

README – Verilog Full Adder 8-bit (Saat Kontrollü)

Proje Hakkında

Bu proje, Verilog HDL kullanarak **clock kontrollü (saat kontrollü)** bir **8-bit tam toplayıcı (full adder)** tasarımı içerir.

Toplama işlemi **her clock sinyali geldiğinde 1 bit** olacak şekilde yapılır. Bu yöntem, donanımsal olarak gerçekçi ve anlaşılması kolay bir yaklaşım sunar.

Modül: Full_adder_8bit

Amaç:

Verilen iki 8-bitlik sayıyı (A ve B) bir taşıma (Cin) ile toplayıp:

- **Sonuç (S)**
- **Her bitin taşıması (Cout)**

değerlerini üretir.

Çalışma Prensipleri

- Toplama işlemi bir sayaç (i) ile bit bit yapılır.
- Her clock (saat) kenarında 1 bit toplanır.
- Taşıma (carry) değeri, bir sonraki bit için güncellenerek devam eder.
- **Toplam 8 clock döngüsünde işlem tamamlanır.**

Değişken	Açıklama
A, B	Toplanacak 8-bit sayı girişleri
S	Toplam sonucu (8-bit)

Cout	Her bit için oluşan taşıma değeri (8-bit)
Cin_current	O anki taşıma (carry in)
Cin_next	Bir sonraki döngü için taşıma değeri
clk	Saat sinyali
i	Hangi bitin toplandığını belirten sayaç

Saat Döngüsü (Clock Cycle)

- **Pozitif Kenar (posedge clk)**
 - S[i] ve Cout[i] hesaplanır.
 - Cin_next güncellenir.
- **Negatif Kenar (negedge clk)**
 - Cin_current, Cin_next'ten alınır. Yani taşıma bir sonraki cycle için hazır olur.

Test Verileri

Kod içinde başlangıç değerleri olarak:

```
A = 8'b0000_0101; // 5
B = 8'b1111_0011; // 243
```

Toplam: A + B = 248, yani 1111_1000

Simülasyon

Kod test bench'e ihtiyaç duymadan kendi içinde test çalıştırır.

Ama istersen aşağıdaki gibi bir simülasyon çıktısı eklersen sonucu da görebilirsin:

```
verilog
KopyalaDüzenle
if (i == 8) begin
    $display("A      = %b", A);
    $display("B      = %b", B);
```

```
$display("Sum    = %b", S);  
$display("Cout   = %b", Cout);  
$finish;  
end
```

Öğrenme Açısından Notlar

- Bu proje, **1-bit full adder mantığını** 8-bit'e genişletme fikrini temel alır.
- Aynı anda tüm bitleri toplamak yerine, clock ile sırayla yapıldığı için **senkron devre yapısını anlamaya yardımcı olur.**
- Taşıma (carry) sıralı şekilde işlendiği için Cin_current ve Cin_next kullanımı önemlidir.