



**ÇİP TASARIM YARIŞMASI  
MİKRODENETLEYİCİ TASARIM KATEGORİSİ  
ÖN TASARIM RAPORU**

**TAKIM ADI: ArchOzu  
BAŞVURU ID: 2007098**

**2024**

# İçindekiler

1. Giriş.....	3
2. Sistem Mimarisi .....	4
2.1 İşlemci Çekirdeği .....	4
2.2 Çevre Birimleri .....	4
2.3 Sistemin Doğrulanması .....	5
3. Tasarım Detayları .....	5
3.1 İşlemci Çekirdeği .....	5
3.1.1 Getir: Bellekten İşlemciye .....	6
3.1.2 Çöz: Anlamlandırma .....	6
3.1.3 Yürüt: Sonucu bul ve kaydet .....	6
3.1.4 Geri Yaz: LOAD Sonucunu Kaydet .....	6
3.2 Bellek Yapısı.....	6
3.3 Bus Yapısı .....	6
3.4 Boot Yapısı .....	6
3.5 Doğrulama.....	6
3.5.1 UVM .....	7
3.5.2 İşlemci Çekirdeği .....	7
3.5.3 Mikrodenetleyici .....	7
3.6 Fiziksel Tasarım .....	7
4. Takım Organizasyonu ve İş Planı.....	8
5. Kaynakça.....	9

# 1. Giriş

Bu proje; açık-kaynak CV32E40P RISC-V çekirdeğine aşağıda belirtilen çevre birimlerini ekleyerek bir mikrodenetleyici yapmayı amaçlamaktadır:

- ✓ UART
- ✓ I2C Master
- ✓ QSPI Master
- ✓ Timer
- ✓ GPIO
- ✓ USB Full-Speed Device (12 Mbps)
- ✓ JTAG

Bu çevre birimlerinin; takım üyeleri tarafından yazılması, açık-kaynak olarak alınmasına tercih edilecektir. Çevre birimlerinin her biri tarafımızca kapsamlı testlere tutulacaktır. Elde edilen mikrodenetleyicinin FPGA implementasyonunun yapılması ve Fiziksel Tasarım çıktılarının elde edilmesi hedeflenmektedir.

[Sistem Mimarisi](#) bölümünde, mikrodenetleyicinin tanımından ve içerdiği birimlerden bahsedilmektedir. Ayrıca, sistemin doğrulamasının nasıl olacağından da bahsedilmektedir.

[Tasarım Detayları](#) bölümünde, mikrodenetleyicinin tasarımı ve doğrulanması detaylıca anlatılmıştır.

[Takım Organizasyonu ve İş Planı](#) bölümünde, öngördüğümüz iş planını detaylıca bulabilirsiniz. Planın seyir akışını bozacağını düşündüğümüz bir sorunla karşılaşırsak planda küçük çapta değişikliğe gidebiliriz.

[Kaynakça](#) bölümünde, raporla ilgili merak ettiğiniz kısımlar hakkında daha detaylı bilgi edinmek için ilgili adresleri IEEE formatında bulabilirsiniz.

