki eleman hafıza ise hem yöneticiden gelen bilgileri alabilir hem de hatta bilgi gönderebilir. Hatta ki eleman mikrodenetleyici veya FPGA ise hem hatta veri gönderebilir hemde hattan veri alabilir ayrıca hattı kontrol etme yetkisine de sahiptir. Şekil 3.2' de örnek bir I2C haberleşme blok diyagramı gösterilmiştir[15-17].



Şekil 3.2. Örnek Bir I2C Haberleşme Sistemi[15]

Şekil 3.2’de görüldüğü üzere hat üzerinde birden fazla master yani yönetici aygıt olabilir. Fakat sistemin o anki kontrolü sadece birisindedir. Hattı yöneten aygıt aynı zamanda seri saat darbelerini de üretir.

3.2.1. I2C Haberleşme Protokolü Terimleri ve Açıklamaları

Tablo 3.1’de I2C seri haberleşme protokolünde kullanılan terimler ve açıklamaları verilmiştir.

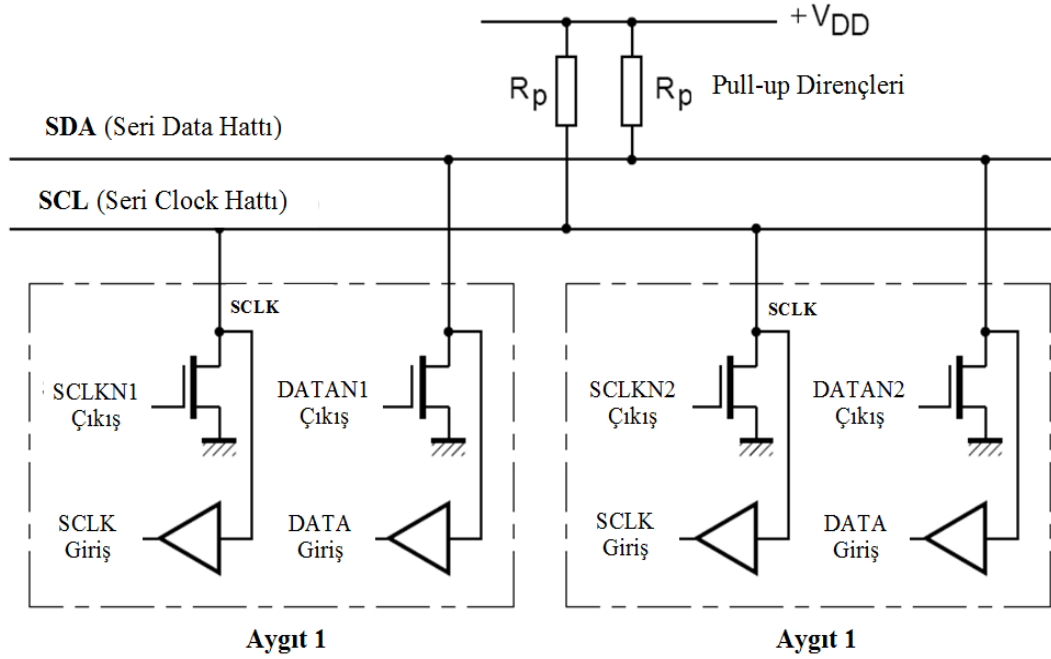
Tablo 3.1. I2C Haberleşme Protokolü Terimleri Ve Açıklamaları[15].

Terim	Tanımlar
Gönderici (Transmitter)	I2C haberleşme protokolünde veri yoluna veri gönderen elemana denir.
Alıcı (Receiver)	I2C haberleşme protokolünde veri yolundan veri alan elemana denir.
Yönetici (Master)	I2C haberleşmesinde veri iletimini yöneten yani başlatan, durduran ve veri iletişimi için gerekli saat darbesini üreten elemana denir.
Yönetilen (Slave)	Yönetici tarafından adreslenen, kontrol edilen elemana denir.
Çoklu-Yönetici (Multi-Master)	Birden fazla yöneticinin aynı anda hattaki veriyi bozmadan hattı kontrol edebilmesi.
Arbitrasyon (Arbitration)	Eğer hatta birden fazla yönetici varsa ve bunlar aynı anda hattın kontrolünü ele almaya çalışırlarsa, bunlardan sadece birinin hattı kontrol etmesi sağlanır ve veri kesilmeden iletilir.
Senkronizasyon (Synchronization)	İki veya daha fazla aygıt için saat darbelerinin eşzamanlı olarak üretilmesi.

3.2.2. I2C Genel Karakteristiği

SDA ve SCL hatlarının her ikisi de iki yönlü çalışabilen hatlardır. Şekil 3.3’de gösterildiği gibi R_p pull-up dirençlerinden dolayı hatta bilgi olmadığı zaman her iki hatta pozitif V_{DD} kaynak gerilimi ile yüksek(lojik bir) seviyesine çekilir[15-17].

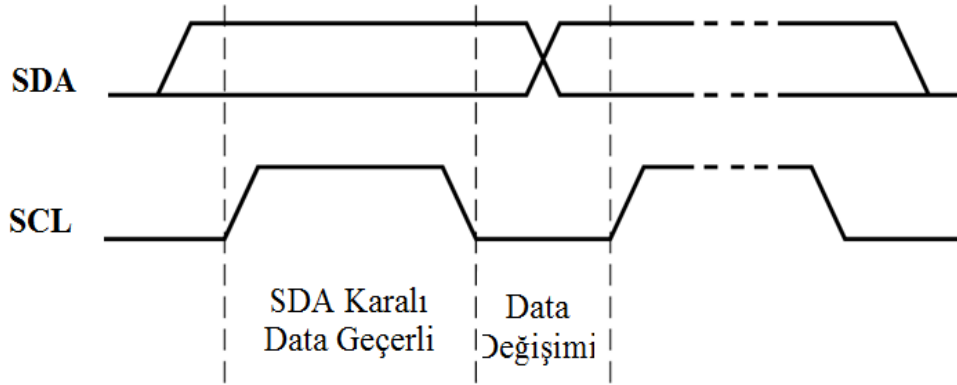
I2C protokolü ile yavaş (slow), hızlı (fast) ve yüksek hızlı (high speed) olmak üzere çeşitli hızlarda veri iletimi sağlanabilir. Yavaş (slow) hızda standart, veri iletişim hızı 100 kbit/s’ye kadar, hızlı (fast) modda veri iletişim hızı 400 kbit/s’ye kadar, yüksek hızlı iletişimde ise veri iletişim hızı 3,4 Mbit/s’ye kadar çıkabilmektedir[15-17].



Şekil 3.3. I2C Ayıtların Hattı Bağlanması[15].

3.2.3. I2C Protokolünde Geçerli Data

Şekil 3.4’de gösterildiği gibi I2C protokolünde datanın geçerli olması için SDA sinyali SCL sinyali yüksek iken değişmez. SDA sinyalindeki değişimler SCL sinyali düşük iken gerçekleşmelidir.[15].

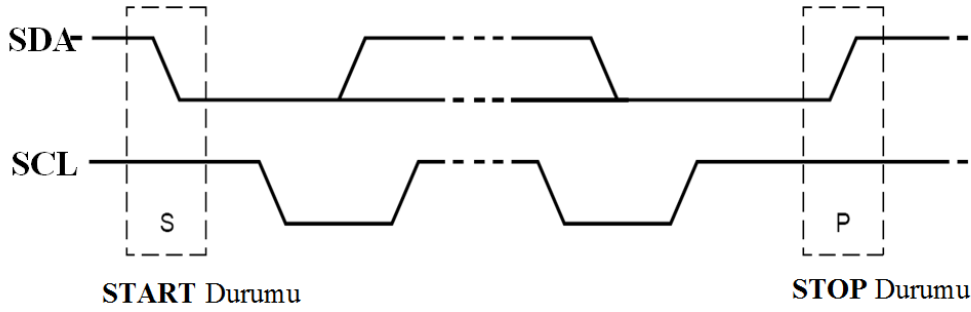


Şekil 3.4. I2C Protokolünde Geçerli Data[15].

3.2.4. I2C Protokolünde START ve STOP Durumları

SCL sinyali yüksek (lojik bir) iken SDA sinyali lojik birden lojik sıfıra geçer ise bu durum START anlamına gelir ve data transferi başlar. START veya tekrarlanan bir

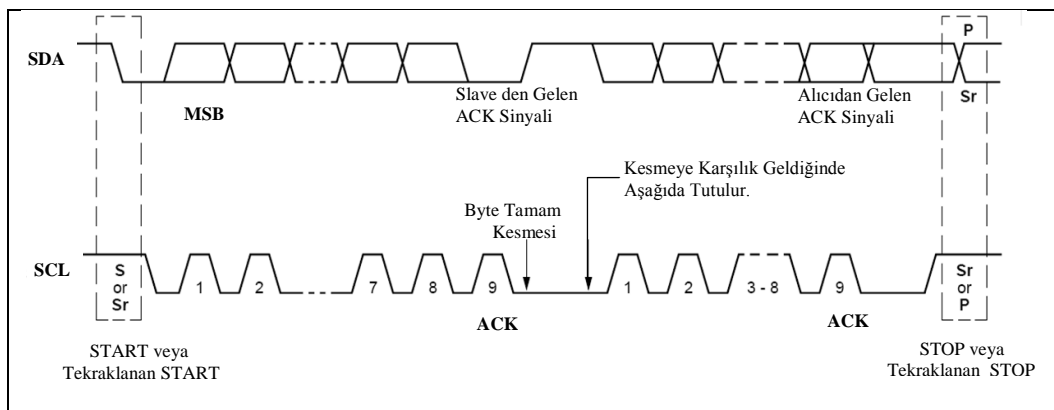
START durumundan sonra SCL sinyali yüksek iken SDA sinyali lojik sıfırdan lojik bire geçerse bu duruma STOP denir ve data transferi sonlandırılır. I2C protokolünde start ve stop durumları şekil 3.5’de gösterilmiştir[15].



Şekil 3.5. I2C Protokolünde START ve STOP Durumları[15].

3.2.5. I2C Protokolünde Data Transferi

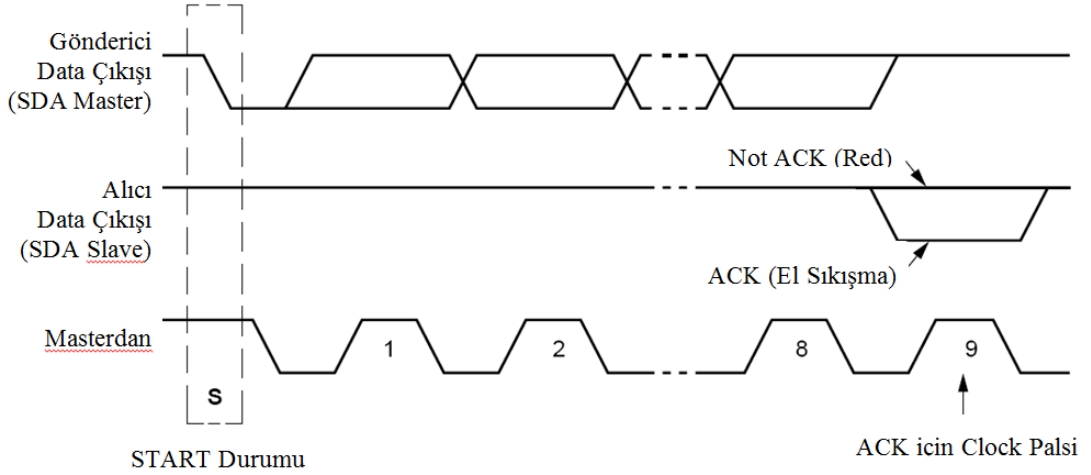
SDA hattında taşınan her byte 8-bit uzunluğunda olmalıdır. Taşınacak bayt sayısı ile ilgili herhangi bir sınırlama yoktur. SDA hattı ile taşınan her bir byte tan sonra kabul biti (ACK) bilgisi alınır. Veri iletimine ilk olarak en değerli bittten (MSB) başlanır. Eğer slave aygıt baytların tamamını alamıyor veya gönderemiyorsa bu durumda SCL, master tarafından düşük (lojik sıfır) seviyesinde tutulur ve slave hazır hale geldikten sonra data iletilir. Şekil 3.6’da I2C Protokolünde data transferi gösterilmiştir.



Şekil 3.6. I2C Protokolünde Data Transferi[15].

3.2.6. I2C Protokolünde ACK (El Sıkışma)

Bu protokolda iletilen, alınan her 8-bit veriden sonra ACK kabul sinyali kullanılmak zorundadır. Kabul için gerekli SCL sinyali yönetici(master) tarafından üretilir. Gönderici, kabul SCL sinyali süresince SDA hattını lojik bire çekerek hattı serbest bırakır. Alıcı kabul SCL sinyali süresince SDA hattını lojik sıfıra çeker. Bu durum şekil 3.7' de gösterilmiştir[17].



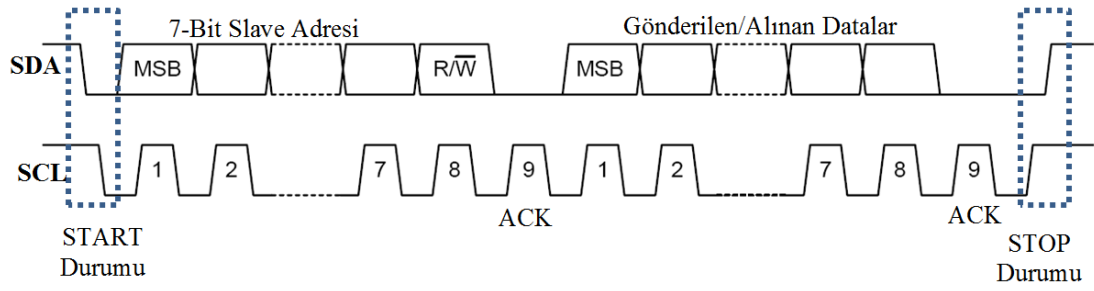
Şekil 3.7. I2C Protokolünde ACK Sinyali[15].

Eğer alıcı tarafından ACK sinyali üretilmez ise gönderici hattı lojik bir seviyesine çekerek iletişimi sonlandırır veya yeni bir Start durumu üretir[15-17].

SCL sinyalinin dokuz numaralı saykılının yüksek periyodu boyunca alıcı data çıkışı (SDA Slave) hattı lojik bir seviyesinde olursa Not ACK sinyali üretilmiş olur. Bu durumda gönderici hattı (SDA Master) lojik bir seviyesine çekerek iletişimi sonlandırır veya tekrarlanan bir Start durumu üretir[15-17].

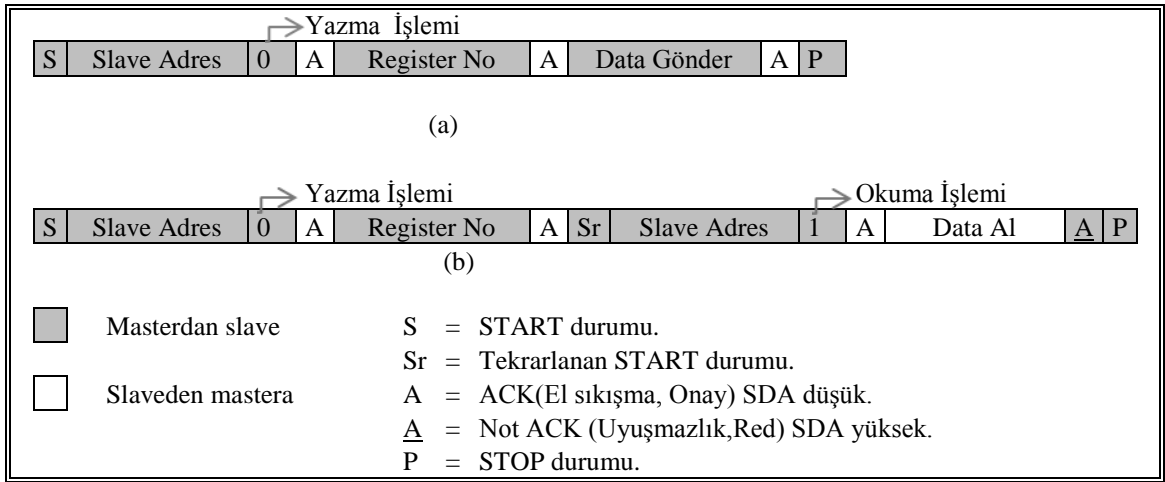
3.2.7. I2C Protokolünde Adresleme Formatı

Şekil 3.8' de I2C protokolünde datanın taşınma adımları gösterilmiştir. START (S) durumundan sonra, slave adresi gönderilir. 7 bit uzunluğundaki bu bitlerden hemen sonra veri yön biti olan R/W biti gelmektedir. Bu bitin '0' olması yazma işlemi, '1' olması ise okuma işlemi yapılacağını göstermektedir. Datalar gönderildikten sonra veri iletimi daima master tarafından üretilen STOP (P) durumu ile sonlandırılmaktadır[19].



Şekil 3.8. I2C Protokolünde Tam Bir Data Transferi[19].

Şekil 3.8’ de I2C protokolünde datanın taşınması sırasında meydana gelebilecek değişik durumlar gösterilmiştir.



Şekil 3.9. I2C Protokolünde Data Transferinde Oluşabilecek Değişik Durumlar[19].

Şekil 3.8’ (a) da Master START şartını oluşturduktan sonra hattaki tüm aygıtlar mastırdan gelecek olan slave adresini beklemeye başlarlar. Sistemin yöneticisi 7-bitlik slave adresini ve bu adrese yazma işlemi yapacağını göstermek için 8. biti 0 gönderir. Hatta ki tüm aygıtlar gelen adres ile kendi adreslerini karşılaştırırlar eğer adres uyuşuyor ise kendisinin iletişime hazır olduğunu belirten ACK (el sıkışma) sinyali gönderilir. Artık master ve slave cihaz haberleşmek için anlaşmıştır. Sonraki adımda master yazma yapacağı slave aygıtın hangi registırlarına yazma işlemi yapacağını gösteren register adresinin gönderir. Slave aygıt tekrar ACK sinyali oluşturur ardından master registerin içeriğine 8-bit bilgiyi gönderir ve slave aygıttan ACK sinyalini bekler. ACK sinyalinden sonra STOP durumu oluşturarak iletişimi sonlandırır[19].

Şekil 3.8' (b) de Master START şartını oluşturduktan sonra 7-bitlik slave adresini ve bu adrese yazma işlemi yapacağını göstermek için 8. biti 0 gönderir. Slave aygıt ACK (el sıkışma) sinyali gönderir. Artık master ve slave cihaz haberleşmek için anlaşmıştır. Sonraki adımda master aygıt SDA hattı aracılığı ile slave aygıtta register adresini gönderir. Slave aygıt register adresini aldığını gösteren ACK sinyali oluşturur. Beklenmeyen bir durum gerçekleşerek master yeni bir START durumu oluşturdu işlem başa döndü. 7-bitlik slave adresini ve bu adresten okuma işlemi yapacağını göstermek için 8. biti 1 gönderir. Slave aygıt, kendisinin iletişime hazır olduğunu gösteren ACK (el sıkışma) sinyali gönderir. Master ve slave cihaz haberleşmek için yeniden anlaşmıştır. Sonraki adımda master slave aygıttan data okumak için çalışıyor ama slave aygıttan cevap gelmediği (Not ACK) için STOP durumu meydana gelir ve iletişim master tarafından sonlandırılır[19].

