



Veri İletişimi Kodlama Tekniklerinin FPGA Üzerinde Gerçekleştirilmesi

Cem Deniz KUMRAL¹, Mevlüt ERSOY²

¹Bilgisayar Mühendisliği, Süleyman Demirel Üniversitesi, Isparta, Türkiye

²Bilgisayar Mühendisliği, Süleyman Demirel Üniversitesi, Isparta, Türkiye
cemkumral34@gmail.com, mevlutersoy@sdu.edu.tr

Özet— Teknoloji altyapısı bulunan her geliştirme ve oluşturulan her sistemde veri iletişimi gerçekleşmektedir. Veri iletişiminin doğru ve uygun altyapılar kullanılarak sağlanması önem taşımaktadır. Hat kodlaması; fiziksel kanalın ve alıcı ekipmanın belirli özellikler için optimal olarak ayarlanmış olan genlik ve zaman-ayrık dijital sinyallerini temsil etmek üzere gerçekleştirilen bir işlemdir. İletilecek olan sinyali iletim ortamının özelliklerine uyumlu olacak biçimlere çevirip, sinyal bozucu etkenlerden daha az etkilenebilecek biçimde ve en az bant genişliği kullanımı sağlayacak şekilde değiştiren sinyal işleme uygulamasıdır. Sistemlerde kullanmak üzere çeşitli hat kodlama tekniği bulunmaktadır. Hangi kodlama tekniğinin kullanılacağı, DC seviyesinin, PSD (Güç Spektral Yoğunluğu) durumunun, bant genişliği gerekliliklerinin, bit hata oranı performansının, saat sinyalinin geri kazanım kolaylığının veya içsel algılama özelliğinin varlığına veya yokluğuna bağlıdır. Bu çalışmada, Manchester, Differential Manchester, NRZ (Non-Return Zero, Sıfıra Dönmeyen), NRZ-Level ve NRZ-Invert hat kodlama tekniklerinin, en yaygın donanım tanımlama dili olan Very High Speed Integrated Circuit Hardware Description Language (VHDL) kullanılarak FPGA kartı üzerinde gerçekleştirilmesi hedeflenmiştir. VHDL kodlamaları VIVADO ortamında gerçekleştirilmiştir. Gerçekleştirilen uygulamanın doğru sonuçlar verip vermediğini kontrol etmek adına, elde edilen tümleşik devreye 8 bitlik veri girişi üzerinden çalışma sağlayan bir simülasyon kodu yazılmıştır. Her sinyal için 100 nanosaniye (ns) girdi süresi tanımlanmış ve saat sinyali 50 ns aralıklarla tetiklenmiştir. Simülasyonun çalıştırılması sonucunda her bir kodlama tekniği için oluşan çıkış sinyalleri, kodlama tekniklerinin kuralları doğrultusunda incelenerek sistemin doğruluğu sağlanmıştır.

Anahtar Kelimeler— Veri iletişimi, Sayısal haberleşme, Kodlama Teknikleri, FPGA, VHDL

Implementation of Data Communication Coding Techniques on FPGA

Abstract— Data communication takes place in every development and every system created with a technology infrastructure. It is important to ensure data communication using correct and appropriate infrastructures. Line coding; It is a process performed to represent amplitude and time-discrete digital signals that are optimally tuned for certain characteristics of the physical channel and receiving equipment. It is a signal processing application that converts the signal to be transmitted into forms that are compatible with the characteristics of the transmission medium and changes it in a way that is less affected by signal interfering factors and provides the least bandwidth usage. There are various line coding techniques for use in systems. Which coding technique to use depends on the DC level, PSD (Power Spectral Density) condition, bandwidth requirements, bit error rate performance, ease of recovery of the clock signal, or the presence or absence of intrinsic sensing capability. In this study, it is aimed to implement Manchester, Differential Manchester, NRZ (Non-Return Zero), NRZ-Level and NRZ-Invert line coding techniques on FPGA board using Very High Speed Integrated Circuit Hardware Description Language (VHDL), the most common hardware description language. VHDL coding was carried out in the VIVADO development platform. In order to check whether the implemented application gives correct results, a simulation code was written to the integrated circuit that provides operation over 8-bit data input. For each signal, an input time of 100 nanoseconds (ns) was defined and the clock signal was triggered at 50 ns intervals. As a result of running the simulation, the output signals generated for each coding technique were analysed in accordance with the rules of the coding technique and the accuracy of the system was ensured.

Keywords— Data communication, Digital communication, Coding Techniques, FPGA, VHDL

