

Bilgisayar Donanımı

Furkan : İSİM
Kaya : SOYİSİM

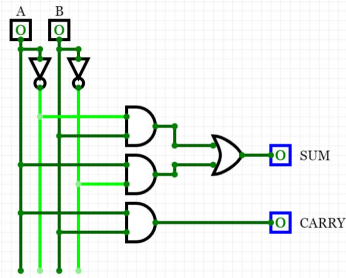
8-bit adder, ALU

191216002 : NUMARA

Merhaba,

Bu raporu BIL003 kodlu Bilgisayar Donanımı dersinde verilmiş olan 02 numaralı ödev için hazırladım. İyi okumalar dilerim.

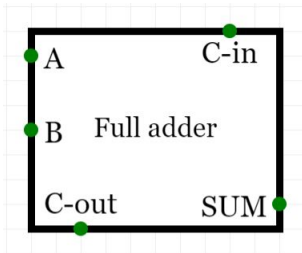
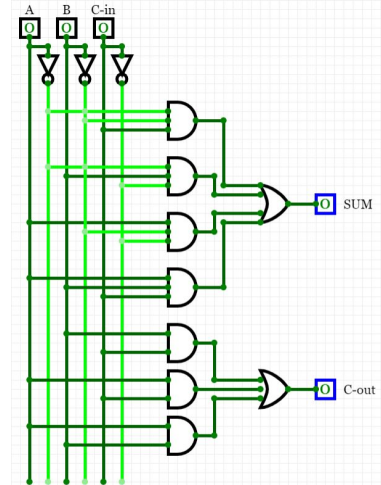
8-bit ADDER



8-bit adder devresi oluştururken işe half adder tasarlayarak başladım ama sonra direkt olarak full adder tasarlayabileceğimiz aklıma geldi ben de bu oluşturduğumu bi kenara bırakıp full adder tasarladım.

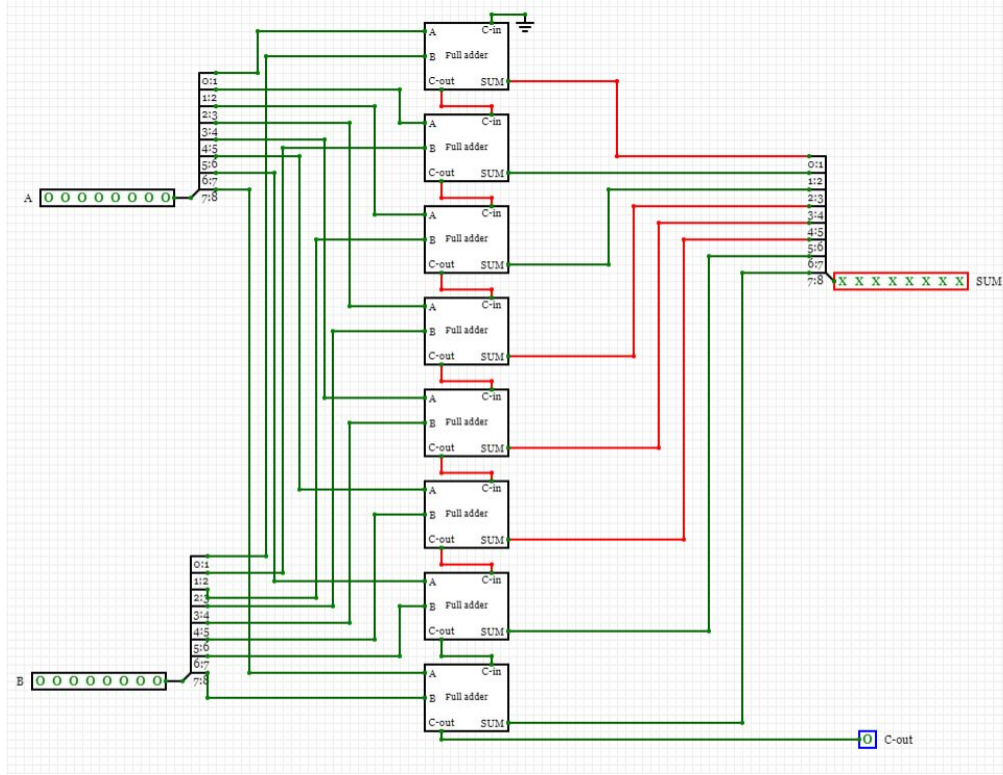
Tam olarak ben de

tasarlamadım aslında Combinational analys toolu ile oluşturdum.