3.2.8. I2C Protokolünde 7-Bit Adresleme Formatı

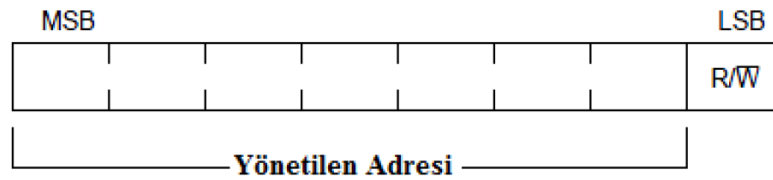
Dijital sistemlerde her aygıtın bir adresi vardır. Master tarafından hangi aygıt ile haberleşmek isteniyor ise o aygıtın adresi I2C protokolü ile gönderilmelidir. I2C adresleme işlemi genellikle, START durumundan sonra gönderilen bayt ile belirlenir. Master tarafından gönderilen bu bayt ile hangi slave aygıtın seçileceği belirlenir. Genel çağrı (General Call) adresi (00000000) bunların dışındadır ve hatta bağlı olan tüm elemanları adresleyebilir. Bu adres kullanıldığında, bütün elemanlar ACK ile cevap verirler. Tablo 3.2'de ilk bayttaki bitlerin açıklamaları verilmiştir[15-17].

Tablo 3.2. İlk Bayttaki Bitlerin Açıklamaları[15].

Slave Adres	R/W Okuma Yazma Biti	Açıklama
0000 000	0	Genel Çağrı.
0000 000	1	Start Byte.
0000 001	X	CBUS Adres.
0000 010	X	Diğer Bus Formatları İçin Ayrılmış.
0000 011	X	İlerki İşlemler İçin Ayrılmış
0000 1XX	X	Hızlı Mod Yönetici Kodu.
1111 1XX	X	İlerki İşlemler İçin Ayrılmış.

1111 0XX	X	10-Bit Slave Adresleme.
----------	---	-------------------------

Şekil 3.8’ de gösterildiği gibi START durumundan sonra ki baytın ilk 7 biti slave adresini oluşturur. 8. bit LSB bitidir ve iletilecek mesajın yönünü belirler. Bu bitin ‘0’ olması, master’ın ilgili slave yazma işlemi yapacağını, ‘1’ olması ise ilgili slave’den okuma yapılacağını gösterir. Adres gönderildiğinde sisteme bağlı tüm elemanlar START durumundan sonraki 7 bit ile kendi adreslerini karşılaştırırlar. Eğer adresler eşleşirse master tarafından R/W bitine bağlı olarak master -alıcı veya master–gönderici olacak şekilde seçilir[15-17].



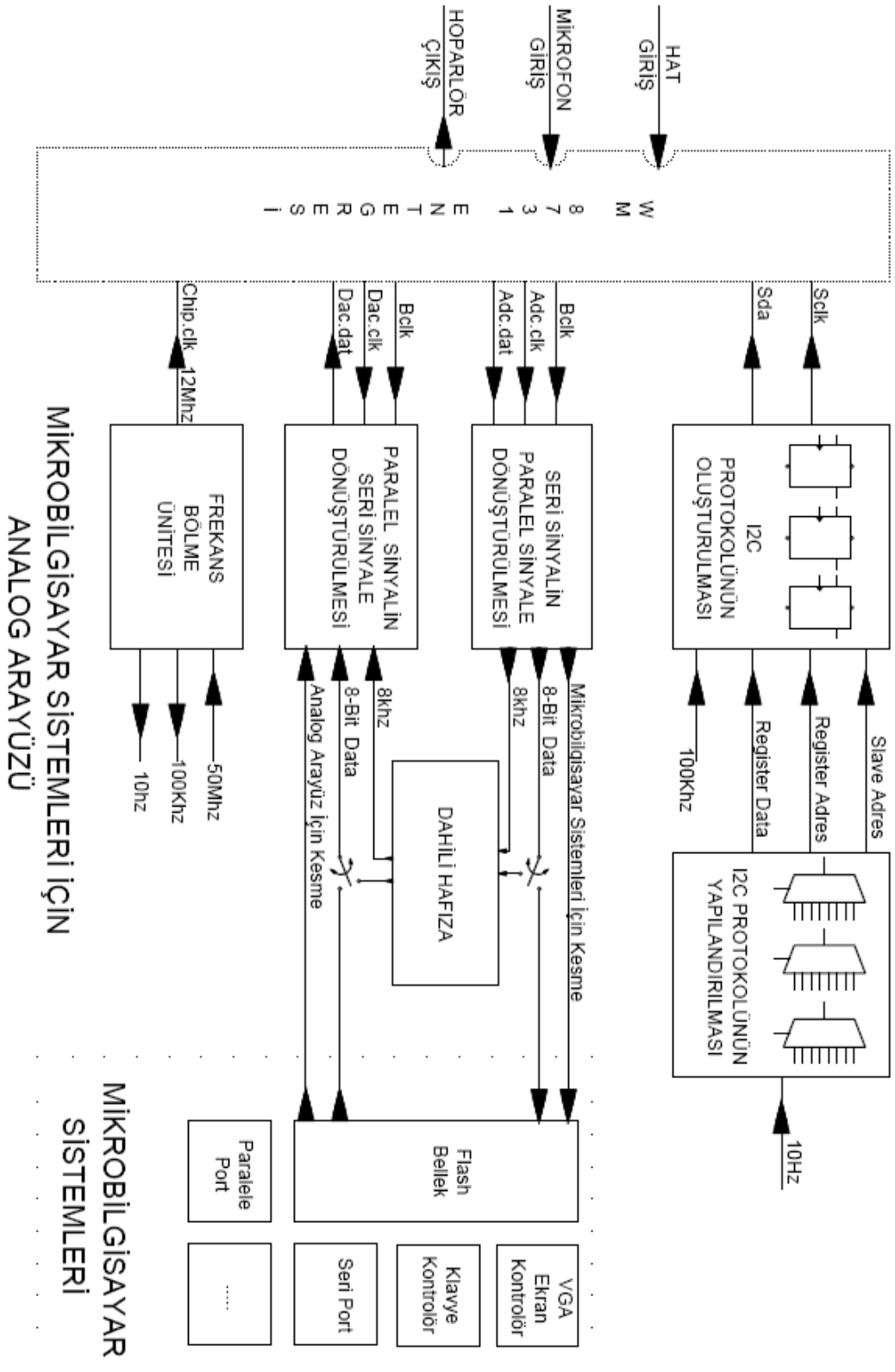
Şekil 3.10. START Durumundan Sonraki İlk Byte[15].

4. YAPILANDIRILABİLİR DONANIM ÜZERİNDE SİSTEMLER VE MİKROBİLGİSAYAR SİSTEMLERİ İÇİN ANALOG HABERLEŞME ARAYÜZ TASARIMI

Bu bölümde FPGA tabanlı sistemler ve mikrobilgisayarlar için eğitsel amaçlı ve modüler olarak tasarlanan analog haberleşme arayüzü hakkında bilgi verilecektir. Gerçekleştirilen sistemin blok şeması şekil 4,1'de verilmiştir.

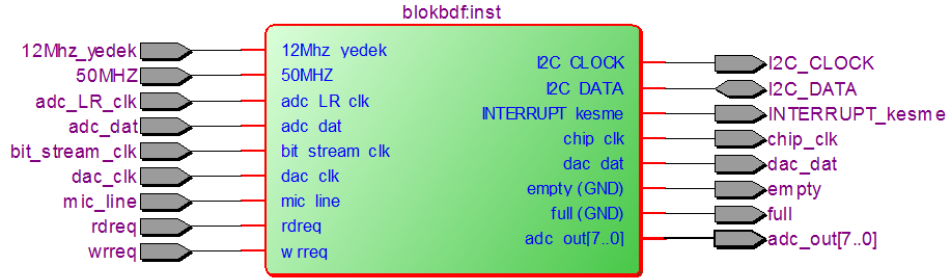
Tasarımı yapılan eğitsel amaçlı analog haberleşme arayüzü ile analog sinyallerin işlenmesinin ve kontrolünün öğrenciler tarafından daha kolay anlaşılabilmesi hedeflenmiştir. Gerçekleştirilen sistem modüler yapıda olup şekil 4.1'de blok şemasında gösterilen her bir blok bir modülü temsil etmektedir. Yapılan çalışmada temel amaç eğitimsel olduğu için, tasarımda kullanılan elemanların içyapısının incelenebilir, kullanıcı tarafından müdahale edilebilir ve geliştirilebilir olmasına özen gösterilmiştir. Eğitsel amaçlı tasarlanan analog haberleşme arayüzü derslerde müstakil kullanılabilceği gibi oluşturulan kesme (interrupt) devresi ile de açık kodlu tüm mikrobilgisayar sistemlerine uyarlanarak da kullanılabilir durumdadır. Öğrencilerin; bu yöntem ile bilgisayarlar ile analog çevre birimleri arasındaki çalışma akışını daha rahat kavrayabilmeleri, motivasyon ve özgüven açısından da üst seviyede kalmaları hedeflenmiştir.

Şekil 4,1'de gösterildiği üzere tasarlanan sistem temel olarak altı modülden meydana gelmiştir. Birinci modülde; 50 Mhz sinyalden sistemin çalışması için gerekli frekanslarda saat darbeleri üretilmiştir. İkinci modülde; FPGA ile çevre birimlerini haberleştirebilmek için I2C kontrolör tasarlanmıştır. Üçüncü modülde; seri sinyal paralel sinyallere dönüştürülmüştür. Dördüncü modülde, paralel datalar dahili hafızaya aktarılırken beşinci modülde ise hafızadan gelen paralel sinyal seri sinyale dönüştürülerek analog çıkışa aktarılmıştır. Son modülde ise tasarlanan analog haberleşme arayüzü BZK.SAU.FPGA mikrobilgisayar sistemine uyarlanmıştır.



Şekil 4.1. Gerçekleştirilen Sistemin Blok Şeması.

Mikrobilgisayar sistemleri için tasarımı yapılan analog haberleşme arayüz modülü şekil 4.2’de verilmiştir. Arayüz dokuz adet giriş pininden, yedi adet çıkış pininden ve bir adet giriş-çıkış pininden meydana gelmiştir.

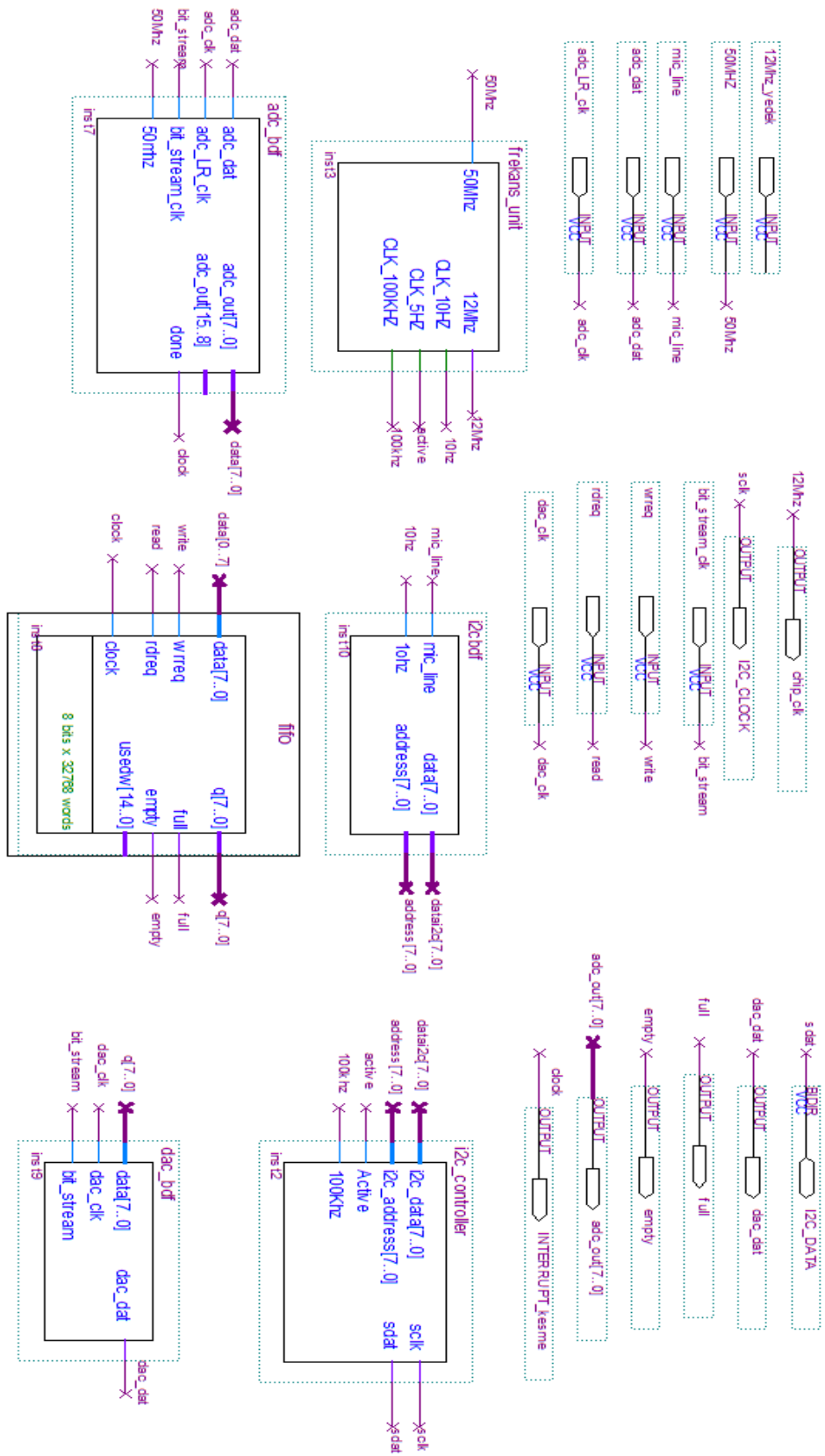


Şekil 4.2. Tasarlanan Analog Haberleşme Arayüz Modülü.

Tasarımı yapılan analog haberleşme arayüz modülü pin tanımlamaları tablo 4.1’de ve modül iç yapısı şekil 4.3’de verilmiştir.

Tablo 4.1. Analog Haberleşme Modülü Pin Açıklamaları.

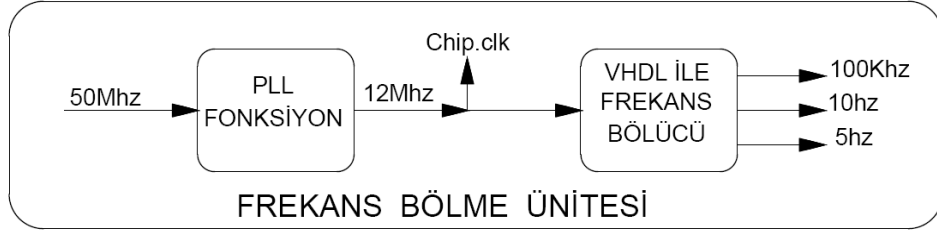
Pin adı	Tipi	Açıklama
12mhz-yedek	Giriş	12Mhz frekanslı saat darbesi yedek girişi.
50mhz	Giriş	50Mhz frekanslı saat darbesi girişi.
Adc_lr_clk	Giriş	Analog dijital dönüştürücü saat darbesi.
Adc_dat	Giriş	Analog dijital dönüştürücüden gelen data.
Bit_stream-clk	Giriş	Bit saat darbesi.
Dac_clk	Giriş	Dijital analog dönüştürücü saat darbesi.
Mic_line	Giriş	0:giriş olarak line seçilir, 1:giriş olarak mikrofon seçilir.
Rdreg	Giriş	Hafızadan oku.
Wrreg	Giriş	Hafızaya yaz.
I2c_clock	Çıkış	I2C protokolü için seri saat darbesi.
I2c_data	Giriş_Çıkış	I2C protokolü seri data akışı.
Interrupt_kesme	Çıkış	Mikrobilgisayar sistemleri için harici kesme sinyali.
Chip_clk	Çıkış	Wm8731 entegresi için 12 Mhz frekanslı saat darbesi.
Dac_dat	Çıkış	Dijital analog dönüştürücüye giden data.
Empty	Çıkış	Hafıza boş.
Full	Çıkış	Hafıza dolu.
Adc_out[7..0]	Çıkış	Analog dijital dönüştürücü çıkışı.



Şekil 4.3. Analog Haberleşme Arayüz Modülü İç Yapısı.

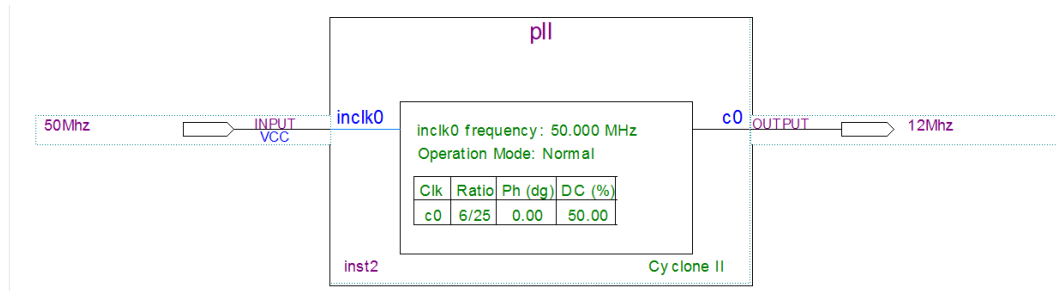
4.1. Frekans Bölme Ünitesi

Bu bölümde sistemin çalışabilmesi için gerekli olan uygun frekanslı saat darbeleri üretilmiştir. Geçekleştirilen sistemin çalışabilmesi için 12Mhz, 100Khz, 10hz ve 5hz frekanslı sinyallerin üretilmesi gereklidir. Frekans bölme ünitesi blok diyagramı şekil 4.4' de verilmiştir.



Şekil 4.4. Tasarlanan Frekans Bölme Ünitesi Blok Diyagramı.

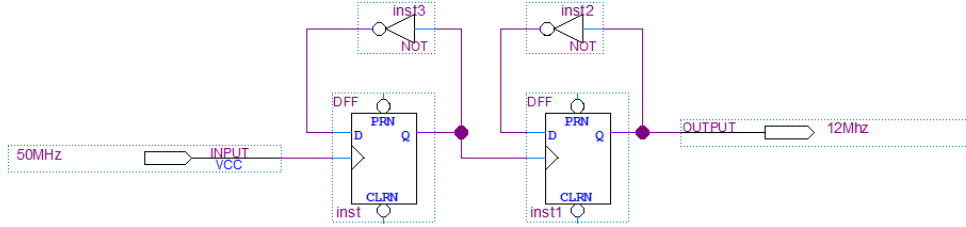
12Mhz frekanslı sinyalin üretilmesinde hazır fonksiyon (PLL) kullanılmıştır. Altera DE2 kiti dahili 50Mhz frekanslı saat darbesi PLL fonksiyona uygulanarak 12 Mhz frekanslı sinyal üretilmiştir. PLL fonksiyon oluşturmak için *Tools* menüsünden *MegaWizard Plug-in Manager* çalıştırılır. Açılan sayfadan genel ayarlar yapıldıktan sonra sırası ile giriş parametreleri ve çıkış parametreleri belirlenerek 12Mhz frekanslı sinyal üretilir. Şekil 4.5' de kullanılan PLL fonksiyon verilmiştir.



Şekil 4.5. Kullanılan PLL Fonksiyon.

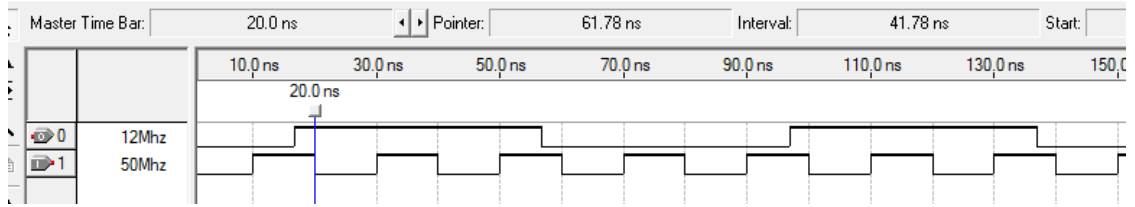
Mikrobilgisayar sistemleri bazı durumlarda sadece bir tane PLL kullanımına izin vermektedir. Mikrobilgisayar sistemi içerisinde hali hazırda kullanılan PLL fonksiyonu ikinci PLL' nin çalışmasına izin vermemektedir. Bu tür çakışmaları engellemek içinde ikinci bir 12Mhz frekans üretici tasarlanmıştır. Tasarlanan frekans bölücü şekil 4.6' da verilmiştir. Tasarlanan analog haberleşme arayüzü herhangi bir

mikrobilgisayar sistemine uyarlandığında PLL iptal edilerek gerekli sinyal frekans bölücünden sağlanır. Frekans bölücünün çıkışından 12.5Mhz alınarak sistemde kullanılır. 12Mhz yerine 12.5Mhz kullanılması sistem çalışmasını etkilememektedir.



