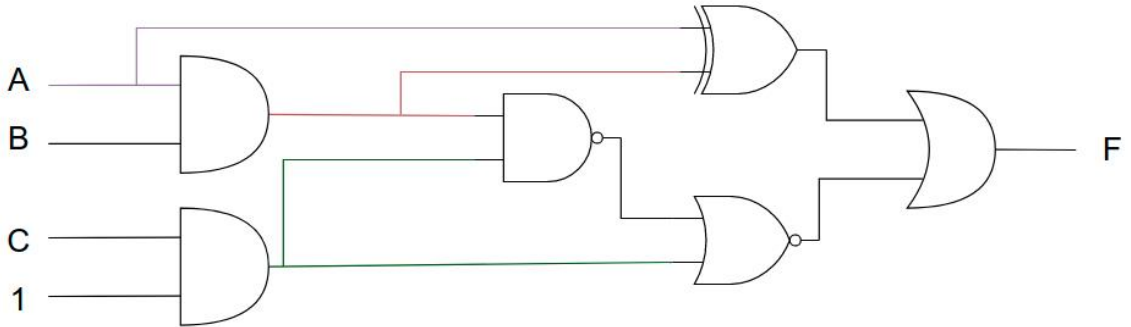




**Not:** Verilen giriş, çıkış ve modül isimlerinin farklı olması durumunda(büyük/küçük harf dahil) puan kırılacaktır.

**1- Kapı Seviyesinde Devre Kodlanması [10 puan]**

Bu kısımda “devre” ( oluşacak dosyanın adı “devre.v” olmalıdır.) isimli bir modül oluşturunuz. Modül kapı seviyesinde kodlanmalıdır, kapı seviyesi ile kodlanmayan modüller değerlendirilmeyecektir.



Yukarıda kapı seviyesinde tasarımı verilen devreyi, Verilog dilinde **kapı seviyesinde** kodlayınız. Kodunuzdaki giriş çıkışlar **büyük harfle** kodlanmalıdır.

*Not: Karışıklığa yol açmaması için aynı kablodan alınan değerler aynı renk ile ifade edilmiştir. A girişinin bağlı olduğu kablo mor, A ve B değerlerinin işlemi sonucu elde edilen çıkış kırmızı, C ve 1 değerlerinin işlemi sonucu elde edilen çıkış ise yeşil ile gösterilmiştir.*

## 2 - Binary Tree [70 puan]

Bu soru verilog ile kapı seviyesinde kodlanmalıdır. Kapı seviyesinde kodlanmayan modüller değerlendirilmeyecektir.

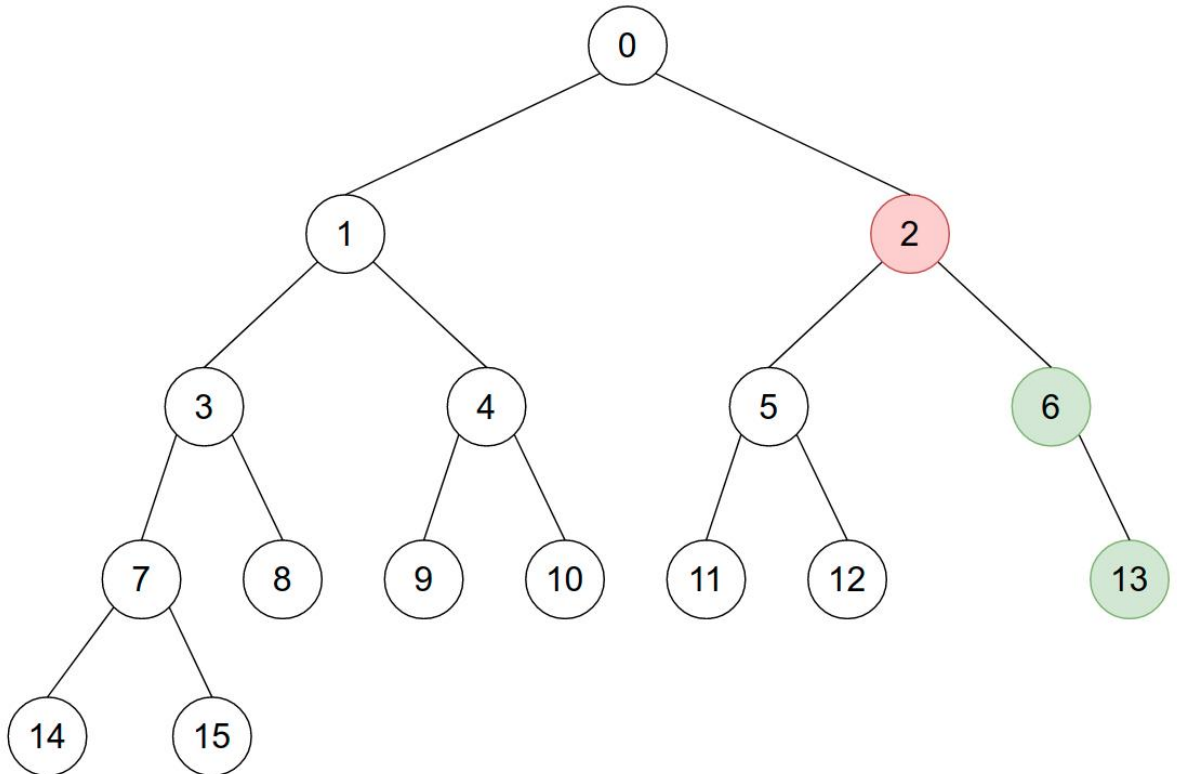
NOT : a, b ve c kısmında görselde belirtilen ağaç yapısı üzerinden sorular çözülecektir.