1. GİRİŞ (INTRODUCTION)

Bilginin iki nokta arasında iletilmesi için analog veya sayısal sinyale çevrilmesi gerekir. Sayısal-sayısal çevirmede sayısal veri sayısal sinyale dönüştürülür [1, 2]. Analog-sayısal çevirmede ise analog veri sayısal sinyale dönüştürülür. Çevirme işleminden elde edilen sinyal paralel veya seri olarak iki nokta arasında iletilir. Her dijital iletimde, verilerin modülasyondan önce yeniden biçimlendirilmesi gerekir [3]. Modülasyon, sinyalin yüksek frekanslı sinüsoid ile karıştırılarak yeni bir sinyal üretme tekniğidir. İletilen sinyalin özelliklerini bozmamak, iletim ortamı üzerinde etkili bir şekilde iletişim kurmak adına hat kodlaması adı verilen işlemler gerçekleştirilmektedir [4, 5]. Hat kodlaması; fiziksel kanalın ve alıcı ekipmanın belirli özellikleri için optimal olarak ayarlanmış olan genlik ve zaman-ayrık sinyali tarafından taşınacak olan dijital sinyali temsil etmekten ibarettir. İletilecek olan sinyali iletim ortamının özelliklerine uyumlu olacak biçimlere çevirip, sinyal bozucu etkenlerden daha az etkilenebilecek biçimde ve en az bant genişliği kullanımını sağlayacak şekilde değiştiren sinyal işleme uygulamasıdır. Başka bir deyişle, bir iletim bağlantısındaki dijital verilerin yani lojik 1 ve 0'ların koaksiyel kablo, optik fiber kablo vb. fiziksel ortamlardan geçirilebilecek gerilim ve akım darbesi dizisine çevrildiği iletişim sistemi için kullanılan anahtar yapı bloğudur [6]. Bir blok olarak, dijital verileri bir girdi olarak alır ve yine dijital verileri çıktı olarak geri döndürür. Dolayısıyla hat kodlamaları bir dijital mantık sistemi olarak nitelendirilebilir. Hat kodlaması, sistemlerde kendi kendine saatleme, hizmet içi hata izleme, hattaki işaretlerin büyük bir DC bileşeninin olmasını engelleme, aynı kablo üzerindeki ses devrelerinin karışmasını önleme, frekansın değiştiği ve özellikle art arda gelmiş pek çok sıfırın bulunduğu durumlarda saat işaretini elde etme gibi amaçlara ulaşmak için kullanılmaktadır. Uygulamalarda kullanmak adına çeşitli hat kodlama tekniği mevcuttur. Hangi kodlama tekniğinin kullanılacağı, DC seviyesinin, PSD (Güç Spektral Yoğunluğu) durumunun, bant genişliği gerekliliklerinin, bit hata oranı performansının, saat sinyalinin geri kazanım kolaylığının veya içsel algılama özelliğinin varlığına veya yokluğuna bağlıdır [7, 8].

Son yıllarda, Field Programmable Gate Array (FPGA) entegre devreleri üzerinde farklı sistem türlerinin uygulanmasında yüksek oranda bir büyüme gözlenmiştir. FPGA'lar programlanabilir mantık blokları, bu blok dizisini çevreleyen giriş-çıkış blokları ve ara bağlantılar olmak üzere düzenlenebilir üç ana bölümden oluşmuştur. Paralel işlem yürütebilme yapısına sahip olması, hızlı prototipleme ve uygun maliyetli çözümler sunan tasarım modellerini beraberinde getirmesi nedeniyle sistem mimarilerinde tercih edilmektedir. FPGA'lar sayısal işaret işleme, radar haberleşme sistemi, uzay, savunma sistemleri, ASIC, medikal resimleme, robotik, ses tanıma, şifreleme ve kod çözme gibi birçok alanda kullanılmaktadır [9].

Bu çalışmada, Manchester, Differential Manchester, NRZ, NRZ-Level ve NRZ-Invert hat kodlama tekniklerinin en yaygın donanım tanımlama dili olan Very High Speed Integrated Circuit Hardware Description Language (VHDL) kullanılarak FPGA kartı üzerinde gerçekleştirilmesi anlatılmıştır. Uygulamada ele alınan 5 kodlama tekniği için de gerekli sadeleştirme işlemleri gerçekleştirilerek bu kodlama tekniklerinin lojik yapılarına ulaşılmıştır. Elde edilen bu lojik yapılara göre VIVADO geliştirme ortamında VHDL ile donanım tanımlama işlemi gerçekleştirilmiş ve kodlamaların çalışması için gerekli olan devre şemaları oluşturulmuştur. Gerçekleştirilen uygulamanın doğru sonuçlar verip vermediğini kontrol etmek adına, elde edilen tümleşik devreye 8 bitlik veri girişi üzerinden çalışma sağlayacak bir simülasyon kodu yazılmıştır. Simülasyonun çalıştırılması sonucunda her bir kodlama tekniği için oluşan çıkış sinyalleri incelenerek sistemin doğruluğu sağlanmıştır. Son olarak yazılan VHDL kod FPGA kartı üzerine aktararak sistem kullanımı ve performans analizi yapılmıştır. Sistemin simülasyon sonuçları ve performans analizi sonuçları "TARTIŞMA VE SONUÇLAR" bölümünde verilmiştir.

2. BİLİMSEL YAZIN TARAMASI (SCIENTIFIC LITERATURE REVIEW)

Srinivasan vd. (2023) çalışmalarında, İkili Faz Kaydırmalı Anahtarlama (BPSK)'yı yüksek hızlı bir FPGA platformu üzerinde uygulamışlardır. Uyguladıkları BPSK modülasyon tekniğini, bir IDE (Entegre Geliştirme Ortamı) olan Xilinx Vivado 2021.01 aracındaki Verilog HDL (Donanım Tanımlayıcı Dili) kullanarak geliştirmişlerdir. Geliştirdikleri BPSK kodunu Vivado ortamında simüle etmişler ve daha sonra osiloskopta donanım uygulama sonuçlarına ulaşabilmek için Basys 3 FPGA kartına aktarım yapmışlardır. Geliştirdikleri tasarım ile FPGA platformu üzerinde daha az alan ve daha az güç kullanımı avantajı elde ettiklerini öne sürmüşlerdir [3].