Şekil 4.6. Tasarlanan Frekans Bölücü.

Frekans bölücünün çıkış dalga şekli şekil 4.7’de verilmiştir.



Şekil 4.7. Frekans Bölücü Çıkış Sinyali.

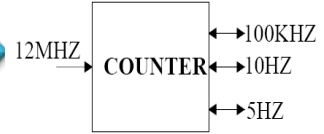
100Khz, 10hz ve 5hz frekanslı sinyalleri elde etmek için 12Mhz frekanslı sinyal giriş olarak kullanıldı ve istenen sinyalleri üretmek için şekil 4.8’de verilen VHDL kodları oluşturuldu. VHDL kodlar derlenerek çalıştırıldığında şekil 4.9’de verilen sinyaller elde edildi.

VHDL Temel Yapısı= Kütüphane Tanımlama + Aygıt Tanımlama + Davranış Tanımlama

```
LIBRARY ieee;  
USE ieee.std_logic_1164.all;  
USE ieee.numeric_std.all;  
USE IEEE.std_logic_signed.all;
```



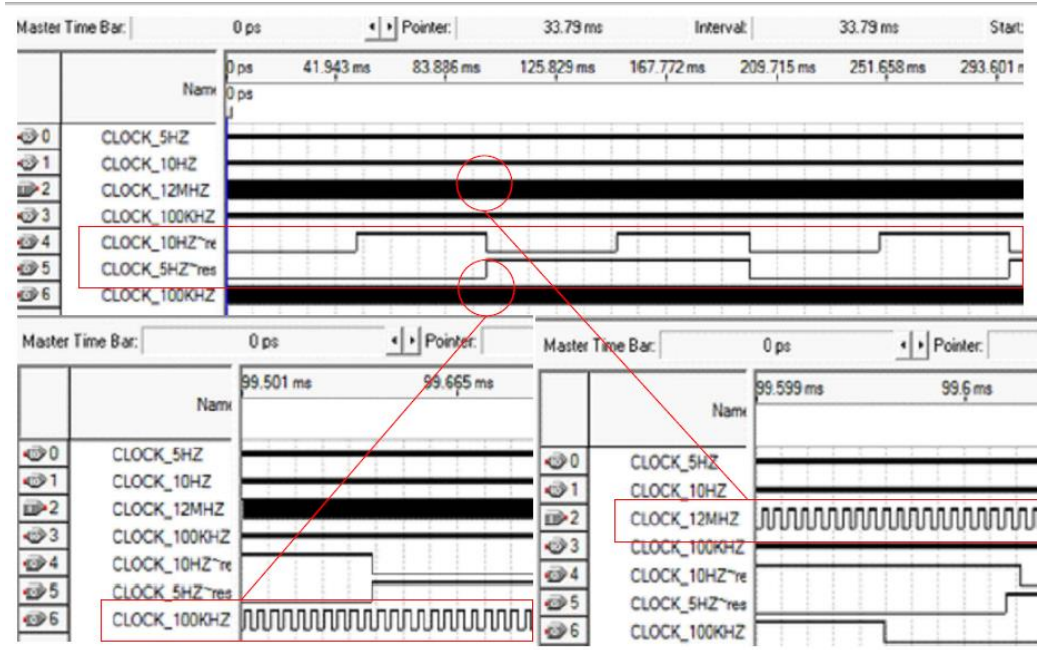
```
ENTITY Frekans_bolucu IS  
PORT (  
    Clk_12Mhz : in std_logic;  
    Clk_5hz   : inout std_logic;  
    Clk_10hz  : inout std_logic;  
    Clk_100Khz : inout std_logic );  
END Frekans_bolucu;
```



```
ARCHITECTURE yapi OF Frekans_bolucu IS BEGIN  
PROCESS(Clk_12Mhz)  
    VARIABLE Clk_5hzV : integer range 0 to 1200000 := 0;  
    VARIABLE Clk_10hzV : integer range 0 to 600000 := 0;  
    VARIABLE Clk_100KhzV : integer range 0 to 60 := 0;  
    BEGIN  
        IF RISING_EDGE(Clk_12Mhz ) THEN  
            IF (Clk_10hzV < 600000) THEN  
                Clk_10hzV := Clk_10hzV + 1;  
            ELSE  
                Clk_10hzV := 0;  
                Clk_10hz <= not Clk_10hz ;  
            END IF;  
            IF (Clk_5hzV < 1200000) THEN  
                Clk_5hzV := Clk_5hzV + 1;  
            ELSE  
                Clk_5hzV := 0;  
                Clk_5hz <= not Clk_5hz;  
            END IF;  
            IF (Clk_100KhzV < 60) then  
                Clk_100KhzV := Clk_100KhzV + 1;  
            ELSE  
                Clk_100KhzV := 0;  
                Clk_100Khz <= not Clk_100Khz ;  
            END IF;  
        END IF;  
    END process;  
END yapi;
```

12MHZ' lik
giriş
sinyalinden
100KHZ, 10HZ
ve 5HZ
frekanslı
sinyal üretildi

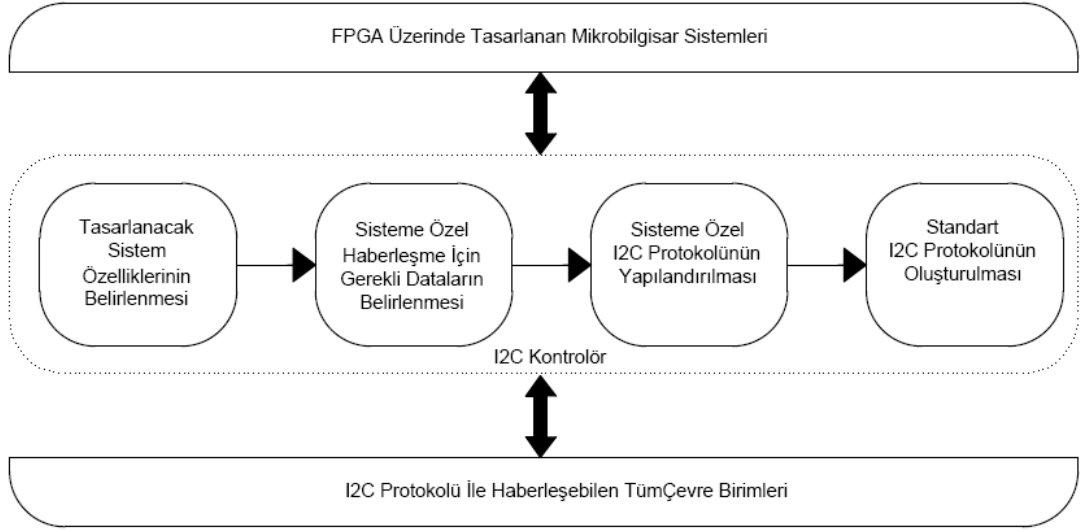
Şekil 4.8. Frekans Bölücü VHDL Kodları.



Şekil 4.9. 12Mhz Sinyalden 100Khz, 10hz Ve 5hz Sinyal Üretilmesi.

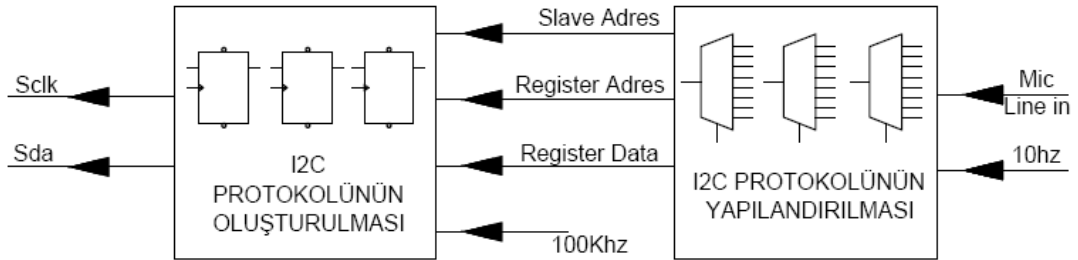
4.2. Gerçekleştirilen I2C Kontrolör

Bu çalışmada Altera DE2 anakartı üzerinde bulunan analog giriş-çıkış için kullanılan Wolfson WM8731 audio kodek entegresi ile FPGA üzerinde tasarlanan mikrobilgisayar sistemlerinin haberleşebilmesi için seri I2C haberleşme protokolü tasarlanmıştır. Bu bölümde aygıtların bir biri ile haberleşebilmesi için tasarlanan, I2C protokolünü kontrol etmek için kullanılan I2C kontrolör hakkında bilgi verilecektir. Gerçekleştirilen I2C protokolü blok diyagramı şekil 4.10'da verilmiştir.



Şekil 4.10. I2C Protokolü Blok Diyagramı.

I2C kontrolör dört bölümden meydana gelmiştir. Birinci bölümde tasarlanacak sistemin frekansı, birim zamandaki örnekleme sayısı, çalışma modu, data hatlarının genişliği gibi özellikler belirlenir. İkinci bölümde WM873 audio kodek entegresinin ürün bilgi sayfalarından faydalanarak belirlenen özelliklere göre gerekli hexadesimal datalar belirlenir. Üçüncü bölümde tespit edilen hexadesimal datalar I2C protokolünde kullanılmak üzere I2C protokolüne uygun olarak yapılandırılır. Dördüncü bölümde; üçüncü bölümden gelen datalar ile SDA ve SCLK sinyallerinden oluşan standart I2C protokolü oluşturulur. İlk üç bölüm I2C protokolünün yapılandırılması olarak düşünüldüğünde gerçekleştirilen I2C kontrolör blok şeması şekil 4.11’de verilmiştir.

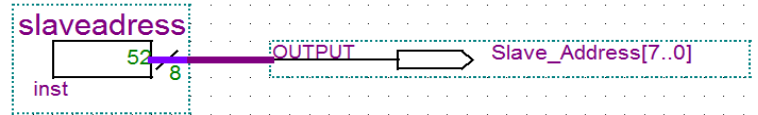


Şekil 4.11. I2C Kontrolör Blok Şeması.

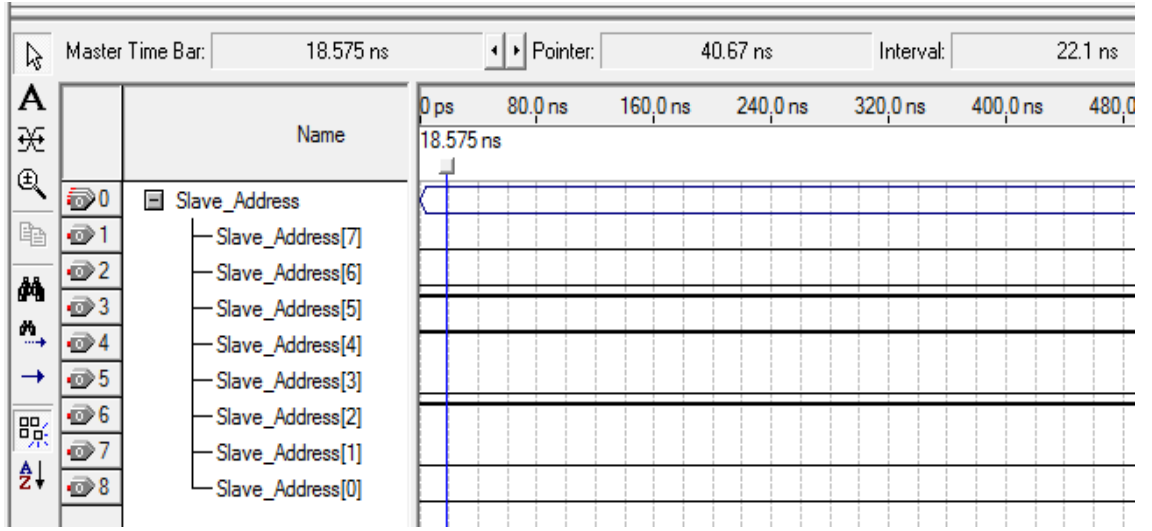
4.2.1. I2C Protokolünün Yapılandırılması

Bu kısımda I2C protokolünde SDA sinyali için gerekli olan slave adres, register adres ve register data bilgileri üretilerek periyodik olarak I2C protokolüne aktarılacaktır. Sistemin girişine 10hz'lik bir saat darbesi ve veri kaynağının mikrofon mu yoksa hat girişi mi olduğunu seçmek için 'Mic/Line in' anahtarı eklenmiştir. 10hz'lik sinyal 12Mhz'lik sinyalden sayıcı kullanılarak elde edilmiştir. 'Mic/Line in' anahtarının konumuna göre ilgili registera gönderilen datanın içeriği değişmektedir. Gerekli dataların üretiminde Wolfson WM8731 audio kodek entegresinin bilgi sayfalarından faydalanılmıştır. Detaylı bilgi üçüncü bölümde verilmiştir. Gönderilecek datalar onaltılık sayı sisteminde 8-bit olarak belirlendikten sonra bu datalar şekil 4.14' de gösterildiği gibi 8x8 multiplexer aracılığı ile 10 hz frekans ile I2C protokolüne gönderilmiştir.

Ürün bilgi sayfalarında Slave adresi Wolfson WM8731 audio kodek entegresi için yazma konumunda '34 hex' ve okuma konumunda '35 hex' olarak belirtilmiştir. Uygulama boyunca WM8731 entegresine yazma işlemi yapıldığından slave adresi sabit '34 hex' (00110100) olarak gönderilmiştir. Şekil 4.12'de Quartus programında slave adresi üretmek için kullanılan şematik devre ve şekil 4.13' de bu devrenin çıkışı gösterilmiştir.



Şekil 4.12. Slave Adresi Üretmek İçin Kullanılan Şematik Devre.



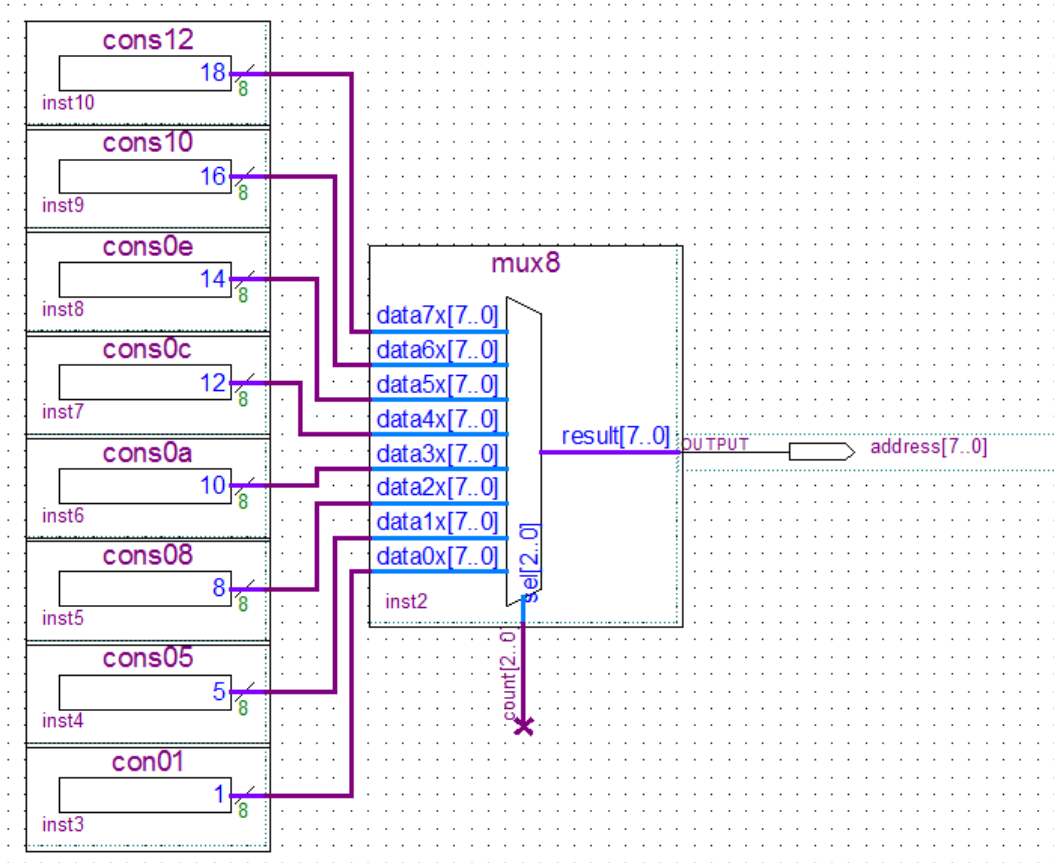
Şekil 4.13. I2C Protokolüne Gönderilen Slave Adres Bilgisi.

Wolfson WM8731 audio kodek entegresi 11 adet 16 bit bağlantılı 8 bit registera sahiptir. İlk 7 bit register adresini gösterirken 8. işlemin okuma mı yoksa yazma mı olduğunu gösterir. Sonraki 8 bit register içerisindeki dataları ifade etmektedir. Bu uygulamada 11 adet registerdan 8 tanesi kullanılmıştır. Kullanılan register adresleri ve kullanım amaçları tablo 4.2’ de verilmiştir.

Tablo 4.2. Kullanılan Register Adresleri ve Kullanım Amaçları.

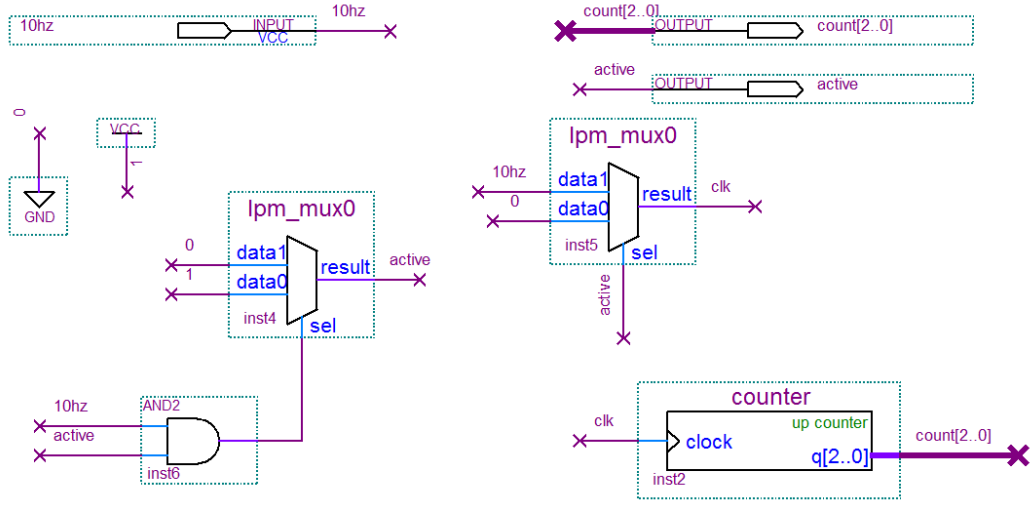
Register No	Hexadesimal Adres	Desimal Adres	Açıklama
1	01	1	Giriş ses seviye bilgilerini içerir
2	05	5	Çıkış ses seviye bilgilerini içerir
3	08	8	Giriş için kullanılacak aygıt seçilir(mic/line).
4	0A	10	Örnekleme frekansı belirlenir
5	0C	12	İstenen bölüm devre dışı bırakılır
6	0E	14	Data formatı ve genişliği seçilir
7	10	16	Örnekleme modu seçilir
8	12	18	Kodek aktif veya pasif yapılır

Tasarımda kullanılacak registerler belirlendikten sonra register adreslerini I2C protokolüne göndermek için 8*8 multiplexer tasarlanmıştır. Register adreslerini I2C protokolüne göndermek için kullanılan 8*8 multiplexer devresi şekil 4.14’de verilmiştir.