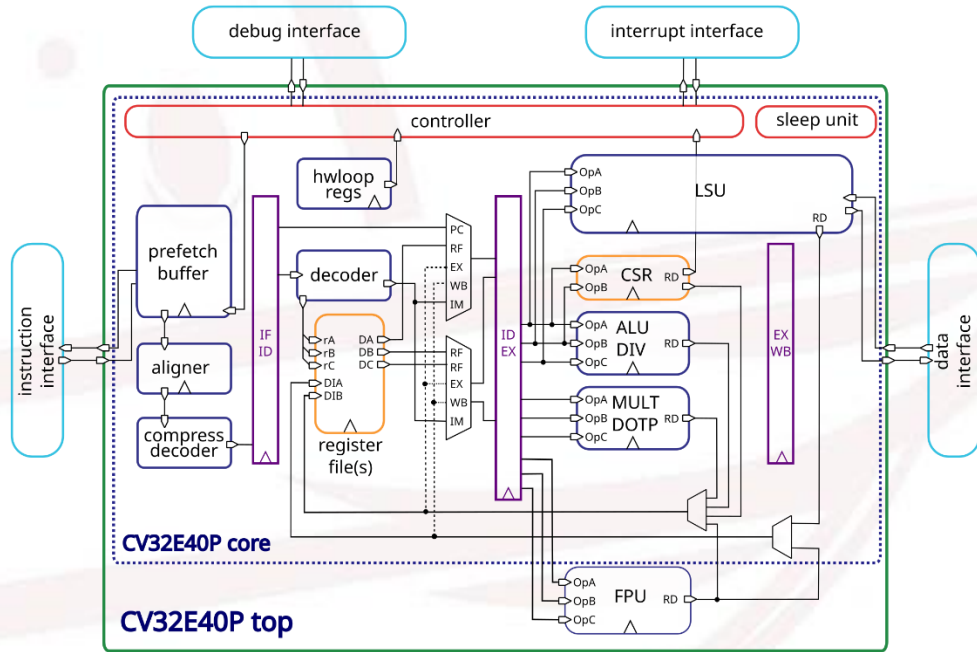


## 2. Sistem Mimarisi

Mikrodenetleyiciler analog ve dijital giriş/çıkışlara sahip olan ve içerisinde mikroişlemci, bellek vb. birçok birim bulunduran elektronik cihazlardır.

### 2.1 İşlemci Çekirdeği

Bu projede, OpenHW Group tarafından tasarlanan CV32E40P işlemci çekirdeğini kullanmaktayız. Bu işlemci çekirdeği, 32-bit RISC-V standardında olup birçok uzantıyı desteklemektedir. İçerisinde bulunan boruhattı 4 aşamalıdır. Mikrodenetleyicilerin öngörülebilir olması istenildiği için boruhattındaki komutlar sırasıyla yürütülür.



Şekil 1 İşlemci Çekirdeği Blok Diyagramı [1]

### 2.2 Çevre Birimleri

Çevre birimleri, Mikrodenetleyicinin dış dünya ile iletişim kurabilmesini sağlar. Bu projede tasarlayacağımız çevre birimleri kısaca şu şekildedir:

- ✓ UART, evrensel asenkron alıcı/verici olup seri iletişim için kullanılan iletişim protokolünü yöneten bir denetleyicidir.
- ✓ I2C, elektronik cihazlar arasında düşük hızlı seri veri iletimi için kullanılan bir protokol olup, I2C Master bu protokole veri alışverişi yapabilen bir denetleyicidir.
- ✓ QSPI Master, hızlı seri veri iletimi için kullanılan bir arayüzü yöneten denetleyicidir.
- ✓ Timer, belirli zaman aralıklarında veya olaylara dayalı olarak işlemleri başlatma ve kontrol etme yeteneğine sahip bir denetleyicidir.
- ✓ GPIO, mikrodenetleyici ile harici cihazlar arasında dijital sinyal iletişimi sağlayan pinlerdir.

- ✓ USB, cihazlar arasında veri aktarımı ve güç sağlama standartlarını yöneten bir denetleyicidir.
- ✓ JTAG, entegre devrelerin test edilmesi, programlanması ve hata ayıklanması için kullanılan bir denetleyici ve iletişim standardıdır.

## 2.3 Sistemin Doğrulanması

Sistemin doğrulanması, Donanım Tasarımında en çok dikkat gerektiren kısımlardan biridir. İşlemci çekirdeği, OpenHW Group tarafından tasarlanmıştır. Detaylar Doğrulama bölümünde anlatılmıştır. UART modülü UVM standardında, geri kalanlar ise genel Verilog test ortamlarında test edilecektir. Sistem, en son genel olarak test edilecektir.

## 3. Tasarım Detayları

Bu bölümde, mikrodeneleyicinin tasarımı ve doğrulanması detaylıca anlatılmıştır.