Rajalakshmi ve Kavitha (2023) çalışmalarında, AC şebekede EMI'nin azaltılmasına yönelik üç fazlı gerilim kaynaklı doğrultucuların (VSR) çalıştırılması için doğrudan sıralı yayılma spektrumu (DSSS) ve frekans atlamalı yayılma spektrumu (FHSS) modülasyon tekniklerini önermişlerdir. DSSS modülasyon tekniğinde ilk kez, üç fazlı VSR'de elektromanyetik parazitini (EMI) azaltılması için ikili faz kaydırmalı anahtarlama ve karesel faz kaydırmalı anahtarlama (QPSK) tekniklerini benimsemişlerdir. Önerilen bu modülasyon tekniklerini gerçekleştirmek için uygun maliyetli bir sahada programlanabilir kapı dizisi (FPGA) kullanmışlardır. Her modülasyon tekniği için ISE Xilinx ortamında bir VHDL kodlaması gerçekleştirmişlerdir. Tasarımları Model-Sim aracıyla doğrulamışlar ve ardından doğrultucu için kapı darbeleri üretmek üzere FPGA kartına aktarmışlardır. Önerilen modülasyon tekniklerini geleneksel sinüzoidal darbe genişliği modülasyonu (SPWM) ile karşılaştırmışlardır. FHSS ve QPSK tekniklerinin güç kalitesini iyileştirdiği ve iletilen EMI seviyelerinin SPWM ile karşılaştırıldığında sırasıyla 22,1 ve 12,4 dB μ V azaldığını göstermişlerdir [4].

Kwiatkowski ve Szplet (2020) çalışmalarında, sahada programlanabilir kapı dizisi (FPGA) cihazında çoklu zaman kodlama hatları (TCL'ler) tabanlı zamandan dijitale dönüştürücü (TDC) uygulamasını daha verimli hale getiren iki ilke sunmuşlardır. Sözde bölümlere ayrılmış bir gecikme hattı ve programlanabilir mantık blok elemanlarının etkin kullanımıyla birçok mantıksal kaynaktan tasarruf edilmesi üzerine bir tasarım yapmışlardır. Ayrıca kıyılmış bir TCL ile ölçüm çözünürlüğünü ve hassasiyetini artırmak ve küresel saat ağının eğrilmesinden yararlanmak için bir tasarım yapmışlardır. Her iki tasarımı da Xilinx tarafından 28 nm CMOS prosesinde üretilen bir Kintex-7 FPGA çipinde uygulamışlardır. Oluşturdukları tasarımları yaygın kullanılan "düz" mimari TCL çözümleriyle ve ayarlanmış bir gecikme hattıyla karşılaştırmışlardır. Tek ve çoklu konfigürasyonlarda çalışan dört tasarımın tamamını kaynak kullanımı, ölçüm çözünürlüğü ve hassasiyet açısından test etmişlerdir. Tek kanallı zaman damgalarına dayalı enterpolasyonlu zaman sayacı kullanılarak tasarlanan TDC'ler ile on adete kadar TCL kullanarak 1 ps'lik ortalama çözünürlük ve 4 ps'nin altında hassasiyet elde edildiği sonucuna varmışlardır [5].

Gupta ve Singh (2016) çalışmalarında, Verilog-HDL yazılım platformunda NRZ, RZ ve Biphase gibi farklı çizgi kodlama şemalarının uygulanmasını sunmuşlardır. Verilog-HDL kodunu, MentorGraphics'in ModelSim aracını kullanarak simüle etmişlerdir. Sonuçlar, yazılım arayüzünde çeşitli dalga formları şeklinde gözlemlenmiştir. Sonuç olarak eğitimler üzerinde bir değer yaratan farklı türdeki çizgi kodlamalarını incelemek için kullanıcı dostu ve yazılım odaklı bir yaklaşımı öne sürmüşlerdir [6].

Singh ve Mishra (2014) çalışmalarında, güvenlik, alan optimizasyonu ve çeşitli kanal ortamlarında verimli dijital iletişimi desteklemek amacıyla Xilinx Spartans-6 XC6SLX45 FPGA platformunda VHDL kullanılarak çeşitli çizgi kodlama şemalarının uygulanmasını amaçlamışlardır. Unipolar RZ ve NRZ, Polar RZ ve NRZ, AMI ve Manchester kodlaması, Pseudo Ternary kodlama ve Mark Inversion çizgi kodlama türlerini Xilinx tasarım araçları ile Spartan-6 FPGA kullanarak

modellemişlerdir. Yaptıkları çalışma sonucunda uygulama yapılan kodlama türlerinin simülasyon sonuçlarını ortaya koymuşlardır [7].

Zuo vd. (2013) çalışmalarında, FPGA tabanlı Manchester kodlayıcı ve kod çözücüyü tasarlamış ve gerçekleştirmişlerdir. Bu tasarımı VHDL programlama dilini kullanarak oluşturmuşlardır. İstenilen kodlayıcı ve kod çözücü şemalarını, Altera geliştirme yazılımı Quartus II 8.0'da simüle ederek test sonuçlarını göstermişlerdir. Uygulamanın doğrulama işlemi için Cyclone II EP2C35F672C6 FPGA yongasını kullanarak gerçekleme yapmışlardır. Çalışma sonucunda, tasarım planının Manchester CODEC'i gerçekleştirmek için başarılı olduğunu ve iyi bir stabilite ile güvenilirliğe sahip olduğunu göstermişlerdir [10].

