



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

1 Aralık 2023  
2023 – 2024 Öğretim Yılı  
Güz Dönemi  
Ödev 2

## [100 puan] gem5 Önbellek Tasarımı

Bu ödevde gem5 simülatörünü kullanarak farklı karmaşıklıklarda önbellek tasarımları yapacaksınız.

gem5 simülatörü (<https://www.gem5.org>), sistem seviyesi mimarisi ve işlemci mikro mimarisi de dahil olmak üzere bilgisayar sistem mimarisi araştırmaları için modüler bir platformdur. Gem5, açık yönetim modeline sahip topluluk tarafından yönetilen bir projedir.

### gem5 Kurulumu

Ödevde başlamadan önce gem5 simülatörünü bilgisayarınıza kurmalısınız. Ödevinizde [COMMIT] commit hashli versiyonu kurmanız gerekiyor.

**Not:** Windows kullanıyorsanız Ubuntu WSL2 üzerine Ubuntu 22.04 distrosu kurarak ya da VirtualBox (<https://www.virtualbox.org>) ile, macOS kullanıyorsanız yine VirtualBox ya da Docker (<https://www.docker.com>) gibi bir program üzerinde x86 Ubuntu 22.04 sanal makine kullanmanız ve ubuntu için olan adımları takip etmeniz gerekmektedir.

Biz Windows kurulumları için WSL2 + Ubuntu 22.04, macOS kurulumları için Docker kullanmanızı öneriyoruz. Ek olarak Ubuntu'ya ilk girdiğinizde paketlerin son versiyonlarını aldığınızdan emin olmak için **sudo apt-get update** komutunu çalıştırmanızda fayda var.

Kütüphaneyi build etmek için gerekli araçları aşağıdaki gibi yükleyin:

```
sudo apt install build-essential
sudo apt install git
sudo apt install gcc g++
sudo apt install m4
sudo apt install scons
sudo apt install zlib1g
sudo apt install zlib1g-dev
sudo apt install python3-dev
```

gem5 kütüphanesini GitHub'dan klonlayın ve 6835f0665744bba3d56921c9406ee97e841b60a0 commit versiyonuna getirin (gem5 23.0.1.0 versiyonu):

(clone atmak yerine <https://github.com/gem5/gem5/archive/refs/tags/v23.0.1.0.zip> adresinden de indirebilirsiniz)

```
git clone https://github.com/gem5/gem5
cd gem5
git checkout 6835f0665744bba3d56921c9406ee97e841b60a0
python3 -m pip install -r requirements.txt
```

Kütüphaneyi elde ettikten sonra gem5 simülatörünü aşağıdaki gibi X86 buyruk kümesi mimarisi için build edin:

```
cd gem5
scons build/X86/gem5.opt -j1
```

Bu şekilde, gem5 simülatörü (gem5.opt) gem5/build/X86 dosya yolunda oluşmuş oldu. Burada daha hızlı build etmek için -j1 argümanında "1" yerine daha fazla iş parçacığı (thread) atayabilirsiniz. Maksimum atayabileceğiniz iş parçacığı sayısını, terminalde **nproc** komutu ile öğrenebilirsiniz.

Ek olarak ödevinize önbelleklerinizi CoreMark başarımlı ölçüm programı ile test etmenizi bekliyoruz. Bunu da aşağıdaki komutlarla derleyebilirsiniz:

```
git clone https://github.com/eembc/coremark
cd coremark
make compile PORT_DIR=linux
```

Bundan sonra ödev kapsamında kendi dosyalarınızı eklemek, eklediğiniz dosyalarla bir sistem oluşturmak ve bu sistem üzerinde program yürütebilmek için **Learning gem5 Tutorial: HelloObject** kısmı ile başlayan dokümanı takip edin. Ödevin ilgili sorusunda tüm dosya/sınıf/parametre/istatistik isimleri ve eklenmesi gereken alt klasörler belirtilecektir.

#### Genel Notlar:

- Simülatöre her yeni dosya eklediğinizde ya da değişiklik yaptığınızda "**scons** build/X86/gem5.opt -j1" ile tekrardan build almanız lazım.
- Simülasyon yaparken işlemci tipi olarak **TimingSimpleCPU** kullanın.
- Simülasyon yaparken **System Call Emulation** modu kullanın.
- Simülasyon istatistiklerini alırken **maxinsts** limitini 10 milyon olarak belirleyin.
- Simülasyon istatistiklerini alırken **CoreMark** kullanın.

#### Kaynaklar

##### Learning gem5 Tutorial

[https://www.gem5.org/documentation/learning\\_gem5/introduction](https://www.gem5.org/documentation/learning_gem5/introduction)

##### gem5 Bootcamp Playlist

[https://www.youtube.com/watch?v=orASbQ02pDw&list=PL\\_hVbFs\\_loVSaSDPr1RJXP5RRFWjBMq\\_q3](https://www.youtube.com/watch?v=orASbQ02pDw&list=PL_hVbFs_loVSaSDPr1RJXP5RRFWjBMq_q3)

##### Learning gem5 Tutorial: Building gem5 (Kurulum)

[https://www.gem5.org/documentation/learning\\_gem5/part1/building](https://www.gem5.org/documentation/learning_gem5/part1/building)

##### Learning gem5 Tutorial: Simple Config (Simülasyon Nasıl Yapılır?)

[https://www.gem5.org/documentation/learning\\_gem5/part1/simple\\_config](https://www.gem5.org/documentation/learning_gem5/part1/simple_config)

##### Learning gem5 Tutorial: HelloObject (Simülatöre Yeni Tasarım Nasıl Eklenir?)

[https://www.gem5.org/documentation/learning\\_gem5/part2/helloobject](https://www.gem5.org/documentation/learning_gem5/part2/helloobject)

##### Learning gem5 Tutorial: SimpleCache (Simülatöre Örnek Önbellek Eklenmesi)

[https://www.gem5.org/documentation/learning\\_gem5/part2/simplecache](https://www.gem5.org/documentation/learning_gem5/part2/simplecache)

#### a) [50 Puan] Doğrudan Eşlemeli Önbellek Tasarımı

Simülatöre sınıf ismi "**DogrudanOnbellek**" olan bir gem5 simülasyon objesi (-ing, SimObject) ekleyiniz.

