**Radyasyona Dayanıklı Hibrit Hesaplama Mimarileri**

**Giriş**

Uzay görev bilgisayarlarının tasarımında esneklik, düşük güç tüketimi ve radyasyona dayanıklılık en önemli hedefler arasındadır. NASA Goddard Space Flight Center tarafından geliştirilen SpaceCube mimarisi, FPGA tabanlı yapısıyla yüksek hesaplama gücü ve yeniden programlanabilirlik özelliklerini tek bir platformda sunar. Bu sayede yapay zekâ destekli veri işleme gibi yoğun görevler, küçük uydular üzerinde dahi gerçekleştirilebilmektedir.

Önceki nesillerde kullanılan Xilinx Kintex UltraScale FPGA’ler, yüksek performans sunsa da SRAM tabanlı bellek yapısı nedeniyle radyasyona karşı hassastır ve ek bir izleme FPGA’sına ihtiyaç duymaktadır. Bu durum sistemin karmaşıklığını ve güç tüketimini artırmaktadır.

Yeni nesil Microchip PolarFire SoC FPGA (MPFS250TS-1FCG1152I) ise flash tabanlı konfigürasyon belleği ve entegre RISC-V işlemci alt sistemi sayesinde aynı işlevleri tek FPGA üzerinde güvenli ve verimli şekilde sağlar. Böylece SpaceCube mimarisi daha sade, düşük güç tüketimli ve dayanıklı bir hale gelir.

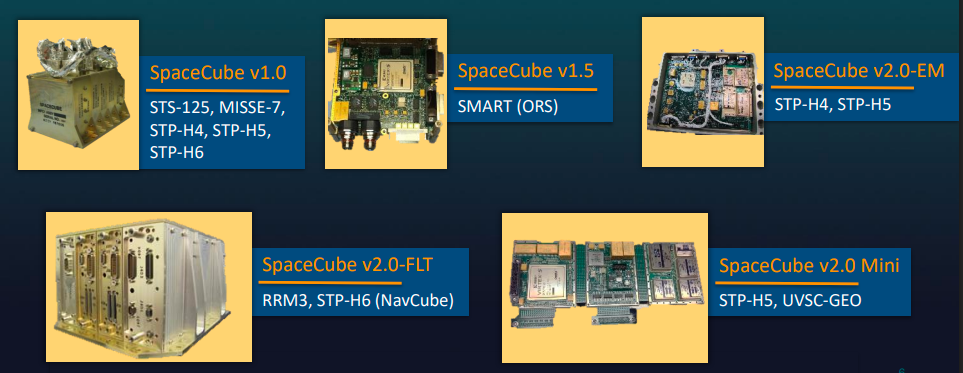
Donanım altyapısının yanında, NOS3 (NASA Operational Simulator for Small Satellites), cFS (core Flight System) ve COSMOS gibi açık kaynaklı araçlar, SpaceCube yazılımlarının geliştirilmesi ve yer testlerinde kullanılmasını mümkün kılar. Bu simülasyon ve test ekosistemi sayesinde uçuş yazılımları, gerçek donanım hazır olmadan da doğrulanabilmekte ve görev öncesinde tümleşik şekilde test edilmektedir.

**SpaceCube Teknolojisi**

**Anahtar Kelimeler: FPGA, SpaceCube, radyasyona dayanıklı, hibrit işlem, uçuş sırasında veri işleme, yeniden programlanabilir sistem, System-on-Chip (SoC), CubeSat, modüler dizayn, hata toleranslı mimari, Xilinx Kintex UltraScale, Microchip PolarFire SoC (MPFS250TS-1FCG1152I), DDR3, DDR4, LPDDR4, NAND Flash, SEU (Single Event Upset), mantık hücreleri, Flip-Flop, BRAM, DSP Slice, Transceiver, SERDES, NOS3, core Flight System (cFS), COSMOS, 42 Dynamics Simulator, SpaceVPX, SpaceVPXLite (VITA 78.1), Control Plane, Data Plane, Utility Plane, RISC-V işlemci alt sistemi, TMR, ECC, Scrubbing, Radiation-hardened FPGA (RT ProASIC3, RTG4)**

Anahtar kelimeler: FPGA, SpaceCube, uçuş sırasında işleme, radrasyon dayanıklı, yeniden programlanabilir sistem.