Nasıl görüneceğini yerleşim görünümde ayarladım ve 8 bit adder devremi tasarlamaya başladım.

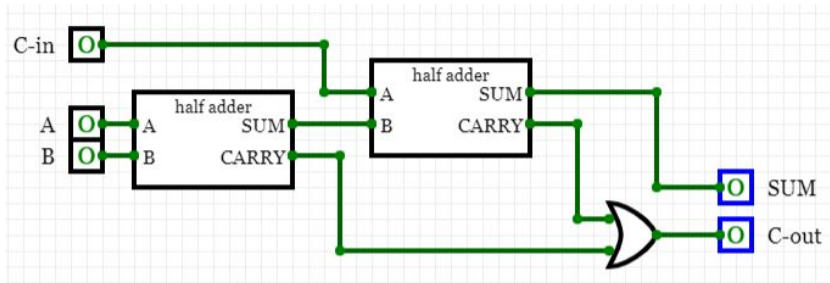
İşe 8 adet full adderı ekleyerek başladım. bunların c-in ve c-out çıkışlarını birbirine bağladım. İlk adderda c-in değerimiz olmadığı için bunu toprak hattına bağladım.



Son adderda ise işlemden gelebilecek olası elde değerimiz için output çıkarttım.

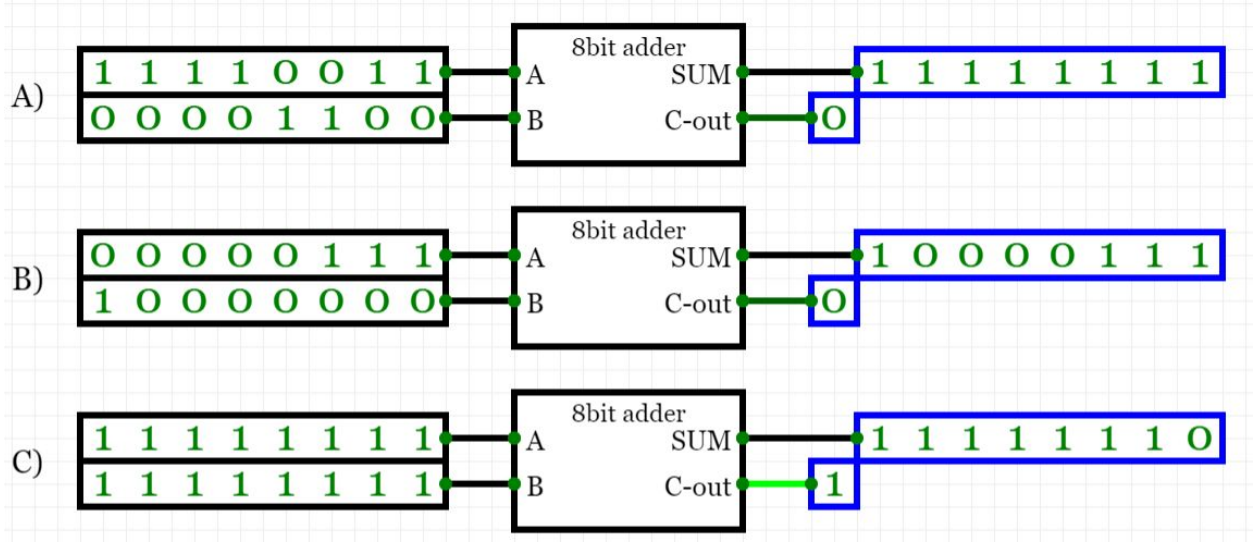
8 bitlik input oluşturdum bunu 8 bit splitter ile tüm adderlara sırasıyla dağıttım.

