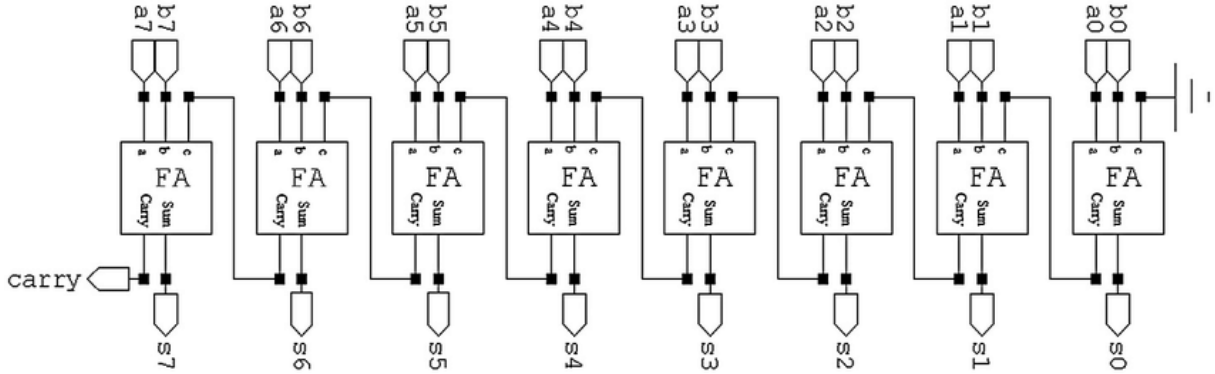


BIL265 PROJE RAPORU

1-) RIPPLE CARRY ADDER:



(Kaynak: <https://www.researchgate.net/publication/283037309/figure/fig2/AS:454461651984390@1485363509931/Eight-bit-Ripple-Carry-adder.png>)

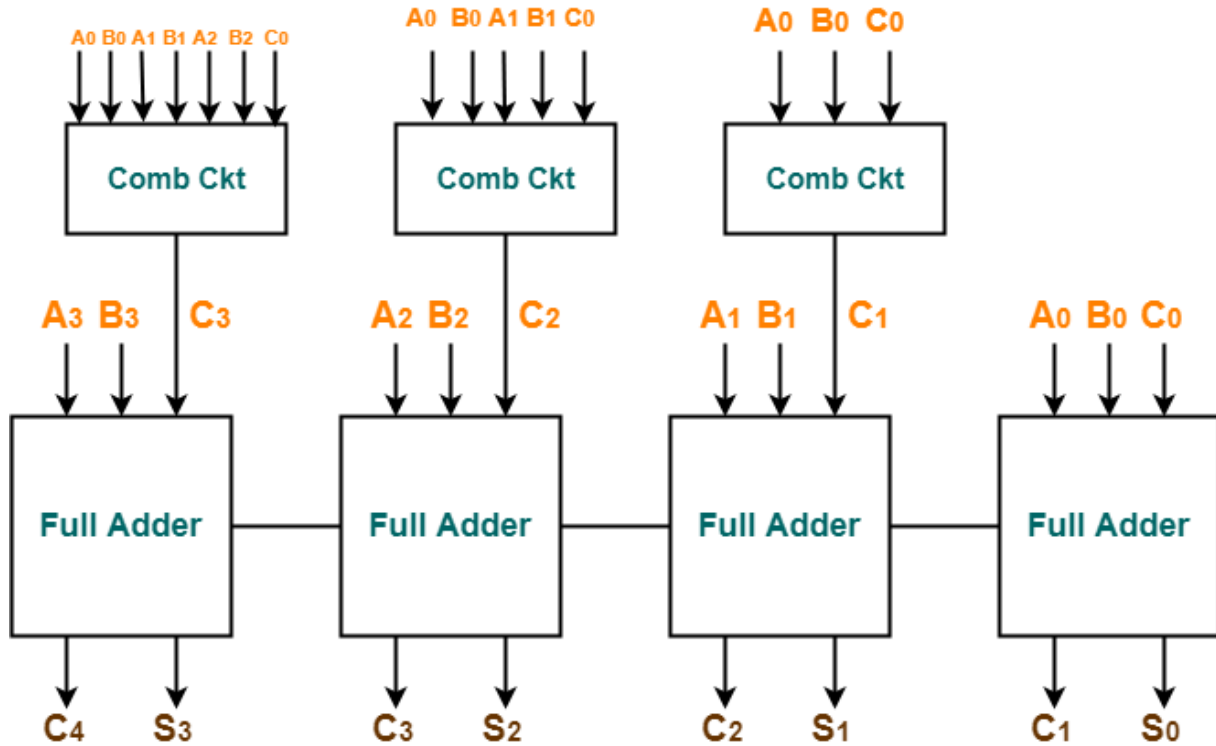
Tasarlanan Ripple Carry Adder, bir sayı sisteminde iki sayıyı toplamak için kullanılan bir elektronik devredir. Bu tip bir toplama devresi, sayıları bitlerine ayırarak toplamayı yapar.

