

# Bil265 Proje Raporu

## Toplayıcılar

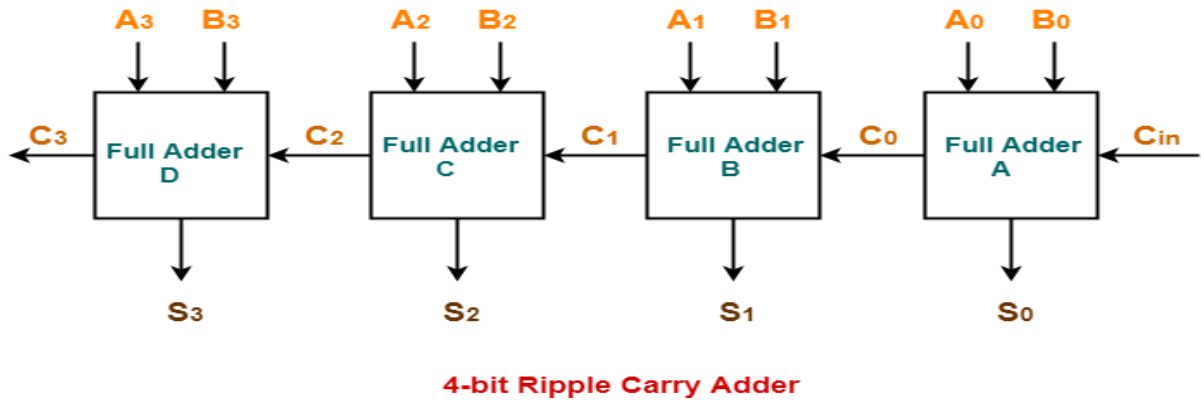
### 1.YÖNTEM

**Yöntem ismi:** Ripple Carry Adder

**Nasıl Çalıştığı:**

Ripple Carry Adder yapısında birbirine sıralı bir şekilde bağlı full adderlar içerir, bu nedenle, her full adder aşamasında elde(carry) üretilecektir. Her full adder aşamasındaki bu elde çıktısı, bir sonraki full addera iletilir ve orada ona bir elde girdisi olarak uygulanır. Bu süreç, son full adder aşamasına kadar devam eder. Böylece, her elde çıktı biti, full adderın bir sonraki aşamasına aktarılır. Bu süreç ne kadar bit toplanmak isteniyorsa o kadar tekrarlanır.

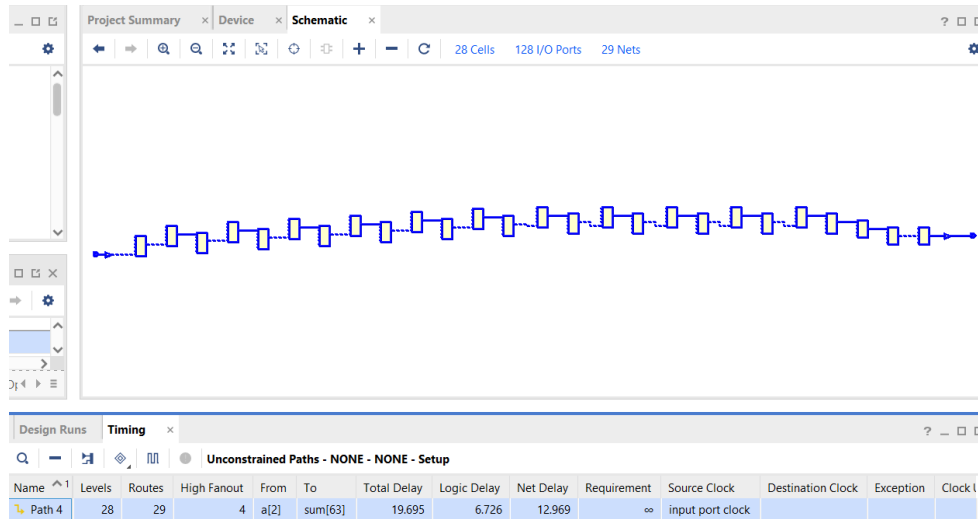
Mimarisi:



**Sentez sonucu kaynak tüketimi:**

LUT 108

**Gecikmesi:** 19.695 (Görselde en uzun yol ve gecikmesi bulunmaktadır)



## 2.YÖNTEM

**Yöntem ismi:** Kogge Stone Adder

**Nasıl Çalıştığı:**

Çalışma prensibini 3 temel adıma ayırırsak:

### 1. Ön işleme

Bu adım, her gelen bitin üretme ve yayma sinyallerinin hesaplanmasını içerir. (Kodda kare modülü)

A(1. N bitlik sayı girdisi) ve (2. N bitlik sayı girdisi)B'deki bit çifti, i 0-N arası bir değer. Bu sinyaller aşağıdaki mantık denklemleriyle verilir:

$$p_i = A_i \text{ xor } B_i \text{ (yayma)}$$

$$g_i = A_i \text{ and } B_i \text{ (üretim)}$$

### 2. İleriye bakma ve Gruplama

Bu adım, her bir bite karşılık gelen eldelerin(carry) hesaplanmasını içerir (Kodda bakınız Cember modülü). (bir önceki adımdan oluşan p'ler)Yayımları ve (bir önceki adımdan oluşan g'ler) üretimleri gruplar, bu gruplama özel bir gruplamadır ve eldeleri içerir. Aşağıdaki mantık denklemleri tarafından verilen ara sinyaller olarak üretir:

$$P_{i:j} = P_{i:k+1} \text{ and } P_{k:j} \text{ (gruplanmış yayılımlar)}$$

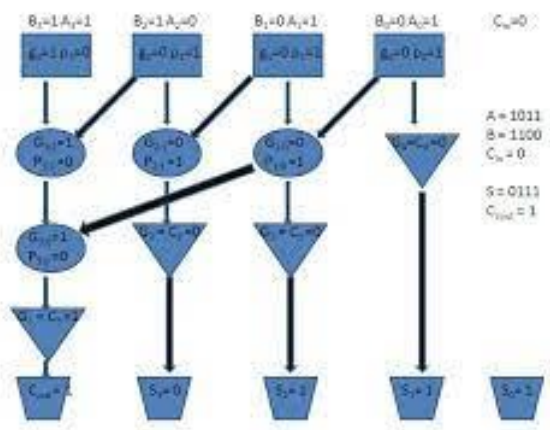
$$G_{i:j} = G_{i:k+1} \text{ or } (P_{i:k+1} \text{ and } G_{k:j}) \text{ (gruplanmış yayılım ve üretimler)}$$

### 3. Son işleme

Bu son adımdır (Kodda bakınız Ucgen modülü).Toplam bitlerin hesaplanmasını içerir. Toplam bitler, aşağıda verilen mantıkla hesaplanır:

$$S_i = p_i \text{ xor } C_{i-1}$$

Mimarisi:



**Sentez sonucu kaynak tüketimi:**

LUT 349

**Gecikmesi:**

17.835

### 3.YÖNTEM

**Yöntem ismi:** Carry Skip Adder

**Nasıl Çalıştığı:**

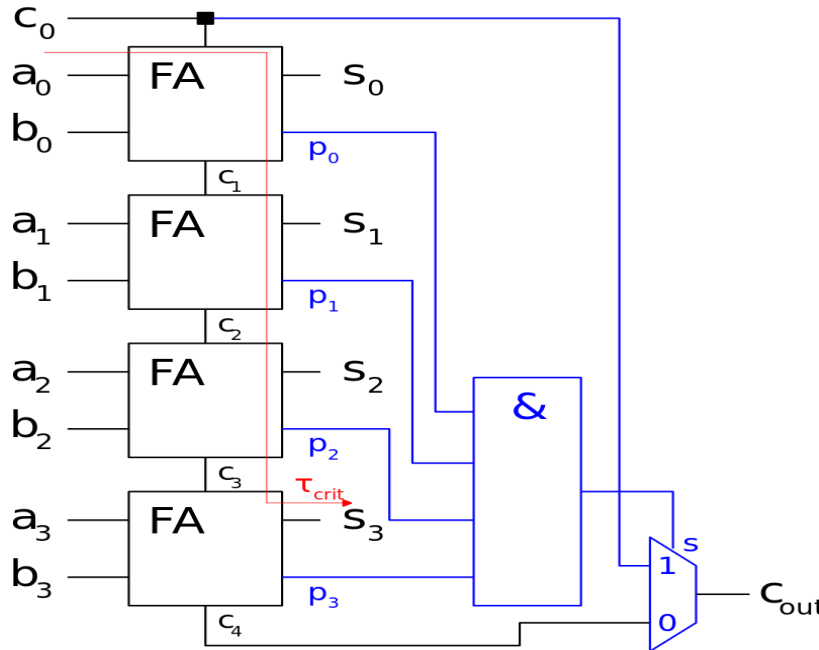
N bit Carry Skip Adder, bir n-bitlik full adder taşıma zinciri, bir n-girişli and kapısı ve bir multiplexer'dan oluşur.

Full adder taşıma zinciri atlama yolu içerir. Bu yol ilk tam toplayıcıda başlar, tüm toplayıcılardan geçer ve sn-1 toplam bitinde biter. Atlama mantığı, bir m-girişli AND geçidi ve bir çoklayıcıdan oluşur.

$$TSK = TAND(m) + TMUX$$

Full adder taşıma zinciri tarafından sağlanan her yayılma biti pi, n-girişli and kapısına bağlanır. Ortaya çıkan bit, ya son taşıma biti cn'yi ya da c0'ı cout'a çeviren multiplexer ın karar biti olarak kullanılır. Yayılma sinyalleri paralel olarak hesaplandığından ve erken erişilebilir olduğundan, bir taşıma atlamalı toplayıcıdaki atlama mantığı için kritik yol, yalnızca çoklayıcı tarafından dayatılan gecikmeden oluşur

**Mimarisi:**



**Gecikmesi:**

19.695

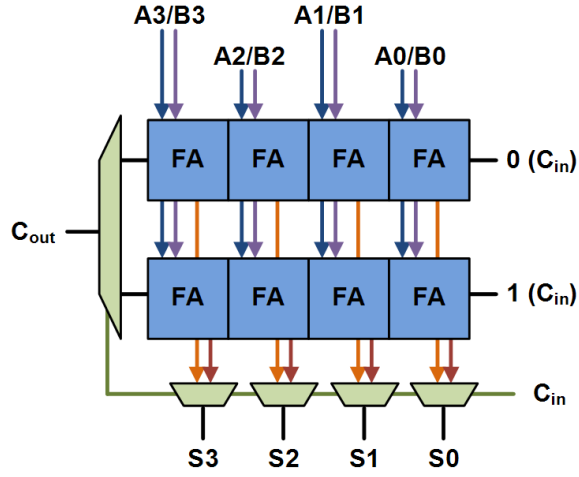
#### 4.YÖNTEM

Yöntem ismi: Carry Select Adder

Nasıl Çalıştığı:

N-bit Carry Select Adder, ripple carry adder ın tersine eldeyi sırayla bitten bit'e yaymaktan kaçınmaktadır. Paralel olarak iki toplayıcımız varsa: birinin elde girişi 0, diğerinin elde girişi 1, o zaman üretilen gerçek elde girişini iki paralel toplayıcının çıkışları arasında seçim yapmak için multiplexer kullanabiliriz. Bu, tüm toplayıcıların hesaplamalarını paralel olarak gerçekleştirebileceği anlamına gelir.

Mimarisi:



Gecikmesi:

19.695