Amrinder vd. (2011) çalışmalarında, güvenlik, alan optimizasyonu ve değişen kanal ortamında iletişimi desteklemesine olanak tanınması nedeniyle, tek bir yonga üzerinde VHDL kullanarak çeşitli çizgi kodlama şemalarının uygulanmasını göstermişlerdir. Kullanılan çizgi kodlama şemaları Unipolar RZ, Polar RZ, NRZ-L, NRZ-I, Manchester, Diferansiyel Manchester, AMI, Psödokterner ve CMI kodlamalarıdır. Universal Line Encoder'ın dalga formlarını ISim (M.81d) aracını kullanarak sunmuşlar ve çalıştırılan çizgi kodlama türlerinin çıktılarının doğruluğunu göstermişlerdir [11].

Ali ve Esraa (2010) çalışmalarında, Alan Programlanabilir Kapı Dizileri (FPGA) üzerindeki yapay bir sinir ağının bir donanım tasarımı sunmuşlardır. Dijital sistem mimarisi, ileriye dönük çok katmanlı bir sinir ağını gerçekleştirmek için tasarlanmıştır. Tasarladıkları mimariyi, Çok Yüksek Hızlı Entegre Devreler Donanım Tanımlama Dili (VHDL) kullanılarak tanımlamışlardır. Çalışma sonucunda yapay sinir ağının FPGA üzerinde gerçekleştirilmesi ile oluşan bellek kullanımı ve sistemin çalışma hızı üzerinde yaptıkları analizleri sunmuşlardır [12].

Mishra ve Saxena (2009) çalışmalarında, güvenlik, alan optimizasyonu ve değişen kanal ortamında iletişimi desteklemesine olanak tanınması nedeniyle, tek bir yonga üzerinde VHDL kullanarak çeşitli çizgi kodlama şemalarının uygulanmasını göstermişlerdir. Kullanılan çizgi kodlama şemaları Unipolar RZ ve NRZ, Polar RZ ve NRZ, AMI ve Manchester kodlamalarıdır. Universal Line Encoder'ın dalga formlarını Modelsim 6.4 aracını kullanarak sunmuşlar ve çalıştırılan çizgi kodlama türlerinin çıktılarının doğruluğunu göstermişlerdir [13].

El-Medany (2008) çalışmasında, Manchester ve Diferansiyel Manchester Coder / Decoder sistemlerinin VLSI donanım uygulamasını sunmuştur. Devrenin donanımını tanımlamak için VHDL (VHSIC Donanım Tanımlama Dili), donanım uygulama görevi için Alan Programlanabilir Kapı Dizileri (FPGA) kullanmıştır. Hedef teknoloji (FPGA) sınırsız sayıda yeniden programlanabildiği için veri hızı kolayca yeniden yapılandırılabilir. Sistemi VHDL kullanılarak tasarlamış ve Xilinx Spartan 3 FPGA Başlangıç kiti kullanılarak donanım üzerine uygulamıştır. Hem Manchester hem de Diferansiyel Manchester için kodlayıcı ve kod çözücüyü, simülasyon amacıyla veya mevcut FPGA kiti kullanılarak donanım ortamında farklı veri girişleri için test etmiştir. Çalışma sonucunda Manchester / Differential Manchester CODEC, ER400TRS Kablosuz Alıcı-Verici Sisteminin seri veri iletimi ve alımı için kullanılabileceğini kanıtlamıştır [14].

3. MATERYAL VE YÖNTEM (MATERIAL AND METHOD)

3.1. Geliştirilen Sistemin FPGA Yazılımı ve Simülasyon Tasarımı

Geliştirilen sistemde Manchester, Differential Manchester, NRZ, NRZ-Level ve NRZ-Invert hat kodlama teknikleri, VHDL mantıksal programlama dili kullanılarak kodlanmıştır. Uygulamada ele

alınan 5 kodlama tekniği için gerekli sadeleştirme işlemleri gerçekleştirilerek bu kodlama tekniklerinin lojik yapılarına ulaşılmıştır. Elde edilen bu lojik yapılara göre VIVADO geliştirme ortamında donanım tanımlama işlemleri gerçekleştirilmiş ve kodlamaların çalışması için gerekli olan devre şemaları elde edilmiştir. Her kodlama tekniği için bir çıkış sinyali tanımlanmış ve tüm mantıksal işlemler aynı görev mimarisi içerisinde gerçekleştirilmiştir. Giriş sinyalleri ile saat sinyalinin sadeleşmiş devre yapısına göre elde edilmiş lojik kapılardan geçirilmesi ile kodlama tekniklerinin mantıksal tasarımları tanımlanmıştır. Bu işlemler sonucunda, tanımlanan çıkış sinyalleri her bir kodlama tekniği için ayrı ayrı elde edilmiştir. Gerçekleştirilen uygulamanın doğru sonuçlar verip vermediğini kontrol etmek adına, elde edilen tümleşik devreye 8 bitlik veri girişi üzerinden çalışma sağlayan bir simülasyon tasarımı oluşturulmuştur. Oluşturulan simülasyon tasarımı da tek bir görev mimarisi içerisinde tanımlanmış ve çalışmada kullanılan tüm kodlama teknikleri için uygulanmıştır. Simülasyon tasarımında her giriş sinyali için 100 nanosaniye (ns) girdi süresi tanımlanmış ve saat sinyali 50 ns aralıklarla lojik “1” ve lojik “0” olacak şekilde tetiklenmiştir. Geliştirilen sistemin FPGA yazılımı Şekil 1’de, simülasyon tasarımı ise Şekil 2’de verilmiştir.

```

architecture Behavioral of lineCoding is
begin
-----MANCHESTER-----
data_out_Man <= data_in xor clk;
-----DIFFERENTIAL MANCHESTER-----
A <= data_in xnor Q;
B <= data_in xor Q;
C <= (((NOT CLK) AND A) OR (CLK AND B));
data_out_difMan <= C xnor '0';
D <= data_out_difMan;
process(CLK)
begin
if rising_edge(CLK) then
Q <= D;
end if;
end process;
----- NRE -----
data_out_NRE <= data_in;
----- NRE_LEVEL -----
data_out_NRE_L <= (NOT data_in);
----- NRE_INVERT -----
process(CLK)
begin
if rising_edge(CLK) then
data_out_NRE_I <= data_in xor data_out_NRE_I;
end if;
end process;
end Behavioral;