SpaceCube, FPGA tabanlı, çok çekirdekli işlemcilerle (CPU,GPU,DSP) hibrit veri işleme gerçekleştirebilen, System-on-Chip (SoC) mimarisine sahip, hafif ve kompakt yapısıyla ileri düzey bir uzay teknolojisidir. Yüksek esnekliği ve yeniden yapılandırılabilir yapısıyla hem yörünge görevlerinde hem de yer testlerinde kullanılmaktadır. Geliştirmeye açık modüler mimarisi sayesinde üniversitelerle yapılacak araştırma projeleri, lisans ve lisansüstü tez çalışmaları için ideal bir platform sunar. Bu yönüyle SpaceCube, akademik camiada hem mühendislik hem de uzay teknolojileri alanında iş birliğine dayalı çalışmalar için önemli bir Ar-Ge altyapısı oluşturmaktadır.

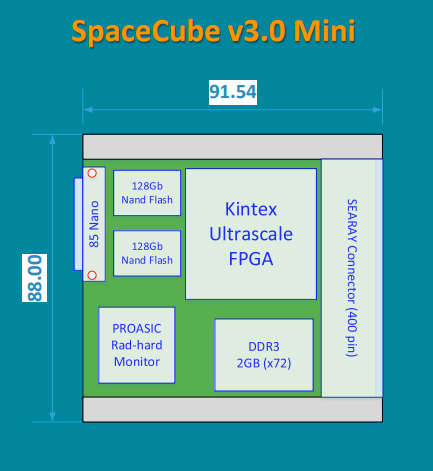


* 1. **SpaceCube Tarihçesi**

SpaceCube projesi, NASA Goddard Space Flight Center bünyesinde, 2006 yılında Internal Research and Development fonu desteğiyle başlatıldı. Amaç, uzay görevleri için yüksek hızlı, düşük maliyetli, düşük güç tüketimi içeren ve yeniden yapılandırılabilir bir veri işleme sistemi geliştirmekti. İlk sürüm olan SpaceCube V1.0 2006-2009 senelerinde Xilinx FPGA tabanlı bir system oplarak geliştirildi. İlk olarak Hubble Uzay Teleskobunda Servicing Mission 4 görevinde, teleskobun göreli konum belirleme ve navigasyon işlemlerini sağlayan RNS( relative navigation sensors) yükünün ana aviyonik işlemcisi olarak kullanıldı. Ardından ISS üzerinde radyasyon etkilerini incelemek üzere deneysel bir yük olarak test edildi. SpaceCube v1.5( geçiş sürümü) donanımsal ve mimari bazı iyileştirmeler içeren bu ara sürüm, yeni nesi sistemin testlerini mümkün kıldı. SpaceCube V2.0, 2014-2015 yıllarında geliştirilen, modüler, yeniden yapılandırılabilir ve çok çekirdekli hibrit işlem mimarisine sahipti. Xilinx Virtex-5 FPGA’LAR,yüksek kapasiteli DDR, NAND, PROM bellek modülleri ve analog izleme sistemleri barındırır. Uçuş sırasında FPGA yeniden yapılandırma ve otonom hata düzeltme(scrubbing) gibi ileri özellikler içerir. SpaceCube V2.0 uygulama alanları, GRSSli Lidar (3D görüntüleme sistemi) sisteminin tüm ver işleme, sensor kontrolü ve gömülü sisttem yönetimi işlemleri iki Virtec -5 FPGA’ya sahip SpaceCube v2.0 tarafından gerçekleştirildi. İkinci olarak Navigator GPS(Çift Bant GNSS Alıcısı) SpaceCube, GPS sinyallerinin alınması, işlenmesi ve konum-hız-zaman hesaplarının yapılmasında merkezi rol oynadı. Orjinal sistemde 4 FPGA artı bir işlemci vardı, SpaceCube v2.0 ile hepsi tek bir Virtex-5 FPGA üzerine entegre edildi. Son olarak da RRM3 robotik görevinde yer aldı. RRM3’ün command and data handling bilgisayarı olarak görev yaptı. Görev boyunca SpaceCube, komut alma dağırma, telemetri verilerini toplama ve gönderme, veri depolama ve aktarım ve analog sensörlerden veri toplama gibi işlevleri yürüttü.

SpaceCube v3.0 mini

SpaceCubev3.0 mini, Nasa tarafından özellikle CubeSat için geliştirilen, yüksek performanslı ve esnek bir uzay bilgisayarıdır. Bu sistemin en büyük farkı, hem çok güçlü olması hem de uçuş sırasında yeniden programlanabilir olmasıdır. Bu projenin amacı yapay zeka gibi işlem gücü isteyen görevleri bile küçük uydularda gerçekleştirebilecek bir altyapı sunmaktır. Tercih edilmesinin temel sebepleri; Hibrit işlem gücüne sahip olmasıdır. FPGA, CPU, GPU ve DSP bir arada bulunur ve çalışır, uçuş sırasında bile görevler değiştirilebilir, radrasyona toleranslıdır uzay ortamında güvenle çalışır, küçük boyutlu olduğunda küp uydularda uyumludur ve son olarak modüler olduğundan dolayı yeni kartlar kolayca eklenebilir.



