

TOBB Ekonomi ve Teknoloji Üniversitesi Bilgisayar Mühendisliği Bölümü

BİL 361 - Bilgisayar Mimarisi ve Organizasyonu

22 Kasım 2022 2022 - 2023 Öğretim Yılı Güz Dönemi Ödev 2

Önbellek Kontrolcüsü ve Dallanma Öngörüsü

a) [125 Puan] Önbellek Kontrolcüsü

Piazzada Kaynaklar/Ödev2 altında bulunan **onbellek.v**, **anabellek.v** ve **tb_onbellek.v** dosyasını inceleyiniz, giriş çıkışlar konusunda anlamadığınız noktaları piazza postunun altından iletebilirsiniz. Bu dosyalardan **anabellek** modülü 32 bayt veri öbek boyutuna sahip, 100 çevrim işlem gecikmeli bir belleğin davranışını gerçeklemektedir. **onbellek** modülü ise 4 bayt hizalı okuma/yazma işlemlerine izin veren ve herhangi bir önbellekleme yapmayan bir durum makinesi içermektedir (tüm istekler anabellekten karşılanır). **tb_onbellek** ise [0, 16383] bayt adreslerine 4 bayt hizalı/hizasız yazma ve okuma isteklerini test etmektedir. Simülasyonu çalıştırıp sonuna kadar ilerlettiğinizde **Tcl Console**'da **[SIM] Test başarılı.** mesajını görmeniz gerekiyor.

Sizlerden **onbellek** modülünün içeriğini **değiştirerek** ya da **yeniden yazarak** (giriş/çıkışlar ve modül ismi sabit kalmalı) bir önbellek kontrolcüsü gerçeklemenizi ve başarım iyileşmesini kıyaslamanızı bekliyoruz (ödev gönderim kısmında istenen rapor içeriğini bulabilirsiniz). Yazacağınız önbelleğin ve kontrolcünün özellikleri aşağıda verilmiştir.

- Toplam önbellek kapasitesi **4KB** olmalı. Modülünüz içinde reg [x:x] hafiza_r [x:x] olarak tanımlayabilirsiniz.
- Önbellek **doğrudan eşlemeli** olmalı.
- Geri yaz (write-back) ile yaz ve yerini ayır (write and allocate) politikaları kullanılmalı.
- Öbek boyutları 32 bayt olmalı.
- Yazma ve okuma erişimleri 4 bayt hizalı yapılmalı.
 dosyadaki size verdiğim obege_yaz ve obekten_oku fonksiyonlarını inceleyebilirsiniz
- Önbelleğiniz içerisinde **iska_sayisi_r** ve **cikarma_sayisi_r** isimli 2 adet ıskalanan veri ve çıkarılan veri sayısını gösteren 32 bit sayaç (reg) bulunmalıdır. Bu sayaçlar önbellek raporunuzda sizlere yardımcı olacaklar.

Önbelleğinizin adres dağılımı aşağıdaki sırada olmalıdır. Doğru bit genişliklerini siz hesaplamalısınız.

Etiket Satir Indisi Bayt Seçimi

İnternetteki verilog kodlarını incelemekte özgürsünüz ancak herhangi bir kodu olduğu gibi almamalısınız (ayrıca kendi aranızda kod paylaşımı yapmamalısınız). Eğer herhangi bir kaynaktan faydalanırsanız (makale, ders videosu, github reposu vb.) ödev raporunuzda alıntı yaparak kodunuzun hangi kısmının buradan esinlendiğini açıklayın. Düzgün referans verdiğiniz ve kodunuz asıl kaynaktan olduğu gibi alınmadığı sürece (fikri anlayıp kendiniz gerçekleyebilirsiniz) intihal olarak değerlendirilmeyecek.

b) [75 Puan] Dallanma Öngörücü

Piazzada Kaynaklar/Ödev2 altında bulunan **ongorucu.v** dosyasını inceleyiniz, giriş çıkışlar konusunda anlamadığınız noktaları piazza postunun altından iletebilirsiniz. **ongorucu** modülü durağan atlamaz tahmini yapan bir dallanma öngörücüyü gerçeklemektedir.