```

Şekil 1. Sistemin FPGA Yazılımı

```

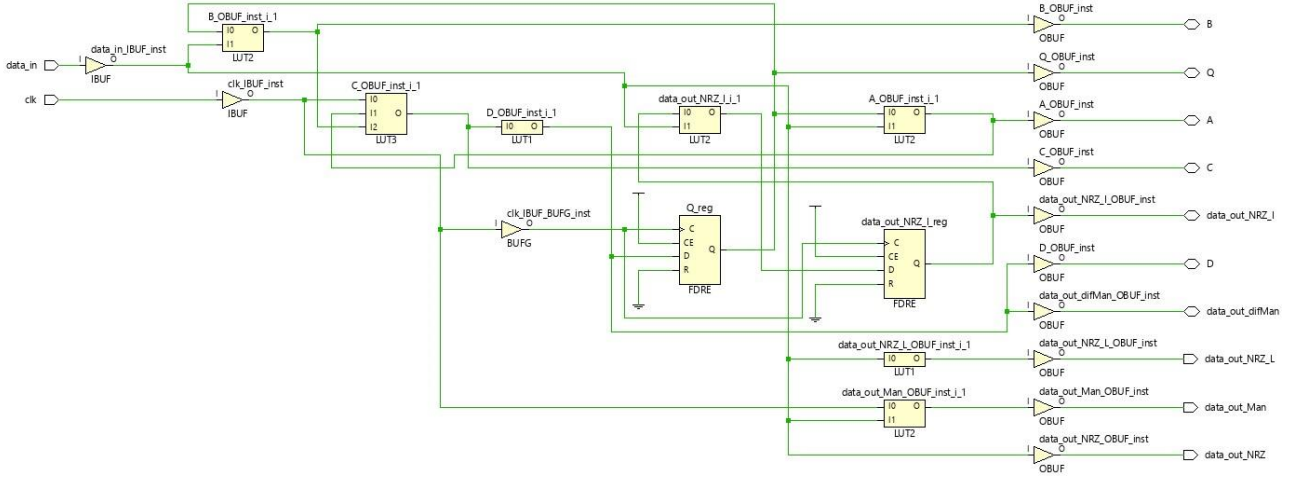
--Inputs
signal CLOCK, DATA_IN, MANCHESTER, DIF_MANCHESTER, NRE,
NRE_Level, NRE_Invert: std_logic;
BEGIN
lineCode: lineCoding PORT MAP (CLOCK, DATA_IN,
MANCHESTER, DIF_MANCHESTER, NRE, NRE_Level, NRE_Invert);
process
begin
CLOCK <= '1';
wait for 50 ns;
CLOCK <= '0';
wait for 50 ns;
end process;
process
begin
DATA_IN <= '1';
wait for 100 ns;
DATA_IN <= '1';
wait for 100 ns;
DATA_IN <= '0';
wait for 100 ns;
DATA_IN <= '0';
wait for 100 ns;
DATA_IN <= '1';
wait for 100 ns;
DATA_IN <= '1';
wait for 100 ns;
DATA_IN <= '0';
wait for 100 ns;
DATA_IN <= '1';
wait for 100 ns;
DATA_IN <= '1';
wait for 100 ns;
DATA_IN <= '0';
wait for 100 ns;
DATA_IN <= '1';
wait for 100 ns;
DATA_IN <= '1';
wait for 100 ns;
end process;
END Behavioral;

```

Şekil 2. Sistemin Simülasyon Tasarımı

3.2. Tümlleşik Sistemin FPGA Devre Yapısı

VHDL ile donanım tanımlama işlemi gerçekleştirilen tasarım VIVADO ortamında sentezlendikten sonra sistemin mantıksal devre tasarımı elde edilmiştir. Elde edilen devre tasarımında veri girişi ve saat sinyali mantıksal ünitelerden geçerek çıkış verilerini oluşturmaktadır. Elde edilen bazı çıkış verileri hem giriş hem çıkış olacak şekilde tanımlanmıştır. Bu veriler bazı kodlama tekniklerinin girişi olarak kullanılmıştır. Tüm lojik ünitelerin işlemleri tamamlandıktan sonra Manchester, Differential Manchester, NRZ, NRZ-Level ve NRZ-Invert kodlama tekniklerinin sinyal çıkışları elde edilmektedir. Geliştirilen sistemin sentezleme işlemi tamamlandıktan sonra elde edilen FPGA devre şeması Şekil 3'te verilmiştir.