### a - Binary Tree Traversal [30 puan]

Bu kısımda “**binary**” ( oluşacak dosyanın adı “**binary.v**” olmalıdır.) isimli bir modül oluşturunuz.

Binary Tree düğümlerden oluşan ve her bir düğümün en az sıfır en fazla iki çocuğunun olabileceği bir veri yapısıdır. Bu soruda ise sizden istenilen, input olarak verilen bir kaynak düğümünden, yine input olarak verilecek yön doğrultusunda iki adım ilerleyip, çıktı olarak hedef düğümünü verebilmektir.

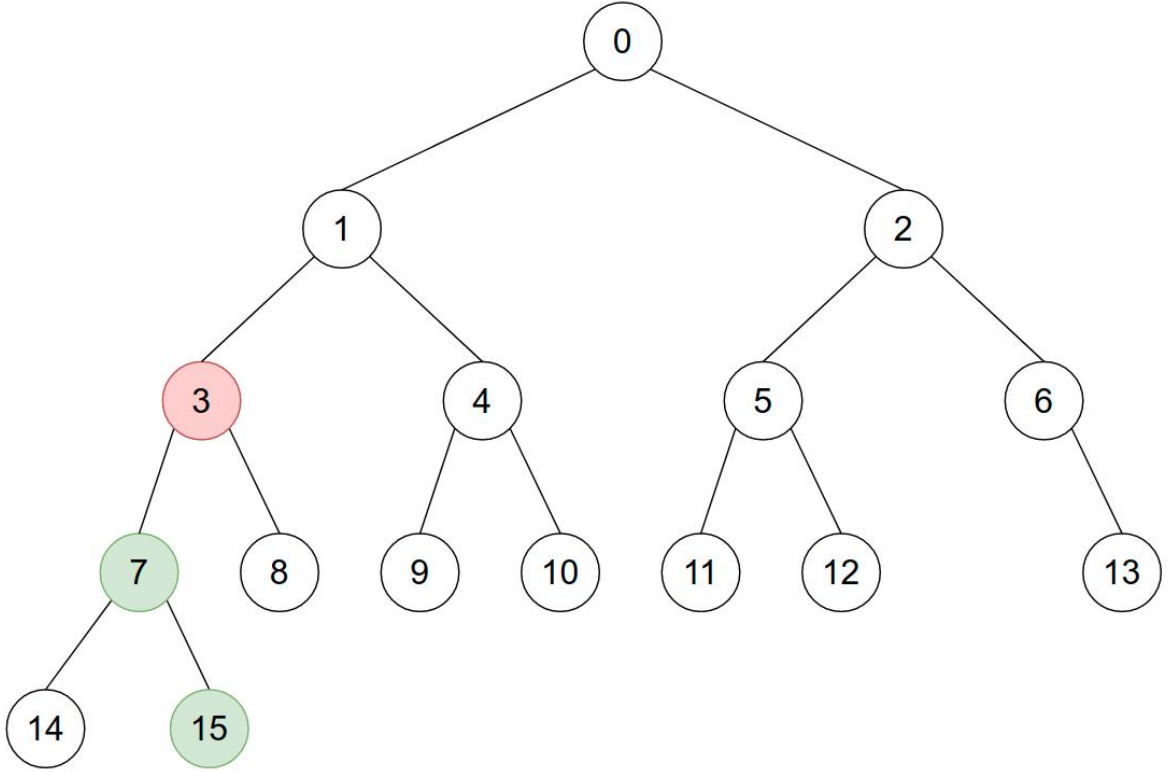
0-16 arası numaralandırılmış 17 adet düğüm arasından yön girişine bağlı olarak ilgili düğüme gidebilmeniz beklenmektedir. Yön girişi 2 bittir ve en anlamlı biti (soldaki ilk adımda sağa mı yoksa sola mı gideceğimizi söylerken, en anlamsız biti (sağdaki) ikinci adımda hangi yöne gideceğimizi söylemektedir. 0 soğa gitmek, 1 sağa gitmek anlamına gelmektedir.