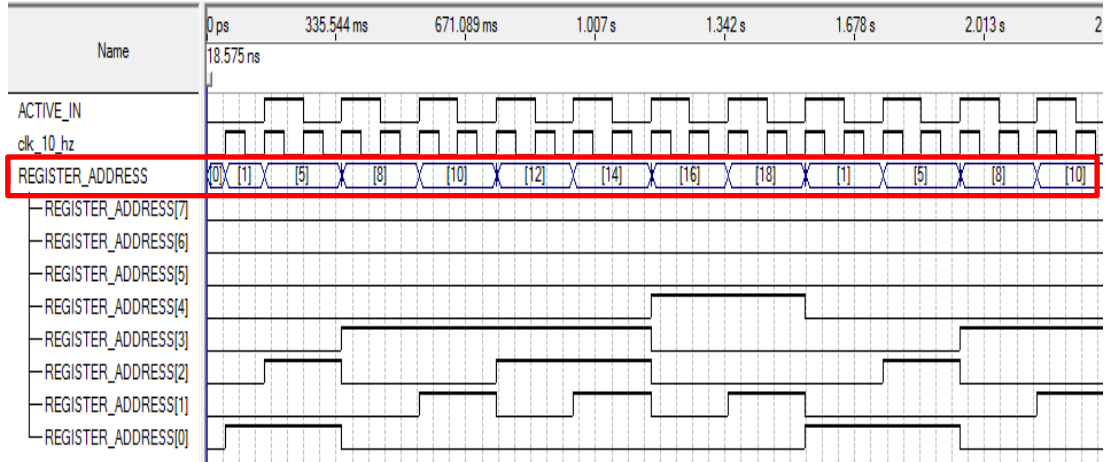
Şekil 4.14. Register Adresi Üretmek İçin Tasarlanan 8*8 Multiplexer.

8*8 multiplexerda seçici uç olarak sürme devresi tasarlanmıştır şekil 4.15’de 3-bitlik sayıcının değeri değişikçe multiplexerin çıkışındaki adres değeri de değişmektedir. Tasarlanan devre ile multiplexer çıkışındaki register adres bilgisi 200ms de bir değiştirilmiştir.



Şekil 4.15. 8*8 Multiplexer Sürme Devresi.

Register adresi üretmek için kullanılan devrenin çıkışından elde edilen 8 bitlik register adres bilgi sinyali şekil 4.16’da verilmiştir.



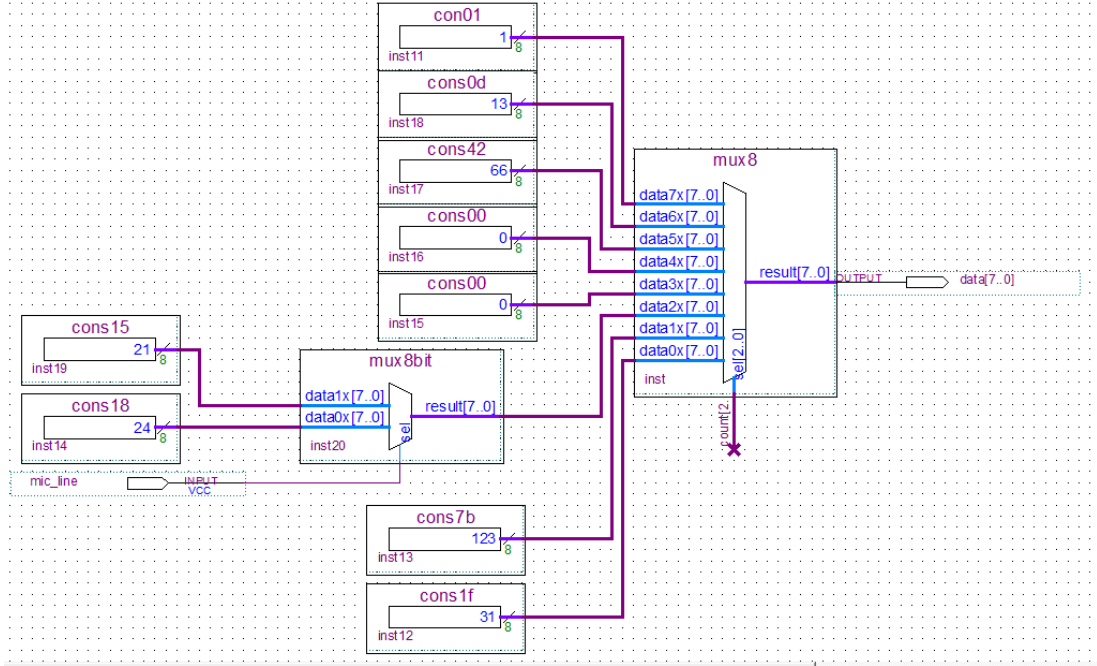
Şekil 4.16. I2C Protokolüne Gönderilen Register Adresi Bilgisi.

Register adres bilgisinden sonra ilgili registra gerekli dataların gönderilmesi gerekir. Register içerisindeki datalar Wolfson WM8731 entegresi bilgi sayfasından faydalanılarak üretilmiştir. Tablo 4.3’ de kullanılan register adresi, register içerikleri ve bu içeriğin anlamı verilmiştir.

Tablo 4.3. Register Adresleri, Register İçerikleri ve Bu İçeriğin Anlamı.

Register Adres (hex)	Register Data (hex)	Register İçeriğinin Anlamı
01	1F	Giriş ses seviyesi ayarlandı.
05	7B	Çıkış ses seviyesi ayarlandı.
08	18	Giriş olarak line in seçildi.
	15	Giriş olarak mikrofon seçildi.
0A	00	Örnekleme frekansı olarak 8Khz seçildi.
0C	00	Devre dışı bırakılan birim yok. Tüm birimler kullanıma hazır.
0E	42	I2S data formatı seçildi. Data genişliği 16 bit olarak belirlendi.
10	0D	Usb mod seçildi.
		Birim zamanda 250 örnekleme yapılacak. Audio kodek entegresine 12Mhz uygulanacak.
12	01	Audio Kodek aktif yapıldı.

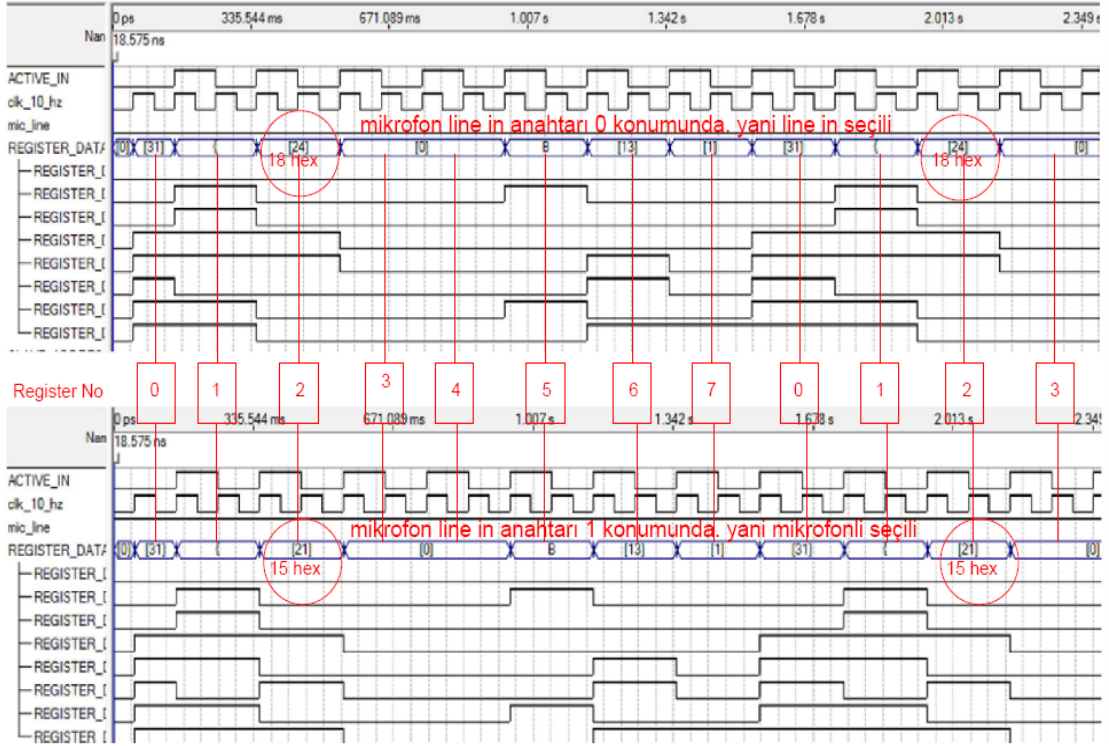
Uygulamada kullanılacak registerlara gönderilecek datalar belirlendikten sonra bu içeriği I2C protokolüne göndermek 8*8 multiplexer tasarlanmıştır. Register içeriklerini I2C protokolüne göndermek için kullanılan 8*8 multiplexer devresi şekil 4.17’de verilmiştir. ‘Line in Mic’ anahtarı ile ses kaydında kullanılacak kaynak seçilir.



Şekil 4.17. Register Data Üretmek İçin Tasarlanan 8*8 Multiplexer.

Multiplexerda seçici uç olarak sürme devresi tasarlanmıştır şekil 4.15. 3-bitlik sayıcının değeri değiştikçe multiplexerin çıkışındaki adres değeri de değişmektedir. Tasarlanan devre ile multiplexer çıkışındaki register adresi bilgisi 200ms de bir değiştirilmiştir.

Register içeriği üretmek için kullanılan devrenin çıkışından elde edilen 8 bitlik sinyal şekli şekil 4.18’de verilmiştir. Şekle dikkat edilecek olursa 8x8 multiplexer seçici ucu ‘000’ iken çıkış ‘1f hex’ desimal karşılığı 31’dir. Seçici uç ‘001’ iken çıkış ‘7b hex’ desimal karşılığı 123’ dür. Seçici uç ‘010’ iken eğer ‘mic/line in’ anahtarı ‘0’ ise çıkış 24 olurken eğer anahtar ‘1’ konumunda yani mikrofon seçili ise çıkış 21 olacaktır. Seçici uç ‘011’ ve ‘100’ konumunda iken çıkış 0 olacaktır. Seçici uç ‘101’ iken çıkış 66 olacaktır. Seçici uç ‘110’ iken çıkış 13 olacak ve son olarak seçici uç ‘111’ iken çıkış 1 olacaktır.

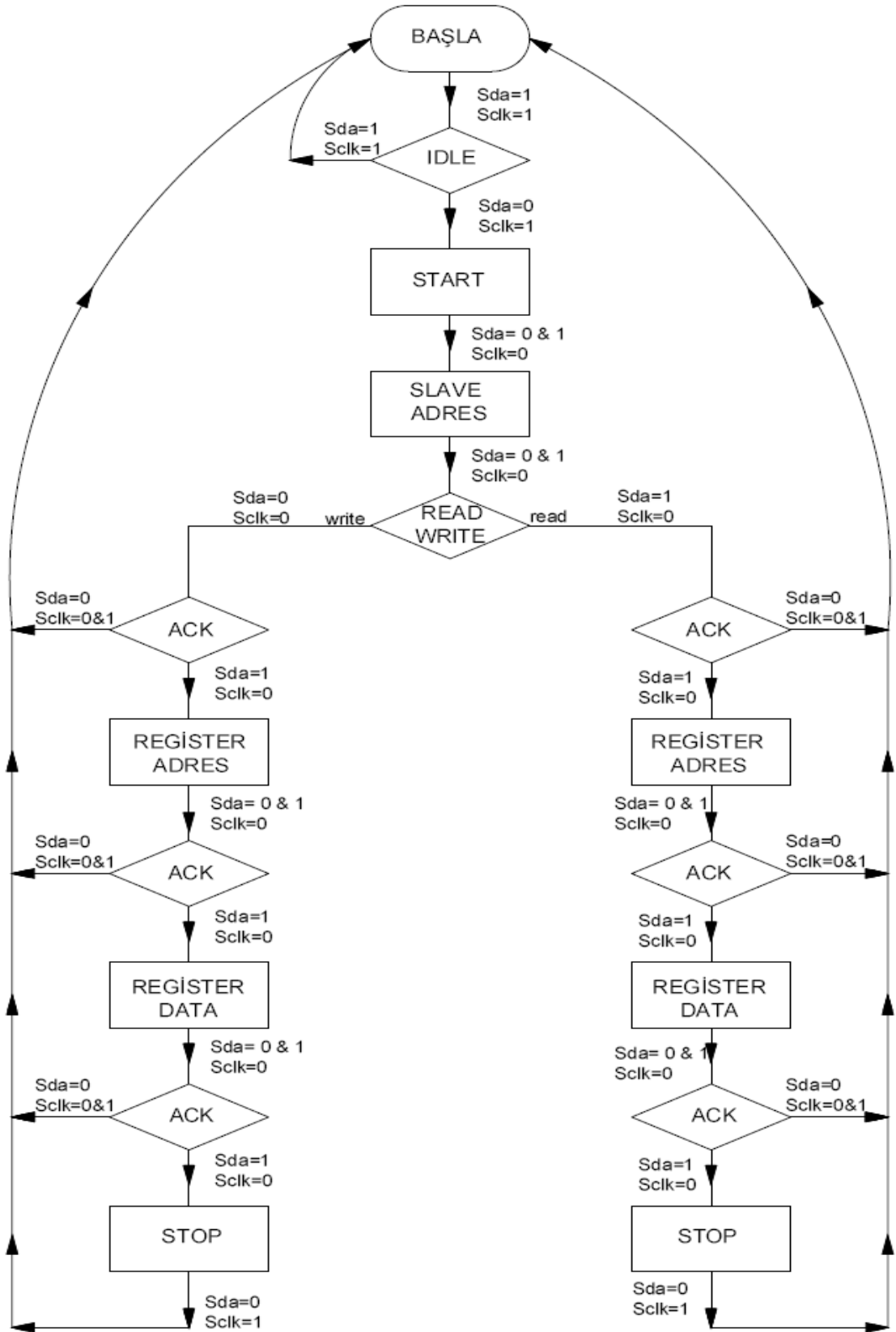


Şekil 4.18. I2C Protokolüne Gönderilen Register İçerik Bilgisi.

4.2.2. Standart I2C Protokolünün Oluşturulması

Bu bölümde I2C protokolünde kullanılan SCLK ve SDA sinyallerinin düzenlenerek standart I2C protokolünün oluşturulması hakkında bilgi verilecektir.

I2C protokolü yapılandırma kısmından periyodik olarak gelen slave adresi, register adresi ve register data bilgileri SDA (serial data) sinyaline aktarılır. Yine bu kısımda I2C protokolüne uygun SCLK (serial clock) sinyali üretildikten sonra 100khz frekansla audio kodek entegresine gönderilir. Tasarlanan I2C protokolünün çalışma akış diyagramı şekil 4.19' da verilmiştir. I2C protokolü hakkında detaylı bilgi 3. bölümde verilmiştir.



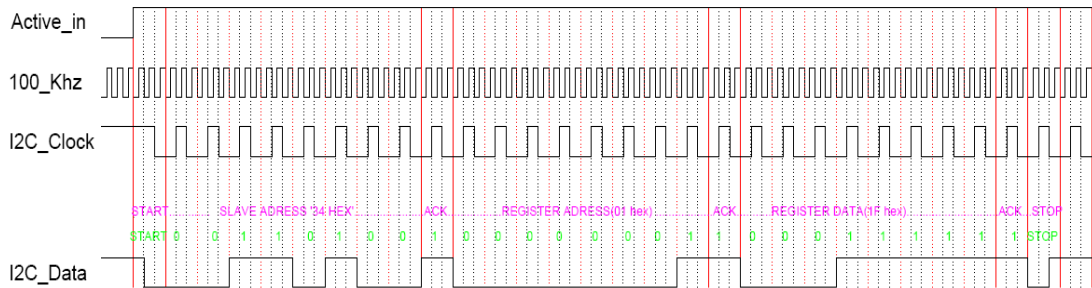
Şekil 4.19. Tasarlanan I2C Protokolü Çalışma Akış Diyagramı

3. bölümde anlatıldığı üzere SCLK sinyali 1 konumunda iken SDA sinyali 1'den 0'a düşerse I2C haberleşme protokolü başlar. SCLK sinyali 1 konumunda iken SDA sinyali 0'dan 1'e yükselirse I2C haberleşme protokolü sonlandırılır. I2C haberleşme protokolü başladıktan sonra tüm adımlar bitene kadar hiçbir şekilde SCLK sinyali 1 iken SDA sinyali değişmemelidir. Yoksa tekrarlanan bir start veya haberleşme protokolünün ortasında stop durumu gelerek haberleşme sonlandırılır. Bu durumu önlemek için SCLK sinyali dizayn edilmiştir.

Tasarlanan sitemde;

1. adım: 'SCLK=SDA=1' boşa bekleme (idle) konumu,
2. adım: 'SCLK=1' ve 'SDA=1 den 0' a start konumu,
- 3.adım: 'SCLK=0' ve 'SDA= 0 veya 1' 7 bit slave adresi gönderilir,
- 4.adım: 'SCLK=0' ve 'SDA= 0 veya 1' 1 bit okuma/yazma bilgisi gönderilir,
- 5.adım: 'SDA= 1' onay konumu slave aygıtın hazır olduğunu belirtir.
- 6.adım: 'SCLK=0' ve 'SDA= 0 veya 1' 8 bit register adresi gönderilir,
- 7.adım: 'SCLK= 1 ve SDA=0'onay konumu register adresi alındı,
- 8.adım: 'SCLK=0' ve 'SDA= 0 veya 1' 8 bit register data gönderilir,
- 9.adım: 'SDA= 1' onay konumu register data alındı,
- 10.adım: 'SCLK=1' ve 'SDA= 0 dan 1' e stop konumu,
- 11.adım: 'SCLK=SDA= 1' boşa bekleme (idle) konumu.

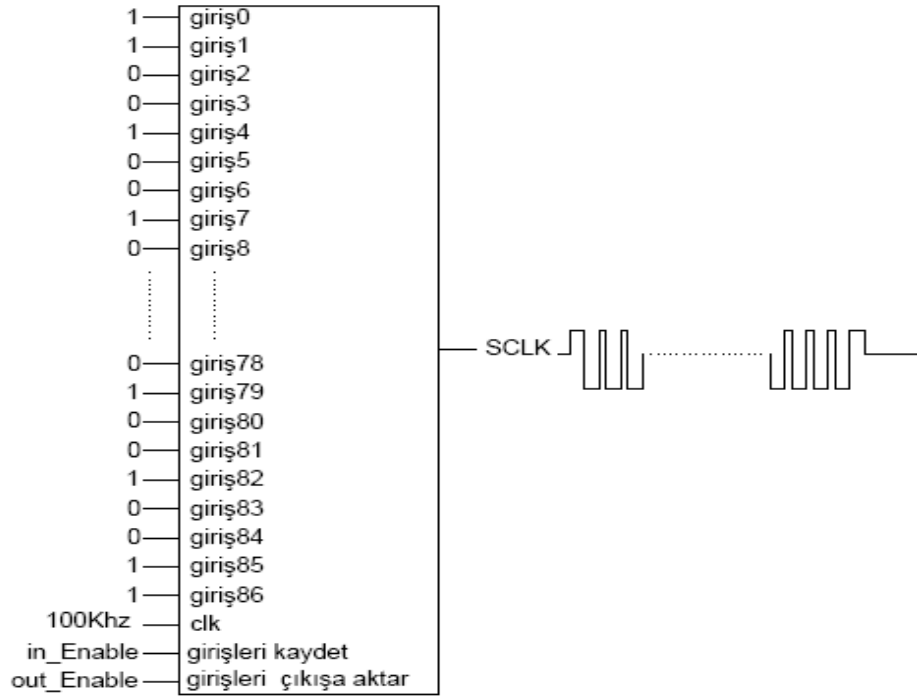
Tasarlanan devrenin SDA ve SCLK çıkış sinyal şekilleri şekil 4.20'de verilmiştir. Şekilde görüldüğü üzere tasarlanan devre ile SDA ve SCLK sinyali hiçbir zaman aynı anda değişmemektedir.