Aynı şekilde diğer değerim içinde oluşturdum ve bunların bitlerini de gerekli adderlara ekledim. Hepsinin sum değerini sırasıyla output splitterında topladım ve 8 bitlik output objemde gösterdim.

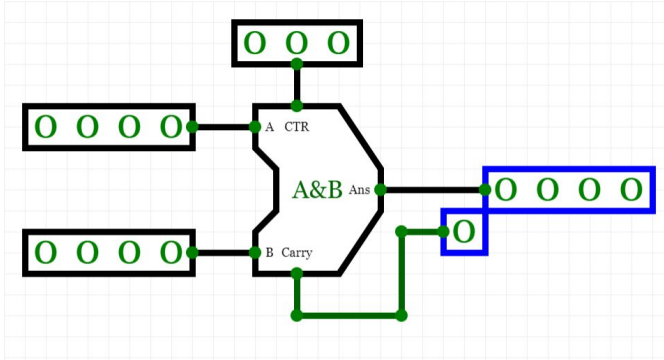


İlk oluşturduğum half adderler da boşa gitmesin diye öğrenmek amacıyla 2 half adder ı birleştirerek full adder oluşturdum.

Ödevde verilen işlemler için 8 bit adder devrimi maine yerleştirdim. A-B-C olarak şıkladım.



ALU



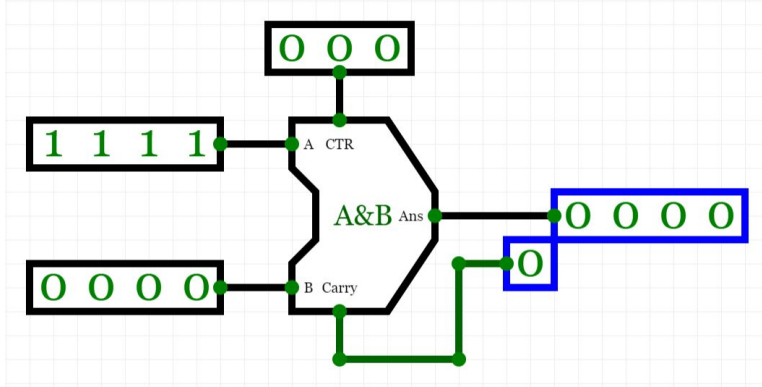
Alu objesini incelemek için main sayfamda solda gördüğünüz gibi devrimi oluşturdum. Control girişinin alabileceği 3 bitlik input sayesinde 2'lik sistemde 8 sayısına kadar ulaşabiliyoruz. ALU'ya da her biri için bir görev atanmış.

Circuitverse'nin ALU birimi için oluşturduğu dokümantasyonundan aldığım tabloyu sağda görebilirsiniz.

Opcode	Opcode Value	Operation
0 0 0	0	A & B
0 0 1	1	A B
0 1 0	2	A + B
1 0 0	4	A & ~B
1 0 1	5	A ~B
1 1 0	6	A - B
1 1 1	7	SLT

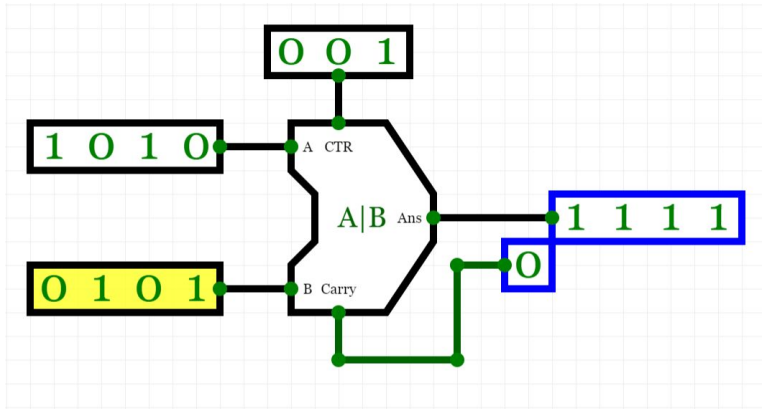
Sırayla inceleyelim;

