BURSA TEKNİK ÜNİVERSİTESİ
MÜHENDİSLİK VE DOĞA BİLİMLERİ FAKÜLTESİ
Bilgisayar Mühendisliği Bölümü

BİLGİSAYAR MİMARİSİ

Hamming Code Projesi

23360859058

Sami Erzincanlı

1. GİRİŞ:

Bu projede, 8, 16 ve 32 bitlik veriler üzerinde Hamming SEC-DED (Single-Error-Correcting, Double-Error-Detecting) kodlama algoritmasını uygulayan web tabanlı bir simülatör geliştirilmiştir.

Geliştirilen simülatör, HTML, CSS ve JavaScript teknolojileri kullanılarak modern bir web arayüzü ile tasarlanmıştır. Uygulama, kullanıcıların ikili veri girişi yapmasına, bu veri için Hamming kodunu hesaplamasına, yapay olarak tek veya çift bit hataları oluşturmasına ve bu hataları tespit edip düzeltmesine olanak tanımaktadır.

Simülatör, Hamming kodlama algoritmasının teorik temellerini pratik bir yaklaşımla sunarak, kullanıcıların hata düzeltme kodları konusunu görsel ve interaktif bir şekilde öğrenmelerine yardımcı olmaktadır. Özellikle parity bitleri, SEC bitleri ve hatalı bitlerin farklı renklerle gösterilmesi sayesinde algoritmanın çalışma mantığı kolayca anlaşılabilmektedir.

Projenin Github Linki: https://github.com/Samierz/Hamming-Code-Simulator

2. PROJE DETAYLARI:

2.1 Kullanılan Teknolojiler

HTML5: Yapısal tasarım

CSS3: Görsel tasarım ve responsive layout

JavaScript (ES6): Ana algoritma ve interaktivite

2.2 Hamming Code Algoritması

2.2.1 Hamming Kodu Oluşturma (buildHammingCode fonksiyonu)

- Gerekli parity bit sayısını hesaplama: $2^p \geq m + p + 1$
- Veri bitlerini 2'nin kuvveti olmayan pozisyonlara yerleştirme
- Her parity bit için XOR hesaplama
- SEC (Secondary) bit hesaplama (çift hata tespiti için)

2.2.2 Hata tespiti ve Düzeltme (detectAndFixError fonksiyonu

Hata Analizi Mantığı:

- **Sendrom = 0, SEC = 0**: Hata yok
- **Sendrom ≠ 0, SEC = 1**: Tek bit hatası (düzeltilebilir)
- **Sendrom = 0, SEC = 1**: SEC bit hatası
- **Sendrom ≠ 0, SEC = 0**: Çift bit hatası (tespit edilebilir, düzeltilemez)

2.3 **Javascript Fonksiyonları**

- *createHammingCode():* Hamming kodu üretimi
- *buildHammingCode(bits):* Hamming algoritması core implementasyonu
- *injectSingleError():* Tek bit hata ekleme
- *injectDoubleError():* Çift bit hata ekleme
- *detectAndFixError():* Hata analizi ve düzeltme
- *getParityPositions():* Parity bit pozisyonlarını hesaplama
- *renderBits():* Görsel bit gösterimi

2.4 Kullanıcı Arayüzü Tasarımı

2.4.1 Görsel Tasarım Özellikleri

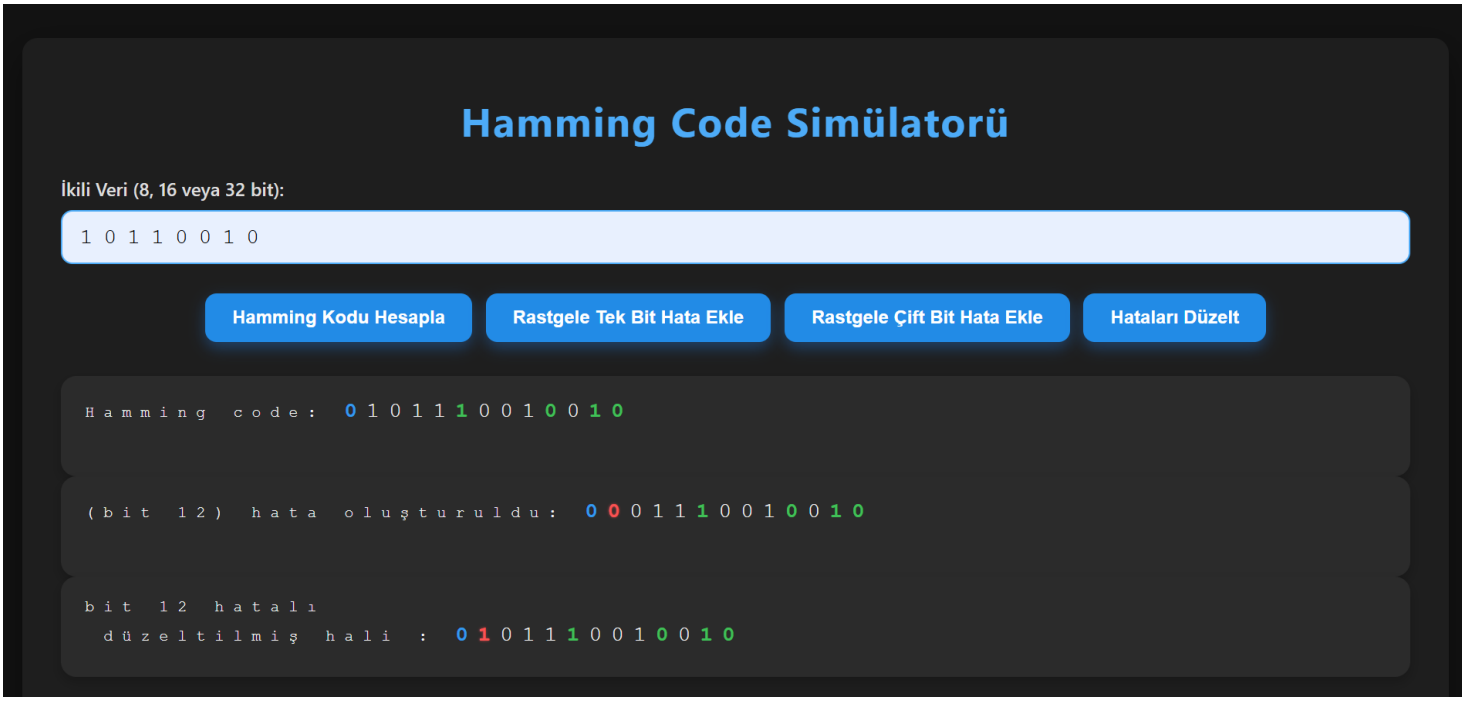
- **Dark Theme**: Modern görünüm için koyu tema kullanımı
- **Renkli Bit Gösterimi**:
  - Yeşil: Parity bitleri
  - Mavi: SEC bit
  - Kırmızı: Hatalı bitler
- **Responsive Layout**: Farklı ekran boyutlarına uyum