Ripple carry adder'in mimarisi, bir dizi toplama ünitesinden oluşur. Her toplama ünitesi, bir çift sayının bir bitlerini toplar ve bir taşıma biti üretir. Bu taşıma biti, bir sonraki toplama ünitesine geçirilir ve orada bir üst basamağın toplama işlemine dahil edilir. Bu şekilde, taşıma bitleri sırayla ilerleyerek tüm sayının toplama işlemi tamamlanır. Ripple carry adder'in avantajları, yapısının basit olması ve yapımının kolay olmasıdır. Ayrıca, bu tip bir toplama devresi çok sayıda bit için de kullanılabilir.

Ancak, ripple carry adder'in bir dezavantajı, toplama işlemi için gereken gecikme süresinin uzun olmasıdır.

Ripple carry adder'in kaynak tüketimi, parametrik olarak tanımlandığı için kullandığı elemanların bit uzunluğuna göre değişebilir.

2-) CARRY LOOKAHEAD ADDER:



Carry Look Ahead Adder Logic Diagram

(Kaynak: <https://www.gatevidyalay.com/wp-content/uploads/2018/06/Logic-Diagram-of-Carry-Look-Ahead-Adder-1.png>) (Resim temsildir projede parametrik olarak tasarlanmıştır.)

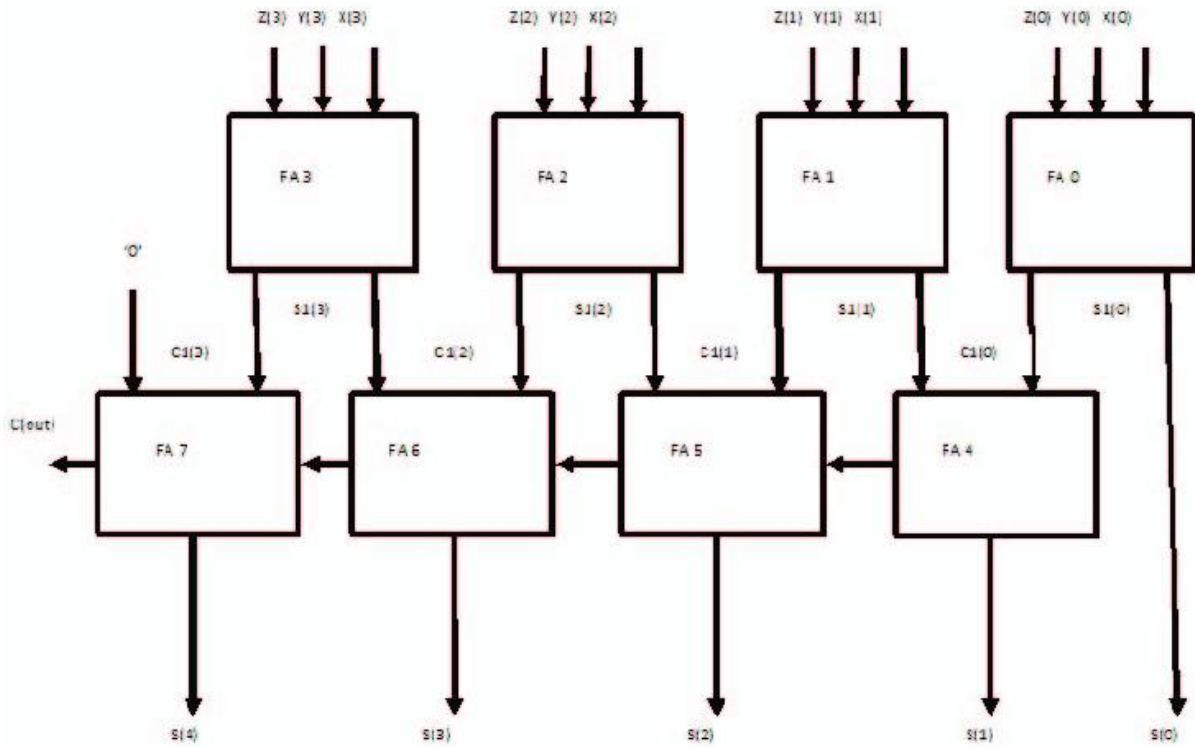
CarryLookAhead adder, sayısal sistemlerde sayıların toplamını hesaplamak için kullanılan bir tasarım tipidir. Bu tasarım tipi, bir sayının basamaklarının birbirlerine eklenmesi işlemini gerçekleştirir ve bir "taşıma" oluştuğunda, sonraki basamağa taşınır. CarryLookAhead adder, ripple carry adder'a göre daha hızlı çalışan bir tasarımdır çünkü taşımaların oluşumu önceden hesaplanır ve böylece ekleme işlemi sırasında taşımaların bir basamaktan diğerine taşınması gerekmez. CarryLookAhead adder'ın çalışma yöntemi şöyle özetlenebilir:

