

CSE331 - Computer Organization HW2



1 bitlik XOR

32 tane yapılıp 32 bit XOR işlemi gerçekleştirilir

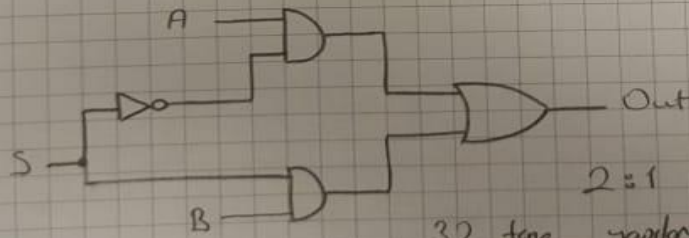


1 bitlik NOR

32 tane yapılıp 32 bit NOR işlemi gerçekleştirilir

Full Subtractor

↳ 32 bitlik full adder kullanılır. Çıkarılacak sayının NOR', alınır full adder'a giden Cin'in de NOR', alınır. full adder farkıyı çıkarılır. Yeni çıkarılacak sayının NOR', alınıp toplanır, 1 eklenir



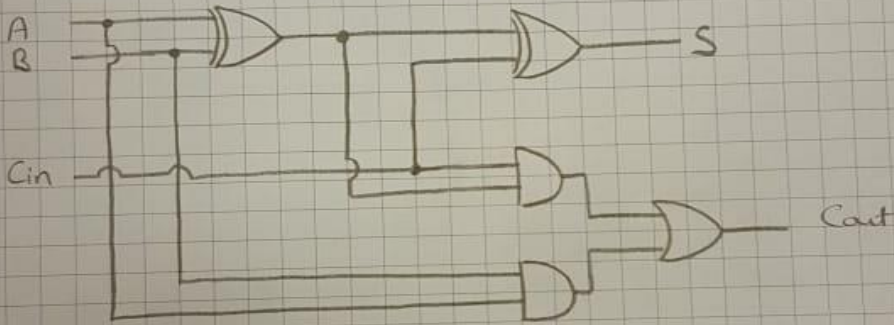
2:1 1 bitlik mux

32 tane yapılarak 2:1 32 bitlik mux elde edilir



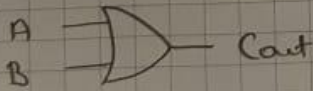
1 bitlik AND

32 tane yapılıp 32 bit AND
işlemi gerçekleştirilir.



1 bitlik full adder

Full adder 32 bitlik yapmak için 32 tane
full adder işlemi gerçekleştirilir.



1 bitlik OR

32 tane yapılıp 32 bitlik OR
işlemi gerçekleştirilir.

AND 32 BIT

32 bitlik and işlemi için 1 bitlik 2 inputlu and kapısı kullandım. 32 kere and kapısını inputların her bir biti için kullanarak 32 bitlik iki sayıyı and yapabildim.

OR 32 BIT

32 bitlik or işlemi için 1 bitlik 2 inputlu or kapısı kullandım. 32 kere or kapısını inputların her bir biti için kullanarak 32 bitlik iki sayıyı or yapabildim.

XOR 32 BIT

32 bitlik xor işlemi için 1 bitlik 2 inputlu xor kapısı kullandım. 32 kere xor kapısını inputların her bir biti için kullanarak 32 bitlik iki sayıyı xor yapabildim.

NOR 32 BIT

32 bitlik nor işlemi için 1 bitlik 2 inputlu nor kapısı kullandım. 32 kere nor kapısını inputların her bir biti için kullanarak 32 bitlik iki sayıyı nor yapabildim.

FULL ADDER 1 BIT

Yukarda görseli olan 1 bitlik full adder tasarladım.

FULL ADDER 32 BIT

32 bitlik full adder tasarlamak için 32 tane 1 bitlik full adder kullandım. Burada yapılan işlemler şu şekildedir:

0. bitler toplanırken carry_in 0 verilir ve sıfırıncı bitler 1 bitlik full addera gönderilir.
 1. bitler toplanırken carry_in kısmı 0. Bitlerin toplanmasından gelen carry_out olacaktır.
 2. bitler toplanırken carr_in kısmı 1. Bitlerin toplanmasından gelen carr_out olacaktır.
- Bu işlem 3.4.5.....31 bitler için gerçekleştirilir.