Şekil 4.20. I2C Protokolü SDA ve SCLK Sinyal Şekilleri.

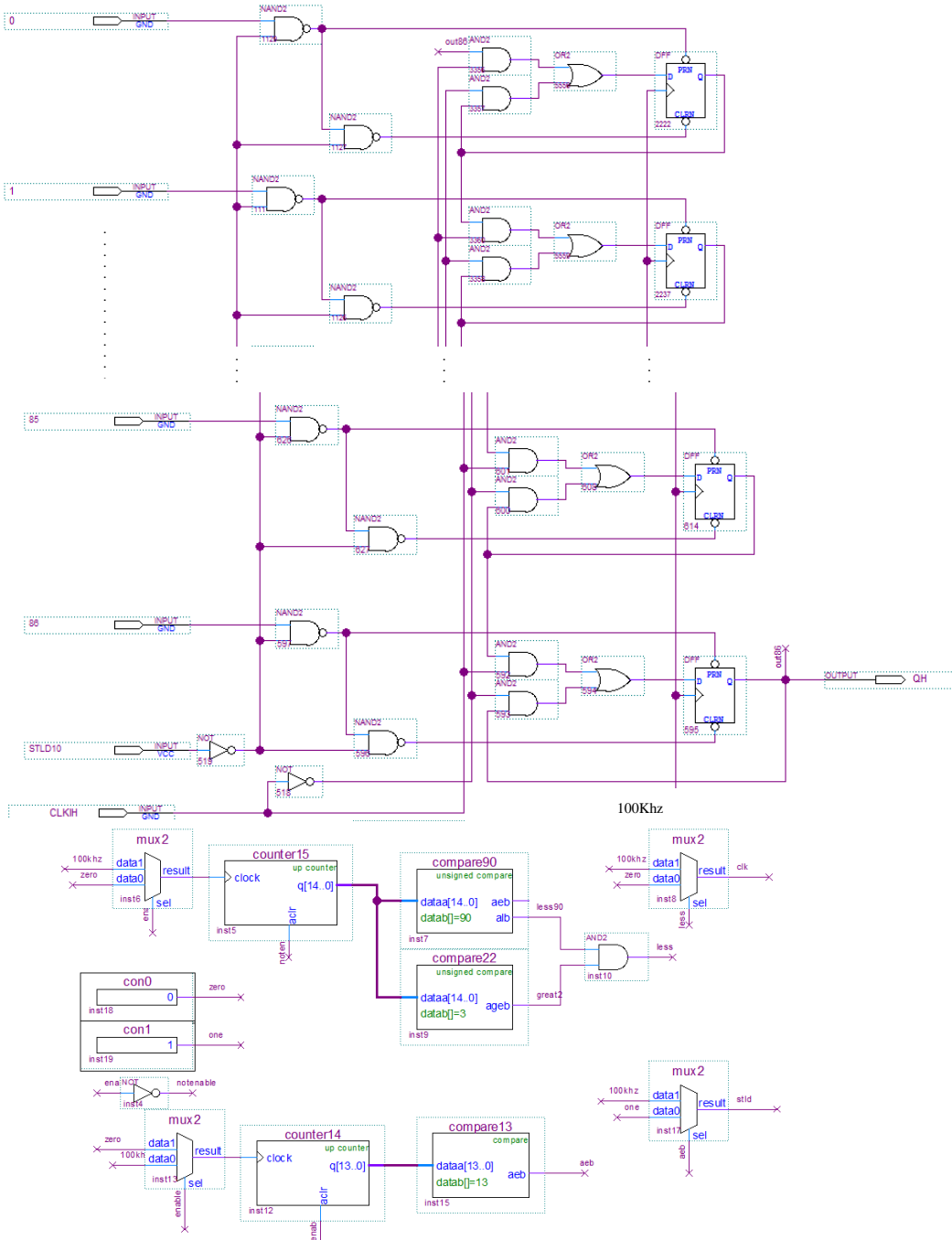
Şekil 4.20’de clock sinyali olarak 100khz’ lik bir clock sinyali kullanılmıştır. Üç clock sinyalinden bir SCLK (i2c_clock) sinyali üretilmiş ve dikey çizgilerle gösterildiği üzere SCLK sinyali 29 adımda tamamlanmıştır. SCLK sinyalini üretmek için $29*3=87$ clock pulsü kullanılmıştır.

SCLK sinyali üretmek için paralel girişleri seri bilgiye dönüştüren bir devre tasarlanmıştır. Girişlerdeki bilgi in_enable girişi aktif iken tek bir saat darbesi ile flip floplara aktarılır. Out_enable sinyali aktif olduğunda 100Khz frekanslı sinyalin her saat darbesinde flip-floplardaki datalar sırası ile çıkışa yani SCLK sinyaline aktarılmış olur. Tasarlanan devrenin blok şeması şekil 4.21’de verilmiştir.



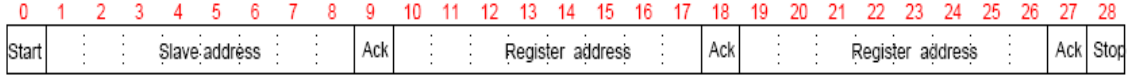
Şekil 4.21. SCLK Sinyal Üretmek İçin Tasarlanan Devrenin Blok Şeması.

SCLK sinyali üretmek için tasarlanan devrenin içyapısı şekil 4.22'de verilmiştir. SCLK ve SDA sinyali üretmek için kullanılan devreler aynı olup sadece girişlere uygulanan sinyaller farklıdır.



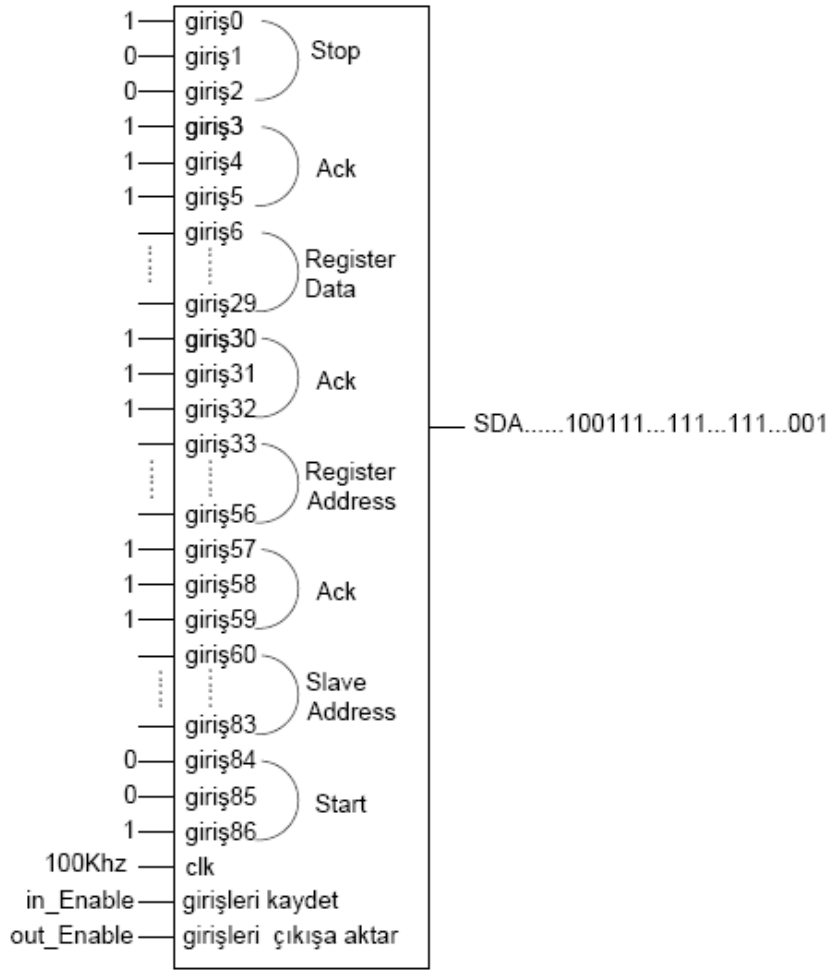
Şekil 4.22. SCLK ve SDA Sinyali Üretmek İçin Tasarlanan Devre.

Gerçekleştirilen sistemde SDA sinyal şekil 4.23’de gösterilen bölümlerden meydana gelmiştir.



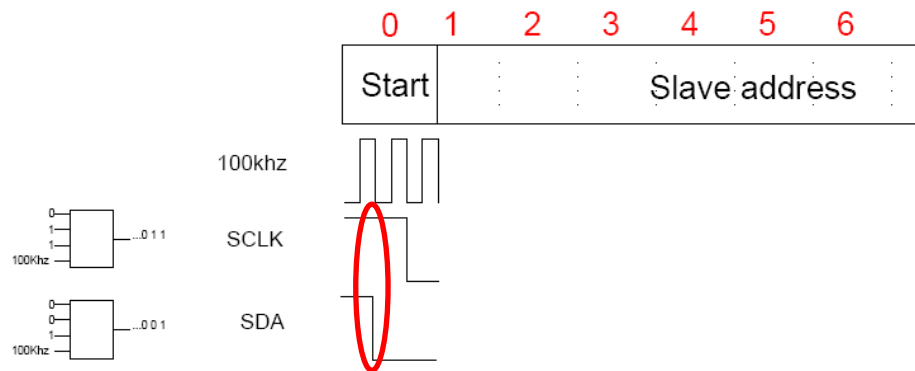
Şekil 4.23. SDA Sinyali İçeriği.

SDA sinyali üretilirken SCLK sinyalinde olduğu gibi 89 tane paralel bilgiyi seri bilgiye dönüştüren shift register devresi tasarlanmıştır. SDA sinyali üretmek için tasarlanan devreye ait blok şema şekil 4.24’ de üretilen SDA sinyal çıkışı şekil4.20’de ve tasarlanan devrenin iç yapısı şekil 4.9’ da verilmiştir. Tasarlanan devre ile paralel 86 giriş 100Khz saat darbeleri ile seri SDA sinyaline aktarılmıştır. Şekilden de anlaşılacağı üzere 100Khz clock pulsü ile çıkışa önce giriş86 en son giriş0 yansıtılmıştır. 86 paralel giriş bilgisi 200 milisaniye aralıklarla aynı anda değişmekte ve SDA sinyalinin periyodik olarak güncellenmesi sağlanmaktadır.



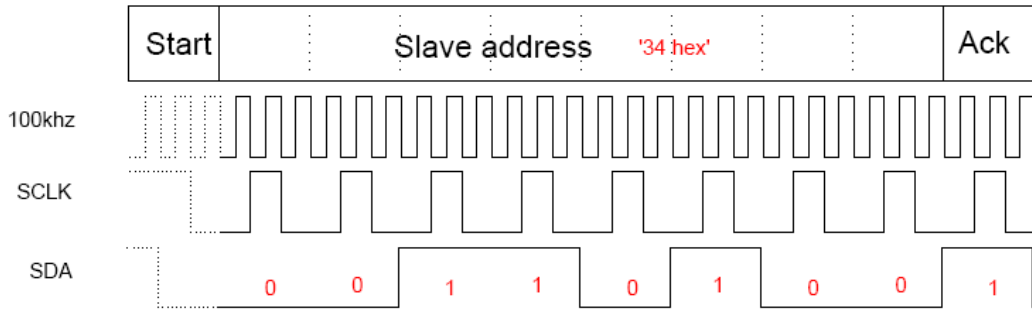
Şekil 4.24. SDA Sinyal Üretmek İçin Tasarlanan Devrenin Blok Şeması.

Şekil 4.25’de gösterildiği üzere SCLK sinyali 1 iken SDA sinyali 0 yapıldığında start şartı sağlanmış olur.



Şekil 4.25. Start Şartının Sağlanması.

Kullanılan audio kodek entegresinin slave adresi ALTERA DE2 kullanıcı kitapçığında '34 hex' olarak belirtilmiştir. Start şartı geldikten sonra slave adresi üretilmiştir. I2C protokolüne uygun olarak master konumundaki aygıt slave aygıttan hazır olduğunu belirten ack sinyalini beklemektedir. SDA sinyaline '34 hex' adresinden sonra slave aygıtın hazır olduğunu gösteren ack sinyali aktarılmıştır. SDA slave adres ve ack sinyalleri şekil 4.26'de gösterilmiştir.



Şekil 4.26 SDA Slave Adres Ve Ack Sinyalleri.

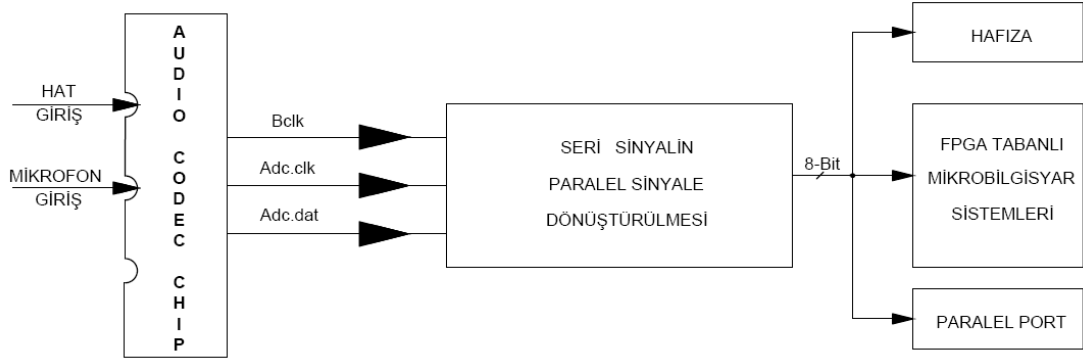
Tasarlanan sistem 200 ms aralıklarla SDA ve SCLK sinyalini güncellemektedir. Bu şekilde register içeriklerindeki değişiklikler eş zamanlı olarak SDA sinyaline yansıtılmıştır. Paralel olarak gelen register adres bilgisi ve register içerikleri slave adresinde olduğu gibi seri bilgiye dönüştürülmüştür. Son kısımda ise SCLK sinyali 1 iken SDA sinyali 0' dan 1'e geçirilerek stop durumuna geçilmiştir. Şekil 4.20'de görüldüğü üzere bekleme konumunda her iki sinyalde 1 durumundadır.

4.3. Seri Bilginin Paralel Bilgiye Dönüştürülmesi

SCLK ve SDA sinyalinin Wolfson WM8731 audio kodek entegresine gönderilmesinin ardından wm8731 entegresi girişinden aldığı analog bilgiyi dijitale dönüştür ve ADC.DAT (Analog Dijital Dönüştürücü Datası) seri sinyalini üretir. .

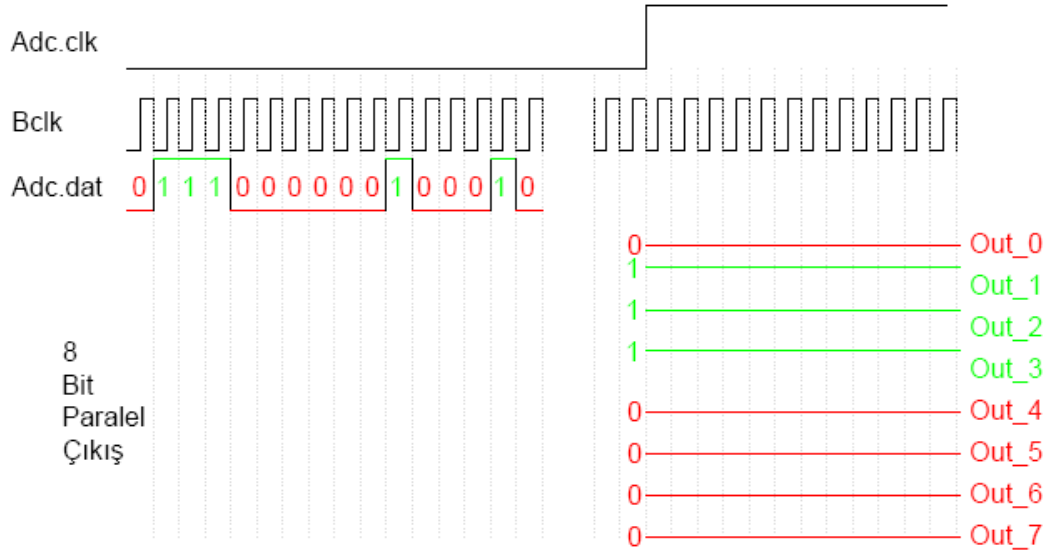
Analog girişten 0 ila 3.3 volt arasında değişen bir gerilim alınmaktadır. '0E hex' register adresine '42 hex' yazıldığı için WM8731 audio kodek entegresi 16 bit data genişliğine sahiptir. Buna göre analog girişten gelen sinyal $3.3/2^{16} = 0.050mV$ hassasiyetle ifade edilebilmektedir. WM8731 audio kodek entegresi '0E hex' register adresine yazılan içeriğe göre 16 bit, 24 bit veya 32 bit olarak analog sinyali

dijitale dönüştürebilmektedir. Bu uygulamada data genişliği olarak 16 bit seçilmiştir. Hafızada daha az yer tutması için 16 bit bilginin en değerli 8 biti alınarak çıkışa aktarılmıştır. Tasarlanan devrenin blok şeması şekil 4.27’de verilmiştir.



Şekil 4.27. Seri Bilginin Paralele Dönüştürülmesi Blok Şeması.

