



**TOBB Ekonomi ve Teknoloji
Üniversitesi**
Bilgisayar Mühendisliği Bölümü
BİL 361 – Bilgisayar Mimarisi ve Organizasyonu

25 Mart 2025
2024 – 2025 Öğretim Yılı
Bahar Dönemi
Ödev 2

[150 puan] Doğrudan Eşlemeli Önbellek Kontrolcüsü Tasarımı

Bu ödevde doğrudan eşlemeli önbellek tasarlayacaksınız.

- Tasarlayacağınız önbellek ile ilgili isterler aşağıdaki gibidir, bu tanımlara uyduğunuzdan emin olmak için kaynaklar altındaki taslak tasarım dosyalarını kullanabilirsiniz.
 - Sistemde bellek adresleri **32 bit** uzunluğundadır.
 - Sistemde **bayt adresleme** kullanılmaktadır.
 - Önbellek satırlarına **doğrudan eşlemeli erişim** yapılmaktadır.
 - Önbellekte bulunmayan yazma istekleri için “**Yaz ve Yerini Ayır**” politikası, önbellekte bulunan yazma istekleri için “**Doğrudan Yaz**” politikası kullanılmaktadır.
 - Önbelleğin toplam boyutu **2KB**’tır.
 - Önbelleğin veri öbekleri **16 bayt**tır.
 - Önbellek Kontrolcüsü - İşlemci arayüzü bağımsız (-ing., decoupled) olarak çalışmaktadır.
 - Önbellek Kontrolcüsü - Anabellek arayüzü bağımsız (-ing., decoupled) olarak çalışmaktadır.
 - Önbellek Kontrolcüsü - Önbellek arayüzü bağımlı (-ing., coupled) olarak çalışmaktadır.
- Kafanızı karıştıran durumlarda piazzada ödev gönderisini kullanmaya çekinmeyin.

Ödev iki kısımdan oluşmaktadır. İlk kısımda önbellek tasarımı Verilog HDL kullanarak gerçeklemeniz beklenmektedir. İkinci kısımda tasarladığınız önbellek ile ilgili sorulara cevap vererek bir rapor hazırlamanız beklenmektedir. Bu kısımların detaylarını sonraki sayfalarda görebilirsiniz.

[100 Puan] a) Önceki sayfada tasarım istekleri verilen önbelleği verilog dilinde “**onbellek.v**” ile tek bir modül olarak tasarlayınız. “**onbellek.v**” için taslak kodunu piazza kaynaklar kısmında bulabilirsiniz. Bu dosyaya giriş / çıkış **isimleri** dışında istediğiniz değişikliği yapabilirsiniz (sadece isimler önemli, atamaları ve genel kodu değiştirmenizde bir sakınca yok).
onbellek.v giriş çıkışları sırasıyla aşağıdaki gibidir.

- **clk** : 1 bitlik saat girişi
- **rst** : 1 bitlik işlemcinizi başlangıç durumuna geri çeviren giriş

- **islemci_istek_gecerli** : 1 bitlik geçerli önbellek isteği olduğunu gösteren **giriş** sinyali
- **islemci_istek_hazir** : 1 bitlik önbellek kontrolcüsünün istek için hazır olduğunu gösteren **çıkış** sinyali
- **islemci_istek_adres** : 32 bitlik önbellek kontrolcüsü istek adres **girişi**
- **islemci_istek_veri** : 32 bitlik önbelleğe yazılacak veri **girişi**
- **islemci_istek_yaz** : 1 bitlik önbelleğe yazma yapılacağını gösteren **giriş** sinyali

- **islemci_cevap_gecerli** : 1 bitlik önbellek kontrolcüsü cevabının geçerli olduğunu gösteren **çıkış** sinyali
- **islemci_cevap_hazir** : 1 bitlik sistemin cevaba hazır olduğunu gösteren **giriş** sinyali
- **islemci_cevap_veri** : 32 bitlik önbellek kontrolcüsü cevap veri **çıkışı**

- **anabellek_istek_gecerli** : 1 bitlik geçerli anabellek isteği olduğunu gösteren **çıkış** sinyal
- **anabellek_istek_hazir** : 1 bitlik anabelleğin istek için hazır olduğunu gösteren **giriş** sinyali
- **anabellek_istek_adres** : 32 bitlik anabellek istek adres **çıkışı**
- **anabellek_istek_yaz** : 1 bitlik anabelleğe yazma yapılacağını gösteren **çıkış** sinyali
- **anabellek_istek_veri** : 128 bitlik anabelleğe yazılacak veri **çıkışı**

- **anabellek_cevap_gecerli** : 1 bitlik anabellek cevabının geçerli olduğunu gösteren **giriş** sinyali
- **anabellek_cevap_hazir** : 1 bitlik kontrolcünün cevaba hazır olduğunu gösteren **çıkış** sinyali
- **anabellek_cevap_veri** : 128 bitlik anabellek cevap veri **girişi**

