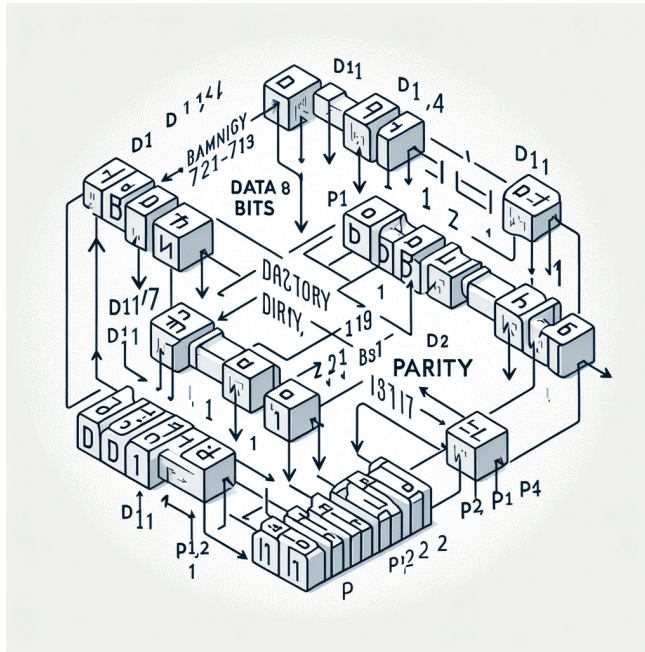


HAMMING ALGORİTMASI (7,4)

Hamming (7,4) kodu, hata tespiti ve düzeltme konusunda kullanılan bir yöntemdir. Bu yöntemde, 4 bitlik veriye 3 bit eklenerek toplamda 7 bitlik bir kod oluşturulur. Bu eklenen 3 bit, verinin doğru şekilde iletildiğini doğrulamak ve olası hataları düzeltmek için kullanılır.

Hamming (7,4) Algoritmasının Açıklaması:

- **Amacı:** Hata tespiti ve düzeltme.
- **Veri Bitleri:** İlk 4 bit (D1, D2, D3, D4) orijinal veri bitleridir.
- **Parite Bitleri:** Eklenen 3 bit (P1, P2, P4) parite bitleridir. Bu bitler, veri bitlerine bağlı olarak hesaplanır.
- **Parite Bitlerinin Hesaplanması:**
 - **P1:** 1, 2, 4. bitlerin XOR'u (D1, D2, D4 gibi).
 - **P2:** 1, 3, 4. bitlerin XOR'u (D1, D3, D4 gibi).
 - **P4:** 2-4. bitlerin XOR'u (D2, D3, D4 gibi).



Parite Bitlerinin Hesaplanması

P1, P2, P4, : Bunlar, belirli veri bitlerinin XOR işlemiyle hesaplanır.

- **P1:** 1, 2, 4. bitlerin XOR'u.
- **P2:** 1, 3, 4. bitlerin XOR'u.
- **P4:** 2, 4. bitlerin XOR'u.

Örnek Hesaplama

Veri Bitleri

- Diyelim ki göndermek istediğimiz veri 1011 olsun. Bu, 4 bitlik veri kısmımızı oluşturur (D1, D2, D3, D4).

Parite Bitlerinin Hesaplanması

Parite bitleri, belirli veri bitlerinin XOR (özel veya) işlemiyle hesaplanır. Hamming (7,4) kodunda 3 parite biti vardır: P1, P2, P4. Her birini aşağıdaki gibi hesaplarız:

- **P1:** 1, 2, 4. bitlerin XOR'u.
 - Bu durumda: D1 (1), D2 (0), D4 (1) $\rightarrow P1 = 1 \text{ XOR } 0 \text{ XOR } 1 = 0$
- **P2:** 1, 3, 4. bitlerin XOR'u.
 - Bu durumda: D1 (1), D3 (1), D4 (1) $\rightarrow P2 = 1 \text{ XOR } 1 \text{ XOR } 1 = 1$
- **P4:** 2-4. bitlerin XOR'u.
 - Bu durumda: D2 (0), D3 (1), D4 (1) $\rightarrow P4 = 0 \text{ XOR } 1 \text{ XOR } 1 = 0$

Sonuç

- Hesaplanan parite bitleriyle birlikte, 7 bitlik Hamming kodumuz şu şekilde oluşur: **P1 P2 D1 P4 D2 D3 D4**
- Bu durumda kodumuz: 0 1 1 0 0 1 1