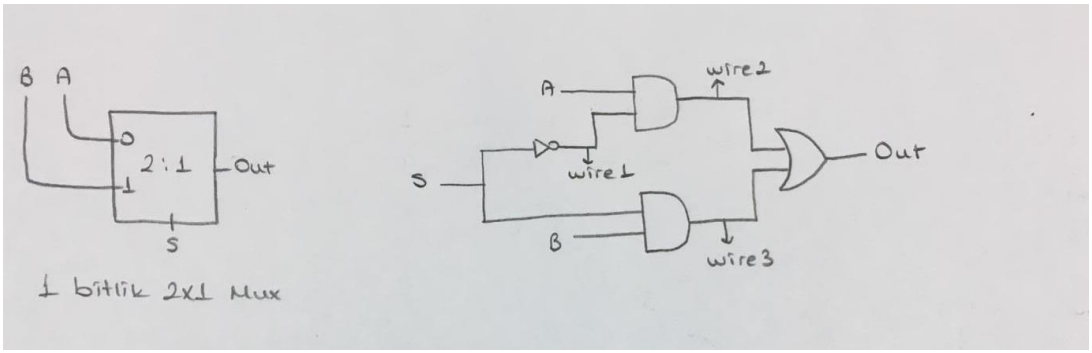
CSE 331-COMPUTER ORGANIZATION

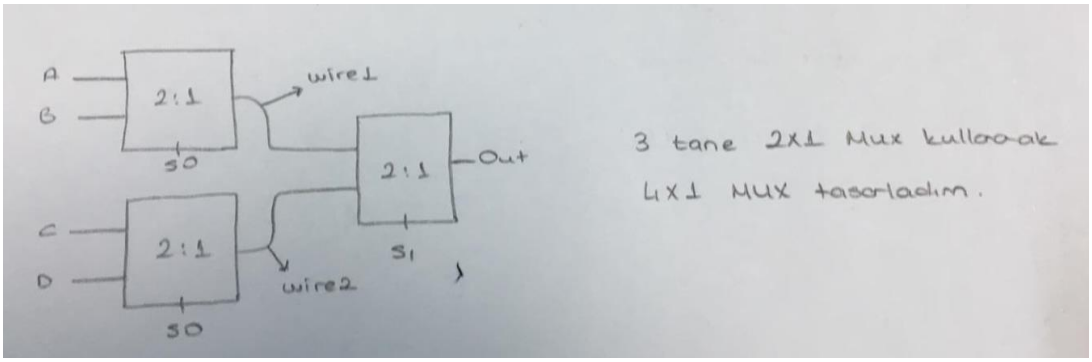
PROJECT2

-MUX_4x1

module MUX_4x1(Out,A,B,C,D,S);



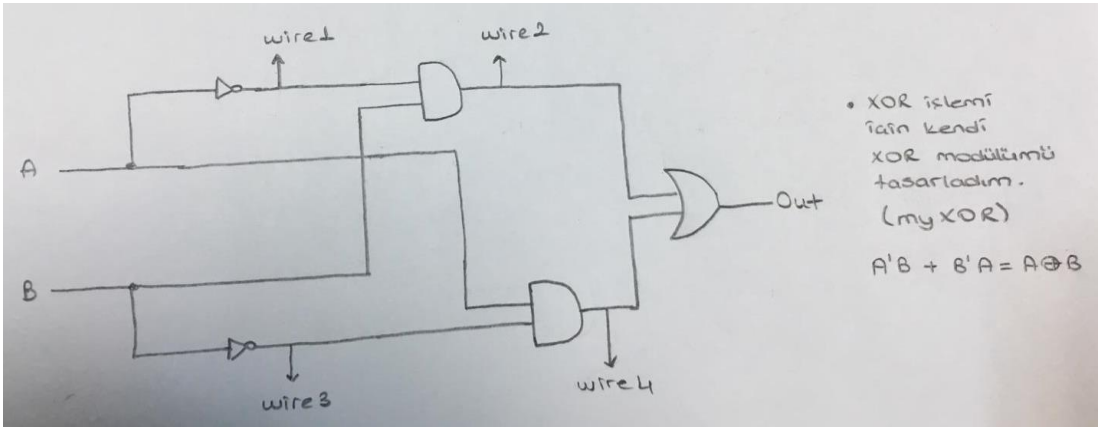
1 Bitlik ALU için 4:1 MUX tasarlamak amacıyla 2:1 1 Bitlik MUX tasarladım. Görseldeki wire değişkenleri **MUX_2x1** modülündeki wire değişkenlerine karşılık gelmektedir.