* 1. SpaceCube Mini Dizaynı

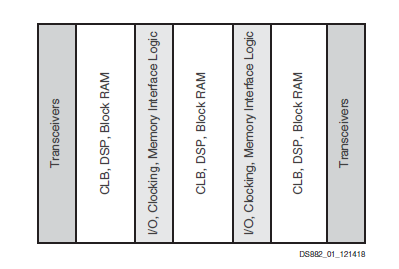
Görsle 1.2 de yer alan SpaceCube v3.0 ın dizaynıdır. Bu kart, üzerinde bir adet Kintex Ultrascale FPGA bulunur, 2 adet 128 gigabit Nand Flash, 1 adet DDR3, 1 adet radrasyon dayanıklı PROASIC monitor, 1 adet dış sistemlerle bağlantıyı sağlayan 400 pinli, yüksek yoğunluklu veri-yönetim SEARAY könektör içerir.

Kintex UltraScale FPGA bu sistemin beynidir. Xilinx’in XQRKU060 radrasyona töleranslı modelidir ve yüksek performanslı, yeniden yapılandırılabilir bir FPGA’dir. Bu FPGA üzerinden gerçek zamanlı veri işleme, sinyal-görüntü işleme, protocol yönetimi ve gömülü işlemciler çalıştırılır. Kintex UltraScale FPGA’in , 726,000 mantık hücresi, 38 Mb BRAM, 2,760 DSP slice ve 32 MGT(Multi-Gigabit Transceivers) (12.5 Gbps’a kadar) kapasitesi vardır.

SpaceCube mimarisinin içerisinde Soft Core Processor’lar yer alır.Bu CPU’lar control ve orkestrasyon beynidir. Ağır hesaplamayı FPGA’daki donanımsal hızlandırıcılar yapar. CPU bunları hazırlar, yönetir, haberleşmeyi yönetir, sistemin işletimsel sağlığını takip eder ve görev akışını yönetir. Bu sistemin uçuş bilgisayarının beyni bu cpudur. Kısa Bu yapıdaki işlemci birimleri, FPGA’in programlanabilir mantığında sentezlenen cpu türleridir. Bu CPU türlerinin uzay ortamında tercih edilmesinin sebepleri ise birinci olarak rad-hard (silicon tabanlı) CPU’lar genelde eski teknolojide kalır. Radraysona dayanıklı FPGA artı soft core işlemciden daha yüksek işlem gücü alınır. Ve radrasyon kaynaklı oluşabilecek hataları düzeltmek için TMR/ECC/Scrubbing ile mimari düzeyde hatalar tölere edilir. İkinci olarak tek bir rad hard CPU ailesine kilitlenmek yerine farklı FPGA’larda aynı mimariyi yeniden sentezleyip kullanabiliriz.

Mantık Hücreleri FPGA’İn temel yapı taşıdır. Her bir hücrede, içinde bir veya birden fazla Look-up Table, Flip-Flop ve çeşitli yönlendirme devreleri barındırır. Dijital devrrelerin tamamı bu hücrelerde oluşturulur. Flip-Flop bileşeni mantık kapıları yardımıyla oluşturulan bellek elemanlarıdır. Sinyal saklama ve zamanlama kontrolü sağlar. Her Flip-Flop bir bitlik veriyi, gelen saat sinyali ile senkronize biçimde saklar. Girişlerine uygulanan sinyal değişmediği sürece çıkış dumunu korurlar. Bu, özellikle senkron sayıcılar, finite state machine konrol devreleri gibi zamanlama hassasiyeti olan yapılar için çok önemlidir. Ayrıca her mantık hücresinde, sinyallerin yönlendirilmesi ve seçimi için MUX gibi seçim devreleri bulunur. Bu bileşenler, farklı LUT’ların veya Flip-Flopların çıktıları arasında seçim yaparak devrenin esnekliğini arttırır.

