



Araştırma Makalesi/Research Article

## Vektör hızlandırıcılı işlemci tasarımı

Abdülkadir Çakır<sup>1</sup>, Necati Çakacı<sup>2</sup>

<sup>1</sup>Isparta Uygulamalı Bilimler Üniversitesi, Teknoloji Fakültesi, Elektrik-Elektronik Mühendisliği Bölümü, 32040, Isparta, Türkiye

<sup>2</sup>Isparta Uygulamalı Bilimler Üniversitesi, Lisansüstü Eğitim Enstitüsü, Elektrik-Elektronik Mühendisliği Anabilim Dalı, 32260, Isparta, Türkiye

### Anahtar Kelimeler

RISC-V İşlemci  
Vektör Hızlandırıcı  
Sayısal Tasarım

### Makale geçmişi:

Geliş Tarihi: 27.03.2025

Kabul Tarihi: 05.12.2025

**Öz:** FPGA ile yapılan gömülü sistem uygulamalarında çeşitli işlemleri kolaylıkla yürütebilmek amacıyla işlemci çekirdeklerinden faydalanılmaktadır. Bu işlemci çekirdekleri FPGA yongasında fiziksel olarak yer alabileceği gibi, bir RTL tasarım kodu ile de FPGA üzerinde sentezlenebilir. FPGA yongasında gömülü olan işlemciler yüksek başarımla çalışabilir ancak özelleştirme seçenekleri kısıtlıdır. Buna karşın RTL tasarım kodu üzerinden FPGA üzerinde çalıştırılan işlemciler, gereksinimlere uygun olarak kolayca düzenlenebilir ve farklı amaçlar için yeniden yapılandırılabilir. Bu çalışmada, Verilog donanım tanımlama dili kullanılarak FPGA uygulamalarında kullanılmak üzere vektör hızlandırıcı yapısına sahip RV32IMB\_Zicond mimarili bir RISC-V işlemci tasarımı geliştirilmiştir. Proje, parametreler üzerinden ihtiyaca uygun olarak yeniden yapılandırılabilir. Tasarlanan sistemde, işlemci ve hızlandırıcı yapılarına ek olarak FPGA kaynaklarını kullanan bellek modülü ve Wishbone arayüzü ile uyumlu UART ve zamanlayıcı çevrebirimleri yer almaktadır. Çalışmada tasarlanan sistem, benzetim ortamında test edilmiş ve bir FPGA geliştirme kartı üzerinde çalıştırılmıştır. Sistemin farklı konfigürasyonları alan ve frekans bakımından karşılaştırılmış ve sentetik test üzerinden başarımlar ölçümü yapılmıştır.

### Atıf için/To Cite:

Çakır A. Çakacı N. Vektör hızlandırıcılı işlemci tasarımı. Uluslararası Teknolojik Bilimler Dergisi, 17(2), 20-27, 2025.

## Processor design with vector accelerator

### Keywords

RISC-V Processor  
Vector Accelerator  
Digital Design

### Article history:

Received: 27.03.2025

Accepted: 05.12.2025

**Abstract:** In FPGA embedded system applications, processor cores are essential for executing various operations efficiently. These cores can either be physically embedded within the FPGA chip or synthesized using an RTL design. Embedded processors offer high performance but limited customization, whereas synthesized processors provide flexibility and reconfigurability for different applications. This study presents a RISC-V processor design with a vector accelerator structure and an RV32IMB\_Zicond architecture, developed using the Verilog hardware description language for FPGA applications. The design is highly parameterized, allowing reconfiguration to meet specific requirements. In addition to the processor and accelerator structures, the system integrates a memory module leveraging FPGA resources, as well as UART and timer peripherals compatible with the Wishbone interface. The system was tested in a simulation environment and implemented on an FPGA development board. Different configurations were evaluated in terms of area, operating frequency, and performance, using synthetic benchmark.

## 1. Giriş

Günümüzde batarya ile çalışan taşınabilir cihazların yaygınlaşması, düşük güç tüketen mikroişlemci ve mikrodenetleyici sistemlerine duyulan ihtiyacı arttırmıştır. Bu sistemler genellikle Reduced

Instruction Set Computer (RISC) mimarilerini temel almaktadır. RISC mimarileri, az sayıda buyruğun basit işlemler gerçekleştirilmesi prensibine dayanmaktadır. Bu sayede işlemci donanımı sadeleşir ve küçülür. İşlemci donanımındaki basitlik, enerji ve başarımlar bakımından verimli yongaların üretilmesine veya Field

\* İlgili yazar/Corresponding author: necaticakaci@outlook.com

Programmable Gate Array (FPGA) gibi programlanabilir mantık devrelerinin içinde çalıştırılabilmesine olanak sağlamaktadır.