FULL SUBTRACTOR 32 BIT

32 bitlik full subtractor tasarlamak için 32 bitlik full adder kullandım. Burada yapılan işlemler şu şekildedir:

Çıkarma işlemi yapılırken çıkartılacak sayının not' ı alınır toplama işlemi yapıp en son 1 ekleme işlemi yapıldığından A-B işlemi için öncelikli olarak B nin her bitini not kapısı kullanarak değiştirdim. 1 ekleme işini carry_in ' e 1 vererek yaptım. Full adder da carry_in ilk değer olarak 0 aldığından 32 bitlik full subtractor fonksiyonun da carry_in' i de not kapısı kullanarak değiştirdim. Son olarak bu değişikliklerle full adder fonksiyonunu çağırdım.

2:1 MUX 1 BIT

Yukarıda görseli 1 bitlik 2:1 mux tasarladım.

ARITHMETIC SHIFT RIGHT 32 BIT

32 bitlik A sayıyı right_shift etmek için 2:1 mux kullandım. Yapılan işlemler şu şekildedir:

4 adet wire değişkenim var. B nin B[0], B[1], B[2], B[3], B[4] bitlerinin 2:1 mux' un seçicileri olarak tasarladım. 32 bitlik kaydırma yapabilmek için 5 adım gerekiyor ve her adımda da 32 tane 2:1 mux vardır.

1. Adım için 2:1 mux' a A[0], A[1], B[0], wr1[0] gönderdim. Yani B[0]' ın değerine göre wr1[0]' a ya A[0] ya da A[1] atanacak. İlk adımda bunlar her bit için tekrarlanacak.

2. Adımda ise artık 2:1 mux' a wr1[0], wr1[2], B[1], wr2[0] gönderilecek. Yani B[0]' ın değerine göre wr2[0]' a ya wr1[0] ya da wr1[2] atanacak.

3. Adımda ise 2:1 lik mux' un girişlerinden biri bir önceki adımda 4. Çıktı olan diğeri ise bir önceki adımın ilk çıktısıdır. Seçici giriş B[2] dir.

4. Adımda ise 2:1 lik mux' un girişlerinden biri bir önceki adımda 8. Çıktı olan diğeri ise bir önceki adımın ilk çıktısıdır. Seçici giriş B[3] tür.

5. Adımda ise 2:1 mux' un girişlerinden biri bir önceki adımda 16. Çıktı olan diğeri ise bir önceki adımın ilk çıktısıdır. Seçici giriş B[4] tür.

Sol taraftaki boşluğa A sayısının most significant bit değeri atanır. Bütün bitler için işlemler yapılır.

SHIFT LEFT 32 BIT

A sayısını or gate' i kullanarak 0 ile bütün bitlerini or' layıp ters şekilde 32 bitlik bir wire' a atadım. Right_shift işleminin aynısını yaptım. Tek fark olarak least significant bit 'e 0 değerini verdim. Ters çevrilmiş wire' ı tekrar 0 ile or' layarak düz çevirdim.

2:1 MUX 32 BIT

2:1 1 bitlik mux lardan 32 tane kullanarak 2:1 32 bitlik mux elde ettim.

8:1 MUX 32 BIT

2:1 32 bitlik mux lardan 7 tane kullanarak 8:1 32 bitlik mux elde ettim.

ALU32

ALU32 de bütün fonksiyonları gerekli parametreleri kullanarak çağırdım ve ayrı ayrı wire değişkenlerine sonuçları atadım.

Bütün wire sonuçlarını 8:1 mux a seçici ve output ekleyerek gönderdim.

8:1 mux seçiciye gelen değere göre de hangisinin outputun atanacağına karar verir.

ALİ HAYDAR KURBAN 151044058