Simülatörün Python dosyası aşağıdaki parametrelere sahip olmalıdır.

**gecikme (Param.Cycles):** Bulunma durumunda yanıtın kaç çevrimde verileceğini belirtir.  
**boyut (Param.MemorySize):** Önbelleğin toplam boyutunu bayt cinsinden belirtir.  
**yazpolitika (Param.String):** Aşağıdaki yazma politikalarından hangisinin kullanılacağını belirtir:

- **YAZVEAYIR:** Yaz ve yerini ayır politikası kullanılır.
- **YAZVEAYIRMA:** Yaz ve yerini ayırma politikası kullanılır.

Ek olarak tasarladığınız gem5 simülasyon objesi aşağıdaki istatistikleri tutmalıdır (**Learning gem5 Tutorial: SimpleCache** içerisinde bunları nasıl tutabileceğinizle ilgili kaynak mevcut):

**bulmaSayisi:** Yapılan isteklerin kaçının önbellekte bulunduğunu sayar.

**iskaSayisi:** Yapılan isteklerin kaçının önbellekte bulunmadığını sayar.

Simülasyon sistemini oluşturduğunuz configuration dosyasına **SistemDogrudan.py** adını verin ve genel notlarda belirttiğimiz kurallara uygun şekilde istatistikleri alın ve stats\_dogrudanonbellek.txt ismiyle kayıt edin.

**Dosyaların ekleneceği yol:** gem5/src/mem/cache

**C++ Dosya isimleri:** dogrudanonbellek.hh, dogrudanonbellek.cc

**Python Dosya ismi:** DogrudanOnbellek.py

**Not: Learning gem5 Tutorial: HelloObject ve SimpleCache** içerisinde bu dosyaları simülatöre eklemenizle ilgili yardımcı kaynaklar ve örnek SConscript'ler var.

## b) [50 Puan] Kümeli Eşlemeli Önbellek Tasarımı

Simülatöre sınıf ismi "**KumeliOnbellek**" olan bir gem5 simülasyon objesi (-ing, SimObject) ekleyiniz. Simülatörün Python dosyası aşağıdaki parametrelere sahip olmalıdır.

**gecikme (Param.Cycles):** Bulunma durumunda yanıtın kaç çevrimde verileceğini belirtir.

**boyut (Param.MemorySize):** Önbelleğin toplam boyutunu bayt cinsinden belirtir.

**yol (Param.Int):** Önbelleğin kaç yollu olacağını belirtir.

**cikarpolitika (Param.String):** Aşağıdaki çıkarma politikalarından hangisinin kullanılacağını belirtir:

- **EUZK:** En uzun süredir kullanılmayana çıkar.
- **RASTGELE:** Rastgele bir yol çıkar.

**yazpolitika (Param.String):** Aşağıdaki yazma politikalarından hangisinin kullanılacağını belirtir:

- **YAZVEAYIR:** Yaz ve yerini ayır politikası kullanılır.
- **YAZVEAYIRMA:** Yaz ve yerini ayırma politikası kullanılır.

Ek olarak tasarladığınız gem5 simülasyon objesi aşağıdaki istatistikleri tutmalıdır:

**bulmaSayisi:** Yapılan isteklerin kaçının önbellekte bulunduğunu sayar.

**iskaSayisi:** Yapılan isteklerin kaçının önbellekte bulunmadığını sayar.

Simülasyon sistemini oluşturduğunuz configuration dosyasına **SistemKumeli.py** adını verin ve genel notlarda belirttiğimiz kurallara uygun şekilde istatistikleri alın ve stats\_kumelionbellek.txt ismiyle kayıt edin.

**Dosyaların ekleneceği yol:** gem5/src/mem/cache

**C++ Dosya isimleri:** kumelionbellek.hh, kumelionbellek.cc

**Python Dosya ismi:** KumeliOnbellek.py

**Not: Learning gem5 Tutorial: HelloObject ve SimpleCache** içerisinde bu dosyaları simülatöre eklemenizle ilgili yardımcı kaynaklar ve örnek SConscript'ler var.

## **Ödev Gönderimi ve Formatı**

Ödevinizde işlemci tasarımı için yazdığınız C++ ile Python kodlarınızı ve elde ettiğiniz örnek istatistikleri uzak platformuna **sıkıştırmadan** yükleyeceksiniz.

### **Gönderilecek Dosyalar (Bunlardan başka dosya değerlendirilmeyecek)**

dogrudanonbellek.hh  
dogrudanonbellek.cc  
DogrudanOnbellek.py  
SistemDogrudan.py  
stats\_dogrudsononbellek.txt  
kumelionbellek.cc  
kumelionbellek.hh  
KumeliOnbellek.py  
SistemKumeli.py  
stats\_kumelionbellek.txt

**Son Teslim Tarihi:** 17 Aralık 2023, 23:59