Mikroişlemcilerle geliştirme süreci, FPGA ile yapılan sayısal tasarımlara nispetle genellikle daha az zahmetlidir. Bu sebepten ötürü, bazı FPGA yongalarında gömülü işlemciler yer almaktadır [1]. Bu işlemciler, FPGA üzerinde sayısal tasarım olarak gerçekleştirilmesi zor algoritma ve uygulamalar için yüksek seviyeli programlama dili kullanarak geliştirme kolaylığı tanımaktadır. Fakat maliyet ve güç tüketimi gibi nedenlerden dolayı her FPGA yongası gömülü işlemci barındırmayabilir. Bu durumda soft-core olarak adlandırılan ve Register Transfer Level (RTL) tasarım kodu ile FPGA üzerinde sentezlenebilen işlemci yapıları kullanılabilmektedir [2]. Söz konusu soft-core işlemci Intellectual Property (IP)'leri, FPGA üreticisi veya üçüncü parti geliştiriciler tarafından sağlanabilir.

Günümüzde FPGA ile birlikte kullanılmak için tasarlanmış birçok açık kaynaklı işlemci projesine internet üzerinden erişilebilmektedir [3]. Bu çalışmaların büyük bir kısmı RISC-V buyruk kümesi mimarisini baz almaktadır. RISC-V, Kaliforniya Berkeley Üniversitesinde 2010 yılında geliştirilmeye başlanan bir buyruk kümesi mimarisidir [4]. RISC-V, diğer yaygın buyruk kümesi mimarilerinin aksine Ticari ve akademik kullanımlar için açık lisansa sahiptir. Özelleştirilebilir ve modüler yapısı ile uygulamaya özel işlemci tasarımlarına olanak sağlamaktadır. RISC-V ekosistemi endüstri ve akademi desteği ile birlikte gün geçtikçe daha fazla büyümektedir.

En basit haliyle bir RISC-V işlemcinin, temel tamsayı buyruk kümesini desteklemesi gerekmektedir. Bu buyruk kümeleri 32-bit için RV32I veya RV32E, 64-bit için RV64I veya RV64E olabilir. I varyasyonlar 32 adet kaydedici içerirken, E varyasyonlar 16 adet kaydedici içermektedir [5].

Bir RISC-V işlemci, temel tamsayı buyruk kümesine ek olarak standart buyruk eklentilerini içerebilir. Bu eklentilerin gerçekleşmesi isteğe bağlı olmakla birlikte, gerçekleştiklerinde çeşitli uygulamalar için başarımlar artışı sağlayabilmektedirler [6]. RISC-V buyruk kümesinde yer alan bazı standart eklentiler Tablo 1'de gösterilmiştir.

Tablo 1. Bazı RISC-V eklentileri

| Eklenti | Açıklama                          |
|---------|-----------------------------------|
| M       | Tamsayı çarpma ve bölme işlemleri |
| A       | Atomik bellek işlemleri           |
| F       | 32-bit kayan nokta işlemleri      |
| D       | 64-bit kayan nokta işlemleri      |
| C       | 16-bit sıkıştırılmış işlemler     |
| B       | Bit manipülasyonu işlemleri       |

RISC-V, standart eklentilerin yanı sıra tasarımcıların kendi belirledikleri özel eklentileri ekleyebilmesine olanak sağlamaktadır. Bu sayede uygulamaya özel işlemci tasarımları gerçekleştirilebilmektedir. RISC-V mimarisinin esnek ve genişletilebilir yapısı, gömülü sistemlerden sunucu sistemlerine kadar çeşitli alanlarda kullanılabilir hale getirmektedir [7].

Bu çalışmada FPGA uygulamalarında kullanmaya uygun, RISC-V mimarili 32-bit bir işlemcinin merkezinde yer aldığı, vektör hızlandırıcı ve çeşitli çevrebirimleri içeren bir sistemin tasarımı hedeflenmiştir. Tasarımda Vivado 2022.2 yazılımı ve Nexys A7-100T FPGA geliştirme platformu kullanılmıştır. Sistem bileşenleri, Verilog 2005 donanım tanımlama dili kullanılarak geliştirilmiştir. Derleyici olarak GNU Compiler Collection (GCC) kullanılmış ve C geliştirme ortamı için gerekli temel kütüphane desteği sağlanmıştır. Sistem bilgisayar ortamında benzetim ile ve fiziksel olarak FPGA kartında çalıştırılıp test edilmiştir.