CarryLookAhead adder, eklenecek sayıların basamaklarının birbirlerine eklenmesi işleminin öncesinde taşımaların oluşumunu hesaplar. Bu işlem, "P" ve "G" adı verilen iki sinyale dayanır. P sinyali, bir taşımının oluşup oluşmayacağını hesaplar ve G sinyali ise, taşımının bir sonraki basamakta oluşup oluşmayacağını hesaplar.

CarryLookAhead adder, hesaplanan P ve G sinyallerine göre eklenecek sayıların basamaklarını birbirlerine ekler. Bu adımda, taşımaların bir basamaktan diğerine taşınması gerekmez çünkü taşımalar önceden hesaplanmıştır.

CarryLookAhead adder, ekleme işleminden sonra oluşan taşımaları bir sonraki basamakta toplar. Bu adımda, ekleme işleminin bir basamaktan diğerine taşınması gerekmez çünkü taşımalar önceden hesaplanmıştır.

3-) CARRY SAVE ADDER:



(Kaynak: <https://www.researchgate.net/profile/Praneet-Raj/publication/323349524/figure/fig2/AS:1064915628478464@1630907079344/4-bit-Carry-Save-Adder.png>) (Resim temsilidir tasarımı parametrik olarak tasarlanmıştır)

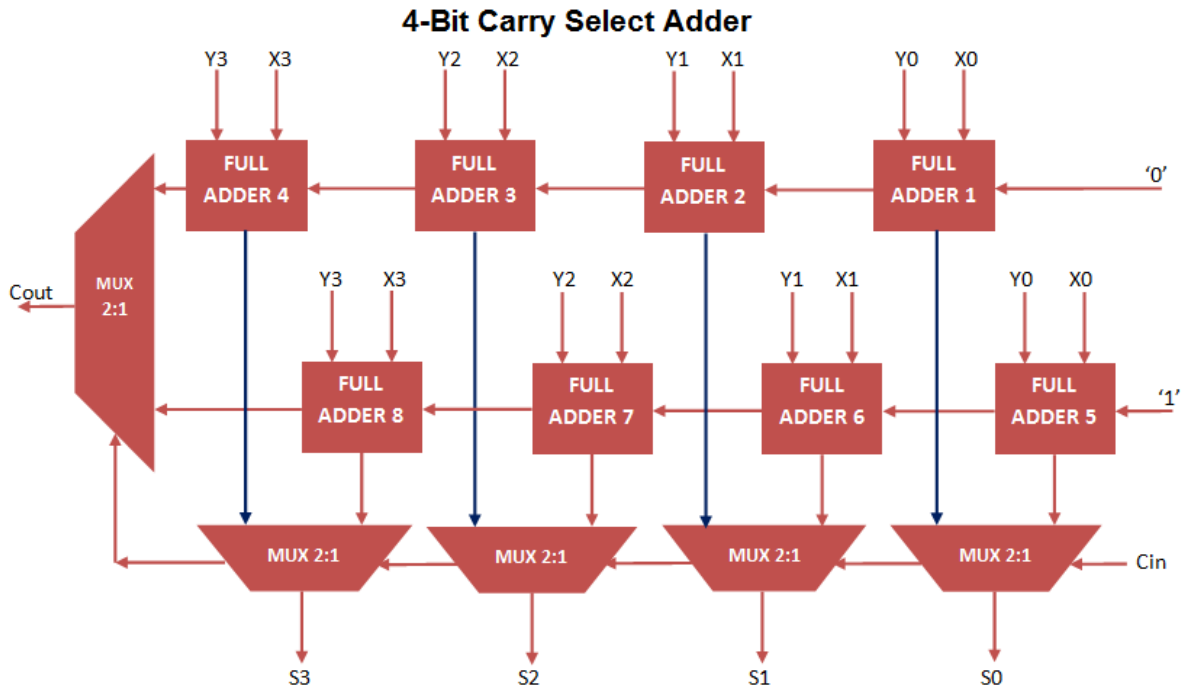
Carry Save Adder (CSA), iki veya daha fazla sayının toplamını hızlı bir şekilde bulmayı amaçlayan bir elektronik devre elemanıdır.

CSA, daha önceki adderlerden daha hızlı çalışır çünkü, ayrıca sayıları toplamak yerine, her bir sayının taşıma bitlerini ayrı bir çıktı olarak verir. Bu sayede, daha sonraki işlemlerde taşıma bitlerinin nasıl işleneceğine dair bir bilgi sahibi olunur ve bu da toplama işleminin hızını artırır.

