RISC-V Mimarisi Yazılım Ekosisteminin Değerlendirilmesi

Serdar Oğuz ATA, Yusuf İbrahim ÖZKÖK

oguzata@aselsan.com.tr, yozkok@aselsan.com.tr

İŞLEMCİ TASARIMI ÇALIŞTAYI '19

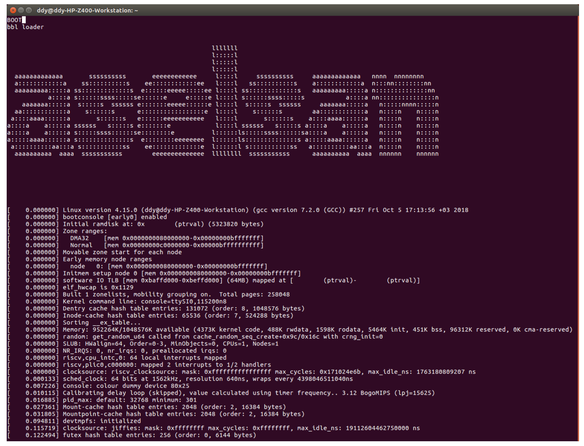
aselsan

GİRİŞ

Yazılım dünyasında yaygın olan açık kaynaklı projeler, donanım dünyasında da yaygınlaşmaya başlamıştır. RISC-V mimarisi de, Berkeley Üniversitesi tarafından 2010 yılında başlatılan açık kaynaklı bir işlemci mimarisi geliştirme projesidir. Proje 2015 yılında kurulan RISC-V vakfı ile dünyaya açılmış ve Google, NVIDIA, Qualcomm, Western Digital, Samsung, Alibaba gibi büyük şirketlerin de katılımıyla büyümüştür.

Şekil 1- RISC-V Vakfı Üyeleri

RISC-V, komut seti mimarisi (ISA) olarak belli bir olgunluğa erişmiş ve bunu destekleyen ekosistem de hızla gelişmektedir. Ürün tasarımlarında bir mimarinin kullanımı yazılım tarafında gelişmiş bir ekosisteme sahip olmasına bağlıdır. Bu makalede RISC-V mimarisi üzerinde geliştirme yapılabilmesi için gerekli yazılım araçlarının yeterliliği bir gömülü yazılım geliştirici gözüyle incelenmiştir. Tüm bu araçlar kullanılarak RISC-V üzerinde çalışan bir Linux imajı oluşturulmuştur



Şekil 2- RISC-V Linux Açılış Kaydı

RISC-V YAZILIM EKOSİSTEMİ

1- Derleyiciler

a. Clang: LLVM projesinde geliştirilen bir C/C++ derleyicisidir. Bu derleyici sadece assembly kodunu oluşturmaktadır ve çıktısının farklı bir derleyici ile makine diline çevrilmesi gerekmektedir.

b. GCC: GCC (GNU Compiler Collection), en yaygın kullanılan ve çok geniş mimari desteği olan bir derleyicidir. GCC, RISC-V mimarisini de aktif olarak desteklemektedir.

2- İşletim Sistemi Desteği

RISC-V mimarisi, Linux işletim sistemi tarafından doğrudan desteklenmektedir. RISC-V'e özel ayarlar, örnek kartların sürücüleri, linux kaynak kodunda ana dalda mevcuttur.

3- Önyükleme Yazılımı

Önyükleme yazılımları mimariye ve donanıma özel olarak geliştirilir. BooM (Berkeley out-of-order Machine) isimli RISC-V mimarili işlemci için BBL (Berkeley Boot Loader) örnek olarak mevcuttur.

4- Hata Ayıklama

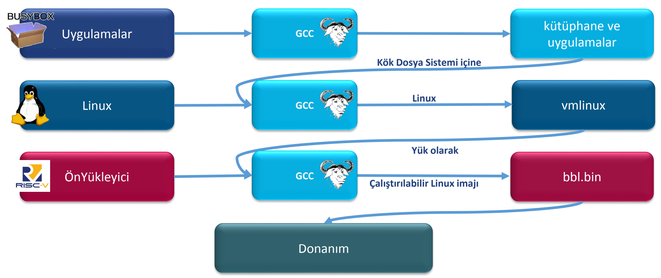
GNU'nun hata ayıklayıcısı olan GDB (GNU Debugger) RISC-V mimarisini desteklemektedir. GDB ile işlemcinin gerektiğinde durdurulup programdaki hataların tespiti mümkündür.