## 2. Sistem Mimarisi

Tasarlanan sistem; işlemci çekirdeği, bellek alt sistemi, vektör hızlandırıcı birimi ve çevrebirim alt sistemi olmak üzere toplam 4 parçadan oluşmaktadır (Şekil 1).



Şekil 1. Projenin genel sistem mimarisi

Bellek alt sisteminde FPGA üzerinde sentezlenebilen ve boyutu parametrik olarak ayarlanabilen bir Random Access Memory (RAM) ve bellek-işlemci arasındaki iletişimi yöneten bellek kontrolcüsü yer almaktadır.

İşlemci çekirdeği, en yüksek konfigürasyonda RV32IMB\_Zicond mimarisini destekleyebilirken, en düşük konfigürasyonda RV32I mimarisini desteklemektedir. İşlemcinin destekleyeceği eklentiler, yapılandırma başlık dosyası üzerinden birbirinden bağımsız olarak seçilebilmektedir.

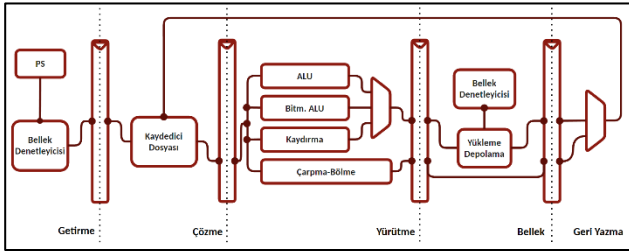
Vektör hızlandırıcı birimi, işlemci çekirdeğine doğrudan bağlıdır ve 8 adet Aritmetik-Logic Unit (ALU) içermektedir. Bu birimde çeşitli işlemler paralel olarak yürütülmektedir. Vektör hızlandırıcının yönetimini işlemci çekirdeği üstlenmektedir.

Çevrebirim alt sistemi; Wishbone kontrolcüsü, Universal Asynchronous Receiver Transmitter (UART) ve zamanlayıcı çevrebirimlerinden meydana gelmektedir. Çevrebirimlerin işlemci çekirdeği ile olan

veri alışverişinde açık kaynaklı bir standart olan Wishbone protokolü kullanılmıştır. UART, tasarlanan sistemin temel girdi ve çıktı işlemleri için kullanılırken zamanlayıcı birimi, başarımlı ölçümü gibi uygulamalarda kullanılmaktadır.

### 2.1. İşlemci çekirdeği

İşlemci çekirdek tasarımında 5 aşamalı boru hattı metodolojisi kullanılmıştır. Tasarlanan mimarinin boru hattı aşamaları; getirme, çözme, yürütme, bellek erişimi ve geri yazma olarak adlandırılmıştır. Konfigürasyon başlık dosyasında tanımlı parametrelere göre gerçekleşen işlemci mikro mimarisi değişiklik gösterebilmektedir. Örneğin, RV32M standart eklentisinin seçilmediği durumda gerçekleşen çekirdekte çarpma-bölme birimi yer almaz. En üst konfigürasyon için işlemci çekirdeğinin şeması Şekil 2'de gösterilmiştir.



Şekil 2. İşlemci çekirdek şeması

Tasarlanan işlemcinin boru hattında bulunan ilk aşama getirmedir. Bu aşamada program sayacı kaydedicisi bulunmaktadır. Program sayacında bellekten okunacak buyruğun adresi tutulur ve bu adres bellek denetleyicisi üzerinden belleğe iletilir. Bellekten okunan buyruk, getirme-çözme aşamaları arasında yer alan boru hattı kaydedicilerine aktarılır. Buyruk, bir sonraki saat çevriminde çözme aşamasına iletilir.

Çözme aşamasında, gereken kaydedici değerleri kaydedici dosyasından okunur. Buyruk ivedi formda değer içeriyorsa, genişletme uygulanır. Çözme aşamasında elde edilen değerler bir sonraki aşama olan yürütmeye aktarılır.