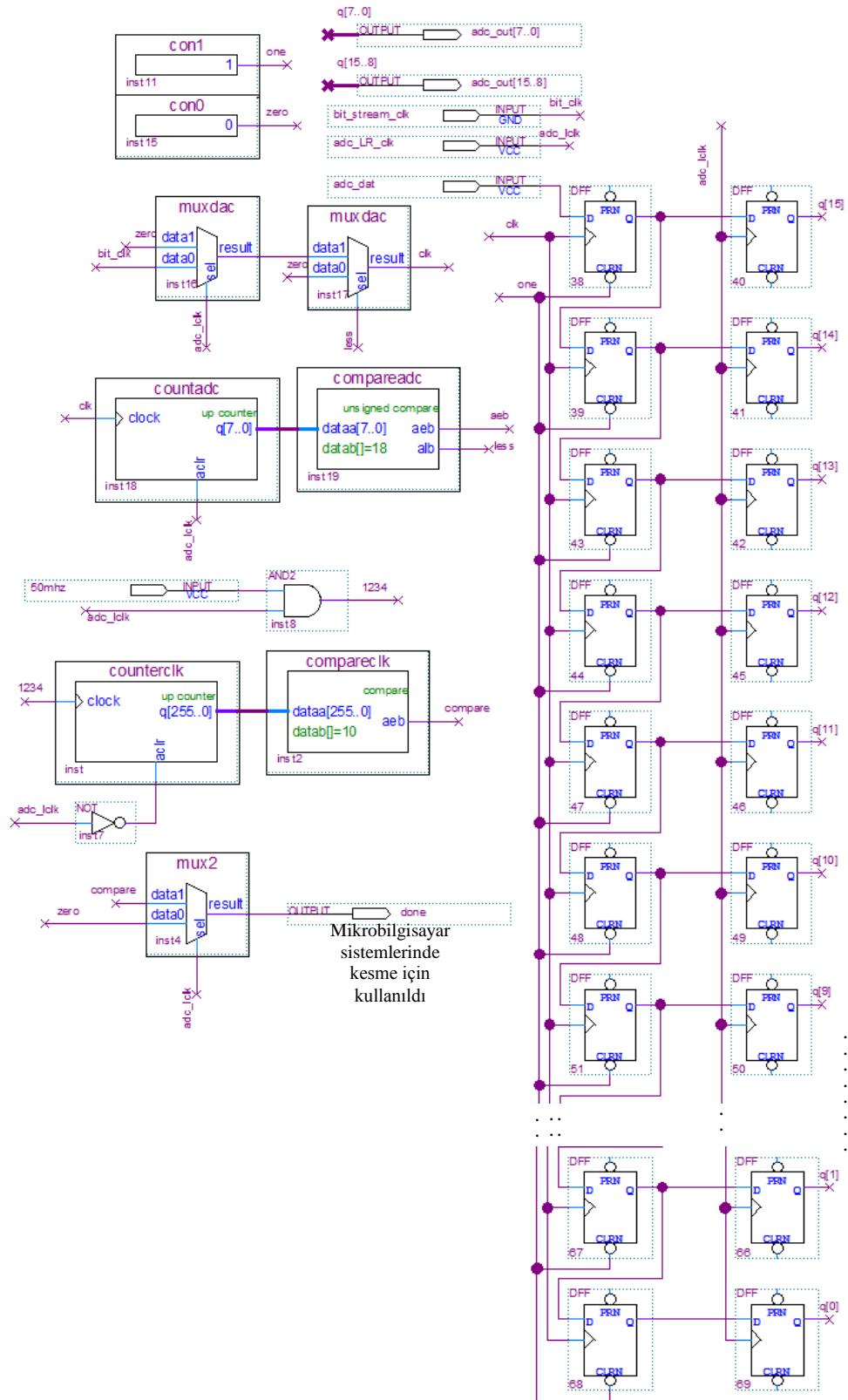
Tasarlanan devrede Adc.clk sinyal frekansı 8khz olarak seçilmiştir. Bclk sinyal frekansı ise 1,5 Mhz’ dir. Şekil 4.29’da gösterildiği gibi gerçekleştirilen devrede 32 adet flip flop kullanılmıştır. Adc.clk sinyali sıfır konumunda iken Bclk sinyalinin ilk 16 saykılında Adc.dat sinyali seri bağlı 16 flip flobun çıkışında 16 bit paralel sinyale dönüştürülür. Adc.clk sinyali 1 konumuna geçtiği anda 16 flip flobun hepsi aynı anda çıkışlarındaki bilgiyi girişlerine bağlandıkları flip floplara aktarır. Buradan anlaşılacağı üzere tasarlanan sistemin örnekleme frekansı Adc.clk sinyaline bağlı olup bu tasarım için 8 khz’ dir. Seri bilginin paralele dönüştürülmesi için tasarlanan devreye ait sinyal şekilleri şekil 4.28’de verilmiştir.



Şekil 4.28. Seri Paralel Dönüştürücü Sinyal Şekilleri

Tasarlanan devrede Adc.dat sinyalindeki 16 bitte çıkışa yansıtılmış ve 8'er bitlik 2 grup oluşturulmuştur. Out_0 ile Out_7 aralığındaki bitler bir grup, Out_8 ile Out_15 aralığındaki bitler bir grup yapılmıştır. Yapılan denemelerde de Out_8 ile Out_15 arasındaki bitlerin sinyali fazla etkilemedikleri gözlenmiş bu nedenle çıkışta sadece Out_0 ile Out_7 aralığındaki en önemli ilk sekiz bit kullanılmıştır. Tasarlanan devreye ait ekran görüntüsü şekil 4.29'da verilmiştir. Tasarlanan devrede; flip-flopların yanısıra Bclk pulsini saydırmak için sayıcı(counter) ve sayıcı çıkışını referans değerlerle karşılaştırmak için karşılaştırıcılar(comparator) kullanılmıştır. Devre üzerinde dikkat edilirse karşılaştırıcının referans değeri '17'den küçükse' yerine '18'den küçükse' olarak seçilmiştir. Çünkü '0E hex' adresindeki registerdan I2S protokolü seçildiğinden Adc.dat sinyalindeki bilgi Bclk sinyalinin 2. Palsi ile başlamaktadır.

8-bit paralel sinyale dönüştürülen seri bilgi buradan paralel porta yönlendirilerek analog sinyal ledler üzerinde gösterildi gibi dâhili hafızaya kayıt edildi ayrıca mikrobilgisayar sistemlerine gönderilerek belleğe yazma uygulaması gerçekleştirildi.



Şekil 4.29. Seri Paralel Dönüştürücü Devresi Ekran Görüntüsü.

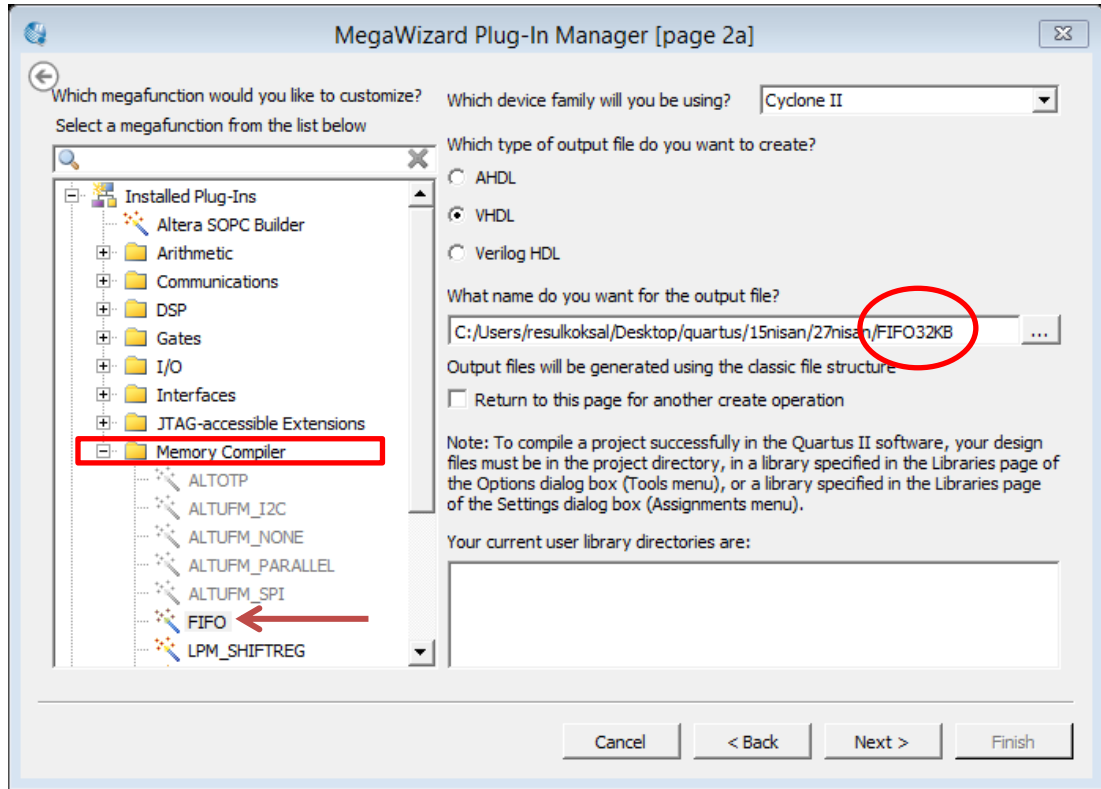
4.4. 8-Bit Paralele Bilginin Hafızaya Kayıt Edilmesi ve Kayıt Edilen Bilginin Hafızadan Okunması

Bu kısımda paralel bilginin kayıt edilmesi için 8-bit genişliğinde 32 Kbyte uzunluğunda FIFO hafıza oluşturuldu. Adım adım anlatırsak;

1.Adım: Menü çubuğundan *Tools/MegaWizard Plug-in Manager* yolu ile Mega fonksiyon sihirbazı açılır.

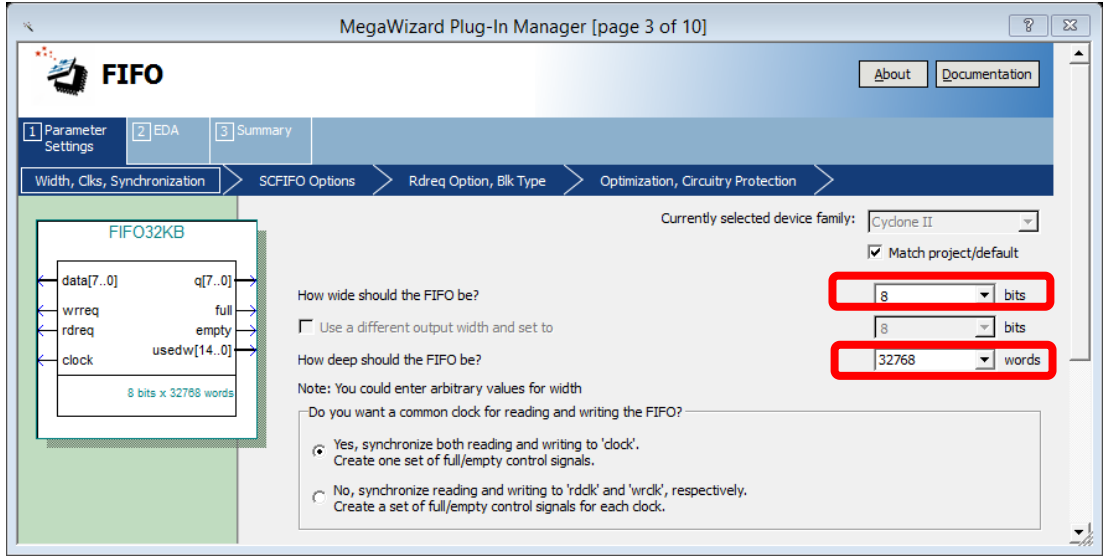
2.Adım: Açılan MegaWizard Plug-in Manager sayfasından 'next ' denir.

3.Adım: Şekil 4.17'den Memory Compiler yazan kısma çift tıklıyoruz, aşağıya doğru açılan listeden FIFO'yu seçiyoruz ve dosya çıkış ismini FIFO32KB diye isimlendirdikten sonra next ikonuna basarak bir sonraki sayfaya şekil 4.30'a geçilir.



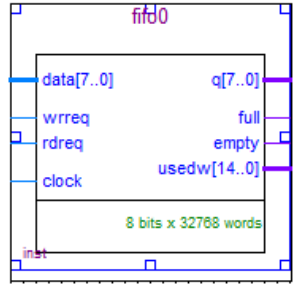
Şekil 4.30. MegaWizard Plug-in Manager Sayfa 2.

4.Adım: Şekil 4.31 de görüldüğü gibi data genişliğini 8 bit ve kelime sayısını 32768 seçerek next ikonuna basarak bir sonraki sayfaya geçilir.



Şekil 4.31. MegaWizard Plug-in Manager Sayfa 3.

5.Adım: Bundan sonra gelen tüm sayfalara next diyerek geçilir ve şekil 4.32’da gösterilen 32 Kbyte FIFO hafıza hazırdır.



Şekil 4.32. 8-Bit 32 Kbyte FIFO Hafıza

6.Adım: Sıra geldi giriş ve çıkış pinlerinin bağlanmasına. Sembolün sol kısmında gösterilen pinler girişleri sağ kısımda gösterin pinler ise çıkışlar temsil etmektedir. Pin bağlantıları ve açıklamaları tablo 4.4’ de verilmiştir.

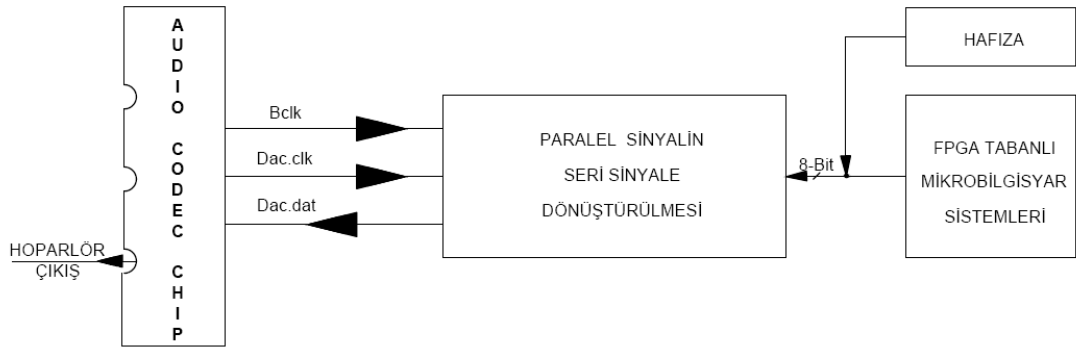
Tablo 4.4: FIFO Hafıza Pinleri ve Açıklaması

Pin Adı	Görevi	Nereye Bağlanacağı
Data[7..0]	Hafizaya Gelen Veriler	Seri Parellel Dönüştürücünün Çıkışına Bağlanacak
Wrreg	Hafizaya Yazma Etkin	İstenen Bir Butona Bağlanabilir.
Rdreg	Hafizadan Okuma Etkin	İstenen Bir Butona Bağlanabilir.
Clock	Kayıt Frekansı	Adc.Lr Clk
Q[7..0]	Çıkışa Aktarılan Datalar.	Seri Parellel Dönüştürücünün Girişine Bağlanacak.

Full	Hafıza Dolu.	İstenen Bir Lede Bağlanır.
Empty	Hafıza Boş.	İstenen Bir Lede Bağlanır.
Usedw[14..0]	Hafızanın Doluluk Oranını	İstenen 14 lede Bağlanır.

4.5. Paralel Bilginin Seri Bilgiye Dönüştürülmesi

Tasarlanan sistemde hafıza çıkışı 8-bittir. Hafızadaki dijital verilerin analog ses sinyallerine dönüştürülebilmesi için hafızadaki 8-bit bilginin seri bilgiye dönüştürülerek WM8731 audio kodek entegresinin Dac.dat girişine uygulanması gerekir. Hafızadaki 8-bit paralel bilginin okunarak seri sinyale dönüştürülmesi ve bu bilginin WM8731 audio kodek entegresine gönderilmesine ait blok şema şekil 4.33’ de verilmiştir.

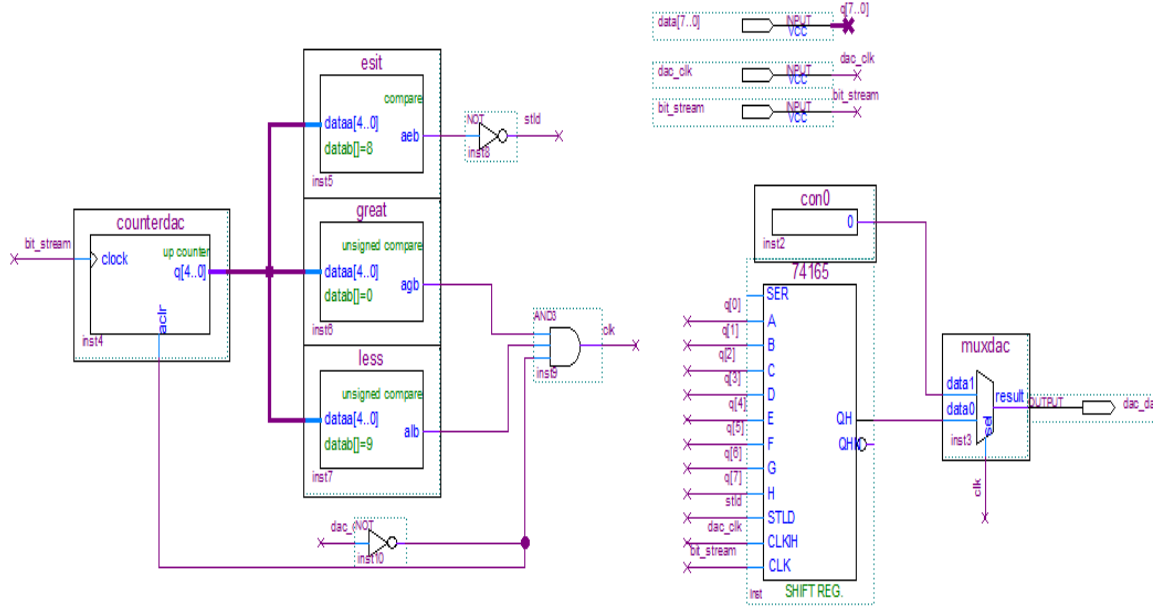


Şekil 4.33. Paralel Bilginin Seri Bilgiye Dönüştürülmesi Blok Şeması

Tasarlanan devrede Dac.clk frekansı 8 khz olarak seçilmiştir. Bclk ise sabit 1.5 Mhz dir. Dac.clk sinyali sıfır konumunda iken Bclk sinyalinin ilk 9 saykılında input olarak kullanılan paralel girişler sıra ile okunmuştur. Dac.clk sinyali 1 konumuna geçtiğinde ise Bclk sinyalinin ilk 9 saykılında daha önce okunan bilgiler Dac.dat sinyaline yansıtılmıştır. Dokuz saykıl kullanılmasını nedeni paralel girişler okunurken ve okunan değerler Dac.dat sinyaline aktarılırken ilk saykılta herhangi bir işlem yapılmadığından 8-bit bilgi için dokuz saykıl kullanılmıştır. Bunun nedeni I2S iletişim formatının kullanılıyor olmasından kaynaklanmaktadır. Bu durum bölüm 2’de anlatılmış olup şekil 2.16’da gösterilmiştir.

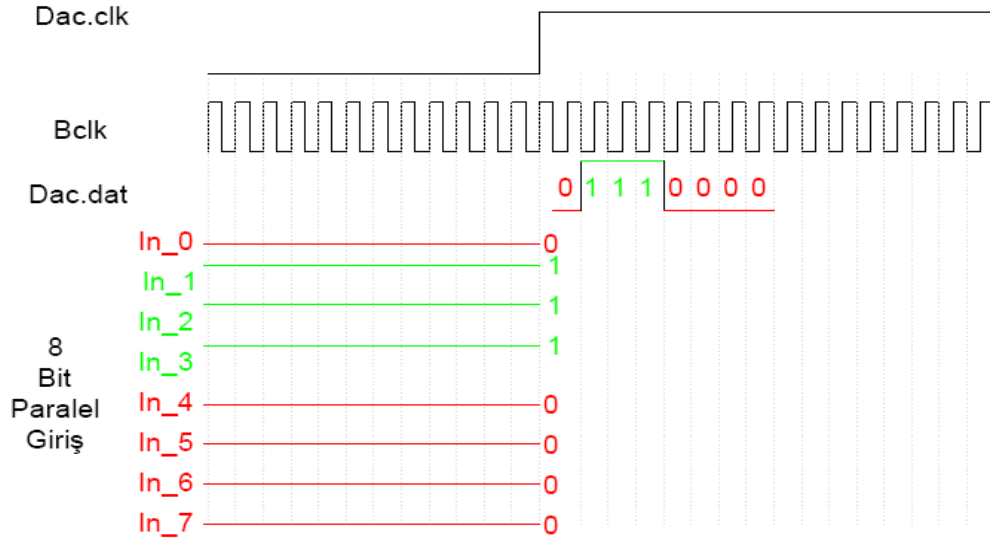
Paralel sinyallerin seri sinyale dönüştürülmesi işleminde 74HC165 entegresinin mimari yapısından faydalanılmıştır. Bclk saykılarını saydırmak için bir counter ve

iki ila dokuzuncu saykılar arasında işlem yapmak için karşılaştırıcılar kullanılmıştır. Bu işlem için tasarlanan devreye ait ekran görüntüsü şekil 4.34’de verilmiştir. Şekil 4.34’ e dikkat edilirse 74165 entegresinin çıkışına bağlanan 2 girişli multiplexer vasıtası ile paralel girişler çıkışa aktarıldıktan sonra dac_clk sinyali 1(high) olduğunda çıkış sıfır olur.



