

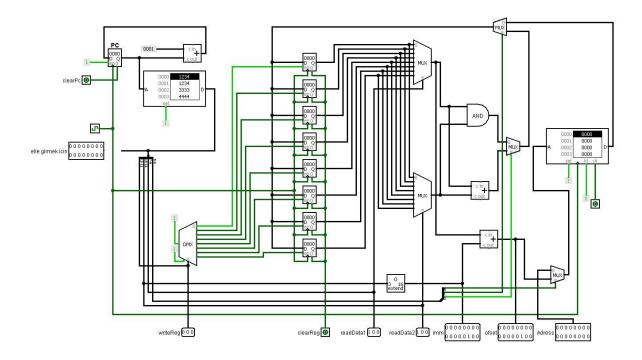
BİLGİSAYAR MÜHENDİSLİĞİ

BİLGİSAYAR ORGANİZASYONU VE MİMARİSİ

PROJE-2 16 BİT İŞLEMCİ TASARIMI

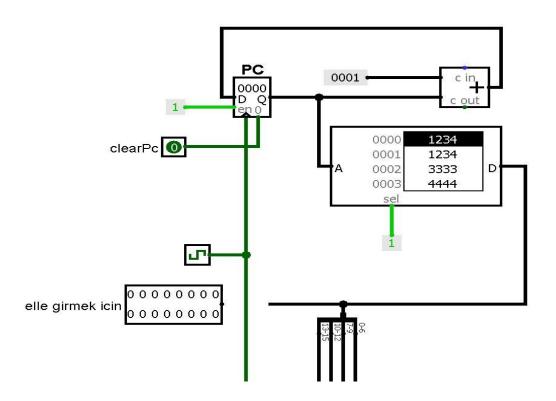
ADEM YAVUZ ÇELİK 1521221055

16 bit işlemci:



Projemde 16 bit işlemci datapath tasarımını logisim kullanarak yaptım. İşlemcimde 8 adet register bulunmaktadır. Komutlarım ve verilerim 16 bittir. Toplama, And , Load işlerimleri bulunmaktadır.

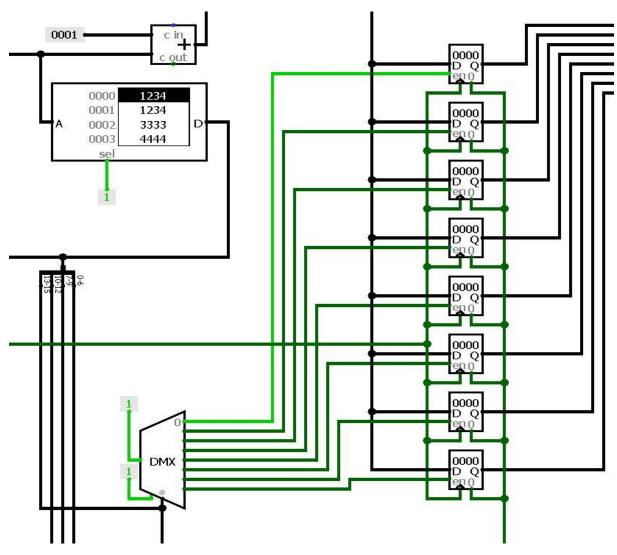
Pc counter ve Instruction memory:



PC counter'm constan olarak enable verdim, çünkü her clockta aktif olarak çalışmasını istiyorum, Counterımdan çıkan değer 2'ye ayrılıyor instruction memory'e gidip komut çalıştırırken aynı zamanda bir sonraki komutun adresini alıp addder yardımıyla bulup o adrese gitmesini sağlıyorum. Clear için pin koydum ve clock koydum. Burda görüldüğü üzere komutlarımı istenilen şekilde splitter yardımıyla ayırıyorum. Ekstradan opsiyonel olarak elle girebilmeniz için bir input koydum burda kendi komutlarınızı çalıştırabilirsiniz.

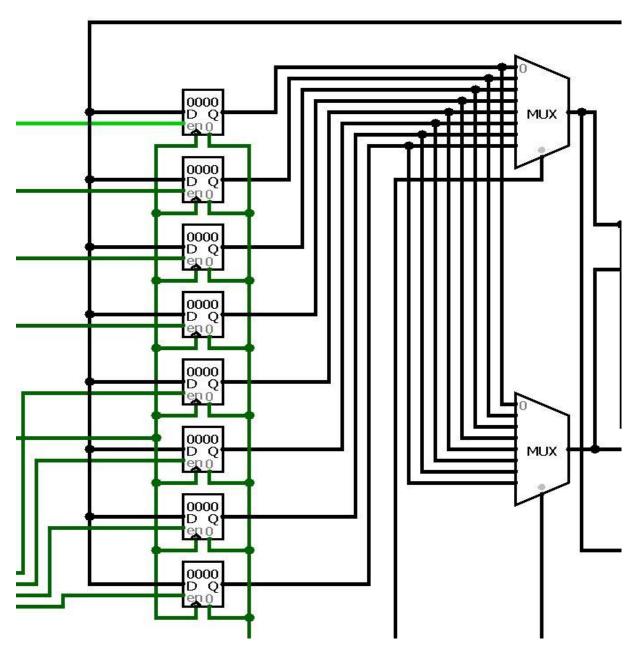
Opcode	R kaynak1	R kaynak2	R Hedef
7	3	3	3

