

Özet ve Amaç:

Bu projede, Swing tabanlı bir arayüz ile kullanıcıdan alınan m-bit uzunluğundaki ikili veri dizisinin SEC-DED (Single Error Correction, Double Error Detection) Hamming kodlaması uygulanmış hâli üretilmekte; kullanıcı tarafından enjekte edilen tek hata düzeltilebilmekte ve çift hata tespit edilebilmektedir.

Not: Proje özellikle 8, 16 ve 32 bitlik veri dizileri üzerinde test edilmiştir ancak dilediğiniz m-bit uzunluğunda veri kullanmaya da uygundur. Amaç, hata düzeltme tekniklerini pratik olarak gözlemlemek ve kavramları pekiştirmektir.

Kurulum & Çalıştırma:

1. Java 8 veya üstü yüklü olmalı.
2. Terminal/komut satırında proje klasörüne gidin:
 - Windows: `cd C:\Users\<KullaniciAdiniz>\Projeler\BLM230-HammingSimulator`
 - macOS/Linux: `cd /Users/<KullaniciAdiniz>/Projeler/BLM230-HammingSimulator`

3.Derleme:

```
javac mimari/*.java
```

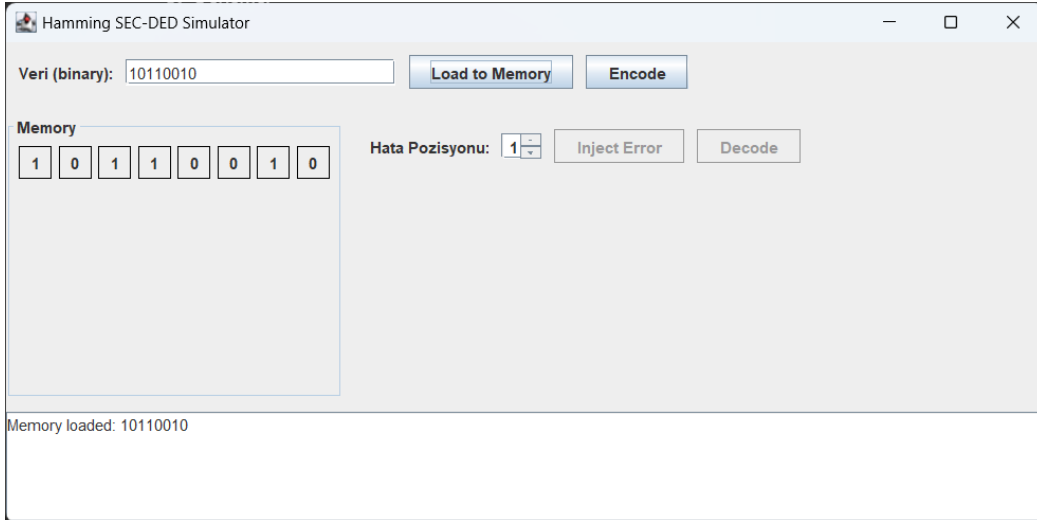
4.Çalıştırma:

```
java mimari.MainUI
```

Arayüz ve İş Akışı:

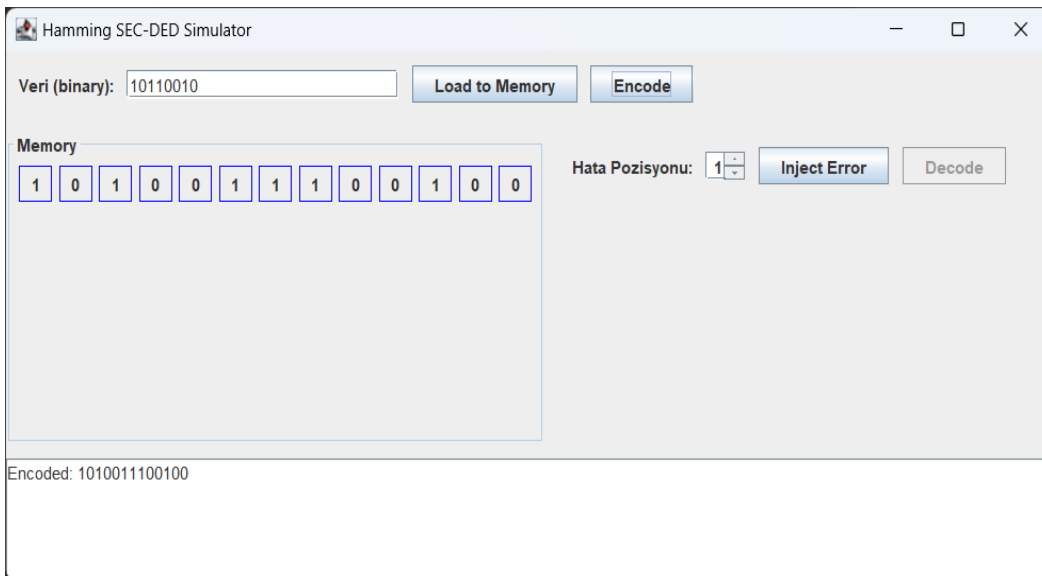
1)Load to memory: Kullanıcı, "Veri (binary)" alanına istediği uzunluktaki 0/1 dizisini girer ve **Load to Memory butonuna tıklar.**