Örnek-1

Örneğin yukarıdaki durumda kaynak\_dugumu girişi 2, ve yön girişi 11 olarak verilmiştir. İlk adımda 2. düğümün sağ çocuğu olan 6. düğüme, ikinci adımda ise 6. düğümün sağ

çocuğu olan 13. düğüme gelmeliyiz çıkış sinyelimiz olan hedef\_dugumu 13 değerini almalıdır.



### Örnek-2

Örneğin yukarıdaki durumda kaynak\_dugumu girişi 3, ve yön girişi 01 olarak verilmiştir. İlk adımda 3. düğümün sağ çocuğu olan 7. düğüme, ikinci adımda ise 7. düğümün sağ çocuğu olan 15. düğüme gelmeliyiz çıkış sinyelimiz olan hedef\_dugumu 15 değerini almalıdır.

#### **Modülünüzün giriş ve çıkış sinyalleri aşağıdaki gibi olmalıdır :**

**kaynak\_dugumu** : 4 bitlik giriş sinyali.

**yon** : 2 bitlik giriş sinyali.

**hedef\_dugumu** : 4 bitlik çıkış sinyali.

NOT : Her zaman 2 bitlik yön girişimiz olacaktır. Bu sebeple belirli sayıda kaynak\_dugumu olabilecek düğümümüz vardır. Bu düğümler dışında bir input girilmeyeceğini varsayabilirsiniz. Örneğin 5 numaralı düğüm size input olarak verilmeyecektir.

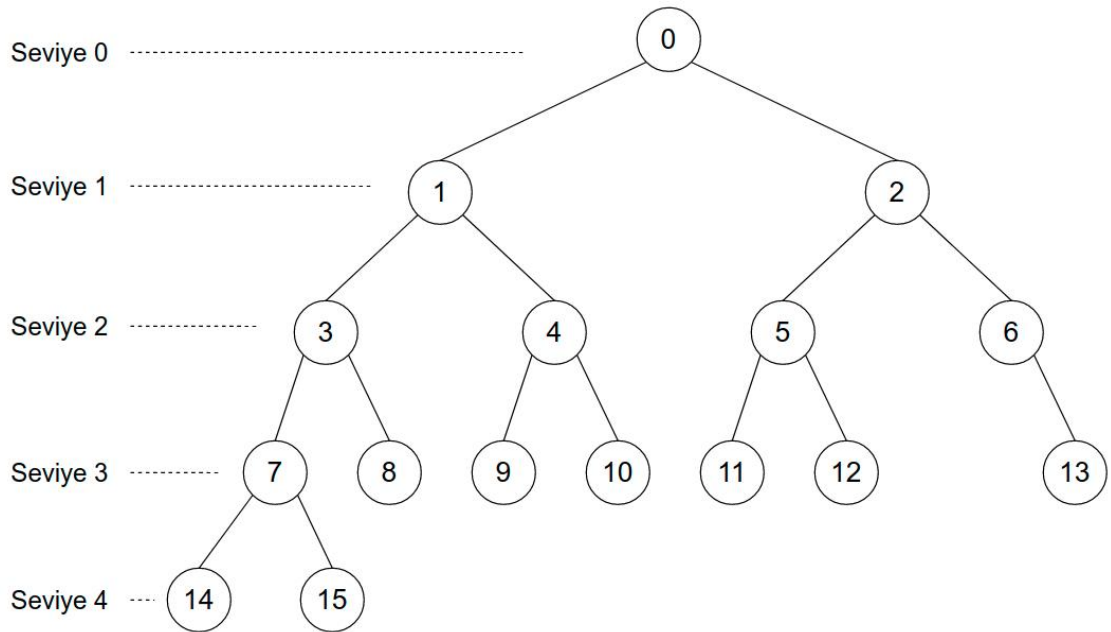
İstisnai bir durum olarak da 2 numaralı düğüm için 10 yön girişi verilirse ve 3 numaralı düğümden sağa gidilmeye çalışılırsa, don't care (önemsenmeyen durum) olarak

düşünebilirsiniz. Bu şart sadece belirtilen düğümler için gerekli olup diğer düğümlerde böyle durumları incelemeye ihtiyaç duymamanız gerektiğine dikkat ediniz.