CSA, özellikle yüksek hızlı işlem yapmayı gerektiren uygulamalar için kullanılır. Örneğin, bir bilgisayarın işlemcisinde veya bir radyo frekansı (RF) ölçüm cihazında kullanılabilir.

Diğer adderlerle karşılaştırıldığında, CSA genellikle daha hızlı çalışır. Ancak, CSA'nın kullanımı daha karmaşık olduğundan, bu devre elemanının tasarımı ve uygulaması daha zordur.

4-) CARRY SELECT ADDER:



(Kaynak: <https://allaboutfpga.com/wp-content/uploads/2016/06/carry-select-adder-VHDL-Code.png>)

Carry Select Adder (CSelA), iki veya daha fazla sayının toplamını hızlı bir şekilde bulmayı amaçlayan bir elektronik devre elemanıdır.

CSelA, toplama işlemini hızlandırmak için Carry Lookahead Adder (CLA) ve Carry Save Adder (CSA) gibi diğer adderlerin özelliklerini birleştirir. CSA'nın çalışma yöntemi, verilen sayıların taşıma bitlerini önceden hesaplayarak işleme sokmaktır. Bu sayede, toplama işlemi sırasında taşıma bitlerinin nasıl işleneceğine dair bir bilgi sahibi olunur ve bu da toplama işleminin hızını artırır. CSelA, toplama işlemi sırasında taşıma bitlerinin hesaplanması için Carry Lookahead Adder'ı kullanır. Ancak, CSA aynı zamanda Carry Save Adder'ı da kullanarak, toplama işlemi sırasında taşıma bitlerinin ayrı bir çıktı olarak verilmesini de sağlar. CSA'nın tasarımında, Carry Lookahead Adder ve Carry Save Adder gibi alt modüller kullanılır.