*Binary değer girildi Load to Memory butonuna basıldı.



2)Encode: Butonuna basıldığında, m veri biti, r parite biti ve bir overall parity biti hesaplanarak kod kelimesi panelde gösterilir.

*Encode sonucu ekran görüntüsü:



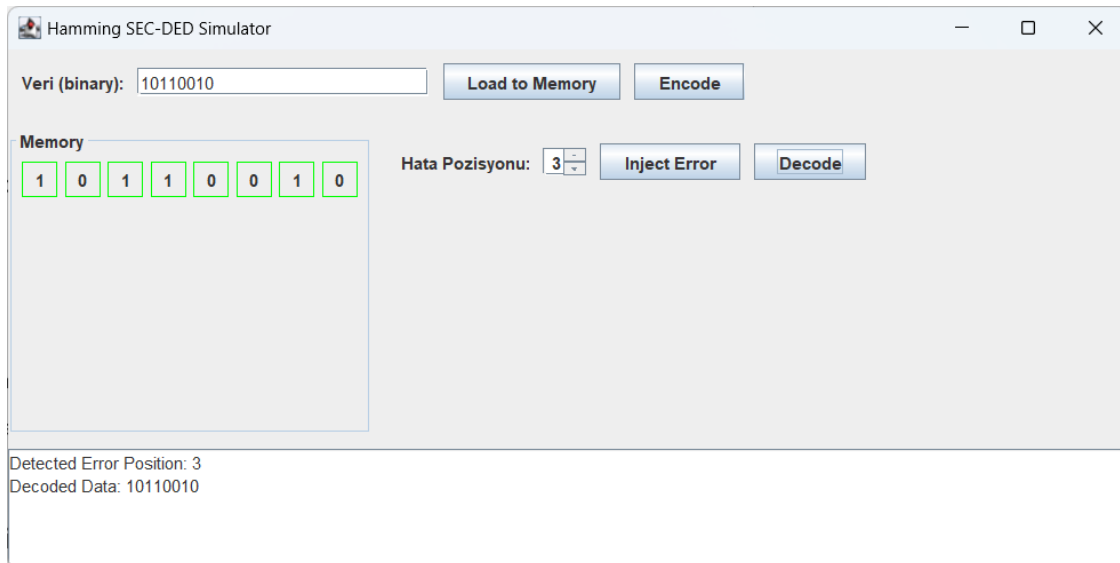
3)Error Injection: Spinner'dan seçilen pozisyondaki bit kırmızı kenarlıklı olarak terslenir.

*Tek hata inject ekran görüntüsü:

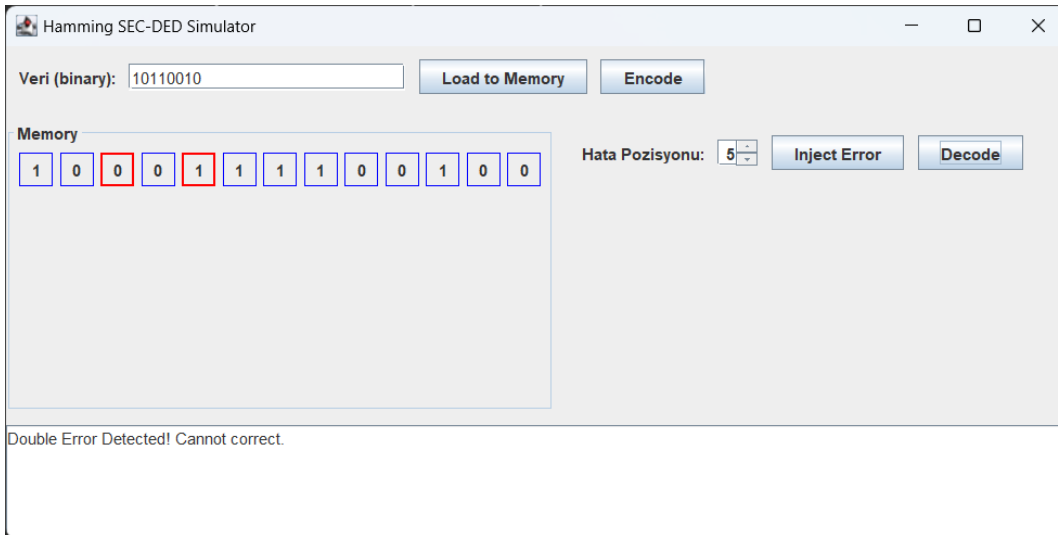
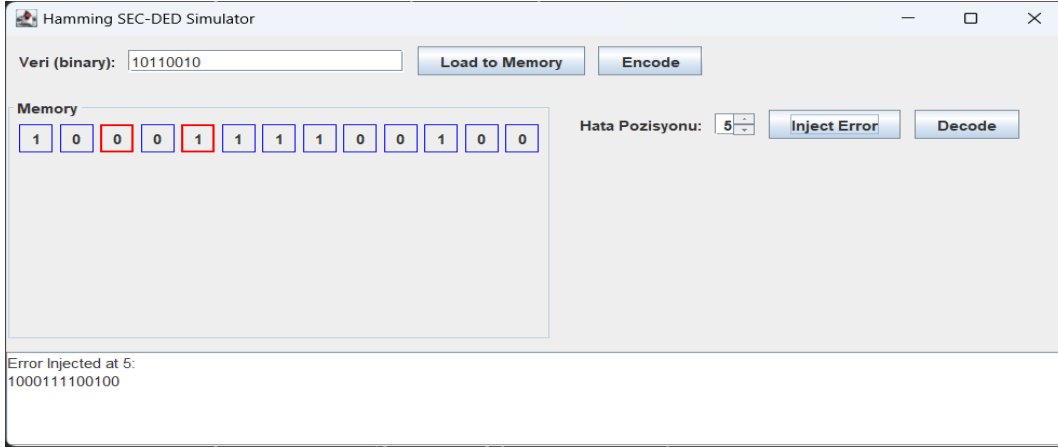


4)Decode: **Decode** butonu ile tek hata tespit edilip düzeltilir veya çift hata durumunda uyarı mesajı verilir.

*Tek hata sonrası decode ekran görüntüsü:



*Çift hata sonrası decode ekran görüntüsü:



5)Test Sonuçları:

Tek Hata Testi

- Girdi: 10110010
- Encode → 1010011100100
- Inject(pos=5) → 1010111100100
- Decode → Detected Error Position: 5, Decoded Data: 10110010

Çift Hata Testi

- Girdi: 10110010

- Encode → 1010011100100
 - Inject(pos=3) → 1000011100100
 - Inject(pos=8) → 1000011000100
 - Decode → Double Error Detected! Cannot correct.
-

6)Linkler

- **GitHub Repo:** [ferideyalcinkaya/BLM230-HammingSimulator: Hamming SEC-DED Simulator for BLM230 Dönem Projesi](https://github.com/ferideyalcinkaya/BLM230-HammingSimulator)
 - **Demo Video:** <https://youtu.be/VideoLinkin>
-

7) Sonuç ve Değerlendirme

Proje, SEC-DED Hamming kodlamasının temel ilkelerini başarıyla uygulamakta ve kullanıcıya hem tek hata düzeltme hem de çift hata tespiti imkânı sağlamaktadır. GUI ve kullanım akışı anlaşılır; tüm gereksinimler karşılanmıştır.

8)Bonus:

“Hamming Code nasıl oluşur?” butonuna tıklayarak hamming code hakkında eğitici bilgi penceresine ulaşabilirsiniz.