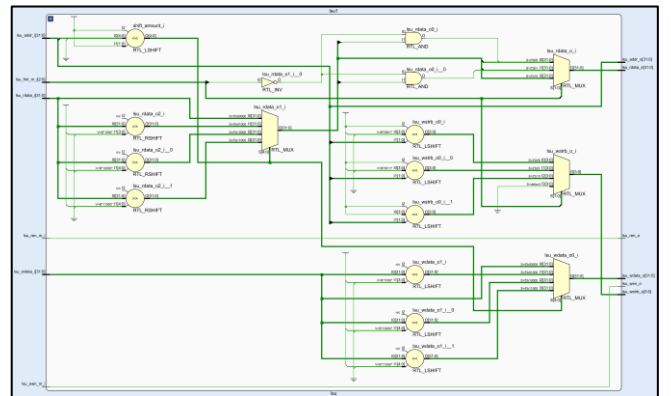
Yürütme aşamasında çeşitli hesaplama birimleri bulunmaktadır. Temel tamsayı buyruk kümesinde yer alan buyruklar Arithmetic Logic Unit (ALU) modülünde işlenir. M standart eklentisi içinde yer alan çarpma ve bölme buyrukları, çarpma-bölme biriminde yürütülür. Çarpma-bölme biriminde her çarpma işlemi 1 saat çevriminde tamamlanmaktadır. Bölme işlemleri ise 32 saat çevriminde tamamlanmaktadır. Bölme işlemi devam ederken boru hattı durdurulur. B standart eklentisi, bit manipülasyonları için destek sağlamaktadır. Bu buyruk kümesi; Zba, Zbb, Zbs, Zbc alt buyruk kümelerinden meydana gelmektedir. B standart

eklentisi altında yer alan çoğu buyruk, bit manipülasyonu ALU modülünde işlenir. Kaydırma ve döndürme işlemleri gerektiren buyruklar, kaydırma modülünde işlenir. ALU, bit manipülasyonu ALU ve kaydırma birimleri kombinasyonel devrelerden meydana gelmektedir ve tek saat çevriminde sonucu bir sonraki aşamaya iletmektedir.

RISC-V Zicnd eklentisi mimariye czero. eqz ve czero. nez olarak adlandırılan iki buyruk eklemektedir. Bu buyruklar temel olarak, bazı koşul ifadelerinin dallanma buyruklarını kullanmadan işlenmesini sağlamaktadır. Dallanma buyruklarının yanlış tahmini işlemci başarımını düşürdüğü için Zicnd buyruklarının kullanıldığı durumda başarımlı iyileşmektedir. Zicnd buyrukları, bir kaynak kaydedicisinin değerinin sıfır olup olmamasına göre bir hedef kaydedicisine sıfır veya bir başka kaynak kaydedicisinin değerini yazmaktadır. Zicnd işlemi ALU modülünde yürütülmektedir.

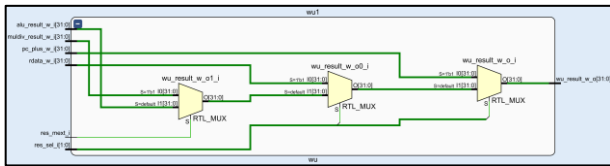
Yürütme biriminin gerçekleşen mimarisi, seçilen buyruk kümesi konfigürasyonu ile doğrudan ilgilidir. RV32I mimarisi gerçekleştirildiğinde yürütme birimi içinde yalnızca ALU modülü yer alır. RV32IM mimarisinin gerçekleştirildiği durumda yürütme birimine çarpma-bölme modülü eklenir. RV32IMB mimarisinin gerçekleştirildiği durumda ise, bit manipülasyonu ALU modülü ve kaydırma modülü yürütme birimine eklenir. Bu durumda temel tamsayı biriminde bulunan kaydırma buyruklarının hesaplanması ALU yerine kaydırma modülünde yapılır. RV32IMB\_Zicnd mimarisinin gerçekleştirildiği durumda Zicnd eklentisinde yer alan iki buyruk için ALU modülüne destek eklenir.

Yürütme aşamasının ardından bellek erişim aşaması yer almaktadır. Bu aşamada yükleme-depolama birimi bulunur. İşlenen buyruğun belleğe erişim gerektirmesi halinde yükleme-depolama birimi, bellek denetleyicisine erişim isteği iletir. Hızasız erişimler yükleme-depolama birimi tarafından desteklenmez. Bu birimin RTL tasarım şeması Şekil 3'te gösterilmiştir.



Şekil 3. Yükleme-depolama birimi

Bellek erişim aşamasının ardından boru hattındaki son aşama olan geri yazma gelir. İşlenen buyruk bir hedef kaydedicisi içermekteyse, bu aşamada hedef kaydedicisine yazılacak değerin nereden geleceği seçilir. Bu değerin kaynağı; ALU, çarpma-bölme birimi, program sayacı veya yükleme-depolama birimi olabilir. Geri yazma birimi, kaynak sinyalleri arasından seçim yapmakla görevli bir çoklayıcı yapısı içerir. Geri yazma biriminin RTL tasarımı şeması Şekil 4'te verilmiştir.