Sizlerden **ongorucu** modülünü **yeniden yazarak** (giriş/çıkışlar ve modül ismi sabit kalmalı) aşağıda varsayımları ve çalışma **figürleri** verilen 2 düzeyli dallanma öngörücüsünü gerçeklemenizi ve başarım iyileşmesi ile ilgili bir rapor hazırlamanızı bekliyoruz (ödev gönderim kısmında istenen rapor içeriğini bulabilirsiniz).

- Modülünüze Ödev1 dokümanında kullandığımız buyruklar gelecek. Bu buyruklardan sadece BEQ, BNE ve BLT için tabloları kullanarak tahmin yapmalısınız, diğer buyruklar için herhangi bir tahminde bulunmayın (yani JAL ve JALR için ek işlem yapmayın).
- Güncelle sinyalleri varsayımsal bir yürüt aşamasından modülünüze iletilecek. Tablolarınızı ve geçmiş yazmaçlarını yalnız bu sinyaller geldikçe güncelleyin.
- Geçmiş değerlerinin tamamı 0, çift doruklu öngörücülerin de tamamı GT değerinden başlatılmalı.
- Eğer atlar tahmini yapıyorsanız, buyruğun atlayacağı adresi de doğru olarak vermelisiniz. Modülünüzün **ps** ve **buyruk** değerlerine erişimi olduğu için hem buyruk tipini anlamanız hem de doğru adresi hesaplamanız mümkün.
- Modülünüz tahminini girişteki **ps** ve **buyruk** ikilisine göre kombinasyonel olarak anında yapmalıdır. Yani tahminlerinizi **always** @* bloğunda yapabilirsiniz. Burada tablolarınıza gelen güncellemeleri hala sıralı mantıkla (**always** @(**posedge clk**)) yapmanız gerektiğini unutmayın.

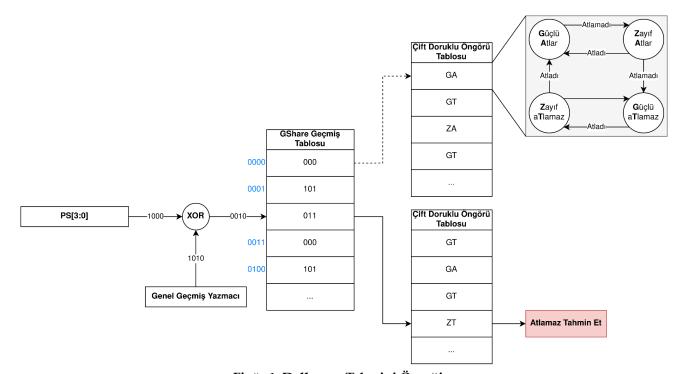
Bu kısmı tam anlamadıysanız **kombinasyonel mantık** (combinational logic) ve **sıralı mantık** (sequential logic) konularını araştırabilirsiniz.

Doğru Çalışan Modül:

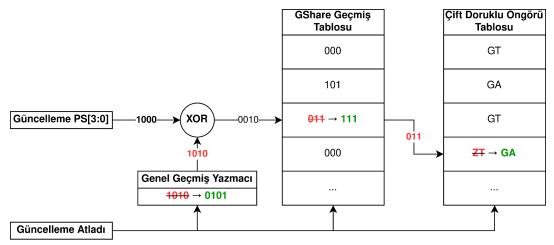
- 0. nanosaniye: ps XX, buyruk XX ..., tahmin XX (önemli değil)
- 2. nanosaniye: ps 0x4000_0000, buyruk BNE ..., tahmin ATLAR (zamanında tahmin)
- 5. nanosaniye: ps 0x4000_0000, buyruk BNE ..., tahmin ATLAR (posedge)