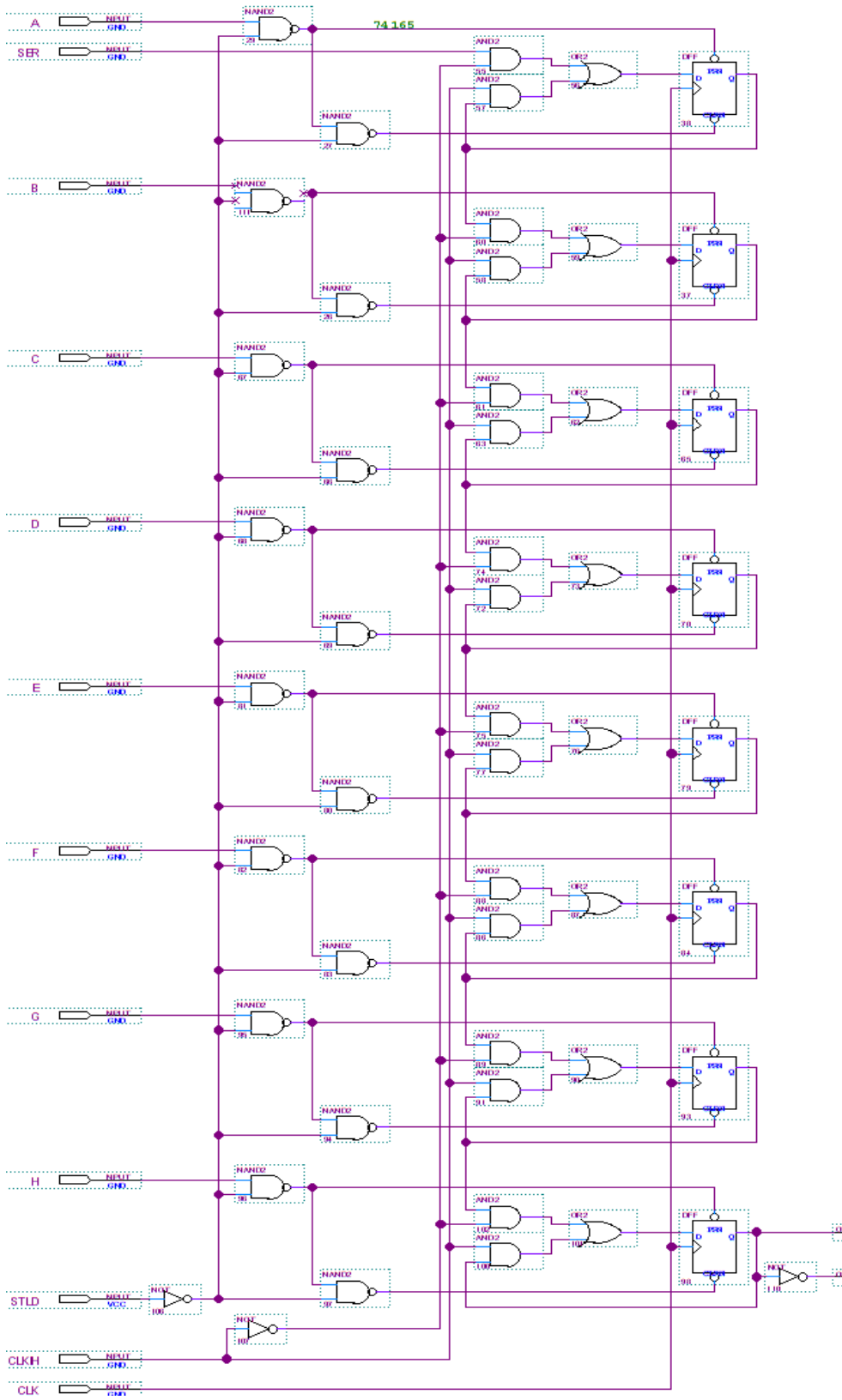
Şekil 4.34. Paralel Seri Dönüştürücü İçin Tasarlanan Devre.

Paralel bilgiyi seri bilgiye dönüştürmek için tasarlanan devreye ait sinyal şekilleri şekil 4.35’de verilmiştir.



Şekil 4.35. Paralel Seri Dönüştürücü Sinyal Şekilleri

Paralel bilgiyi seri bilgiye dönüştürmek için tasarlanan devrede kullanılan 74HC165 entegresini iç yapısı şekil 4.36.'de verilmiştir.

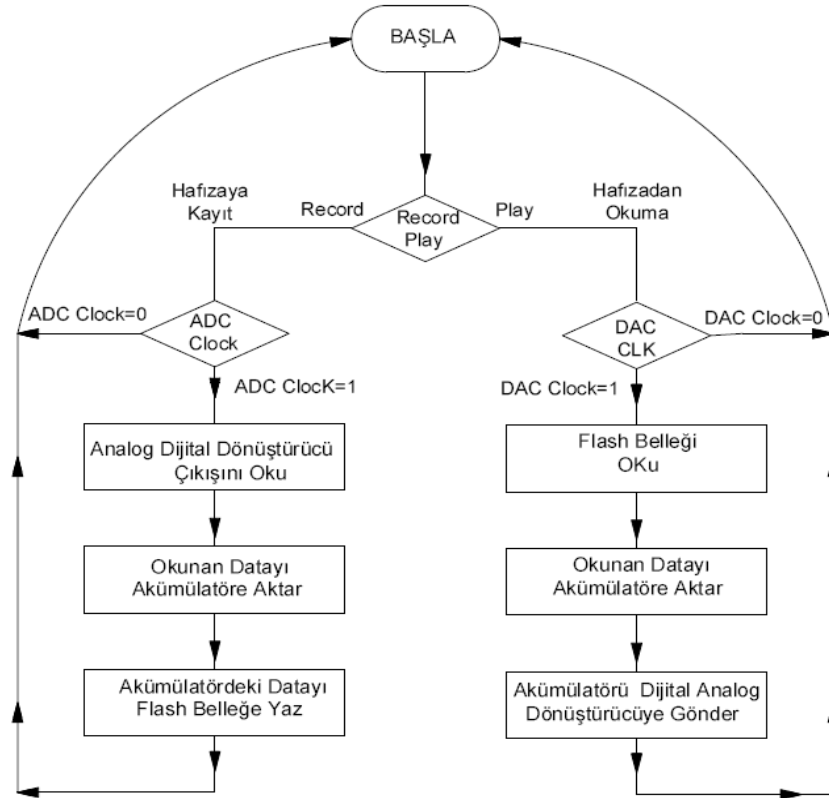


Şekil 4.36. 74HC165 Entegresi İç Yapısı.

4.6. Tasarlanan Analog Haberleşme Arayüzünün BZK.SAU.FPGA Mikrobilgisayar Sistemine Uyarlanması

Bu kısımda yapılandırılabilir donanım üzerindeki mikrobilgisayar sistemleri için tasarlanan analog haberleşme arayüzünün BZK.SAU.FPGA mikrobilgisayar sistemine uyarlanması hakkında bilgi verilecektir.

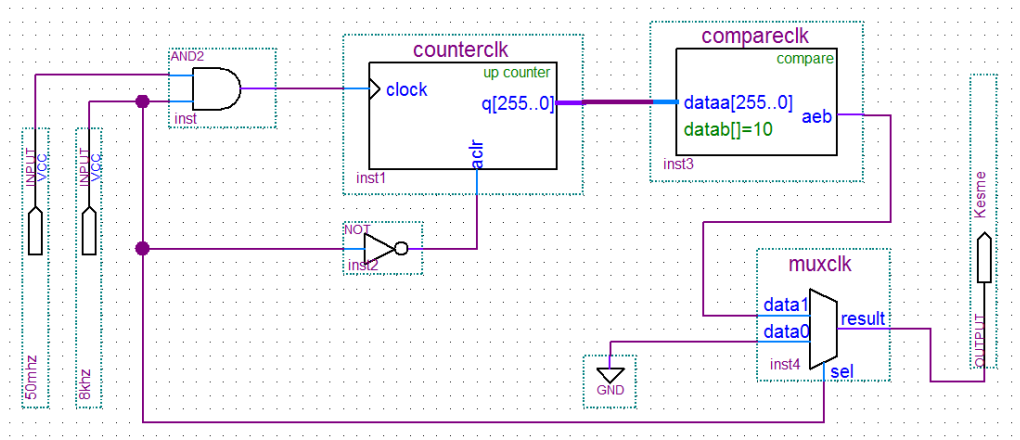
Bu kısım iki aşamalı olarak tasarlanmıştır. Birinci aşamada analog haberleşme arayüzünden gelen 8-bit dijital data BZK.SAU.FPGA mikrobilgisayar sistemi tarafından alınarak hafızada saklanırken ikinci aşamada BZK.SAU.FPGA mikrobilgisayar hafızasından alınan dijital dataların analog haberleşme arayüzüne gönderilmesi işlemi gerçekleştirilmiştir. Aşamaların detaylarına geçmeden sistem akış şemasını oluşturmak, hem kendi işimizi kolaylaştıracak hem de yaptığımız işin anlaşılmasını kolaylaştıracaktır. Şekil 4.37’de sistem akış şeması verilmiştir.



Şekil 4.37. Sistem Akış Şeması.

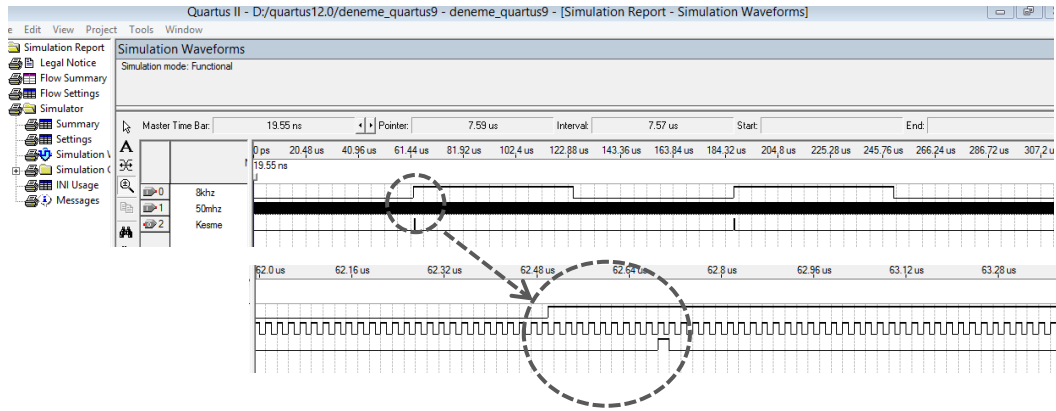
4.6.1. Mikrobilgisayar Sistemleri İçin Kesme Sinyalinin Üretilmesi

Tasarlanan analog haberleşme arayüzü ile analog girişten alınan analog sinyal dijital 8-bit paralel bilgiye dönüştürülerek mikrobilgisayar sistemlerinin işleyebileceği bir forma dönüştürülür. Mikrobilgisayar sisteminin işleyebileceği veriler hazırlandıktan sonra analog haberleşme arayüzü mikrobilgisayar sistemine hazırlanan verileri okuması için bir kesme sinyali gönderir. Kesme sinyali üretilmez ise mikrobilgisayar sistemi sürekli olarak analog girişleri kontrol etmek zorunda kalır. Kesme sinyali üretmek için kullanılan devre şekil 4.38’de verilmiştir.



Şekil 4.38. Kesme Sinyali Üretme Devresi

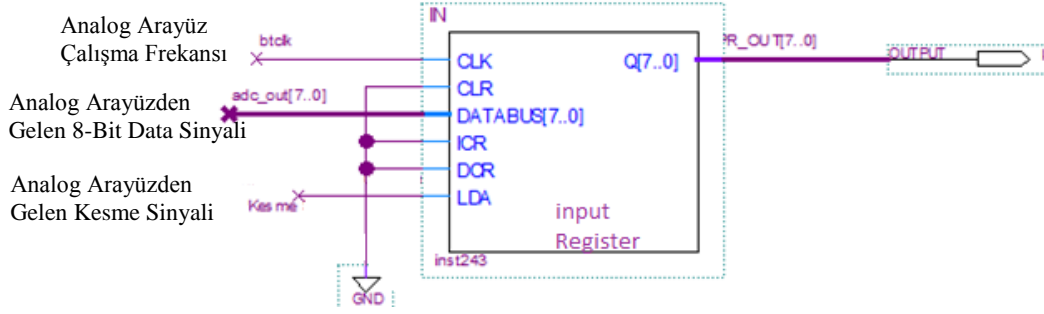
Şekil 4.38’de verilen kesme devresinin simülasyonu yapıldığında şekil 4.39’da verilen kesme sinyali elde edilir. 50Mhz frekanslı saat darbesi geliştirme kiti tarafından üretilen saat darbesini, 8KHz frekanslı saat darbesi de adc.clk sinyalini ifade eder.



Şekil 4.39. Üretilen Kesme sinyali

4.6.2. Tasarılan Analog Haberleşme Arayüzünün BZK.SAU.FPGA Mikrobilgisayar Sistemine Donanımsal Olarak Bağlanması

Üretilen kesme sinyali, 8-bit data sinyali ve 1,5Mhzlik çalışma frekansı şekil 4.40'da gösterildiği gibi BZK.SAU.FPGA Mikrobilgisayar sistemine donanımsal olarak bağlanır.



Şekil 4.40. Analog Arayüzden Gelen Sinyallerin BZK.SAU.FPGA Mikrobilgisayar Sistemine Donanımsal Olarak Bağlanması.

4.6.3. Tasarılan Analog Haberleşme Arayüzünün BZK.SAU.FPGA Mikrobilgisayar Sistemine Yazılımsal Olarak Bağlanması

Tasarlanan analog haberleşme arayüzünden gelen sinyaller BZK.SAU.FPGA mikrobilgisayar sistemine bağlandıktan sonra mikrobilgisayar sisteminin input registerdaki dataları alması ve işleyebilmesi için gerekli program kodları yazılmıştır. BZK.SAU.FPGA mikrobilgisayar sistemi 59 komuttan oluşan komut setinden dokuz adet komut kullanılmıştır. Bu komutlar ve kullanım şekilleri aşağıda anlatılmıştır[3,5].

IN Komutu: IN komutu BZK.SAU.FPGA mikrobilgisayar sisteminin input registeri okumasını sağlayarak sisteme veri girişini sağlayan komuttur. IN komutu ile alınan bilgiler akümülatöre atılır.

STA Komutu: Akümülatördeki veriyi alarak istenilen adrese yazar. STA \$0030H komut satırı akümülatördeki veriyi 0030H adresine yazar.

LDA Komutu: İstenilen adresteki veriyi akümülatöre yazar. LDA \$0030H komut satırı 0030H adresindeki veriyi akümülatöre yazar.

CMP Komutu: Karşılaştırma komutudur. Akümülatördeki veri ile istenilen veriyi karşılaştırır. CMP #BC00H akümülatördeki veri ile BC00H verisini karşılaştırır.

BZR Komutu: CMP komutu ile yapılan karşılaştırmanın sonucuna göre dallanma yapar. BZR B komut satırı karşılaştırmanın sonucunda veriler aynı değerde ise B etiketine programı dallandırır. Değilse bir alt satırdan program akışı devam eder.

BRA Komutu: Koşulsuz dallanma yapar. BRA A komut satırına geldiğinde Program A etiketine dallanır.

HLT Komutu: Kesmeleri kontrol et. Kesme varsa kesmeye git.

INCR Komutu: Akümülatör içeriğini bir artır.

OUT Komutu: OUT komutu BZK.SAU.FPGA mikrobilgisayar sisteminin output registeri okumasını sağlayarak sistemden çevre birimlere veri aktarılmasını sağlayan komuttur.

Kullanılan komutların açıklamaları da verildikten sonra yazılan program kodu ve açıklaması aşağıda Tablo 4.1.'de verilmiştir.

Tablo 4.5. Program Kodu.

Satır No	Etiket	Program Kodu	Açıklama
1	A	IN	İnput registeri oku
2		STA@0030h	Okunan değeri 30hex adresinde bulunan monitör başlangıç adresi(A000)'ya yaz
3		LDA\$0030h	30 hex adresinde bulunan adres bilgisini akümülatöre aktar
4		INCR	Akümlatörü (adresi) bir artır
5		STA \$0030h	Akümlatör (adres) bilgisini 30hex adresine yaz
6		CMP #BC00h	Akümlatör içeriği BC00hex(monitör bitiş adresi) ise
7		BZR B	B' ye dallan
8		HLT	Kesmeleri kontrol et
9		BRA A	A'ya git
10	B	LDA #A000	A000h adresindeki datayı akümülatöre aktar
11		STA \$0030h	Akümlatör değerini 30hex adresine yaz
12		LDA @0030h	0030h adresinde bulunana adres bilgisini akümülatöre aktar
13		OUT	Akümlatörü output registera aktar.

14		LDA \$0030H	0030H adresindeki veriyi akümülatöre aktar
15		INCR	Akümülatörü bir artır
16		STA\$0030	Akümülatör değerini 30hex adresine yaz
17		CMP #BC00h	Akümülatör içeriği BC00(monitör bitiş adresi) hex ise
18		BZR C	C' ye dallan
19		BRA B	B'ye git
20	C	END	Program sonu

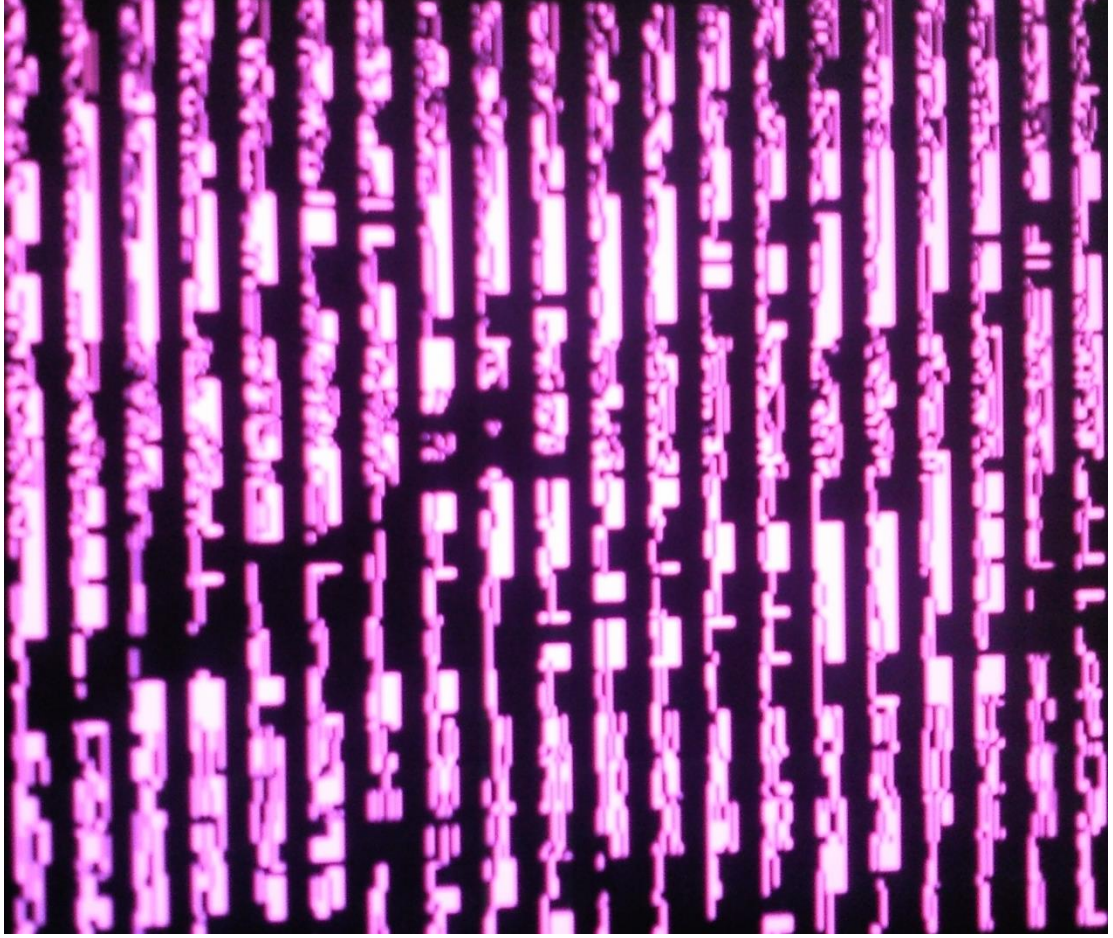
Analog haberleşme arayüzü ile BZK.SAU.FPGA Mikrobilgisayar sisteminin donanımsal ve yazılımsal olarak bağlanması ile iki sistem birbiri ile haberleşebilir duruma getirilmiştir.