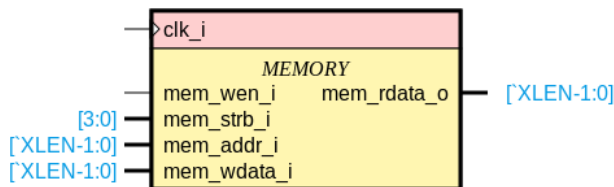


Şekil 4. Geri yazma birimi

## 2.2. Bellek yapısı

Tasarlanan sistem, büyük ve veri için tek RAM bellek içerecek biçimde, Von Neumann mimarisinde tasarlanmıştır. Bellek FPGA için sentezlenebilmekte ve boyutu parametrik olarak ayarlanabilmektedir. Varsayılan bellek boyutu 32 KB olarak belirlenmiştir.

İşlemci çekirdeğinden belleğe giden, buyruk okuma ve veri yükleme-depolama istekleri bellek kontrolcüsü tarafından sıralanmaktadır. Bir yükleme-depolama işlemi yürütülürken buyruk okuma işlemi durdurulmaktadır. Tasarlanan belleğin üst modül bağlantıları Şekil 5'te gösterilmiştir.



Şekil 5. Bellek üst modülü

Belleğin programlanması, sentezleme aşamasından önce RISC-V makine kodunu içeren bir HEX dosyasının belleğe gömülmesi ile gerçekleştirilmektedir.

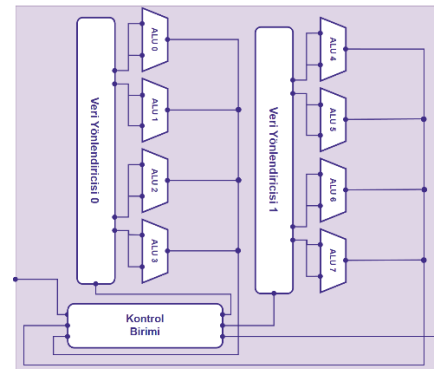
### 2.3. Vektör hızlandırıcı

Vektör hızlandırıcı birimi, sistemin genel hesaplama başarımını arttırmak amacıyla tasarlanmıştır. Bu birim, aynı anda birden fazla aritmetik ve mantıksal işlemi gerçekleştirebilmektedir. Hızlandırıcı birimi sayesinde işlemci yükü dağıtılır ve işlemciye ait bazı görevler hızlandırıcı birim tarafından yerine getirilir. Bu sayede sistem genelinde başarımlar artışı sağlanır. Tasarlanan vektör hızlandırıcı birimi 8 adet ALU içermektedir. Tasarlanan ALU yapıları çeşitli aritmetik ve mantıksal işlemleri desteklemektedir. Desteklenen işlemler Tablo 2'de listelenmiştir. Sınırlı kaynaklara sahip FPGA

platformlarında sistemi gerçekleyebilmek için, çarpma işlemleri konfigürasyon dosyasından devre dışı bırakılabilmektedir. Tasarlanan vektör hızlandırıcı biriminin yapısı Şekil 6'da gösterilmiştir.

**Tablo 2. Vektör hızlandırıcının desteklediği işlemler**

| İşlem    | Açıklama                       | İşlenen adedi<br>× işlenen bit<br>genişliği |
|----------|--------------------------------|---------------------------------------------|
| VEC_ADD  | Vektör toplama işlemi          | 16×32                                       |
| VEC_SUB  | Vektör çıkarma işlemi          | 16×32                                       |
| VEC_AND  | Vektör ve işlemi               | 16×32                                       |
| VEC_XOR  | Vektör özel veya işlemi        | 16×32                                       |
| VEC_OR   | Vektör veya işlemi             | 16×32                                       |
| VEC_NOT  | Vektör değil işlemi            | 8×32                                        |
| VEC_SLL  | Vektör sola kaydırma           | 16×32                                       |
| VEC_SRL  | Vektör sağa kaydırma           | 16×32                                       |
| VEC_SRA  | Vektör aritmetik sağa kaydırma | 16×32                                       |
| VEC_MUL  | Vektör çarpım (Düşük 32-bit)   | 16×32                                       |
| VEC_MULH | Vektör çarpım (Yüksek 32-bit)  | 16×32                                       |



Şekil 6. Vektör hızlandırıcı birimi

Vektör hızlandırıcı biriminin çalışması, işlemci tarafından yönetilmektedir. İşlemci, vektör hızlandırıcıya hesaplanacak değerleri yükler ve yapılacak işlemi belirtir. Vektör hızlandırıcı birimi tek çevrimde sonuçları hesaplar. Vektör hızlandırıcının ürettiği sonuçlar, işlemci tarafından okunur. Vektör hızlandırıcı birimi bellek haritasına eşlemeli biçimde tasarlanmış ve bellek uzayının 0x30000000 adresine yerleştirilmiştir.