### 3.1 İşlemci Çekirdeği

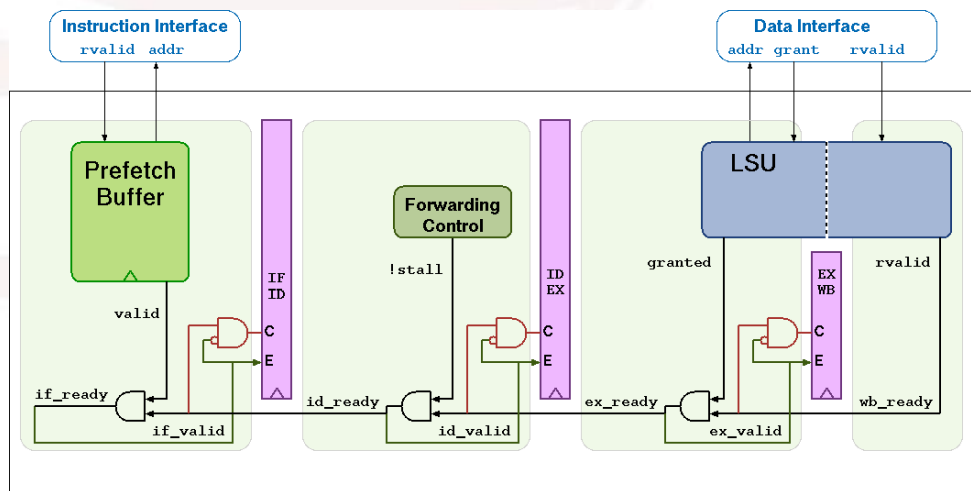
CV32E40P, temel komutlara ek olarak başka komutları da desteklemektedir.

Bunlardan bazıları:

- ✓ Sıkıştırılmış
- ✓ Çarpma ve bölme
- ✓ Performans Sayacı
- ✓ Virgüllü sayı

CV32E40P, 4 aşamaya sahiptir. Sırasıyla:

1. Getir (IF)
2. Çöz (ID)
3. Yürüt (EX)
4. Geri Yaz (WB)



Şekil 2 Boruhattı [2]

### 3.1.1 Getir: Bellekten İşlemciye

Komut belleğinden, oldukça küçük büyüklükteki İGİÇ (İng. FIFO) prensibine sahip arabelleğe (İng. Buffer) komut hizalı bir şekilde getirilir. Daha sonra getirilen bu komut, arabellekten Çöz aşamasına aktarılır.

### 3.1.2 Çöz: Anlamlandırma

Bu aşamada komut, kategorize edilir. Bu kategorizasyon sırasında, ilgili komutun türü, hangi birimlere ihtiyacı olduğu, hangi yazmaçları okumak/yazmak istediği vb. kriterler belirlenir. Böylece benzer komutların aynı alt görevleri aynı devrelerden farklı parametrelerle işlenebilir. Devrelerin birden fazla komut için kullanılabilir olması alan tasarrufu, yerleştirme kolaylığı, maliyet vb. birçok konuda olumludur.

### 3.1.3 Yürüt: Sonucu bul ve kaydet

Aritmetik işlemler, virgüllü sayı işlemleri ve daha birçok işlem hesaplanır. LOAD dışındaki komutların sonucu bu aşamada kaydedilir. Dallanma koşullarının sağlanıp sağlanmadığına bakılır. Bazı komutlar 1'den fazla döngü (İng. Cycle) gerektirebilir. Böyle durumlarda, boruhattı bloke edilir.

### 3.1.4 Geri Yaz: LOAD Sonucunu Kaydet

Veri belleğinden veri çekilmesini sağlayan LOAD komutunun sonucu kaydedilir. İstenirse virgüllü sayı komutlarının sonucu da bu aşamada kaydedilebilir.

## 3.2 Bellek Yapısı

Mikrodenetleyicimizde iki tane bellek bulunacaktır. Bu bellekler şu şekilde sınıflandırılmıştır:

- ✓ 8 kB komut belleği
- ✓ 8 kB veri belleği

Çevre birimlerinin kullanımı için gerekli adresler tanımlanarak CV32E40P ile bağlantısı kurulacaktır.

## 3.3 Bus Yapısı

Çevre birimleri ile işlemciyi birbirine bağlamak için APB protokolü kullanılacaktır. İş planında aksaklıklar çıkarsa zaman kazanmak için açık-kaynak kullanılabilir.

