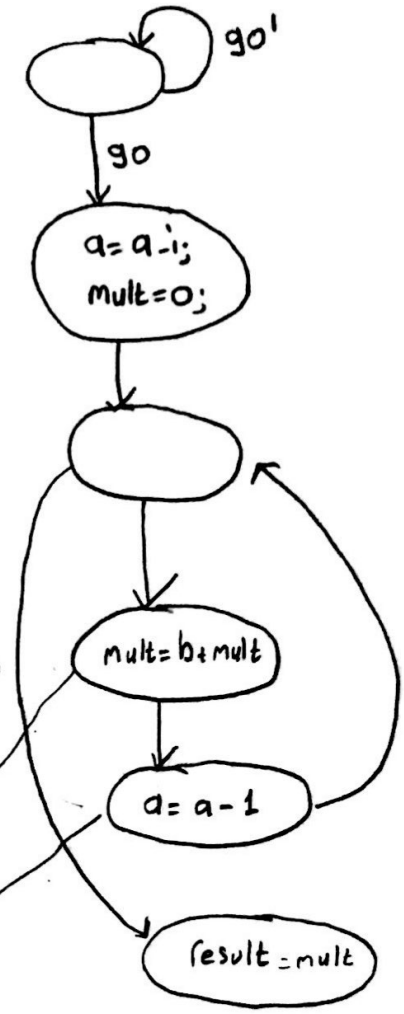
First State Diagram

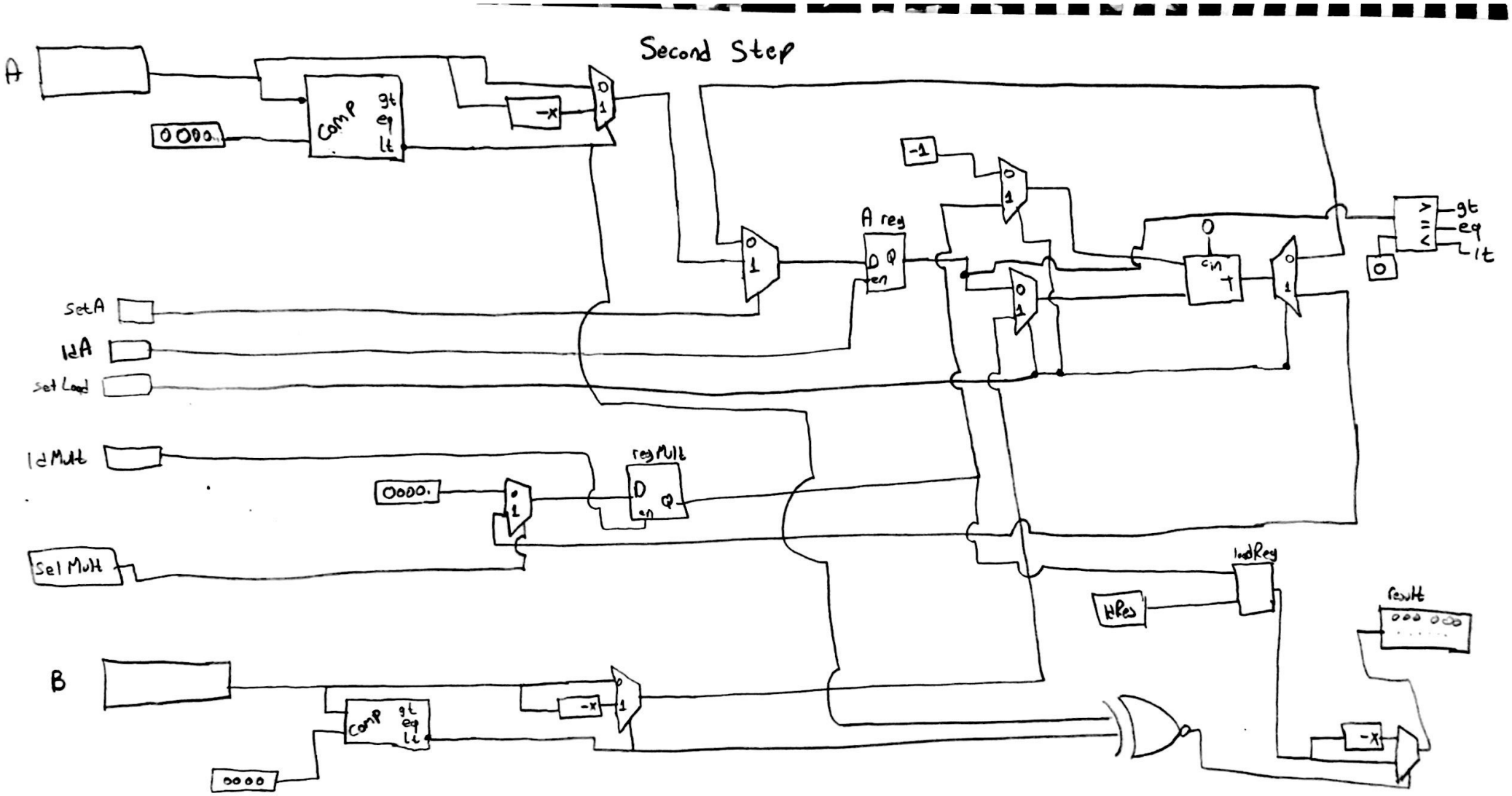
161044063
Zafer ALTAY

```
while (! go);
mult = 0;
while (a>0){
    mult= mult+b;
    a= a-1;
}
```