Block RAM (BRAM), FPGA içerisinde yer alan, yüksek hızlı ve doğrudan erişilebilir bir bellek mimarisidir. BRAM, özellikle geçici veri saklama, buffer, FIFO yapıları ve önbellek işlemleri için kullanılır. **XQRKU060 FPGA'**de toplam **38 Megabit (Mb)** BRAM kapasitesi vardır, bu da yaklaşık **4.75 Megabyte (MB)** anlamına gelir. Bu kapasite ayrı BRAM bloklarına dağıtılmıştır ve bu bloklar kullanıcı ihtiyacına göre dinamik olarak adreslenebilir. BRAM bloğu, çeşitli genişlik ve derinlik kombinasyonlarında yapılandırılabilir. Bu esneklik, hem düşük bant genişlikli control verisi hem de yüksek hızlı medya verisi için aynı altyapıyı kullanmaya olanak tanır. BRAM gömülü işlemciler için program kodu ve veri belleği olarak ya da veri işleme algoritmlarında geçici saklama alanı olarak yaygın şekilde kullanılır. BRAM, FPGA içinde dış belleğe gitmeden yüksek hızda erişim sağlayarak gerçek zamanlı sistemler için gecikmeyi düşürür ve verimliliği arttırır. BRAM sinyal işleme, iletişim protokolleri gibi zamanda kritik operasyonları kesintisiz bir şekilde çalıştırmamıza olanak tanır.



* 1. FPGA içinde kullanılan başlıca bloklar

Her FPGA tabanlı kartların temel blokları ortaktır. FPGA çipi dikey sütunlar halinde düzenlenmiştir. Her bir sütun, farklı görevlerde özelleşmiş alt bloklar içerir. Bu bloklar, haberleşmeyi, sinyal işlemeyi, kontrolü ve bellekleri gösterir.

Transceiver’lar, FPGA ile dış dünya arasında hızlı veri alışverişi sağlarlar. Uzay uygulamalarında kamera verisinin gigabit hızında aktarımı, uydudan yere veri gönderimi ve farklı kartlar arasında yüksek hızlı alışverişi destekler. GTH blokları, her biri bağmsız alıcı ve verici devrelerden oluşur. Verici tarafı, temel olarak bir parallel-to-serial dönüştürücüdür. Alıcı ise temelde bir serial-to-parallel dönüştürücüdür. Transceiver Şunları içerir; SERDES, PLL (PHASE LOCKED LOOP), Clock and Data Recovery (CDR), Equalizer/ Pre-emphasis, Protocol-specific logic, FIFO/Buffers.

Digital Signal Processing bloğu,yüksek hızlı sayısal işlem görevlerini donanım düzeyinde gerçekleştirmek üzere tasarlanmış özel bir yapıdır. Bu bloklar genellikle çarpma, toplama, akümülasyon, filtreleme ve sinyal dönüşümleri gibi işlemleri gerçekleştirmek için kullanılır. Çarpmalar ve görüntü işleme görevleri FPGA içinde yüksek hızda ve düşük gecikmeyle çalıştırılabilir. XQRKU060 DSP blokları sadece matematiksel işlem birimleri değil, aynı zamanda akıllı yönlendirme ve karar verme mekanizmalarını da donanımsal olarak destekleyen güçlü bir dijital sinyal işleme altyapısı sunar.

Configurable Logic Block (CLG), dijital devre tasarımı açısından temel yapı taşıdır. Mantık Hücreleri bu yapı içerisinde yer alır. Temel mantıksal işlemler bu yapı bloğunun içinde gerçekleşir. XQRKU060 FPGA 8 LUTs ve 16 flip-flops bulundurur.