## 2.4. Çevrebirimler

Çevrebirim alt sistemi UART, zamanlayıcı ve Wishbone kontrolcüsünden meydana gelmektedir. UART çevrebirimi, sistemin çıktılarının fiziksel olarak gözlemlenebilmesi ve çeşitli test sonuçlarının elde edilebilmesi için sisteme dahil edilmiştir. Zamanlayıcı birimi, süre ölçümü gerektiren uygulamalarda kullanılmak amacıyla tasarlanmıştır. Tasarlanan çevrebirimler ve işlemci arasındaki iletişimin yönetimi için bir veri yolu kontrolcüsü de tasarlanmıştır. Tasarlanan veri yolu kontrolcüsü, açık kaynaklı bir standart olan Wishbone arayüzü ile uyumludur. Çevrebirim alt sisteminin modül bağlantı şeması Şekil 7'de gösterilmiştir.



Tasarım tanımı aşamasında proje isterleri ve çalışmanın kapsamı belirlenmiştir. Tasarım girişi aşamasında, daha önceden hazırlanmış proje taslağı baz alınarak Verilog donanım tanımlama dili ile sistemi oluşturan bileşenler modellenmiştir. Bu aşamada, RTL tasarımı yer alan bağlantıların, FPGA'nın fiziksel bağlantılarına eşlenmesini sağlayan kısıt dosyası da hazırlanmıştır. Modellenen sistem bileşenleri benzetim aşamasında test edilmiştir. Benzetim işlemlerinde, projede kullanılan FPGA platformunun geliştirme ortamı olan Vivado yazılımından faydalanılmıştır. Benzetim sonucunda doğrulanmış sistem bileşenleri, bir sonraki aşamada sentez işlemine tabi tutulmuştur.

Sentezleme aşamasında, sentez aracı tasarım kodunu girdi olarak alır ve bir Netlist dosyası üretir. Bu işlem sonucunda tasarım, FPGA kaynaklarına dönüştürülmüş olur. Sentezleme aşamasının ardından gerçekleştirme aşamasına geçilir. Bu aşama, birkaç adımdan meydana gelir. Gerçekleme aşamasında opsiyonel olarak hız, alan ve güç tüketimini iyileştirmek için tasarım optimizasyonları yürütülebilir. Gerçekleme aşamasının bir sonraki adımında sentezlenen ve optimize edilen tasarım FPGA mantık hücrelerine yerleştirilir. Bu adımın ardından başka opsiyonel optimizasyonlar yürütülebilir [10].

Gerçekleme aşamasının ardından yönlendirme aşamasına geçilir. Bu aşamada, FPGA mantık hücrelerine yerleştirilen tasarımın iç bağlantıları yapılır. Yönlendirme aşamasından sonra bazı optimizasyonlar opsiyonel olarak yürütülebilir. Son aşamada, gerçekleştirilen tasarım üzerinden "bitstream" dosyası üretilir. Bu dosya FPGA'yı programlamak için kullanılır.

Proje kapsamında yapılan çalışmada Vivado FPGA akışı kullanılmıştır. Sistem başarımını arttırmak için varsayılan sentez stratejisi "performans optimizasyonu" olarak değiştirilmiştir.

Tasarlanan sistemin kolay biçimde test edilebilmesi için C programlama dili desteği eklenmiştir. Bunun için, bağlayıcı betiği (linker script), assembly başlangıç programı, sürücü kütüphanesi ve vektör hızlandırıcı kütüphanesi gibi bazı yardımcı program ve kütüphaneler geliştirilmiştir. Hazırlanan bağlayıcı betiği, derleyiciye bellek haritasını belirtmek için kullanılmaktadır. Assembly başlangıç kodu, sistem açılış ayarlarının yapılması ve C kodundaki "main" fonksiyonuna dallanmayı sağlamak için yazılmıştır. Geliştirilen kütüphaneler, sistemde yer alan çevrebirimler ve hızlandırıcı yapıları için yardımcı fonksiyonlar sağlamaktadır. Derleyici olarak RISC-V GNU Compiler Collection (GCC) kullanılmıştır. Derleyici tarafından üretilen ikilik tabandaki program kodları,

onaltılık tabana dönüştürülerek işlemci belleğine yerleştirilerek programlama gerçekleştirilmektedir.

#### 4. Bulgular

Tasarlanan sistemin bileşenleri, işlemci çekirdeği, vektör hızlandırıcı ve çevrebirimler olmak üzere ayrı ayrı sentez aracından geçirilmiş ve elde edilen sonuçlar gözlemlenmiştir. Buna ek olarak çekirdeğin farklı konfigürasyonları kaynak tüketimi bakımından incelenmiştir. Çevrebirimler tarafından kullanılan Look-Up Table (LUT), Flip-Flop ve Digital Signal Processing (DSP) kaynak sayısı Tablo 3'te gösterilmiştir.