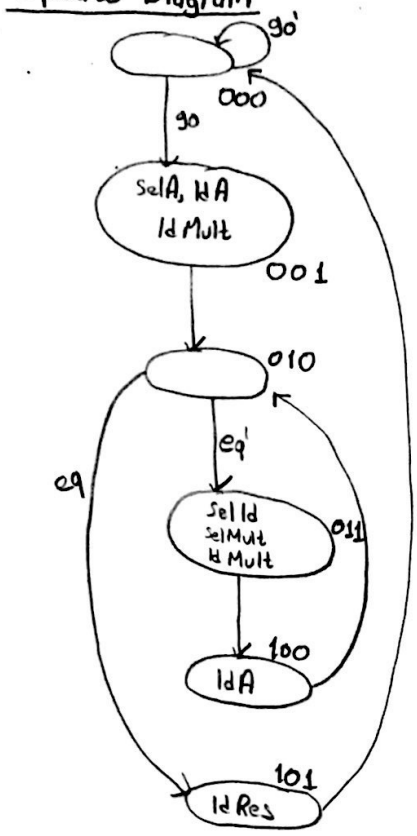


First Step

2 state yaptım çünkü toplama işlemlerini sırayla yapıyorum. Tek statede yaparsam adder içinde hata alabilirim.



Update Diagram



Step 3 and Step 4

P_2 P_1 P_0	SelA	Id A	selMult	WMult	selLoad	IdRes
0 0 0	0	0	0	0	0	0
0 0 1	1	1	0	1	0	0
0 1 0	0	0	0	0	0	0
0 1 1	0	0	1	1	1	0
1 0 0	0	1	0	0	0	0
1 0 1	0	0	0	0	0	1
1 1 0	unused					
1 1 1						

$SelA = P_2' P_1' P_0$ $\quad Id\,A = P_2' P_1' P_0 + P_2 P_1' P_0'$

$Sel\,Mult = P_2' P_1 P_0$

$Id\,Mult = P_2' P_1' P_0 + P_2' P_1 P_0 = P_2' P_0$

$Sel\,Load = P_2' P_1 P_0$ $\quad Id\,Res = P_2 P_1' P_0$

P_2	P_1	P_0	gt	eq	lt	go	N_2	N_1	N_0
0	0	0	x	x	x	0	0	0	0
0	0	0	x	x	x	1	0	0	1
0	0	1	x	x	x	x	0	1	0
0	1	0	0	1	0	x	1	0	1
0	1	0	1	0	0	x	0	1	1
0	1	1	x	x	x	x	1	0	0
1	0	0	x	x	x	x	0	1	0
1	0	1	x	x	x	x	0	0	0
unused	1	1	0	x	x	x	x	0	0	0
	1	1	1	x	x	x	x	0	0	0