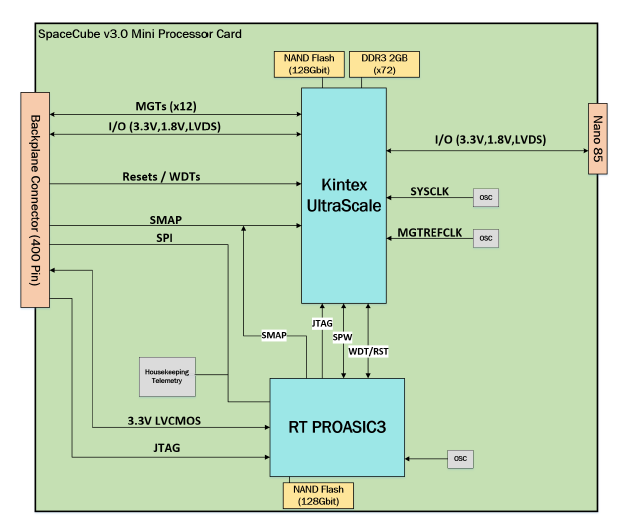
I/O Logıc, birimi FPGA’nın dış dünya ile veri alışverişini sağlayan en önemli donanım bileşenlerindendir. I/O pinleri hem çıkış hem giriş olarak kombinasyonel hem de senkron yapı olarak yapılandırılabilir. Tüm pinler DDR desteklidir, kısaca hem saat sinyalinin yükseliş anında hem de saat sinyalinin düşüş anında veri aktarımı gerçekleştirilebilir. Her bir giriş ve çıkış, programlanabilir gecikme unsurları olan input delay ve output delay bloklarıyla donatılmıştır. Bu, özellikle DDR bellek, yüksek hızlı ADC/DAC arayüzleri veya zamanlama hassasiyeti gerektiren dış bileşenlerle haberleşmede kritik önem taşır. Bu bloklar kademeli yerleştirilerek daha uzun gecikmeler elde edilebilir. FPGA içerisindeki yüksek hızlı seri haberleşme ihtiyacını desteklemek için her I/O pininde SERDES birimleri bulunur. Serdes (serial/deserializer), gelen parallel veriyi seri bir şekilde ya da seri gelen veriyi parallel bir şekilde iletebilir. Bu sayede bazı haberleşme protokolleri için transceiver kullanmadan doğrudan SELECTIO üzerinden hızlı veri alışverişi yapılabilir. Böylece, transceiver’lar gibi yüksek güç tüketen kaynaklar ayrılmadan, daha ekonomik IO blokları üzerinden yüksek hızlı bağlantılar kurulabilir.

Saat sinyali oluşturma ve dağıtımını sağlayan elemanlar, hafıza arayüzünü ve giriş/çıkış devrelerini barındıran sütunlara bitişik konulmuştur. Bu sıkı bağlı i/o ve saat oluşturma i/onun hafıza arayüzüne ve başka i/o protokollerine düşük gecikmeli saat ile veri alışverişi yapmasına olanak tanıyor. Her CMT(CLOCK MANAGEMENT TILE) bloğu içerisinde 1 tane MMCM(MIXED-MODE CLOCK MANAGER) iki tane ise PLL(PHASE-LOCKED LOOP) vardır. MMCM, geniş aralıktaki frekansları sentezleyici ve aynı zamanda gelen saat sinyallerindeki jitterı filtreleyici olarak görev yapar. Jitter, gelen saat sinyallerinin bozulması, zamanlama kararsızlığıdır. Sinyalin gelmesi gereken zamandan erken ya da geç gelmesi durumudur ki bu da zaman kararsızlığı oluşturur. PLL, MMCM’e benzerdir ama esneklik olarak daha katıdır. Phase-Locked Loop, frekans çoğaltma, senkronizasyon, saat dengeleme gibi kritik işlemleri yapan analog-dijital hibrit devredir. Atanmış hafıza arayüz devrelerine (DDR4,DDR3) gerekli saat sinyalleini sağlar. Yapısı MMCM’e benzerdir. PLL yapısı, giriş sinyalini alır bir phase detectore sokar. Bu dedektör giriş sinyaliyle pll’in ürettiği output clock’u karşılaştırarak bir error sinyali üretir. Bu üretilen error sinyali VCO (voltage-controlled oscillator) besler. Bu error sinyaline göre saat sinyalinin frekansı arttırılır veya azaltılır ta ki pll’in giriş sinyaliyle çıkış sinyallerinin fazları ve frekansları eşit olana kadar.

SpaceCube v3.0 mini dizaynında kullanılan NAND flashlar, uçucu olmayan kalıcı veri saklama üniteleridir. Elektrik kesildiğinde bile verileri korur ve yüksek yoğunlukta veri saklamak için idealdir. Bilimsel veriler, görüntü sensor çıktıları, yük görevlerine ait işlem sonuçları bun and belleklerde saklanır. Yüksek bant genişlikli görevlerde ara kayıt elemanı olarak kullanılır. FPGA için konfigürasyon dosyaları, işletim sistemi ve sistem yazlımı burada tutulut. Boot sırasında FPGA, configürasyon dosyalarını burdan alır. İki bellek üzerinde de aynı konfigürasyon yazılıdır, bunun sebebi SEU( Single Event Upset) gibi radrasyondan dolayı oluşan bit bozulmalarına karşı bir önlem almaktır. Eğer iki konfigürasyon dosyasında da bozulmalar varsa, işlemci ilk konfigürasyonu PROM’dan alarak rollback yapılır. Hata töleranslı depolama yapar. Aynı zamanda yeri veri aktarılırken tampon görevi görür. Yükleme/indirme bant genişliği sınırlı olduğu için büyük miktarda veri önce NAND flashlara yazılır, sonar zamanlamaya göre yere indirilir.