Tablo 3. Çevrebirimlerin kaynak tüketimi

| Birim       | LUT | Flip-Flop |
|-------------|-----|-----------|
| UART        | 387 | 693       |
| Zamanlayıcı | 34  | 97        |

DSP kaynağının yetersiz olduğu FPGA platformları için vektör hızlandırıcı biriminden çarpma işlemi kaldırılabilir. Çarpma işlemi içeren ve içermeyen vektör hızlandırıcı biriminin kaynak tüketimi Tablo 4'te verilmiştir.

Tablo 4. Vektör hızlandırıcının kaynak tüketimi

| Durum            | LUT  | Flip-Flop | DSP |
|------------------|------|-----------|-----|
| Çarpma etkin     | 3412 | 557       | 32  |
| Çarpma devredışı | 3004 | 558       | 0   |

Çekirdeğin farklı eklenti yapılandırmalarının kaynak tüketimine etkisi Tablo 5'de gösterilmiştir. Beklenildiği üzere minimum konfigürasyon (RV32I) en az kaynak tüketen mimari olurken, maksimum konfigürasyon (RV32IMB\_Zicond) en fazla kaynak tüketen mimari olmuştur.

Tablo 5. İşlemci çekirdeğinin kaynak tüketimi

| Mimari         | LUT  | Flip-Flop | DSP |
|----------------|------|-----------|-----|
| RV32I          | 1162 | 534       | 0   |
| RV32IM         | 1972 | 765       | 4   |
| RV32IMB        | 2906 | 774       | 4   |
| RV32IMB_Zicond | 3196 | 780       | 4   |

Tüm sistem, Nexys A7-100T FPGA için maksimum konfigürasyona göre gerçekleştirildiğinde toplamda 6768 LUT, 2122 flip-flop ve 36 DSP kaynağı harcamaktadır. Sistem 156 mW dinamik ve 98 mW statik olmak üzere toplam 254 mW güç harcamaktadır. İşlemci çekirdeğinin konfigürasyonlara göre maksimum çalışma saat frekansları Tablo 6'da verilmiştir.

Tablo 6. İşlemci çekirdeğinin maksimum frekansı

| Mimari         | Maksimum Çalışma Frekansı (MHz) |
|----------------|---------------------------------|
| RV32I          | 70                              |
| RV32IM         | 60                              |
| RV32IMB        | 60                              |
| RV32IMB_Zicond | 60                              |

İşlemci başarımını test etmek için, gömülü sistemlerde yaygın olarak kullanılan CoreMark sentetik başarımlar ölçüm programı kullanılmıştır. CoreMark, Embedded Microprocessor Benchmark Consortium (EEMBC) tarafından 2009 yılında geliştirilen bir mikroişlemci başarımlar ölçüm aracıdır [11].

CoreMark programının çalıştırılabilmesi için tasarlanan işlemci sistem mimarisi ile uyumlu olacak biçimde tekrar uyarlanması (port edilmesi) gerekmektedir. Bunun için “core\_portme.c” ve “core\_portme.h” dosyaları üzerinde gerekli düzenlemeler yapılmıştır.

CoreMark, işlemcinin maksimum konfigürasyonunda Nexys A7-100T FPGA ile 60 MHz frekansında 500 iterasyon çalıştırılmıştır. Buna göre işlemci, saniyede 29 iterasyon tamamlamış ve 0.48 CoreMark/MHz başarımlarına ulaşmıştır. CoreMark programının çıktısı Şekil 11’de gösterilmiştir.

```
UART initialization complete.
Benchmark starting...
2K performance run parameters for coremark.
CoreMark Size      : 666
Total ticks        : 1040339905
Total time (secs): 17
Iterations/Sec     : 29
Iterations         : 500
Compiler version   : GCC12.1.0
Compiler flags     : -o3
Memory location    : STACK
seedcrc           : 0xe9f5
[0]crclist        : 0xe714
[0]crcmatrix      : 0x1fd7
[0]crcstate       : 0x8e3a
[0]crcfinal       : 0xa14c
Correct operation validated. See README.md for run and reporting rules.
```

Şekil 11. İşlemci CoreMark sonucu

## 5. Sonuçlar

Bu çalışmada, FPGA platformlarında kullanılmak üzere özelleştirilebilir 32-bit RISC-V mimarili işlemci, vektör hızlandırıcı ve çevrebirimleri içeren bir sistem tasarımı yapılmıştır. Proje, Nexys A7-100T FPGA kartı ve Vivado yazılımı kullanılarak Verilog dili ile geliştirilmiştir. Projede işlemci çekirdeği modüler yapıda tasarlanmıştır. Bu sayede kullanıcı, FPGA platformuna ve gerçekleştireceği uygulamaya göre çekirdek mimarisini yeniden yapılandırabilmektedir. Tasarlanan sistemin C programlama dili ile kullanılabilmesini sağlamak için çeşitli kütüphane ve yardımcı dosyaları içeren “baremetal” geliştirme ortamı hazırlanmıştır.