Yanlış Çalışan Modül:

- 0. nanosaniye: ps XX, buyruk XX ..., tahmin XX (önemli değil)
- 2. nanosaniye: ps 0x4000_0000, buyruk BNE ..., tahmin XX (tahmin yapılmamış)
- 5. nanosaniye: ps 0x4000_0000, buyruk BNE ..., tahmin ATLAR (posedge, geç tahmin)



Figür 1: Dallanma Tahmini Örneği



Figür 2: Tablo ve Yazmaç Güncelleme Örneği

NOT: İlk eriştiğimiz tablonun sayaçlar değil ilgili satırın geçmişi olduğuna dikkat edin. Bu geçmiş tablosunu kullanarak satırın çift doruklu öngörücü tablosunun ilgili satırına erişiyoruz. Derste gördüğünüz klasik GShare öngörücüden biraz daha farklı.

Ödev Gönderimi ve Formatı

Gönderdiğiniz tüm tasarımların **sentezlenebilir** verilog standartlarında yazılması beklenmektedir. Tasarımlarınızın keyfi bir FPGA kartı için sentezlenebilir olduğunu kontrol edin. Modülleriniz simülasyon üzerinden kontrol edileceğinden implementasyon ve bitstream aşamaları **gerçekleştirilmeyecek**. Bu aşamalar için "**constraints**" dosyaları oluşturmakla **vakit kaybetmeyin**.

Sentez çıktılarını almak için bir çok farklı yol var ancak daha önce yapmadıysanız aşağıdaki adımları izleyebilirsiniz:

- -Her şık için proje ağacından ilgili modüle sağ tıklayıp "**Set as Top**"ı seçin. Eğer zaten ilgili modül kalın fontla yazılıysa (zaten hedef olarak seçiliyse) bu adımı atlayabilirsiniz.
 - (a için **onbellek.v** ve **b** için **ongorucu.v** dosyaları seçilmelidir)
- -Vivado'da sol kısımdaki menüden "**SYNTHESIS**" altında bulunan "**Run Synthesis**" seçeneğine tıklayın.
- -Senteziniz tamamlandığında sentez raporunuzu (vds dosyasını) kayıt edin ve gönderiminize ekleyin. Raporunuz sentez bittikten sonra aşağıdaki yolda oluşacaktır.

"{PROJE_KLASORU}/{PROJE_ISMI}.runs/synth_1/{MODUL_ISMI}.vds"

[25 Puan] (Bonus) Ödev Raporu

a ve **b** şıkları için tasarladığınız modüllerin sizlere ilk başta verilen modüllere oranla başarım iyileşmesini ölçünüz ve tartışınız. Ayriyeten varsa zorlandığınız kısımlara ve nasıl çözdüğünüze kısaca değinin.

Başarım kıyaslaması için sizlere daha çok fikir vermek açısından bir kaç örnek veriyorum. İyileşmeyi gösterdiğiniz sürece aklınıza gelen her şekilde yapabilirsiniz kesinlikle bu yöntemlere kısıtlı değilsiniz:

- **a şıkkı için** paylaştığım örnek **tb_onbellek**'i düzenleyerek farklı erişim örüntüleri için bulma oranlarını, aynı örüntüler için tüm isteklerin yanıtlanmasının kaç çevrim sürdüğünü tartışabilirsiniz.
- **b** şıkkı için Ödev1 buyrukları ile yazdığınız basit bir döngü için doğru tahmin oranını, varsayımsal bir dallanma gecikmesi için (*örn.*, 2) başarımın ne kadar arttığını gösterebilirsiniz.

ödev2_rapor.pdf isimli ödev raporunuzu, düzenlenmiş kodlarınızı (modül ve dosya isimlerini değiştirmeden) ve yukarıda nasıl alacağınız açıklanan tasarım sentez raporlarınızı uzak platformuna sıkıştırmadan yükleyin.

Son Teslim Tarihi: 18 Aralık 2022, 23:59