DDR3, bellek sistemin geçici veri işleme ve yüksek bant genişliğine sahip görevleri için kullandığı ana çalışma belleğidir. Her saat sinyalinde bir yükselen kenar birde düşen kenar olmak üzere iki kere veri aktarımı gerçekleştirebilir. Bu hem daha uzun bit genişliğine sahipo veri aktarmaya olanak sağlar hem de veri iletim hızını önemli ölçüde arttırmış olur. Senkron ve uçucu olan bir DRAM türüdür. İşlemci programları çalıştırırken ana RAM olarak burayı kullanır. FPGA, memory interface controller ile DDR3 ile direct bir bağlantı sağlar. DMA modülü CPU’yu meşgul etmeden veri taşıyabilir. İşlemci veri işlemek için bellek ihtiyaacını DDR3’ün üzerinde sağlar.

Searay, yüksek performanslı FPGA ile başka modüller arasında yüksek hızlı, güçlü sinyal bütünlüğü ve çoklu pin sayısı sağlayan bir konnektördür. FPGA, bu konnektör sayesinde LVDS, SERDES, SPI, UART ve I2C gibi sinyaller ile dış dünyadaki yardımcı kartlarla bağlantı kurar.

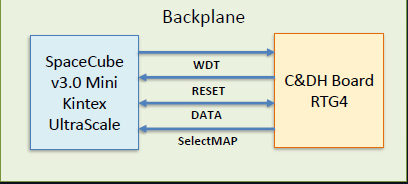


* 1. **SCv3.0 Mini Processor Card**

SpaceCubev3.0 hata-töleranslı mimari, bir sistemde oluşabilecek hatalara ragmen operasyonu devam ettirmeye yönelik oluşturulan mimari tasarımıdır.

**RT ProASIC3**, SpaceCube v3.0 Mini sisteminde güvenli açılış ve hata yönetimi için kullanılan, **radiation-hardened**, **flash-tabanlı**, düşük güç tüketen ve güvenilir bir FPGA’dır. Bu çip, **yüksek performanslı ama radyoaktif etkilere açık** olan Kintex UltraScale FPGA’yi **izler, konfigüre eder ve korur. Görev control sistemi SpaceCube’a komut gönderir. Bu komutlar ilk önce RT ProASIC3’e gelir, yorumlanır sonar Kintex’e aktarılır ya da bozuksa sistem tarafından müdahale yapılır. RT ProASIC3 selectmap protokolünü kullanarak Kintex FPGA’e konfigürasyon yükler. Bu sayede NAND flashlardan konfigürasyon dosyaları selectmap ile Kintex FPGA’e aktarılır. ProASIC3 aynı zamanda Kintex’I izler. Eğer Kintex uzun süre cevap vermezse watchdog süresi dolar sistemi resetler. RT ProASIC3 uzay aracından bir komutla konfigüre edilebilir. Uçuş sırasında konfigüre edilebilmesi, dinamik operasyonlara uygunluğunu destekler.**

**RT ProASIC3, Stand-Alone Operation modunda çalışabilir. Bu terim bu FPGA’in sistemin geri kalanından bağımsız şekilde çalışabilme yeteneğidir. Kintex FPGA de herhangi bir radrasyondan dolayı kitlenme durumunda RT ProASIC3 tek başına Kintex’I control eder reset atar ve nand flashtan bitstreami alıp yükler. Tek başına sistemden bağımsız olarak kendi başına bu işlemleri düzenleyip yapabilir.** RT ProASIC3, sistemdeki Kintex FPGA’nın **konfigürasyon bütünlüğünü korumak için** iki farklı yöntemle **scrubbing (düzenli kontrol ve düzeltme)** işlemi yapar. Diğer öbür işlem ise smart scrubbing. Bu işlem readback ile Kintex’in konfigürasyon dosyalarını okur, burada bitlerde bir hata bulursa bunu düzeltir. Hatalı bit’ler ECC ile tespit edilir. Yalnızca Kintex FPGA’I değil aynı zamanda NAND Flashlerinde konfigürasyon dosyalarını kontrol eder ve bunlar düzeltir.