5- Simülatör/Emülatörler

RISC-V mimarili işlemciler ve donanımlar henüz yaygın olmadığı için, donanımı taklit eden araçlara erişim önemlidir.

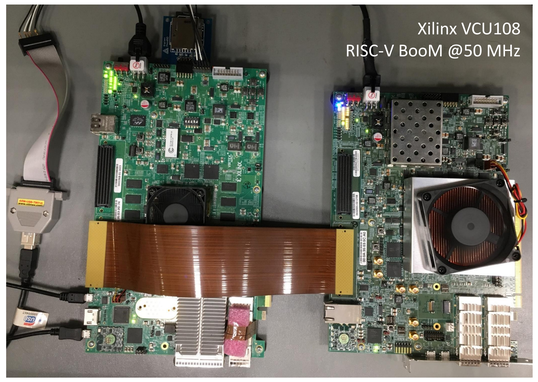
a. Spike: RISC-V komut seti simülatörüdür. Spike, RISC-V komut seti mimarisi için "altın referans" olarak görülmektedir. Bu simülatör, işlemcinin davranışını komut komut taklit ederek gerçeğe uygun analiz imkanı sağlar.

b. QEMU: Donanım davranışını fonksiyonel olarak taklit eden açık kaynaklı bir işlemci emülatörüdür. QEMU, spike gibi komut komut akışı taklit etmez. Bu yüzden spike'tan daha hızlı çalışır ancak komut seviyesinde analiz imkanı sağlamaz.



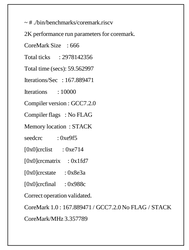
Şekil 3- İşletim Sistemi Oluşturma Adımları

Şekil 4- Hata Ayıklama Ortamı



Şekil 5- FPGA Tabanlı Geliştirme Ortamı

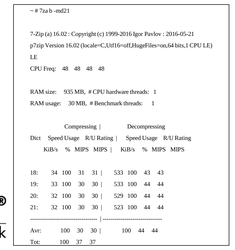
Uygulama Yazılımları - Başarım Değerlendirme Araçları (Benchmark)



Benchmark Uygulamaları

- CoreMark
- Dhrystone
- Whetstone
- 7Zip Benchmark
- NBench

CoreMark®
An EEMBC Benchmark



Şekil 6- CoreMark Sonuçları

Şekil 7- 7zip Sonuçları

Bu çalışma kapsamında oluşturulan işletim sistemi imajı üzerinde başarım ölçümleri alınmıştır. İşlemci pipeline ve ISA performansı değerlendirmesinde en çok kullanılan CoreMark sonuçları (Şekil-6) ve bellek erişim başarımı ölçümüne daha uygun bir benchmark uygulaması olan 7Zip algoritma başarım değerleri (Şekil-7) verilmiştir. Çalışmanın amacı işlemci başarımı değerlendirilmesi olmadığı için sonuçların farklı işlemcilerle karşılaştırmaları verilmemiştir.

SONUÇ

Bu çalışmanın önemi; sadece açık kaynaklı ve serbestçe erişilebilen araç ve kodlar kullanılarak yine aynı şekilde kaynak koddan sentezlenmiş RISC-V mimarili işlemci üzerinde gömülü yazılım geliştirme sürecinin tüm adımlarının uygulanmış olmasıdır.

Sonuç olarak çalışmamızda en temel yazılım ekosistemi bileşenlerinin, RISC-V için açık kaynak olarak mevcut ve erişilebilir olduğu ortaya konulmuştur. Böylelikle RISC-V açık kaynak işlemci mimarisinin herkes tarafından rahatlıkla kullanılabileceği herhangi bir ticari ürüne bağlı kalınmadan bir işlemci ekosistemine sahip olunabileceği gösterilmiştir. Aynı zamanda yapılacak katkılar ile de bu ekosistemin gelişimine destek olunabileceği değerlendirilmektedir.