2.4.2 Etkileşim Elementleri

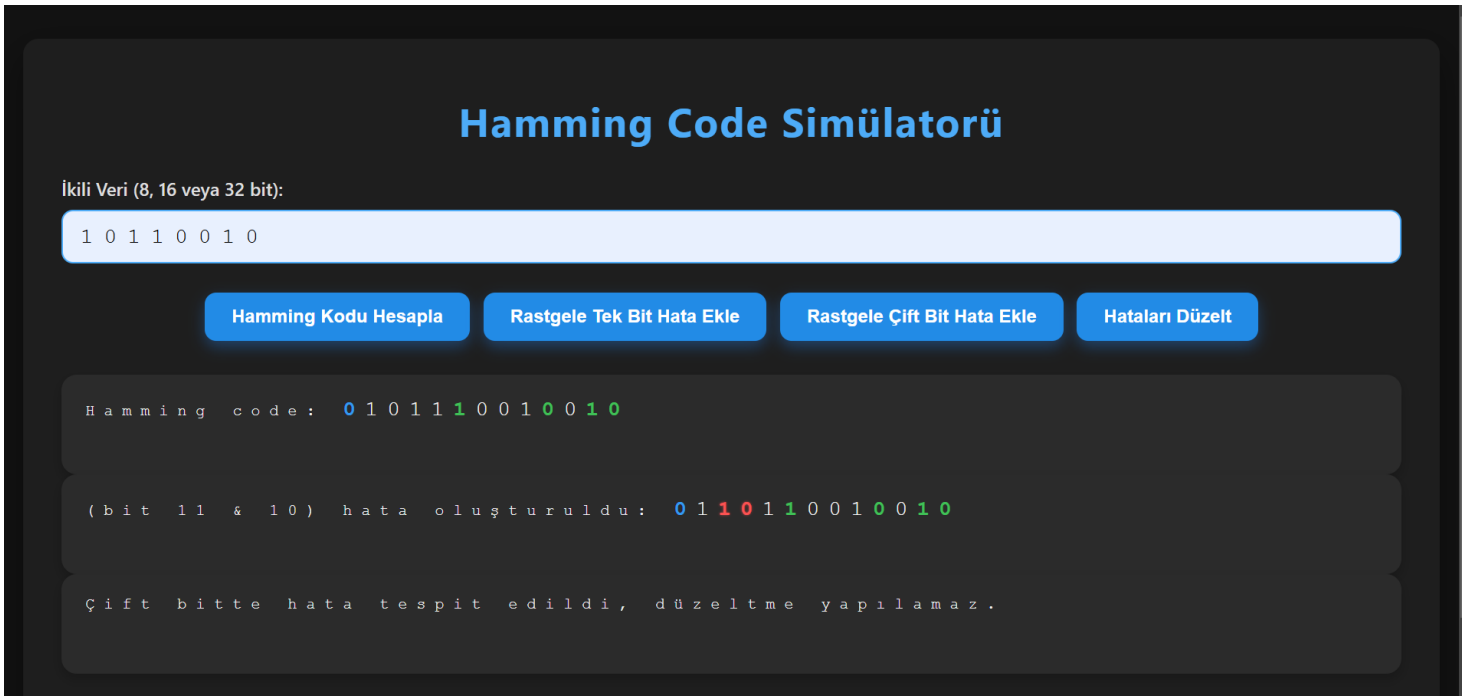
- Veri giriş alanı (input validation ile)
- 4 ana işlev butonu
- 3 sonuç gösterim alanı

2.5 Ekran Görüntüleri

Sitenin giriş ekranı



Girilen bir veri için oluşturulan hamming code ve o veriye eklenen rastgele tek bit bir hatanın düzeltimi



3. Sonuç

Bu projede, Hamming SEC-DED kodlama algoritmasının web tabanlı bir simülatörü başarıyla geliştirilmiştir. Uygulama, hem eğitim amaçlı kullanım için hem de algoritmanın pratik anlaşılması için etkili bir araç sunmaktadır. Modern web teknolojileri kullanılarak oluşturulan arayüz, kullanıcıların Hamming kodlama konseptlerini görsel olarak kavramalarına yardımcı olmaktadır.

Proje, tüm belirlenen gereksinimleri karşılamakta ve ek özellikler ile kullanıcı deneyimini artırmaktadır. Gelecek çalışmalarda, daha gelişmiş hata düzeltme algoritmaları ve performans analizi özellikleri eklenebilir.