* 1. **Companion Card Operation**

Companion Card Operation mimarisinde ana kontrolcü yerine, veri işleme görevini üstlenen yardımcı bir kart gibi çalışır. Sistem kontrolü RTG4’de kalır. Bu, sistemin daha kararlı çalışmasını sağlar. SpaceCube daha çok yük görevlerine odaklanır. Selectmap üzerinden uzaktan konfigürasyon yapılabilir. RTG4 tabanlı C&DH karrtı tüm sistemi resetleyebilir, konfigüre edebilir ve izleyebilir. Fault tolerant sistem mimarisiyle bu mimari arasındaki farklar. Fault tolerant mimarisinin amacı sistemin kötü koşullarda bile tek başına çalışabilmesini hedeflerken bu mimari, daha çok işlem gücüne odaklanmaktadır.

SpaceCube geliştirme sürecine başlandığında ilk başta masaüstü aplikasyonlarla başlanır. Bu aplikasyonlar operasyonal algoritmaları barındırır ayriyetten Nos3 sanal makinasıyla beraber Cfs framework kullanılarak uçuş yazılımı geliştirilir. Geliştirilern yazılımların hepsi Nos3 sanal makinasıyla beraber uydu sanki yörüngedeymiş gibi simüle edilir. Simülasyonlar başarıyla tamamlandıktan sonra geliştirme kartı adımına geçilir. Bu adımda spacecube geliştirme kartı üzerinde donanım ve yazılım denenir. Bu kart, uçuş donanımına benzer FPGA + işlemci içerir ama laboratuvar kullanımına uygundur. FPGA üzerinde hızlandırıcı modüller, CPU üzerinde cFS tabanlı flight software çalıştırılır. Burada: driver geliştirme, donanım-yazılım entegrasyonu, ilk performans ölçümleri yapılır. Geliştirme kartıyla başarılı olan tasarım uçuş modeline birebir benzer olan mühendislik modelinde geliştirilen kart fiziksel olarak yörüngedeymiş gibi test edilir(tc gönderme, tm alma , çevresel testler). Uçuş modeli ise mühendislik modelinde yapılan tesler başarılı bir şekilde gerçekleştirildikten sonra geçilen aşamadır. Bu model artık yörüngeye fırlatılmaya hazır ve nazır duruma geçirilmiştir. Son aşama ise uydu üzerindeki operasyonlardır, uçuş bilgisayarı uydunun tüm yükünü çekmeye başlar ve telemetri, telekomut alışverişi başlanır. FPGA reconfiguration veya yazılım güncellemeleri güvenli uplink protokolleri (ör. CFDP, CLTU) ile gönderilir. Uçuş sırasında fault tolerant, sağlık izleme ve otonom operasyonlar kritik hale gelir.

**Yazılım Geliştirme ve Simülasyon**

**NOS3**

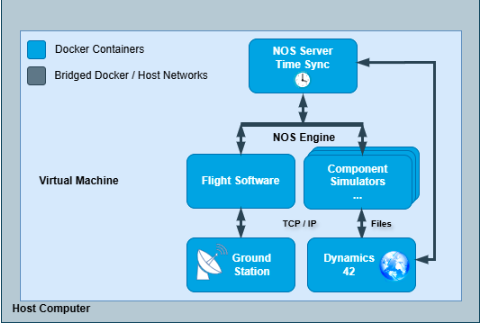
Küp uydular, düşük maliyetli ve hızlı geliştirmehedefleriyle tasarlanan, genellikşe üniversiteler, araştırma merkezleri tarafından yapılan uydulardır. Ancak bu tür görevlerde şu tür problermler karşımıza çıkar:

* Gerçek donanım pahalıdır ve geç gelir
* Geliştirme süreci parallel yürütülemez çünkü donanım hazır olmadan yazılım test edilemez.
* Geliştirme ekipleri sınırlı sayıda donanım paylaşmak zorundadır

Bu durumda NASA tarafından geliştirilen NOS3 çerçevesi, donanım henüz hazır değilken bile yazılım geliştirmeyi ve test etmeyi mümkün kılar, gerçek donanım simülasyonu sağlar, yer istasyonu yazılımı, uçuş yazılımı ve donanım simülasyonunu entegre çalıştırır ve geliştirme ekibine, uçuşa hazır yazılımı önceden doğrulama ve test etme imkanı verir.

NOS3**,** uydu projelerinde geliştirme sürecini hızlandıran, donanım bağımlığını azaltan ve yazılımın doğruluğunu erken evrede test etmeye olanak sağlayan bir simülasyon ve test altyapısıdır.özellikle CubeSat’lar için kullanılan açık kaynaklı bir uydu görev simülasyon çerçevesidir. Amaç, uzaya çıkmadan önce uydu görevlerinin yer istasyonu ile olan haberleşmesi, yazılım testi, donanım davranışları gibi birçok bileşenin yer simülasyonu ortamında entegre olarak test edilmesini sağlamaktır.