$N_2 = P_2' P_1 P_0' gt' eq\, lt' + P_2' P_1 P_0 = P_2' P_1 (P_0' gt' eq\, lt' + P_0) = P_2' P_1 (P_0' + P_0)(gt' eq\, lt' + P_0) \Rightarrow$

$N_2 \Rightarrow P_2' P_1 P_0 + P_2' P_1 gt' eq\, lt'$

$N_1 = P_2' P_1' P_0 + P_2' P_1 P_0' gt.eq' lt' + P_2 P_1' P_0'$

$N_0 = P_2' P_1' P_0' go + P_2' P_1 P_0' gt' eq\, lt' + P_2' P_1 P_0' gt\, eq' lt'$

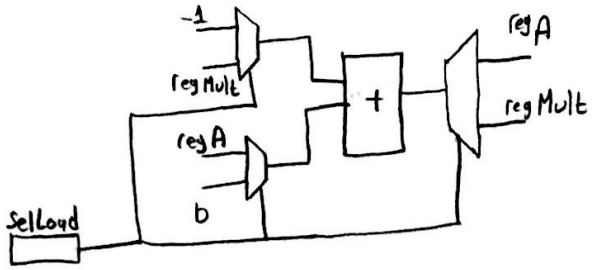
Report

Step 1: Koda göre durumları belirleyip diyagramı çizdim. Datapath içinde tek adder kullanacağım ve adder'a gelen sayıları muxlarla belirlediğim için 2 farklı state'e böldüm. (a = a-1 ile mult = mult+b). ✓

Step 2 = Öncelikle kullanıcıdan aldığım inputları 0 ile Comparator'e yolladım.

Comparator'ün lessthan çıkışını, inputun kendisi 0 negatifi 1 ayağına bağlanan mux'un select biti yaptım. Böylelikle sayı 0'dan küçük olursa devreye pozitife çevirerek yollamış oldum. Bu iki inputun lessthan çıkışlarını devrenin sonunda tekrardan XNOR'layarak, sonucun kendisini ve negatifini yolladığım result multiplexerin select biti yaptım. Eğer iki bitin değeri aynıysa XNOR sonucu 1 olacağı için devreden gelen orijinal pozitif sayıyı outputa yollamış oldum, XNOR değeri 0 olduğu durumda 1 negatif 1 pozitif sayımız var demektir. Sayılardan birini devreye pozitif'e çevirerek yolladığım için şimdi sonucun negatife çevrilmiş halini seçip sonuca yollamış olarak negatif sayılarla da devrenin çalışmasını sağlayarak bonus görevi de yerine getirdim.

Devre içinde işlemleri yaparken 2 tane mux ve 1 tane demux kullandım



Bu kısımda eğer selLoad 0 olursa -1 ile regA yı topladım, demux içinde aynı select biti olduğu için regA'ya kaydettim. SelLoad 1 olursa, regMult'un değeri ile b'yi toplayıp tekrardan regMult'a kaydetmiş oldum. Hepsinin select bitini aynı yaparak yapılacak işlemi ve kaydedilecek yeri seçmiş oldum.

Sel Load'ın her 0 olduğu durumda regA değeri 1 azalacağı için kontrolünü A registerinin enable bitini kontrol ederek sağladım. Aynı durumu regMult için de gerçekleştirerek devrenin toplama işlemlerinin adımlarını kontrollü bir şekilde yapmasını sağladım. Her regA değerini 0 ile kıyaslayarak, comparatörün çıkış bitlerini datapath'in outputu, control unit'in inputu yaptım.

Step 3= İnput ve Outputları belirlediğim için state diagramı güncelleyip doğruluk tablosu oluşturdum. Doğruluk tablosunda control unit'in outputları sadece State'lere bağlı olduğu için tabloyu iki ayrı şekilde yaptım.

Step 4= Doğruluk tablolarında ki ifadeleri boolean expressionlara çevirdim.

Step 5= Boolean expressionlara göre control unit'i tasarlayıp, datapath ile birleştirdim.

Total = Projenin tüm adımlarını sağlamış olup ek olarak bonus görevi de yerine getirdim.