Sistem bileşenlerinin ayrı ayrı sentezlenmesi ve farklı konfigürasyonlardaki işlemci çekirdeğinin kaynak tüketiminin incelenmesi, tasarımın esnekliğini ve optimizasyon potansiyelini ortaya koymuştur. Elde edilen sonuçlar, vektör hızlandırıcının çarpma birimi devre dışı bırakılarak kaynak kullanımında önemli ölçüde azalma sağlanabileceğini göstermiştir. Ayrıca, işlemci çekirdeği konfigürasyonunun (RV32I’den RV32IMB\_Zicnd’a) LUT, flip-flop ve DSP kaynakları üzerindeki etkileri analiz edilmiştir. Sistem

entegrasyonu sonrasında yapılan testler, tasarlanan sistemin başarılı bir şekilde çalıştığını ve belirtilen performans hedeflerine ulaştığını doğrulamıştır.

Proje kapsamında yapılan sistem tasarımı, işlemci mikro mimarisinde yapılacak iyileştirmeler, farklı bellek mimarilerinin kullanımı ve çevrebirimlerin arttırılması gibi hususlarla geliştirilebilir.

## Teşekkür

2023-YL1-0201 numaralı proje ile bu çalışmayı maddi olarak destekleyen Isparta Uygulamalı Bilimler Üniversitesi Bilimsel Araştırma Projeleri Yönetim Birimi Başkanlığı’na teşekkür ederiz.

## Kaynaklar

- [1] AMD. Zynq 7000 Product Brief. <https://www.amd.com/content/dam/amd/en/documents/products/adaptive-socs-and-fpgas/soc/zynq-7000-product-brief.pdf> (Erişim Tarihi: 25.03.2025)
- [2] Tong JG, Anderson ID, Khalid MA. Soft-core processors for embedded systems. *2006 International Conference on Microelectronics, Dhahran, 14-17 May 2006*.
- [3] Höller R, Haselberger D, Ballek D, Rössler P, Krapfenbauer M, Linauer M. Open-source risc-v processor ip cores for fpgas—overview and evaluation. *8th Mediterranean Conference on Embedded Computing (MECO)*, Montenegro, 10-14 June 2019.
- [4] Waterman A, Lee Y, Patterson DA, Asanovic K. The RISC-V instruction set manual, volume I: User-level ISA, version 2.0. *EECS Department, University of California, Berkeley*, Tech. Rep. UCB/EECS-2014-54, 4.
- [5] Patterson D, Waterman A. The RISC-V Reader: an Open Architecture Atlas. Strawberry Canyon, California, USA, 2017.
- [6] Harris S, Harris D. *Digital Design and Computer Architecture, RISC-V Edition*. Elsevier, USA, 2022.
- [7] Ditzel DR. Accelerating ML Recommendation With Over 1,000 RISC-V/Tensor Trocessors On Esperanto's ET-SoC-1 Chip. *IEEE Micro*, 42(3), 31-38, 2022.
- [8] OpenCores. WISHBONE System-on-Chip (SoC) Interconnection Architecture for Portable IP Cores. [https://cdn.opencores.org/downloads/wbspec\\_b4.pdf](https://cdn.opencores.org/downloads/wbspec_b4.pdf) (Erişim Tarihi: 25.03.2025)
- [9] Li J, Yang L, Feng X, Xing Y, Chen Z, Wan P. *UART Controller with FIFO Buffer Function Based on APB Bus. In 2022 IEEE 16th International Conference on*

*Anti-counterfeiting, Security, and Identification (ASID)*, 1-4, 2022.

- [10] Bruno F, Eschemann G. *The FPGA Programming Handbook: An essential guide to FPGA design for transforming ideas into hardware using SystemVerilog and VHDL*. 2nd Edition. Packt Publishing Ltd, 2024.
- [11] Gal-On S, Levy M. Exploring coremark a benchmark maximizing simplicity and efficacy. The Embedded Microprocessor Benchmark Consortium, 2012.