AND durumu $(000)_2 \mid (0)_{10}$:



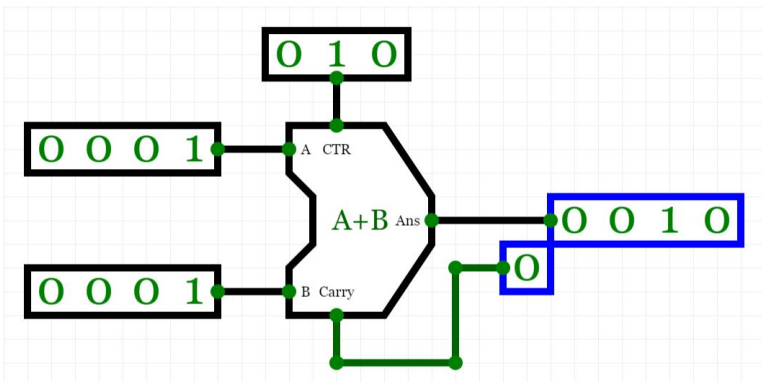
Beklenildiği gibi her bir biti diğer girişimizdeki 0 yok ettiği için sonucumuz 0000 oldu.

OR durumu $(001)_2 \mid (1)_{10}$:



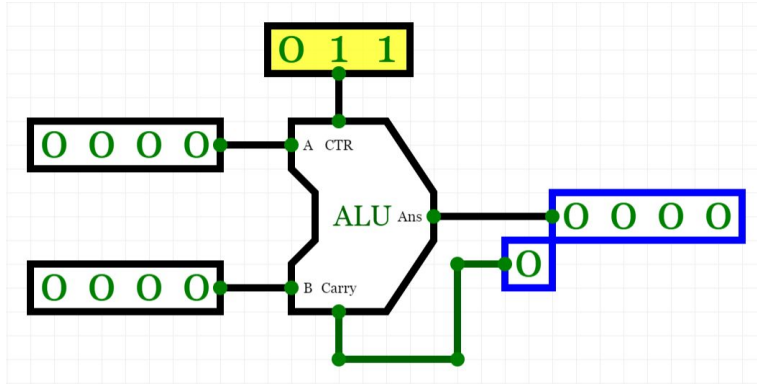
Veya kapısı gibi çalıştı ve herhangi biri 1 ise sonucu 1 döndürdü.

ADDER durumu $(010)_2 \mid (2)_{10}$:



Adder gibi çalıştı. 1 ile 1 i toplattım 2 sonucunu verdi.

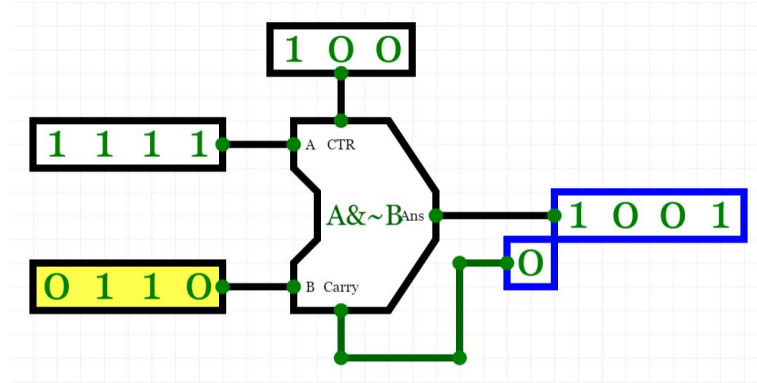
++ durumu $(011)_2 \mid (3)_{10}$:



Bu değer ile aluyu çalıştıramadım dökümantasyonunda da herhangi bir bilgi yoktu 3 ü neden atladıklarını bilmiyorum ama idle durumu için bırakmış olma ihtimallerini düşünüyorum. Vikipadiadan baktım ki ++ durumu imiş ama