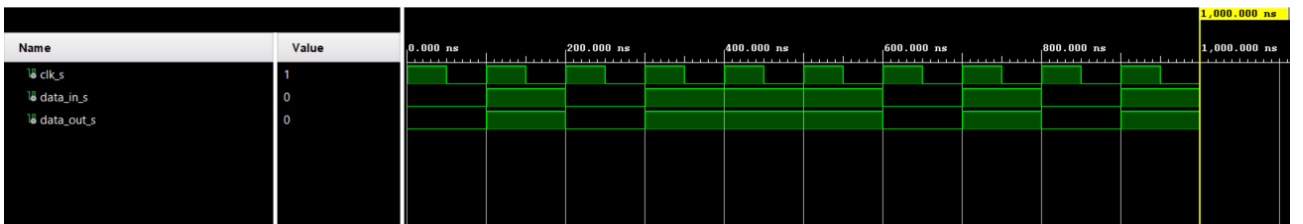
Şekil 3. Sistemin FPGA Devre Şeması

4. TARTIŞMA VE SONUÇLAR (DISCUSSION AND RESULTS)

Oluşturulan simülasyon tasarımı her kodlama türü için ayrı ayrı çalıştırılmış ve sinyal sonuçları elde edilmiştir. Her bir kodlama türünün ve tümlleşik sistemin simülasyon sonuçları aşağıda ayrı başlıklar halinde verilmiştir.

4.1. NRZ Kodlama FPGA Simülasyon Sonuçları

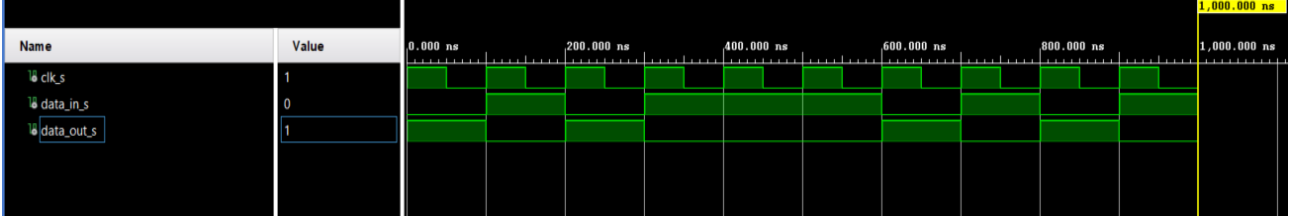
NRZ kodlamanın mantıksal çalışma prensibinde saat sinyali tetiklemesi sonrası giriş sinyalinin aynısı çıkışa aktarılmaktadır. Örneğin giriş sinyali lojik "1" ise çıkış sinyali de lojik "1" olarak elde edilir. NRZ kodlama tekniğinin FPGA simülasyon sonuçları Şekil 4'te verilmiştir.



Şekil 4. NRZ Kodlama FPGA Simülasyon Sonuçları

4.2. NRZ-Level Kodlama FPGA Simülasyon Sonuçları

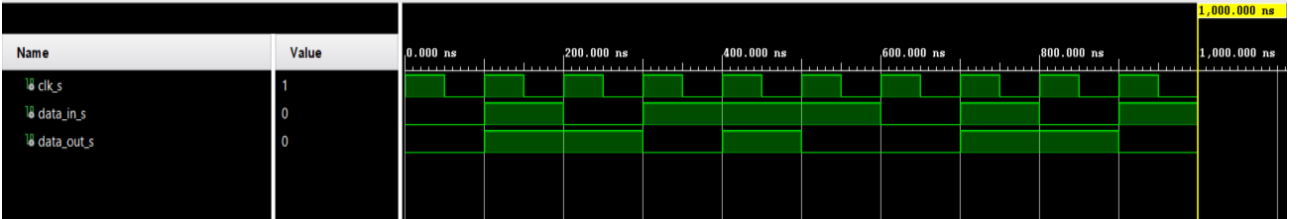
NRZ-Level kodlamanın mantıksal çalışma prensibinde saat sinyali tetiklemesi sonrası giriş sinyali lojik “0” ise çıkışa lojik “1” değeri, giriş sinyali lojik “1” ise çıkışa lojik “0” değeri aktarılmaktadır. NRZ-Level kodlama tekniğinin FPGA simülasyon sonuçları Şekil 5’te verilmiştir.



Şekil 5. NRZ-Level Kodlama FPGA Simülasyon Sonuçları

4.3. NRZ-Invert Kodlama FPGA Simülasyon Sonuçları

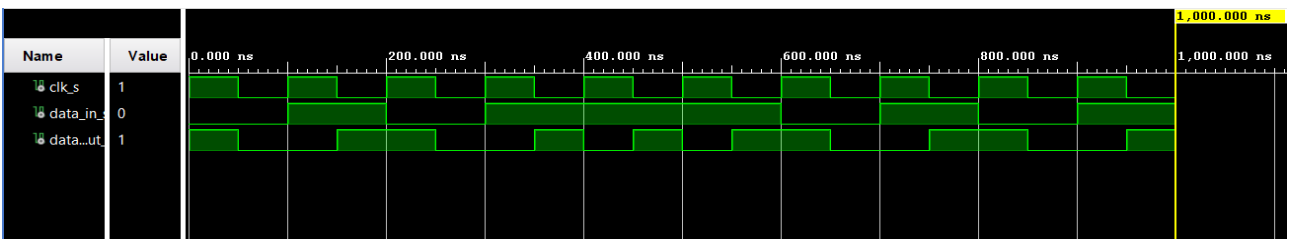
NRZ-Invert kodlamanın mantıksal çalışma prensibinde saat sinyali tetiklemesi sonrası giriş sinyali lojik “1” ise çıkış değeri önceki durumun tersi olarak değişmektedir. Giriş sinyali lojik “0” ise çıkış değeri önceki durumdaki değerini korumaktadır. NRZ-Invert kodlama tekniğinin FPGA simülasyon sonuçları Şekil 6’da verilmiştir.