- **onbellek_istek_gecerli** : 1 bitlik geçerli önbellek isteği olduğunu gösteren **çıkış** sinyali
- **onbellek_istek_yaz** : 1 bitlik önbelleğe yazma yapılacağını gösteren **çıkış** sinyali
- **onbellek_istek_veri** : 128 bitlik önbelleğe yazılacak veri **çıkışı**
- **onbellek_istek_adres** : 7 bitlik önbellek istek adres **çıkışı**
- **onbellek_cevap_veri** : 128 bitlik önbellekten okunan veri **girişi**

Önemli Not - 1: Tasarlanan kontrolcünün basit olması için sistem tarafından yapılan erişimlerin hepsinin kelime boyutunda olduğunu ve kelime boyutunun 4 bayt olduğunu varsayın.

Önemli Not - 2: Önbellek modülü sadece önbellek satırlarının verisini tutmaktadır. Önbellek satırlarının durumunu ve gerekli diğer bilgileri tutan bir yapı tasarlamak size bırakılmıştır.

Önemli Not - 3: Önbellek sıralı mantık ile çalışmaktadır ve önbellek kontrolcüsü tarafından oluşturulan isteklere saatin yükselen kenarında cevap vermektedir.

Önemli Not - 4: Anabellek sıralı mantık ile çalışmaktadır ve önbellek kontrolcüsü tarafından oluşturulan isteklere 25 çevrim sonra cevap vermektedir.

Önemli Not - 5: Önbellek kontrolcüsü - İşlemci ve Önbellek kontrolcüsü - Anabellek arayüzü hazır ve geçerli sinyallerini kullanarak iletişim sağlar. Bu sinyallerin nasıl çalıştığını anlamak için, “ready-valid handshake” protokolünü inceleyebilirsiniz.

[50 Puan - BONUS] b) a şıkkındaki tasarımını simülasyon ortamında çalıştırın ve aşağıdaki soruları cevaplayın. Sonuçlarınızı kısa bir rapor haline getirin.

1. [5 Puan] Testbench'te verilen programı önbellek kullanan ve kullanmayan sistemlerde deneyin. Sonuçlarınızı raporlayın.
2. [10 Puan] Tasarımınıza iki adet istatistik sayacı ekleyin. Önbellekte bulamama sayısını takip etmek için "**sayac_bulamama**" isimli sayacı ve toplam erişim sayısını takip etmek için "**sayac_erisim**" isimli sayacı kullanın.

Testbench'te verilen programı çalıştırın ve önbellekte bulma oranınızı raporlayın.

3. [5 Puan] Doğrudan eşleme yöntemini kullanan bir önbellek kontrolcüsü ile bulma oranını en az kaç yapabilirsiniz? Bu oranı elde edecek bir program yazın. Programı ve simülasyonda elde ettiğiniz önbellekte bulma oranını raporlayın.
4. [12 Puan] Sistem gecikmesini testbench'teki program için iyileştirmek istiyorsunuz. Bu programın bellek erişimlerini göz önünde bulundurarak önbellek tasarımınızın hangi parametrelerinin değiştirilmesi gerektiğinden bahsedin. (Önbellek doğrudan eşlemeli olmalı.)
5. [13 Puan] Sisteminizde bir LLM modeli çalıştırmaya karar verdiniz ve çalıştırdığınız modelin çok sayıda matris çarpması yaptığını gözlemlediniz. Kodu incelediniz ve matris çarpması sırasında zamanda ve konumda yerelliğin yüksek olduğunu fark ettiniz. Bunları göz önünde bulundurarak, programın çalışma süresini kısaltmak için önbellek tasarımında yapabileceğiniz değişikliklerden bahsedin. (Önbellek doğrudan eşlemeli olmak zorunda değil.)
6. [5 Puan] Önbellek kontrolcüsünü gerçeklerken yaşadığınız zorluklardan bahsedin. Yaşadığınız zorluklara nasıl çözüm bulduğunuzdan bahsedin.

Ödev Gönderimi ve Formatı

Ödevinizde **önbellek tasarımı** için yazdığınız **Verilog dosyaları** ve **önbellek tasarımlarınızın sentez çıktısı** bulunmalıdır. Ödevinizi uzak platformuna **sıkıştırmadan** yükleyeceksiniz.

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

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

- Proje ağacından ilgili modüle (islemci.v) sağ tıklayıp "**Set as Top**"ı seçin. Eğer zaten ilgili modül kalın fontla yazılıysa (zaten hedef olarak seçiliyse) bu adımı atlayabilirsiniz.
- 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"

Yüklenecek dosyalar: onbellek.v, onbellek.vds, rapor.pdf

Son Teslim Tarihi: 21 Nisan 2025, 23:59