## 3.4 Boot Yapısı

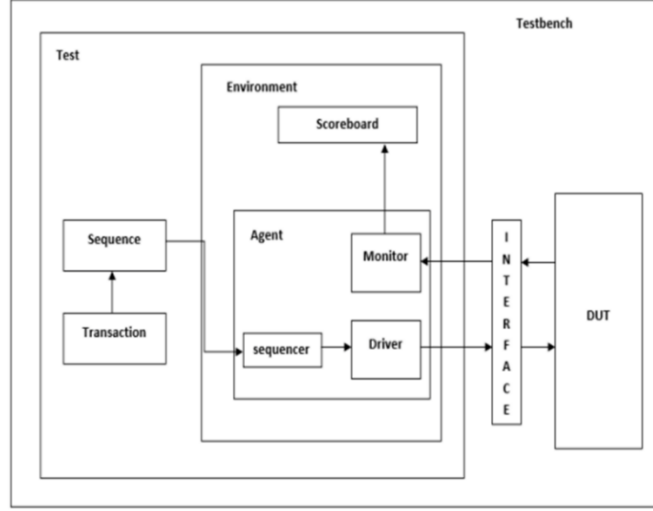
İlk başta, mikrodenetleyici QSPI Master üzerinden 1kB'lık ROM'da tutulan Bootloader kodunu komut belleğine aktaracaktır. Daha sonrasında, komut belleğinden yürütmeye başlayacaktır.

## 3.5 Doğrulama

Bu bölümde UVM ve mikrodenetleyicinin doğrulaması hakkında bilgi verilmiştir.

### 3.5.1 UVM

UVM veya Universal Verification Methodology, yarıiletken endüstrisinde kullanılan dijital tasarımların doğrulaması için standartlaştırılmış bir metodolojidir. Bu, modüler, ölçeklenebilir ve tekrar kullanılabilir doğrulama ortamları oluşturmak için bir çerçeve ve yönerge sağlar. UVM, modern dijital tasarımların karmaşıklığını yönetmede ve doğrulama sürecini kolaylaştırmada etkinliği nedeniyle geniş bir şekilde benimsenmiştir.



Şekil 3 UVM Test Ortamı Mimarisi [3]

### 3.5.2 İşlemci Çekirdeği

İşlemci Çekirdeğinin testleri OpenHW Group tarafından yapılmıştır. Siemens EDA Onespin araçlarından RISC-V ISA İşlemci Doğrulama aracı kullanılarak CV32E40P'nin tamamı doğrulanmıştır.

### 3.5.3 Mikrodenetleyici

C/C++ üzerinden RISC-V derleyicisi çalıştırılıp RISC-V kodu QSPI çevre birimine atılacaktır. Program çalıştığında buradaki ROM'dan okuyarak çalışmaya başlayacaktır.

## 3.6 Fiziksel Tasarım

Fiziksel Tasarım, sırasıyla şu aşamalardan oluşmaktadır:

1. RTL Tasarımı: Devrenin davranışsal modeli yazılır.
2. Sentez: Devrenin ağ listesi (İng. netlist) çıkartılır.
3. Zemin Planlanması (İng. Floorplanning): Güç dağılımını ve makroların yerleştirilmesini içerir.
4. Yerleştirme: Tüm hücreler optimal bir şekilde yerleştirilir.
5. Saat Ağacı Sentezi (İng. CTS): Saat sinyali tüm devreye dengeli bir şekilde dağıtılır.
6. Yönlendirme: Hücreler arası tel bağlantıları yapılır.