5. SONUÇLAR VE ÖNERİLER

IEEE bilgisayar topluluğu ve bilgisayar bilimleri müfredatı düzenleme birliği tarafından tavsiye edilen müfredat içeriğinde yer alan “Tüm öğrenciler pratik çalışma ile teorik bilgiyi bir bütün halinde öğrenmek zorundadırlar” ifadesi derslerin işlenmesinde pratik çalışmaların gerekliliğine vurgu yapmaktadır[1]. Teorik bilgilerin pratik uygulamalara dönüştürülmesi hem yüksek maliyet hem de uzun tasarım sürecinden dolayı her zaman mümkün olmayabilir. ‘Gömülü Sistemler’, ‘Bilgisayar Mimarisi’ ve ‘İşletim Sistemi’ gibi derslerde pratik uygulama eksikliği FPGA geliştirme kartları kullanılarak minimal seviyeye indirilebilir. FPGA geliştirme kartları ile öğrenciler ve dijital sistem geliştiriciler tasarımlarının çalışmasını sanal ortam yerine gerçek ortam şartlarında görme imkânına sahip olurlar. Yeniden yapılandırılabilir donanımlar olan FPGA’ ler; tekrar tekrar programlanabilmeleri sayesinde öğrencilere, eğitim kurumlarına ve firmalara ek bir maliyet getirmeden mevcut kitler üzerinde değişik uygulamalar yapma imkânı sunar.

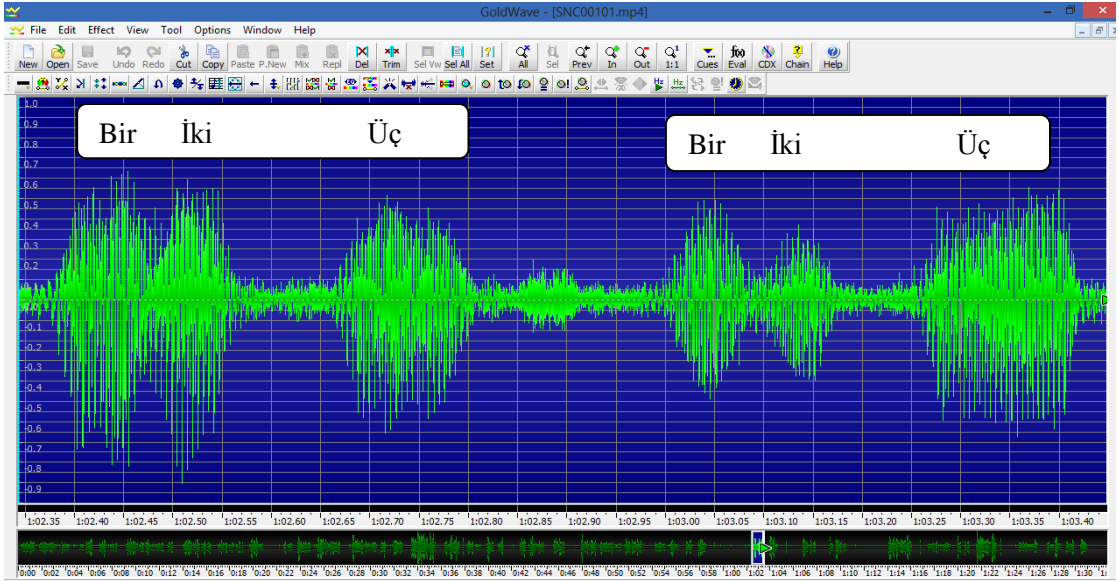
Bu tez çalışmasında yapılandırılabilir donanım üzerindeki sistemler ve eğitimsel amaçlı mikrobilgisayar sistemleri için analog haberleşme ara yüz tasarımı gerçekleştirilmiştir. Tasarımı gerçekleştirilen analog haberleşme arayüzü ile mikrobilgisayar sistemlerine analog sinyalleri kontrol ve işleyebilme özelliği kazandırılarak analog giriş çıkışa sahip çevre birimlerinin kontrol edilmesi sağlanmıştır. Tasarımı gerçekleştirilen analog haberleşme arayüzü kullanılarak sıcaklık, basınç, nem gibi analog fiziksel değişkenler kontrol edilebilir.

Tasarlanan analog haberleşme arayüzü kullanılarak BZK.SAU.FPGA mikrobilgisayar sisteminin analog sinyalleri kontrol etme ve işlemesi özelliği kazandırılmıştır. Tasarlanan analog haberleşme arayüzünden alınan dijital veriler BZK.SAU.FPGA Mikrobilgisayar sisteminde hem şekil 5.1’de gösterildiği gibi monitöre yazdırılmıştır.

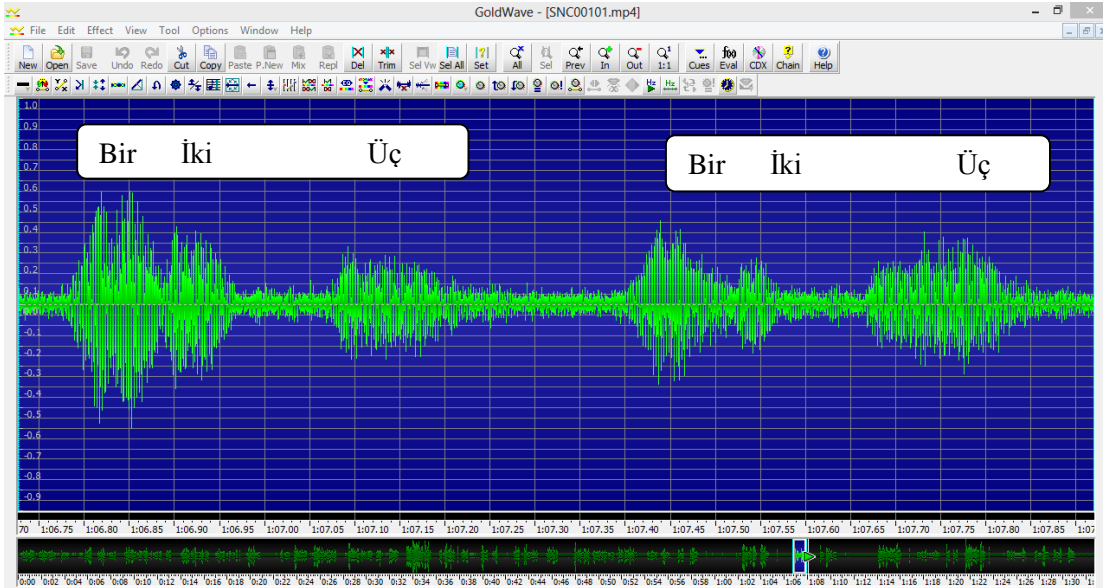


Şekil 5.1. Analog Haberleşme Arayüzünden Gelen Dijital Verilerin BZK.SAU.FPGA Mikrobilgisayar Sisteminde Monitöre Yazdırılması.

Tasarlanan analog haberleşme arayüzünden alınana dijital veriler BZK.SAU.FPGA Mikrobilgisayar sistemine kayıt edilmiştir. Şekil 5.2' de verilen analog ses giriş sinyali BZK.SAU.FPGA Mikrobilgisayar sisteminde hafızaya kayıt edilmiştir. Daha sonra mikrobilgisayar sisteminin hafızasındaki dijital veriler tasarlanan arayüz ile alınarak analog sinyale çevrilmiş ve çıkışa gönderilmiştir. Hafızadaki dijital veriler analog sinyale dönüştürüldüğünde ise şekil 5.3' de verilen ses çıkış sinyalleri elde edilmiştir



Şekil 5.2: Analog Giriş Sinyali.



Şekil 5.3: Analog Çıkış Sinyali.

Şekil 5.2 ve şekil 5.3’ de görüldüğü üzere analog giriş sinyali ile analog çıkış sinyali birbirine benzemekte fakat sinyal genlikleri farklılık göstermektedir. Bunun nedeni data genişliğinin 16 bit yerine 8 bit seçilmesinden kaynaklandığı düşünülmektedir.

Bu tez çalışmasında temel amaç eğitimsel olup öğrencilere dijital sistemler ile analog çevre birimleri arasındaki haberleşmenin nasıl gerçekleştiğinin kavratılması amaç edinilmiştir. Bu nedenle tasarlanan sistemin tamamının öğrencilerin kolayca anlayabileceği temel yapıda olmasına, küçük modüllerden oluşmasına, modüllerin çalışmasının ayrı ayrı gözlenebilir olmasına, tasarlanan sistemin müdahale edilebilir ve geliştirilebilir olmasına gayret edilmiştir.

Eğitimsel çalışmalarda soyut kavramların öğrenciler tarafından anlaşılması daha zor olmaktadır. Bu çalışmada tümevarım metodu kullanılarak modüler bir yapı oluşturulmuştur. Herbir modül bir basamak yapılarak anlaşılması ve yapılması zor gibi algılanan analog sinyallerin dijital sistemlerde işlenmesi ve kontrol edilmesi işlemi tüm öğrencilerin rahatlıkla kavrayabilecekleri bir forma indirgenmiştir. Tasarımı yapılan analog haberleşme arayüzü modüler yapıda olduğu gibi her bir modül de kendi içerisinde alt modüllerden meydana gelmiştir. Modüler yapı ile öğrenciler sistemin karmaşasında boğulmadan komple bir sistem tasarımı yerine istedikleri bölümleri kendi tasarladıkları devre ile değiştirerek sistemin nasıl çalıştığını gözlemleyebileceklerdir. Tasarlana sistemin modüler yapıda olmasının en büyük avantajlarından biriside öğrencileri sistemin karmaşasından kurtararak onlara özgüven kazandırmasıdır.

Bu tez çalışmasında tasarımı yapılan analog haberleşme arayüzü ile yapılandırılabilir donanım üzerindeki (FPGA) mikrobilgisayar sistemlerinin analog sinyalleri işlemedeki eksikliklerinin giderilmesi amaçlanmıştır. Tasarlanan sistem altı modülden meydana gelmiştir. Birinci modülde; sistemin çalışması için gerekli uygun frekanslı saat darbeleri üretilmektedir. İkinci modülde; yapılandırılabilir donanım ile analog çevre birimlerini haberleştirmek için standartlara uygun seri I2C haberleşme protokolü tasarlanmıştır. Üçüncü modülde; analog girişten seri olarak alınan datalar 8-bit paralel sinyale dönüştürülmüştür. Dördüncü modülde; 8-bit paralel veriler hafızaya kayıt edilmiştir. Beşinci modülde; hafızadan alınan 8-bit paralel veriler seri sinyale dönüştürülerek analog çıkışa aktarılmıştır. Altıncı modülde; tasarlanan analog haberleşme ara yüzü BZK.SAU.FPGA mikrobilgisayar sistemine uyarlanmıştır.

Gerçekleştirilen analog arayüz uygulaması ile öğrencilerin, bilgisayarlar ile analog çevre birimleri arasındaki çalışma akışını daha kolay kavrayabilmeleri, motivasyon ve özgüven açısından da üst seviyede kalmaları hedeflenmiştir.

Bundan sonraki çalışmalarda geliştirilen analog haberleşme arayüzü kullanılarak BZK.SAU.FPGA mikrobilgisayar sistemi ile sıcaklık, basınç, nem, ses gibi analog sinyaller ile kontrol edilebilen sistem tasarımları yapılabilir. Yine geliştirilen analog haberleşme arayüzü kullanılarak analog çıkış veren kameradan gelen video sinyali sayısal sinyale dönüştürülerek kameradan gelen veri ile hafızaya önceden kayıt edilen veriler yapay sinir ağları kullanılarak karşılaştırılır bu yöntem ile yüz tanıma veya plaka okuma gibi görüntü işleme işlemleri de gerçekleştirilebilir.

KAYNAKLAR

1. The Joint Task Force on Computing Curricula, IEEE Computer Society and Association for Computing Machinery, Computin Curricula, (2001).
2. Özcan Demirel, ÖĞRETME SANATI. Pegem Yayıncılık, Ankara, 2003, s:51.
3. OZTEKIN, H., Eğitim Amaçlı Yapılandırılabilir Modüler Donanım Üzerine Gömülü İşletim Sistemi Tasarımı, Doktora, Sakarya Üniversitesi, Bilgisayar ve Bilişim Mühendisliği, 2012.
4. <http://www.fpganedir.com/FPGA/index.php> (Erişim Tarihi: 09/06/2014).
5. ÖLMEZ, E., Fpga Tabanlı Mikrobilgisayar Mimarisi Kullanılarak Dc Motor Sürücü Tasarımı Ve Uygulaması, Y. Lisans, Bozok Üniversitesi, Mekatronik Mühendisliği, 2012.
6. <http://tr.wikipedia.org/wiki/Fpga>(Erişim Tarihi:20/07/2014).
7. OZTEKIN, H., Bilgisayar Mimarisi Simülatörü Tasarımı, Y. Lisans, Sakarya Üniversitesi, Bilgisayar ve Bilişim Mühendisliği, 2009.
8. TRIPATHI, J. N., et al., Soft-Reconfiguration Management for Operating Systems with Multiprocessor Architecture, International Conference on Advanced Computing and Communications, Guwahati-Assam, (2007) pp:573-577.
9. LU, Z., ZHANG, X., SUN, C., An Embedded System with uClinux based on FPGA, IEEE Pacific-Asia Workshop on Computational Intelligence and Industrial Application, Wuhan, (2008), Vol. 2, pp: 691–694.
10. GARDEZI, A. F., AHSAN, M. N., MASUD, S., Framework for Performance Analysis of RTOS-Enabled Embedded Systems on FPGA, International Symposium on Performance Evaluation of Computer & Telecommunication Systems, Istanbul-TURKEY, (2009), Vol. 41,pp:35-40.
11. PELLIZZONI, R., CACCAMO, M., Real-Time Management of Hardware and Software Tasks for FPGA-based Embedded Systems, IEEE Transactions on Computers, Vol. 56, No. 12, pp. 1666-1680, Dec. 2007.
12. HUERTA, P., et al., Operating System for Symmetric Multiprocessors on FPGA, International Conference on Reconfigurable Computing and FPGAs, Cancun, (2008) pp:157-162.
13. HAN, W., et al., Mapping Real Time Operating System on Reconfigurable Instruction Cell Based Architectures, 15th Annual IEEE Symposium on

- Field-Programmable Custom Computing Machines, Napa-CA, (2007) pp:301-304.
14. T.C. MEB, Elektrik Elektronik Teknolojisi, Mikrodenetleyiciyle Analog İşlemler, 2012.
 15. Philips Semiconductors, The I2C - Bus Specification, Version 2.1, January 2000.
 16. AKTAŞ, F., Kablosuz Algılayıcı/Eyleyici Ağlarla Denetim Sistemi Tasarımı, Y. Lisans, Kocaeli Üniversitesi, Fen Bilimleri Enstitüsü, Elektronik ve Bilgisayar Eğitimi, 2012.
 17. ATMACA, S., I2C Bus Seri İletişim Protokolü İçin Veri İzleme Sistemi, Y. Lisans, Sakarya Üniversitesi, Elektronik ve Bilgisayar Eğitimi, 2002.
 18. KILINÇ, M., Uzaktan Erişimli Sistem Odası Kontrolü, Y. Lisans, Bozok Üniversitesi, Mekatronik Mühendisliği, 2012.
 19. HWANG, E., Implementing an I2C Master Bus Controller in a FPGA, 2008.
 20. GÜNEROĞLU, A., Fotovoltaik Sistemlerde Fpga Kullanımı, Y. Lisans, Kocaeli Üniversitesi, Elektrik Eğitimi, 2008.
 21. B., Çelik., Fpga Tabanlı Robotik Kontrol, Y. Lisans, Uludağ Üniversitesi, Elektronik-Haberleşme Eğitimi Programı, 2013.
 22. Peker, Z., Fpga İle Veri Gizleme Uygulamaları, Y. Lisans, Uludağ Üniversitesi, Elektronik Mühendisliği Anabilim Dalı, 2013.
 23. http://web.itu.edu.tr/orencik/BilgMimYenYakl2007/Mehmet_Aktas/FPGA_Mimarisi_Rapor.pdf (Erişim Tarihi: 11/06/2014).
 24. PEDRONI, V.A., Circuit Design With VHDL, MIT Press, Cambridge Massachusetts, London, England.
 25. PERRY, D., L., VHDL Programming by Example , Fourth Edition, McGraw-Hill.
 26. BHASKER, J., A VHDL Primer, American Telephone and Telegraph Company Bell Laboratories Division, P T R Prentice Hall, Englewood Cliffs, New Jersey 07632.
 27. Chu,P.,P., Embedded Sopc Desigh With Nios II Processor And Vhdl Examples, Cleveland State University, A John Wiley & Sons, Inc., Publication, 2011.

28. ASPENDEN, P., j., VHDL Tutorial, Eda Consultant, Ashenden Designs Pty. Ltd., Elsevier Science (USA), 2004.
29. AYATA, F., Fpga Tabanlı Üç Eksenli Robot Kol Kontrolü, Y. Lisans, Süleyman Demirel Üniversitesi, Elektronik ve Bilgisayar Eğitimi Ana Bilim Dalı, 2013.
30. DE2 Development and Education Board User Manual, Version 1.4, Altera Corporation, 2006.
31. WM8731/WM8731L, April 2004 rev 3.4, Wolfson Mikroelektronik Plc.
32. Quartus II Subscription License, 2100@194.27.212.50
33. Quartus II Introduction Using Schematic Designs For Quartus II 12.0, Altera Corporation, 2012.
34. H. Oztekin, F. Temurtas, E. Olmez, and A. Gulbag "FPGA-Based Flash Memory Controller for BZK.SAU.FPGA10.1 Microcomputer Architecture Design as an Educational Tool" International Journal of Computer and Communication Engineering, Vol. 1, No. 3, pp. 241-245, September 2012.
35. H. Oztekin, F. Temurtas and A. Gulbag, "A modular approach to VGA Monitor Controller for BZK.SAU.FPGA10.1 microcomputer architecture design," Int. Proc. Computer Science and Information Technology(ICICA 12), vol. 24, pp. 27-31, February 2012.
36. H. Oztekin, F. Temurtas and A. Gulbag, "BZK.SAU.FPGA10.1: A modular approach to FPGA-based micro computer architecture design for educational purpose," Comput. Appl. Eng. Educ.. doi: 10.1002/cae.20553, in press.
37. H. Oztekin, F. Temurtas and A. Gulbag, "BZK.SAU: Implementing a hardware and software-based Computer Architecture simulator for educational purpose," Int. Conf. Computer Design and Applications (ICCD 10, vol.4, pp.V4-90-V4-97.

ÖZGEÇMİŞ

1980 yılında Kırşehir’de doğan Resul KÖKSAL, ilk, orta ve lise öğrenimini Kırşehir’de tamamlamıştır. 1998 yılında kazandığı Gazi Üniversitesi Teknik Eğitim Fakültesi Elektrik Öğretmenliği bölümünü 2002 yılında başarıyla bitirmiştir.

2002 yılında T.C. Başbakanlık Devlet İstatistik Kurumunda iş hayatına başlamış ve 2004 yılında Milli Eğitim Bakanlığına geçiş yapmıştır. Halen Yerköy Teknik ve Endüstri Meslek Lisesinde elektrik elektronik teknolojisi alan öğretmeni olarak çalışmakta olan Resul KÖKSAL evli ve bir çocuk babasıdır.

İletişim Bilgileri

Adres: Yerköy Teknik ve Endüstri Meslek Lisesi, Yerköy/YOZGAT.

Telefon: (505) 292 42 58

E-posta: resulkoksal@hotmail.com