

VLSI II

I. PROJE

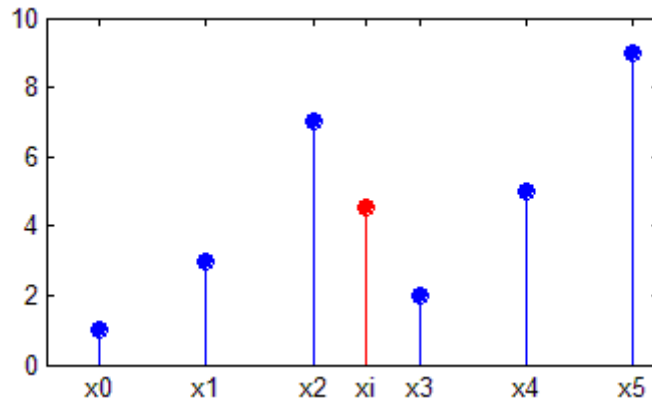
CUBIC INTERPOLATOR TASARIMI

1. Projenin Tanımı

Proje, 24 bitlik sayısal bir işareti alıp, sayısal işaretin örnekleme frekansını iki katına çıkartan, aynı zamanda aşağıda anlatılmış olan üçüncü dereceden interpolasyon algoritmasını uygulayarak çıkış işaretinin ideale yaklaşmasını sağlayan devre tasarımıdır. Devre tasarlanmadan önce MATLAB kullanılarak algoritma test edilecek, daha sonra Verilog dili ile yazılan kodun çıktıları MATLAB çıktılarıyla karşılaştırılacaktır.

2. Üçüncü Dereceden İnterpolasyon Algoritması

Üçüncü dereceden interpolasyon algoritması aşağıda anlatılmıştır. İşlem adımlarını tamamlayarak katsayıları bulmanız gerekmektedir.



Üçüncü dereceden interpolasyon için eşit aralıklarla örneklenmiş $f(x)$ işaretinin ardışık örneklerinden 4 tanesi (örn. x_1 x_2 x_3 x_4) alınarak $f(x_1)$, $f(x_2)$, $f(x_3)$ ve $f(x_4)$ noktalarından geçen

3. derece polinom fonksiyonunun bulunması, sonra bu fonksiyonun x_i noktasındaki (x_2 ve x_3 noktalarının orta noktasındaki) değerinin elde edilmesi gerekmektedir. Bunun için izlenecek yöntem aşağıda verilmiştir:

1. Aralıkların eşit ve 1 olduğu, x_1 değerinin 0 olduğu varsayılır. Buna eksenî öteleme ve genişletme denir. Bu varsayım hesap kolaylığı sağlayacaktır.

2. $f(x) = ax^3 + bx^2 + cx + d$ olmak üzere:

$$f(x_1) = f(0) = y_1 = d$$

$$f(x_2) = f(1) = y_2 = a + b + c + d$$

$$f(x_3) = f(2) = y_3 = 8a + 4b + 2c + d$$

$$f(x_4) = f(3) = y_4 = 27a + 9b + 3c + d$$

eşitlikleri yazılarak aşağıdaki denklem takımı oluşturulur:

$$\underbrace{\begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 8 & 4 & 2 & 1 \\ 27 & 9 & 3 & 1 \end{bmatrix}}_A \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} \longrightarrow \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = A^{-1} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix}$$

3. Esas bulunmak istenen katsayılar a,b,c ve d değil $f(x_i) = f\left(\frac{3}{2}\right) = \frac{27}{8}a + \frac{9}{4}b + \frac{3}{2}c + d$

olduğu için denklem takımı yeniden düzenlenir:

$$\underbrace{\begin{bmatrix} \frac{27}{8} & 0 & 0 & 0 \\ 0 & \frac{9}{4} & 0 & 0 \\ 0 & 0 & \frac{3}{2} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_K \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = K A^{-1} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} \longrightarrow \begin{bmatrix} \frac{27}{8}a \\ \frac{9}{4}b \\ \frac{3}{2}c \\ d \end{bmatrix} = K A^{-1} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix}$$

$$\begin{bmatrix} \frac{27}{8}a \\ \frac{9}{4}b \\ \frac{3}{2}c \\ d \end{bmatrix} = K A^{-1} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} k_{11} & k_{12} & k_{13} & k_{14} \\ k_{21} & k_{22} & k_{23} & k_{24} \\ k_{31} & k_{32} & k_{33} & k_{34} \\ k_{41} & k_{42} & k_{43} & k_{44} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} k_{11}y_1 + k_{12}y_2 + k_{13}y_3 + k_{14}y_4 \\ k_{21}y_1 + k_{22}y_2 + k_{23}y_3 + k_{24}y_4 \\ k_{31}y_1 + k_{32}y_2 + k_{33}y_3 + k_{34}y_4 \\ k_{41}y_1 + k_{42}y_2 + k_{43}y_3 + k_{44}y_4 \end{bmatrix}$$

buradan da $k_i = \sum_{n=1}^4 k_{ni}$ ($K A^{-1}$ matris çarpımının i. sütunun toplamı) olmak üzere;

$$f(x_i) = f\left(\frac{3}{2}\right) = \frac{27}{8}a + \frac{9}{4}b + \frac{3}{2}c + d = y_1k_1 + y_2k_2 + y_3k_3 + y_4k_4$$

olarak bulunur. Kısacası k katsayıları bulunduktan sonra $f(x_i)$ değeri yukarıdaki formülden hesaplanacaktır.

4. Projenin Ayrıntıları

Projede yaratılacak olan devrenin giriş çıkışları şu şekilde olacaktır:



clk: 10 MHz saat işareti

reset: Asenkron, aktif düşük

data_in: 24 bitlik clk/2 frekansında örneklenmiş giriş işareti

out: 24 bitlik çıkış işareti

data_in verisinin clk/2 frekansında örneklenmiş olması, bu işaretin clk frekansında örneklendiğinde (değişim periyodu örnekleme periyodunun yarısı olduğu için) iki clk darbesinde bir değişmesi anlamına gelir. Yani 100ns'lik clk periyoduna sahip bir testbench kodu şu şekilde olacaktır:

```
always #50 clk = ~clk;
```

```
initial begin
```

```
    ...
```

```
    clk reset değişimleri ve ilk değerleri
```

```
    ...
```

```
    data_in = 1;
```

```
    #200 data_in = 2;
```

```
    #200 data_in = 5;
```

```
    ....
```

```
end
```

Hatırlamanız gereken bir diğer nokta ise sizin çıkışa ortadaki değeri vermiş olduğunuzdur. Yani interpolasyonu hesapladığınız an çıkışa verebilirsiniz, ama interpolasyondan sonraki

verinin interpolasyonu hesaplamaya yarayan verilerden birisi olması gerekmektedir. Kısacası çıkış verisi bir interpolasyon – bir gerçek giriş şeklinde gitmek zorundadır.