Carry Lookahead Adder, toplama işlemi sırasında taşıma bitlerinin hesaplanması için kullanılır. Carry Save Adder ise, toplama işlemi sırasında taşıma bitlerinin ayrı bir çıktı olarak verilmesini sağlar. Diğer adderlerle karşılaştırıldığında, CSA genellikle daha hızlı çalışır. Ancak, CSA'nın tasarımı ve uygulaması daha karmaşıktır ve bu nedenle, diğer adderlerden daha pahalı olabilir. Sonuç olarak, Carry Select Adder, hızlı sayı toplama işlemleri gerektiren uygulamalarda kullanılır ve diğer adderlerden daha hızlı çalışır.

Tasarımı ve uygulaması pek çok alt modül bulundurduğu için daha karmaşıktır ve bu nedenle diğer adderlerden daha pahalı olabilir. CSA'nın tasarımında Carry Lookahead Adder, MUX, Carry Save Adder gibi alt modüller kullanılır.

ADDERLERİN KARŞILAŞTIRILMASI

Projede yapılan tasarımların tümü de birkaç sayıyı toplamanın yapılması için kullanılan sayısal elemanlardır. Ancak bu adder türleri, birbirlerinden farklı yöntemlerle çalışır ve farklı özelliklere sahiptir.

Ripple carry adder, basit bir toplama işlemini gerçekleştirir ve bir önceki çıktının taşıma bitini kullanarak bir sonraki işlemin toplamını hesaplar. Bu, bir sayının birkaç basamaklı olduğu durumlarda daha yavaş çalışabilir, ancak yapısı kolaydır ve düşük maliyetli bir seçenektir.

Carry lookahead adder ise, bir önceki çıktının taşıma bitini kullanmaz ve taşıma bitlerini önceden hesaplayarak daha hızlı çalışır. Bu adder türü, daha büyük sayıları hızlı bir şekilde toplamak için kullanılabilir ancak yapısı daha karmaşıktır ve daha pahalıdır.

Carry save adder ise, toplama işlemini parçalara bölerek hızlı bir şekilde gerçekleştirir. Bu adder türü, daha büyük sayıları toplamak için kullanılabilir ancak çıktının düzgün bir şekilde okunması için ek adımlar gerektirir.

Carry select adder ise, iki farklı adder türünü kullanarak toplama işlemini gerçekleştirir. Bu adder türü, toplama işleminin hızını arttırmak için kullanılabilir ancak yapısı daha karmaşıktır ve daha pahalıdır.

Genel olarak, hangi adder türü kullanılacağı, toplama işleminin boyutu, hızı ve maliyeti gibi faktörlere göre belirlenir.

TESTLER:

Yazılan testbench dosyaları bütün modüller için aynı durumları test etmektedir. Ve bu durumlar (2 çok bitlik sayı girişi) rastgele belirlenmiştir. (FullAdder modülü için ayrı tb dosyası vardır.)

KATKI:

Mert Başar Yıldırım (211101039)(mertbasaryildirim@etu.edu.tr)

- Bazı modüllerde hiyerarşik olarak kullanılan FullAdder modülünü yazdı.
- CarryLookAhead modülünü yazdı.
- CarryLookAhead modülü için testbench dosyası yazdı.
- uart_fpga modülünü yazdı.
- uart_fpga modülü için gerekli "constrainsforfpga" dosyasını yazdı.
- Raporu yazdı. (Herkes kendi kısmıyla ilgili bilgi verdi, bu kişi word belgesinde toplayıp pdf'e çevirdi)

Pelin Buçukoğlu (191101059)(pbucukoglu@etu.edu.tr)

- CarrySaveAdder modülünü yazdı.
- CarrySaveAdder modülü için testbench dosyası yazdı.
- Github hesabı oluşturdu.

Ahmet Tanrikulu (201201066)(ahmettanrikulu@etu.edu.tr)

- CSelA8 (CarrySelectAdder) modülünü yazdı.
- CSelA8 modülünde kullanılan 4 bitlik "RippleCarryAdder", "MUX2to1" ve "FullAdder" alt modüllerini yazdı.
- CSelA8 modülü için test bench dosyası yazdı.

Egemen Çorbacı (211201026)(mcorbaci@etu.edu.tr)

- FPGA'yı ödünç aldı.
- RippleCarryAdder modülünü yazdı.
- RippleCarryAdder modülü için testbench dosyası yazdı.