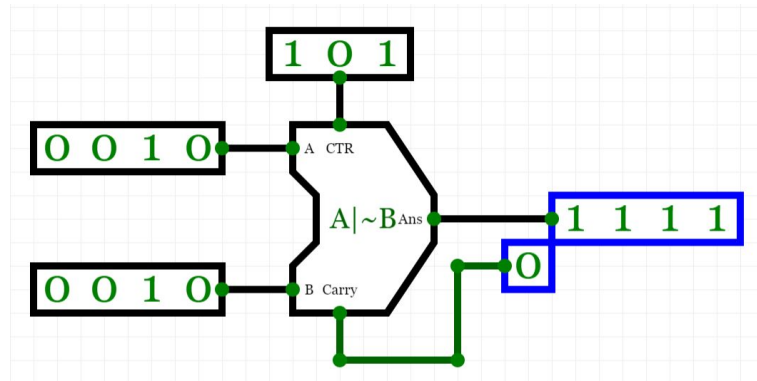
circuitverste bulunmuyor.

NAND durumu $(100)_2 \mid (4)_{10}$:



NAND kapısı gibi davrandı. AND kapısının 1 olması gereken yerde 0, 0 olması gereken yerde 1 oldu.

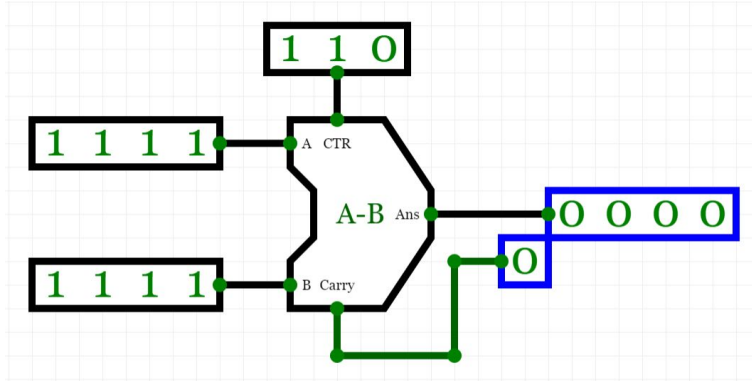
NOR durumu $(101)_2 \mid (5)_{10}$:



Kendi dökümantasyonunu vs araştırdım ama $\mid \sim$ sembolünün ismini bulamadım xor olarak sonuç tutmadı $\& \sim$ olduğu gibi n olabilir dedim ama nor kapısı olaması için 1 1 in de 0 olması

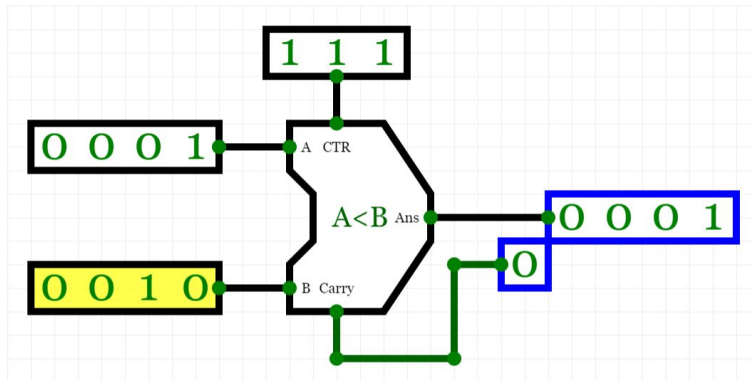
gerekirdi. Bunun hakkında araştırma yapacağım.

SUBTRACT durumu $(110)_2 \mid (6)_{10}$:



13 den 13 çıkarırsak 0 kalır bu da çalışıyor. :)

LESS THAN durumu $(110)_2 \mid (6)_{10}$:



1 < 2 i kontrol etti ve sonuç true olduğu için 1 çıktısını verdi.

Raporumu okuduğunuz için teşekkürler. İyi günler dilerim.