4:1 MUX tasarlarken 3 tane 2:1 MUX kullandım ve şekildeki gibi tasarladım. Görseldeki wire değişkenleri MUX_4x1 modülündeki wire değişkenlerine karşılık gelmektedir.

-myXOR

module myXOR(Out,A,B);



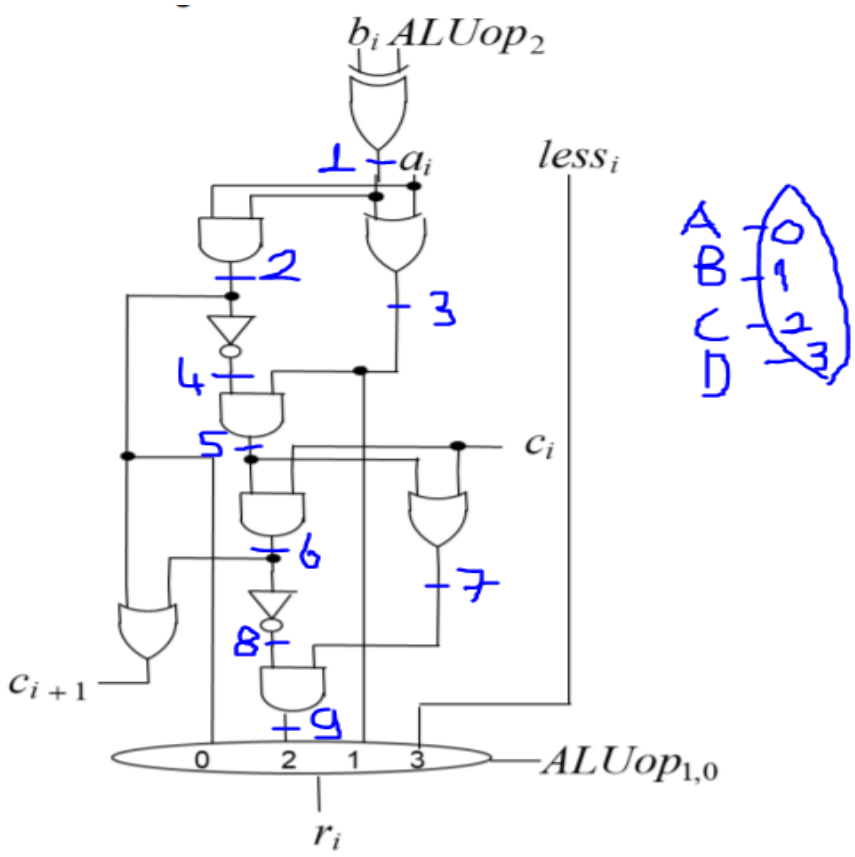
XOR işlemi için kendi XOR modülümü görseldeki gibi tasarladım. Görseldeki wire değişkenleri myXOR modülündeki wire değişkenlerine karşılık gelmektedir.

Bu modüllerin ardından 1 bitlik ALU tasarımı için yeni bir modül oluşturdum(ALU_1Bit).Bu modülü oluştururken proje PDF'indeki ALU tasarımını kullandım.