## 4. Takım Organizasyonu ve İş Planı

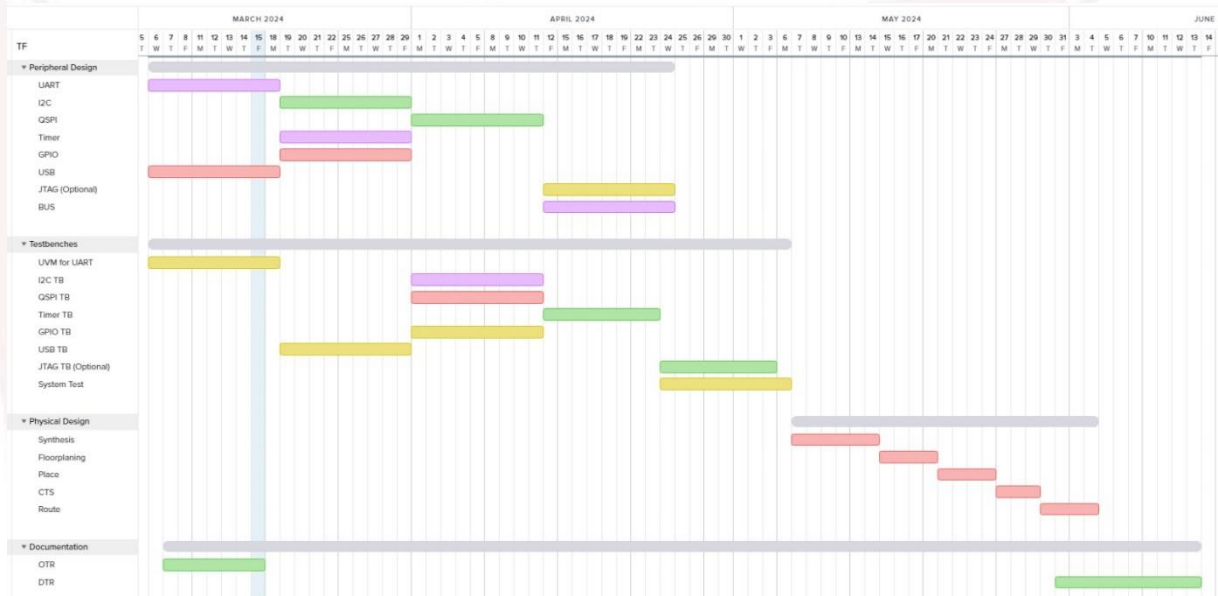
Takımımız, danışmanımız da dahil olmak üzere toplamda 5 kişiden oluşmaktadır. Takımdaki herkes, Özyeğin Üniversitesi'nde okumaktadır. Danışmanımız Cengiz Emre Dedeoğlu, Bilgisayar Mühendisliği bölümünde doktora yapmaktadır. Metin Arda Köker, Bilgisayar Mühendisliği bölümünde; Ahmet Utku Erşahin ise Elektrik-Elektronik Mühendisliği bölümünde yüksek lisans yapmaktadır. Kutay Bulun ve Taha Gemici, Elektrik ve Elektronik Mühendisliği bölümünde lisans öğrencileridir.

Ahmet Utku Erşahin; UART ve Timer çevre birimlerinin RTL tasarımlarından, I2C Master çevre biriminin test ortamından sorumludur.

Metin Arda Köker; USB ve GPIO çevre birimlerinin RTL tasarımından, QSPI çevre biriminin test ortamından sorumludur. Projenin Fiziksel Tasarım kısmını ikili beraber yapacaktır.

Taha Gemici; geriye kalan çevre birimlerinin RTL tasarımlarının yapılmasından, JTAG çevre biriminin doğrulanmasından, tüm sistemin FPGA üzerindeki implementasyonundan ve raporların hazırlanmasından sorumludur.

Kutay Bulun, geriye kalan RTL tasarımların doğrulanmasından ve JTAG çevre biriminin RTL tasarımından sorumludur. İlk başta UART için UVM standardına uygun test modülü oluşturacaktır. Devamında, geriye kalan çevre birimleri için test modülleri geliştirecektir.



Şekil 4 İş Planı



## 5. Kaynakça

- [1] «CORE-V CV32E40P User Manual - Introduction,» OpenHW, 2023. [Çevrimiçi]. Available: <https://docs.openhwgroup.org/projects/cv32e40p-user-manual/en/latest/intro.html>. [Erişildi: 12 Mart 2024].
- [2] «CORE-V CV32E40P User Manual - Pipeline Details,» OpenHW, 2023. [Çevrimiçi]. Available: <https://docs.openhwgroup.org/projects/cv32e40p-user-manual/en/latest/pipeline.html>. [Erişildi: 12 Mart 2023].
- [3] M. K. N. Vivekananda T, «Design and Verification of the UART and SPI protocol using UVM,» 2022. [Çevrimiçi]. Available: <https://www.irjet.net/archives/V9/i9/IRJET-V9I918.pdf>.