Register'a yazdırma:

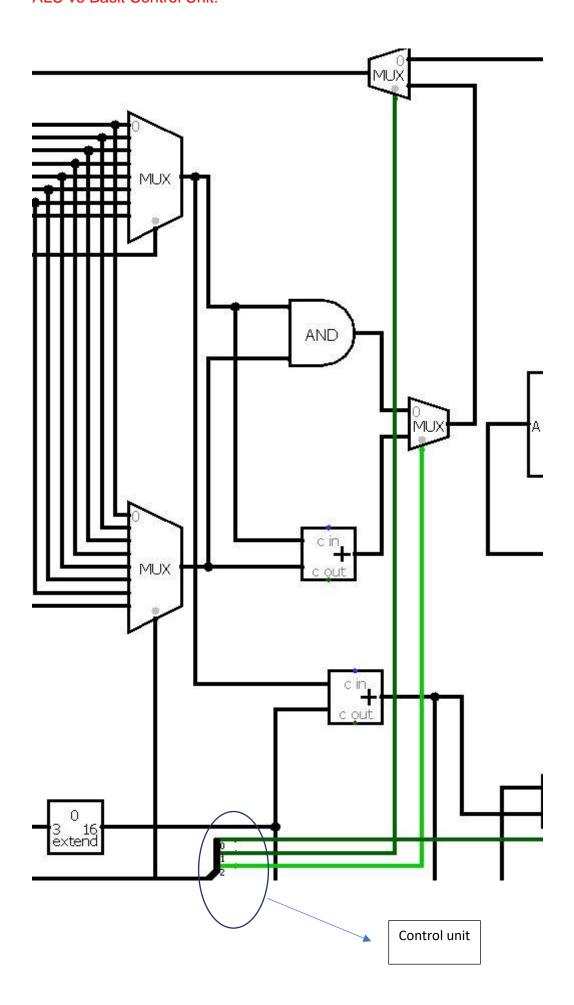


Burda splitter ile ayırdığım son 3 bitimi bir demuxa veriyorum sebebi 8 register'ım var hangi registerı seçme bilgisini selection uçlarına girdiğim 3 bit ile demux'a veriyorum.Sadece LOAD,ADD,AND komutlarım olduğu için her daim aktif olcak demux'um ve input'um da 1 olcak.Alttan verdiğim 3 bit ile hangi register'ımı seçtiysem demux'a verdiğim input ile onu enable ediyorum böylece o register aktif olup memory'den veya register'ımdan dönecek olan değeri yazdırabileyim.

Register'dan okuma:

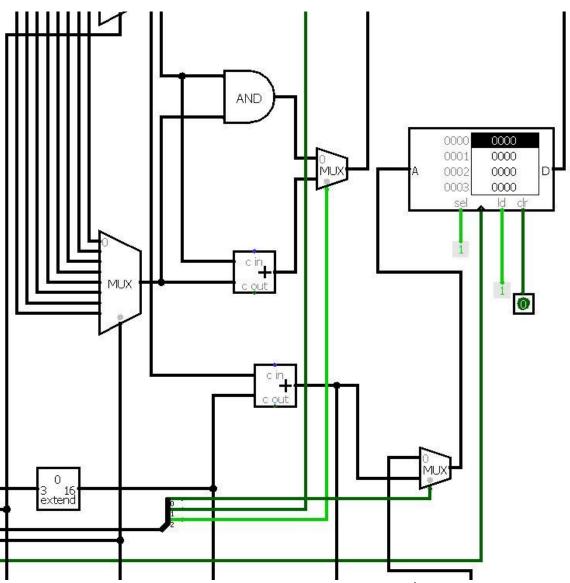


8 register'ımdan hangisini okuyacaksam mux 'un selection girişine girdiğim 3 bit ile seçebiliyorum.2 mux var çünkü birisinden ilk datamı diğerinden 2. datamı alabilmek için.Sonrasında çıkış oalrak'da aluya gönderiyorum.



Burda gelen datalarım and veya add işlemine sokuluyor. Sonrasında ise register'a yazılmaz üzere çıkıyor. Control unitimde default olarak 000 konumunda memory'e el ile girmiş olduğum adresteki datayı yüklüyor. 100 durumunda offset olarak aldığım değer ile register'dan okuduğum adres değerini toplatıp memory'e gönderiyorum böylece LDR R0,[R6,#4] işlemini gerçekleştirebiliyorum. Eğer benim ortadaki bitim 2. bitim 1 durumunda ise add/and işlemlerini yapabiliyorum. 010,110,011,111 burdanda görüldüğü üzere şayer 2. bitim aktif edilmişse 1. bitimin bir önemi kalmamakta load işlemlerini otomatik iptal etmekte. 3. bitim ise add veya and yapabilmesi için Alu çıkışıma konulmış mux'un selection uçlarıdır.

Memory:



Burda ld ucum hep aktif çünkü store yapmam gerekmiyor. İstediğim zaman clear edilmem için clear pinim ve her daim seçili ve clock palsim.