ALU_1Bit

module ALU_1Bit(Result,A,B,Cin,Less,ALU_OP,Cout);

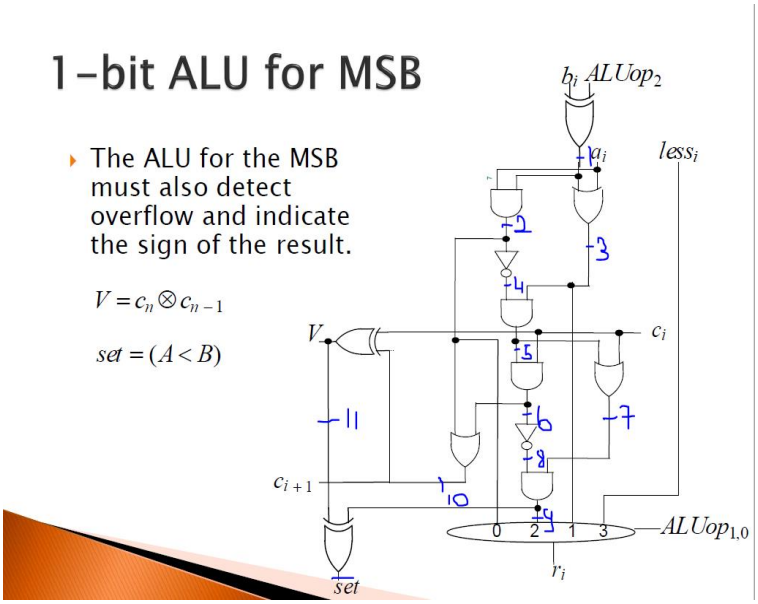
1 Bit ALU modülü oluştururken input olarak 1 bitlik A, B, Cin, Less değişkenlerini ve 3 bitlik ALU_OP değişkenini kullandım. Output olarak ise Result ve Cout değişkenlerini kullandım.



Görseldeki sayılar ALU_1Bit modülündeki wire değişkenlerine karşılık gelmektedir. Şeklin yan tarafında ise tasarladığım MUX u çizdim. Örneğin görseldeki MUX un 0'ı, tasarladığım MUX daki A inputuna karşılık gelmektedir.

-ALU_MSB_1Bit

module ALU_MSB_1Bit(Result,A,B,Cin,Less,ALU_OP,Cout,Set);



Set on less then işlemi yapabilmek için 32 bit ALU oluştururken kullanmak üzere 1 Bit MSB ALU oluşturdum. Bunu 32 Bit ALU da son bit için kullandım. Buradan çıkan set'i 32 bitlik ALU'nun ilk bitine less inputu olarak gönderdim.

Görseldeki sayılar ALU_MSB_1Bit modülünde kullandığım wire değişkenlerine karşılık gelmektedir.

-ALU_32Bit

module ALU_32Bit(Result,A,B,Cin,Less,ALU_OP,Cout);

Bu modüldeki outputlar 32 Bitlik Result değişkeni ve 1 Bitlik Cout değişkenidir. İşlem sonuçları Result değişkeninde tutulur. İnputlar ise üzerinde işlemlerin yapılacağı 32 Bitlik A, B değişkenleri, opcodeları tutacak olan 3 Bitlik ALU_OP değişkeni ve 1 Bitlik Cin ile Less değişkenleridir. Parametre sıralaması yukarıda gösterildiği gibidir.

32 Bit ALU oluştururken bu modülün içerisinde 31 kez 1 Bit ALU modülünü çağırdım. İlk bitten sonra diğer bitlerin Cin değişkenlerine bir önceki bitin Cout sonucunu verdim. Bu sebeple wireCinCout adında 31 bitlik bir wire kullandım. 32 Bit ALU'nun msb biti için ise ALU_MSB_1Bit modülünü kullandım. Buradan çıkan Set outputunu ilk bite Less değişkeni yerine gönderdim. Diğer tüm bitlerde Less değişkenine 0 verdim.

-ALU_32BitTestBench

module ALU_32BitTestBench();

Bu modülün içerisinde tasarladığım 32 Bit ALU'yu test etmek için ALU_32Bit modülünü şu şekilde çağırdım.

