BURSA TEKNİK ÜNİVERSİTESİ

DERS	BLM230 BİLGİSAYAR MİMARİSİ
İSİM	Abdullah Çelik
ÖĞRENCİ NUMARASI	22360859059
KONU	Hamming SEC-DED Code Simülatörü
Projenin github linki	https://github.com/Oxhi1/Oxhi1-Hamming-SEC-DED-Kodlama-ve-Hata-D-zeltme-Sim-lat-r-8-16-32-Bit-Python-Tkinter-GUI

GİRİŞ

Bu proje, **Bursa Teknik Üniversitesi - Bilgisayar Mühendisliği** bölümünde yürütülen **Bilgisayar Mimarisi** dersi kapsamında geliştirilmiştir. Projede, veri bütünlüğünü sağlamak amacıyla **Hamming SEC-DED (Single Error Correcting, Double Error Detecting)** algoritması temel alınarak bir hata tespit ve düzeltme simülatörü geliştirilmiştir.

Kullanıcı, 8, 16 veya 32 bitlik ikili verileri sisteme girebilir. Sistem bu verileri Hamming kodlamasıyla belleğe yazar. Ardından kullanıcı, belirli bit(ler)i yapay olarak bozabilir. Program bu bozulmuş veriyi analiz eder ve **tekli bit hatalarını otomatik düzeltir**, **çiftli bit hatalarını ise tespit eder**.

KULLANILAN PROGRAMLAMA DİLİ VE TEKNOLOJİLER

- **Python 3.10+**
- **Tkinter**: Arayüz tasarımı için kullanılmıştır.
- **math** ve **random** kütüphaneleri
- Bit manipülasyon ve hata düzeltme işlemleri Python diliyle doğrudan gerçekleştirilmiştir.

FONKSİYON AÇIKLAMALARI

hamming_sec_ded_encode(data)

- Verilen 8/16/32 bitlik veriyi Hamming kodlamasına göre kodlar.
- Parity bitlerini ve genel parity'yi (P0) hesaplayarak kodlanmış veriyi döndürür.

inject_error()

- Kullanıcının belirttiği bit veya bitleri yapay olarak bozar (0 → 1, 1 → 0).
- Bozulan bitleri kırmızı renkle GUI üzerinde gösterir.

analyze_and_correct()

- Bozulmuş veriyi analiz ederek sendrom kelimesini oluşturur.
- Hata türüne göre kullanıcıya durumu bildirir:
    - **Tekli hata** tespit edilirse, otomatik düzeltir ve düzeltilen biti yeşil gösterir.
    - **Çiftli hata** varsa, tespit edilir ama düzeltilemez.

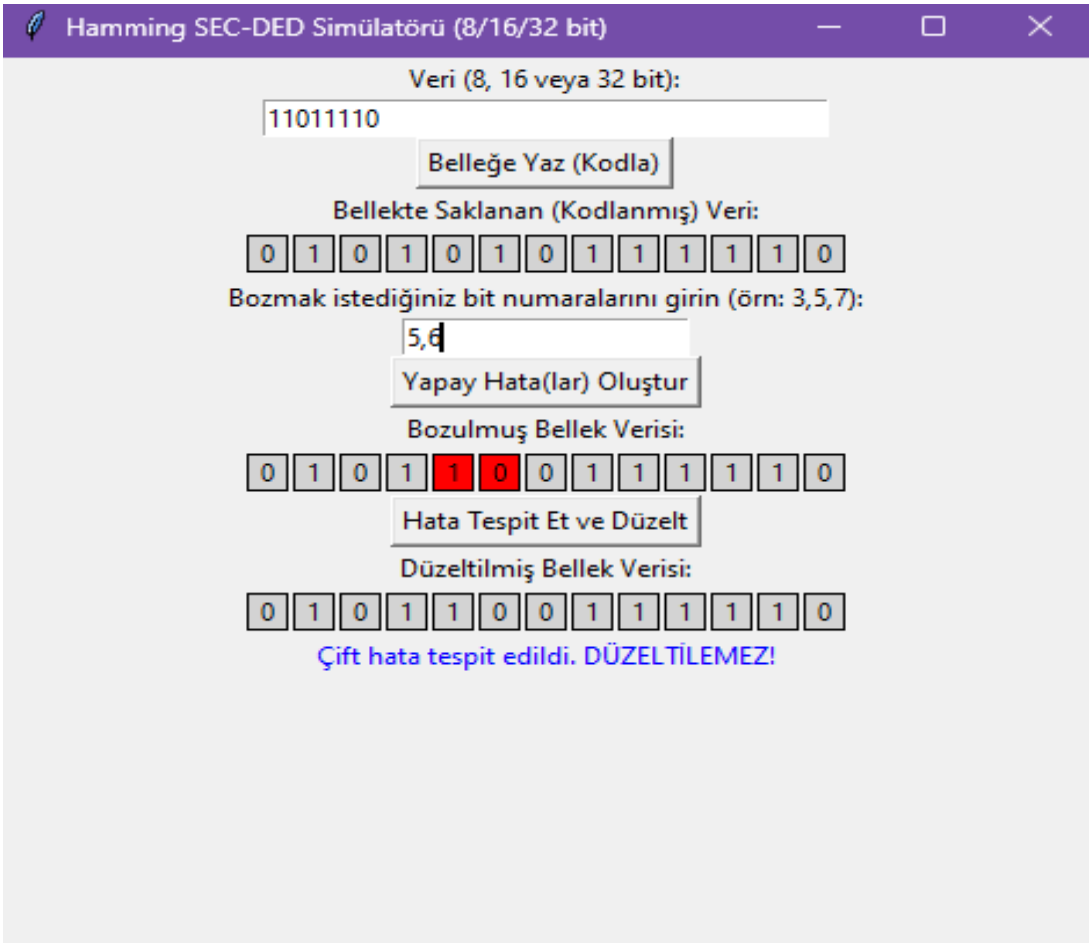
show_bits(frame, bits, highlight_index=None, highlight_color="red")

- Bit dizilerini GUI üzerinde bloklar halinde görsel olarak gösterir.
- İstenirse belirli bit(ler) renklendirilerek vurgulanır.

ARAYÜZ KULLANIMI

Uygulama, Tkinter arayüzü ile kullanıcı dostu bir şekilde tasarlanmıştır. Aşağıdaki adımlarla kullanılır:

1. **Veri Girişi**
    - 8, 16 veya 32 bitlik sadece 0 ve 1 içeren veri girilir.
    - Belleğe Yaz (Kodla) butonuna tıklanır.
2. **Kodlanmış Bellek Verisi**
    - Hamming kodu uygulanmış veri GUI üzerinde görüntülenir.



3. **Hata Oluşturma**

- Hangi biti bozmak istersiniz? kutusuna 1’den başlayarak bozulması istenen bit(ler) girilir.
- Virgül ile birden fazla bit girilebilir (örn. 3,7).
- Yapay Hata Oluştur butonuna tıklanır.

4. **Hata Tespit ve Düzeltme**
   - Hata Tespit Et ve Düzelt butonuna basıldığında:
     - **Tekli hatalar** yeşil renkle düzeltilir.
     - **Çiftli hatalar** tespit edilir ve kullanıcı uyarılır.