### b- Binary Tree Seviye Belirleme [20 puan]

Bu kısımda ise “seviye” ( oluşacak dosyanın adı “seviye.v” olmalıdır.) isimli bir modül oluşturunuz.

Binary Tree, a şıkında da belirttiğimiz gibi düğümlerden oluşan ve her bir düğümün en az sıfır en fazla iki çocuğunun olabileceği bir veri yapısıdır. Bunun yanında her bir düğüm ağaç içerisinde belirli bir seviyede bulunmaktadır. Örneğin,



Yukarıdaki ağaç yapısında 5 farklı seviye (0-5 arası) bulunmaktadır. Bu soruda input olarak gelen **dugum** girişinin seviyesini hesaplamanız gerekmektedir.

Örneğin **dugum** adlı giriş 5 ise **dugumun\_seviyesi** çıkışı da 2 olmalıdır.

**Modülünüzün giriş ve çıkış sinyalleri aşağıdaki gibi olmalıdır :**

**dugum** : 4 bitlik giriş sinyali.

**dugumun\_seviyesi** : 3 bitlik çıkış sinyali.

### c- Binary Tree Seviye Farkı Hesaplama [20 puan]

Bu kısımda ise “**fark**” ( oluşacak dosyanın adı “**fark.v**” olmalıdır.) isimli bir modül oluşturunuz.

Bu soruda hiyerarşik modelleme kullanmanız gerekmektedir. (2-a ve 2-b’de yazdığınız modülleri kullanmalısınız. Ek modül kullanamazsınız.) Giriş olarak iki düğüm ve iki yon gelmektedir. Daha sonra “**binary**” modülünü iki giriş için de kullanarak iki tane **hedef\_dugumu** bilgisi elde etmelisiniz. Daha sonra bu düğümlerin seviyelerini “**seviye**” isimli modülü kullanarak hesaplamalısınız. Çıkış olarak ise bu iki seviye arasındaki farkı vermeniz beklenmektedir.

**Modülünüzün giriş ve çıkış sinyalleri aşağıdaki gibi olmalıdır :**

**dugum\_1** : 4 bitlik giriş sinyali.

**dugum\_2** : 4 bitlik giriş sinyali.

**yon\_1** : 2 bitlik giriş sinyali.

**yon\_2** : 2 bitlik giriş sinyali.

**seviye\_farkı** : (kaç bit olduğunu siz hesaplamalısınız) çıkış sinyali.

### 3 - Asansör [20 puan]

Bu kısımda ise “**asansor**” ( oluşacak dosyanın adı “**asansor.v**” olmalıdır.) isimli bir modül oluşturunuz. Soru kapı seviyesinde kodlanmalıdır, kapı seviyesinde kodlanmayan modüller değerlendirilmeyecektir.

Bu soruda sizden istenilen 3 katlı bir bina için asansör tasarlamanız.(Bina katları 0, 1, 2, 3) Tasarlanacak asansörün hareketi input olarak gelecek olan butona bağlıdır ve şu şekildedir :

buton → 00 : Asansör 1 kat yukarı çıkmalı.

buton → 01 : Asansör 3 kat yukarı çıkmalı.

buton → 10 : Asansör 2 kat aşağı inmeli.

buton → 11 : Asansör 1 kat aşağı inmeli.

Asansör belirtilen butonlar haricinde hareket edememektedir. Kural dışı durumlar için, asansörün üst kat sınırını aşmaya çalışıyorsa çıkış olarak 3, en alt kat sınırı aşmaya çalışılıyor ise çıkış olarak 0 verilmelidir. Örneğin asansör 1. kattayken buton girişi 10 olursa, asansör iki kat aşağı inemeyeceği için 0. katta durmalıdır.

**Modülünüzün giriş ve çıkış sinyalleri aşağıdaki gibi olmalıdır :**

**bulundugu\_kat** : 2 bitlik mevcut katı gösteren giriş sinyali.

**buton** : 2 bitlik giriş sinyali.

**durdugu\_kat** : 2 bitlik asansörün son konumunu gösteren çıkış sinyali.