Şekil 6. NRZ-Invert Kodlama FPGA Simülasyon Sonuçları

4.4. Manchester Kodlama FPGA Simülasyon Sonuçları

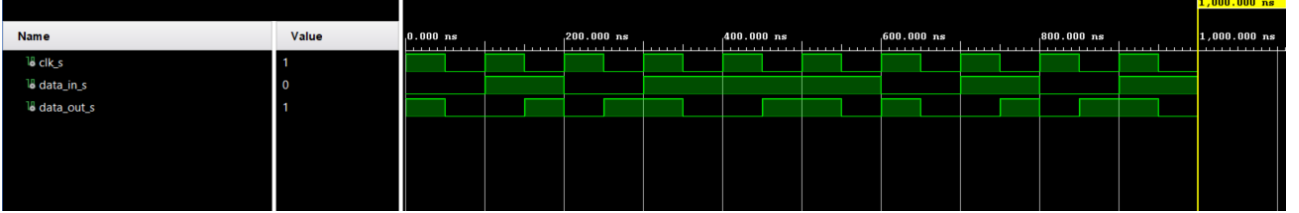
Manchester kodlamanın mantıksal çalışma prensibinde saat sinyali tetiklemesi sonrası sinyalin ortasında seviye değişimi gerçekleşmektedir. Giriş sinyalinin lojik “0” ve lojik “1” olması durumlarına bağlı olarak çıkış sinyali iki farklı şekilde oluşmaktadır. Giriş sinyalinin lojik “0” olduğu durumlarda “Z” şeklinde, giriş sinyalinin lojik “1” olduğu durumlarda ise “ters Z” şeklinde çıkış sinyali oluşmaktadır. Manchester kodlama tekniğinin FPGA simülasyon sonuçları Şekil 7’de verilmiştir.



Şekil 7. Manchester Kodlama FPGA Simülasyon Sonuçları

4.5. Differential Manchester Kodlama FPGA Simülasyon Sonuçları

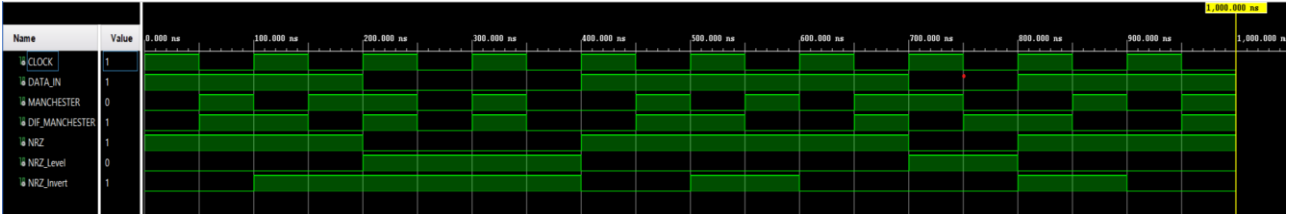
Differential Manchester kodlamanın mantıksal çalışma prensibinde saat sinyali tetiklemesi sonrası sinyalin ortasında seviye değişimi gerçekleşmektedir. Giriş sinyali lojik “0” ise çıkış sinyalinin şekli bit başında değişime uğrar. Giriş sinyali lojik “1” ise çıkış sinyali önceki durumdaki şeklini korumaktadır. Differential Manchester kodlama tekniğinin FPGA simülasyon sonuçları Şekil 8’de verilmiştir.



Şekil 8. Differential Manchester Kodlama FPGA Simülasyon Sonuçları

4.6. Geliştirilen Sistemin Tümlleşik FPGA Simülasyon Sonuçları

Tümlleşik sistemin simülasyon tasarımının çalıştırılması sonucunda verilen giriş değerleri doğrultusunda oluşan çıkış değerleri her kodlama tekniğinin mantıksal çalışma prensibi kapsamında incelenmiştir. Sistemin verdiği çıkış sinyallerinin tutarlı olduğu gözlemlenmiş ve geliştirilen sistemin doğruluğu kanıtlanmıştır. Tüm kodlama tekniklerinin tümlleşik olarak FPGA platformu üzerinde sentezlenmesi sonrasında elde edilen simülasyon sonuçları Şekil 9’da verilmiştir.



Şekil 9. Tümlleşik Simülasyon Sonuçları

4.7. FPGA Kaynak Kullanım Sonuçları

Oluşturulan tasarım BASYS3 FPGA kartına aktarılmış ve tasarımın kart üzerinde kullandığı mantıksal kaynak kullanım oranları elde edilmiştir. FPGA kartı kapsamında kullanılan LUT, Flip-Flop (FF), giriş-çıkış portları (IO) ve Buffer (BUFG) sayılarına ulaşılmış ve toplam kaynak miktarına göre oranlamaları yapılmıştır. Geliştirilen sistemin FPGA kaynak kullanım sonuçları Tablo 1’de verilmiştir.

Tablo 1. FPGA Kaynak Kullanım Sonuçları

Kaynak Türü	Kullanım	Toplam Kaynak Miktarı	Kaynak Kullanım Oranı (%)
LUT	4	20800	0.02
FF	2	41600	0.01
IO	12	106	11.32

Tablo 1’de görüldüğü üzere oluşturulan tasarım FPGA kartının kaynaklarını çok düşük oranda kullanmaktadır. Bu sebeple sistemin çalışma performansı ve tutarlılığı yüksektir. Tasarım FPGA kartına yüklendikten sonra bile kart üzerinde farklı görev tanımlamaları için kullanıma uygun olan kaynak miktarı oldukça fazladır. Bu sebeple tasarım sorunsuzca genişletilebilir ve sisteme çok sayıda kodlama ve modülasyon tekniği eklenebilir.