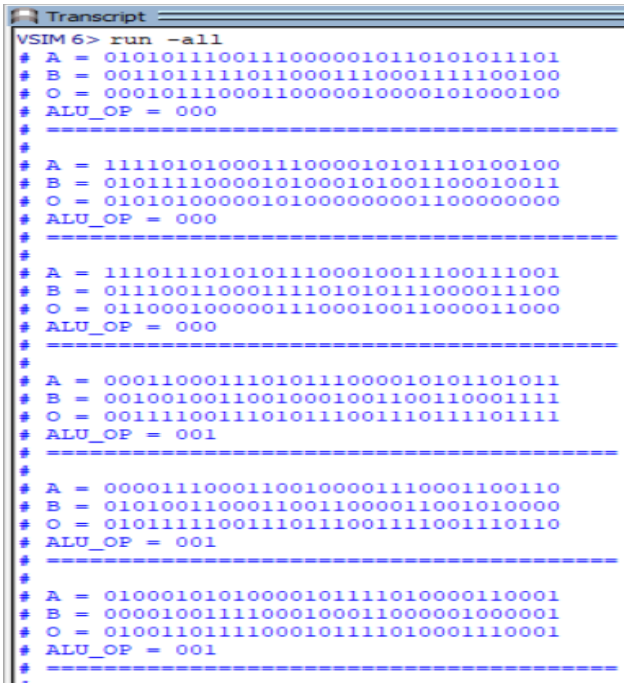
ALU_32Bit my32BitALU(Result,A,B,ALU_OP[2],Less,ALU_OP,Cout);

Burada Cin değişkeni yerine ALU_OP[2] verdim. Çünkü ADD işlemi yapılırken ilk elde bitinin 0, SUBTRACT ve SET ON LESS THEN işlemi yaparken 1 olması gerekmektedir. AND ve OR işleminde bu bit önemli değildir. Bu yüzden ilk elde bitini yapacağım işleme göre vermem gerekmektedir. Slaytlarda verilen opcodlara göre ADD işlemi ile SUBTRACT ve SET ON LESS THEN işlemlerini ayıran bit ALU_OP'un 2. Bitidir. Bu yüzden Cin yerine ALU_OP[2] verdim.

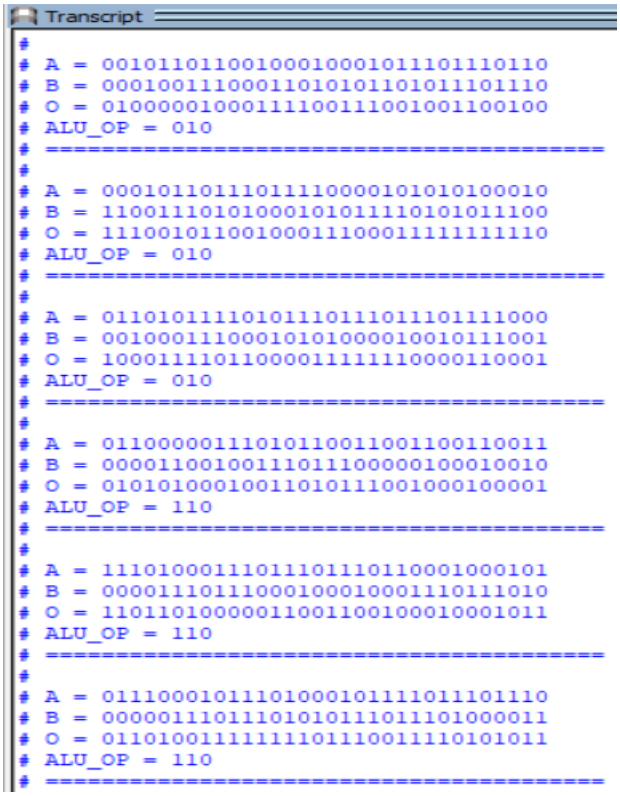
Test ederken AND işlemi için ALU_OP ‘u “000”, OR işlemi için “001” , ADD işlemi için “010” , SUBTRACT işlemi için “110” ve SET ON LESS THEN işlemi için “111” verdim. Her işlemi 3 kez farklı inputlar ile denedim. Sırasıyla inputları(A,B), Result’ı ve ALU_OP’u yazdırdım.

Test Sonuçları

-AND VE OR



-ADD VE SUBTRACT



-SET ON LESS THEN

Kullandığım GATE Sayısı

MUX_4x1 : 12 tane

myXOR : 5 tane

ALU_1Bit : 26 tane

ALU_MSB_1Bit : 37 tane

ALU_32Bit : 843 tane

MEDİNE ASLAN

161044015