5. ÇIKARIM (Conclusion)

Bu çalışmada, Manchester, Differential Manchester, NRZ, NRZ-Level ve NRZ-Invert hat kodlama tekniklerinin en yaygın donanım tanımlama dili olan Very High Speed İntegrated Circuit Hardware Description Language (VHDL) kullanılarak FPGA kartı üzerinde gerçekleştirilmesi anlatılmıştır. Gerçekleştirilen uygulamanın doğru sonuçlar verip vermediğini kontrol etmek adına, elde edilen tümleşik devreye 8 bitlik veri girişi üzerinden çalışma sağlayan bir simülasyon kodu yazılmıştır. Her sinyal için 100 nanosaniye (ns) girdi süresi tanımlanmış ve saat sinyali 50 ns aralıklarla tetiklenmiştir. Simülasyonun çalıştırılması sonucunda her bir kodlama tekniği için oluşan çıkış sinyalleri, kodlama tekniklerinin kuralları doğrultusunda incelenerek sistemin doğruluğu sağlanmıştır.

Kullanılan FPGA devresinin sonsuz kez düzenlenebilir olması ve VHDL ile büyük bir esneklik elde edilmesi sayesinde bir tasarımın kısa zamanda değiştirilerek birçok farklı uygulamada kullanılabileceği görülmüştür. Gerçekleştirilen çalışmada farklı çalışma prensiplerine sahip kodlama türleri uygulanmış ve simülasyon sonuçları sunulmuştur. Literatürde bulunan diğer çalışmalara kıyasla bu çalışmada, oluşturulan VHDL tasarım, arka planda oluşan mantıksal devre tasarımı ve her kodlama türünün simülasyon sonucu ayrı ayrı verilmiş ve açıklanmıştır. FPGA’ların geleneksel işlemcilerin sahip olmadığı hız, güvenlik ve paralel işlem yapabilme kabiliyetine sahip olması sebebiyle sinyal kodlama teknikleri uygulamalarında kullanımının doğru ve tutarlı sonuçlar getirdiği görülmüştür. Gelecek çalışmalarda daha fazla kodlama türü ve daha büyük giriş verileri içeren uygulamalar gerçekleştirilebilir. Kodlama tekniklerinin kullanıldığı büyük sistemlerde FPGA kullanımı sağlanarak daha performanslı ve tutarlı sistemler elde edilebilir.

KAYNAKLAR (REFERENCES)

- [1] Gürfidan, R., & Ersoy, M. (2020). A New Hybrid Encryption Approach for Secure Communication: GenComPass. *International Journal of Computer Network and Information Security*, 12(4), 1-10.
- [2] Gürfidan, R., & Ersoy, M. (2021). Blockchain-based music wallet for copyright protection in audio files. *Journal of Computer Science & Technology*, 21.
- [3] Srinivasan, S., Kavitha, M., Rani, G. V., Manoharan, L., Terence, E., & Siva, A. V. (2023, March). Implementation of Digital Modulation Techniques in High-Speed FPGA Board. In *2023 Second International Conference on Electronics and Renewable Systems (ICEARS)* (pp. 21-26). IEEE.
- [4] Rajalakshmi, A., & Kavitha, A. (2023). Suppression of EMI using cost-effective FPGA-based digital communication modulation techniques in power converters. *IETE Journal of Research*, 69(6), 3711-3722.
- [5] Kwiatkowski, P., & Szplet, R. (2020). Efficient implementation of multiple time coding lines-based TDC in an FPGA device. *IEEE Transactions on Instrumentation and Measurement*, 69(10), 7353-7364.

- [6] Gupta, Ankit, Gurashish Singh. "Implementation and Analysis of Different Line Coding Schemes using Verilog." *International Journal of Science, Engineering and Technology Research (IJSETR)* 5.2 (2016): 395-401.
- [7] Singh, V., & Mishra, B. (2014). FPGA implementation of various lines coding technique for efficient transmission of digital data in communication. *Polar*, 11(01), 11.
- [8] Singh, Gurashish, vd. "FPGA implementation of different NRZ line coding schemes." 2016 1st India International Conference on Information Processing (IICIP). IEEE, 2016.
- [9] Ersoy, M., & Kumral, C. D. (2021). Bilgisayar Destekli FPGA Tabanlı Sayısal Sistemler Deney Seti Tasarımı. *Bilişim Teknolojileri Dergisi*, 14(3), 301-312.
- [10] Y. H. Zuo vd., "Design and Implementation of Manchester CODEC Based on FPGA", *Applied Mechanics and Materials*, Vol. 273, pp. 805-809, 2013.
- [11] Kaur, Amrinder, Mandeep Singh & Balwinder Singh. "VHDL IMPLEMENTATION OF UNIVERSAL LINE ENCODER-DECODER FOR COMMUNICATION."
- [12] Ali, Haitham Kareem, Esraa Zeki Mohammed. "Design artificial neural network using FPGA." *IJCSNS* 10.8 (2010): 88.
- [13] Mishra, P. K., & Saxena, S. (2009). A novel approach for VHDL implementation of universal line encoder for communication. *Journal of Scientific Research*, 36(1), 30-40.
- [14] El-Medany, Wael M. "FPGA implementation of RDR Manchester and D-Manchester CODEC design for wireless transceiver." 2008 National Radio Science Conference. IEEE, 2008.