NOS3, gerçek donanım yerine yazılım tabanlı donanım benzetmeleri sunar. Geliştirilmekte olan uçuş yazılımın uyduya yüklenmeden yazılım ile test edilmesini sağlar.



* 1. NOS3 Mimari

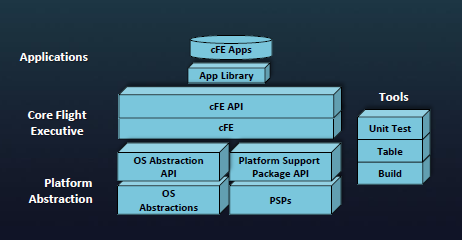
Her uydu içerisinde, aracın farklı bölümlerini temsil eden çeşitli “bileşen” (component) simülatörleri bulunur. Bu bileşen simülatörleri, bit ve byte seviyesinde donanım davranışlarını modelleyen donanım kütüphanesi (HWLIB) üzerinden arayüzler sağlayan **NOS Engine middleware** aracılığıyla iletişim kurar. **Uçuş yazılımı, aslında uzayda çalışmadığını fark etmez.**

**NOS Engine, simülasyon ortamlarında kullanılmak üzere tasarlanmış bir mesaj tabanlı arak atman yazılımıdır. Modüler yapısı sayesinde güçlü bir çekidek katmanı sunar ve bu katman I2C,SPI ve CAN Bus gibi berlirli haberleşme protokollerini simüle edecek şekilde genişletilebilir. İleri özellikler olarak zaman senkronizasyonu, veri manipülasyonu ve hata enjekte etme gibi özellikler sayesinde, simülasyon bileşenlerini bağlamak ve test etmek için hızlı, esnek ve yeniden kullanılabilir bir sistem sağlar. İki temel nesne modeline dayanır, Bu modeler; Node (düğüm) ve Bus(veriyolu) nesneleridir. Her node, sadece bağlı olduğu bus içerisindeki diğer node’larla iletişim kurabilir. Node, sistemde mesaj gönderip/alabilen her tür uç noktadır. Bus ise aynı anda birden fazla düğümün bağlı olabiileceği iletişim grubudur. Her düğüm bir veriyoluna ait olmalıdır. Her düğüm, sadece bağlı olduğu bus içerisindeki diğer nodelarla iletişim kurabilir. Farkı buslar arasında doğrudan bir iletişim yoktur. Böylece her bus bir sandbox ortamı gibi çalışır.**

Flight Software(uçuş yazılımı), uydunun görevi boyunca uyduyu opera etmek ve yöneltmekten görevlidir. NOS3 cfs kullanarak kullanıcıların baştan sona kadar bir uzay aracının görevini simüle etmesine olanak tanır. NOS3’ün uçuş yazılımı default olarak cfs kullanmaya ayarlanmıştır. Ama CFS haricinde kullanıcıların keşfetmesi için entegre edilen başka uçuş yazılımı da vardır.

Cfs, Nasa’nın çok merkezli bir konfigürasyon kontrol kurulu yönetiminde açık kaynak olarak yayımlanan, yeniden kullanılabilir yazılım hizmetleri ve uygulamarından oluşan uçuş yazılımı çerçevesidir. Temel bileşeni olan core flight executive (Cfe), uygulama çalışma zamanını ve hizmet API’lerini sağlayarak uygulamaların eş zamanlı olarak ve kararlı bir şekilde çalışmasını sağlar. Platform soyutlama katmanı (OSAL) ve platform destek paketleri (PSP) aracılığıyla, linux, RTEMS, VxWorks gibi farklı gerçek zamanlı işletim sistemleri ile SpaceCube veya CSPv1 gibi donanımlar arasında “bir kez yaz, her yerde çalıştır” yaklaşımı sunar. Katmanlı mimarisi; Yürütme, yazılım veri yolu, olay, zaman ve tablo hizmetlerini açıkça ayırarak hem bakım kolaylığı hem de işlevsel genişletirebilirlik sağlar. Cfs; endüsrti, akademi ve diğer uzay ajanslarının katkı ve desteğiyle sürekli evrilmekte, CubeSat’tan büyük görev sınıfı uydu projelerine kadar güvenilir uçuş yazılımlarının hızlı geliştirilmesini mümkün kılmaktadır.

Cfs’in üç temel katman bulunur. Bunlar yukarıda da bahsedildiği gibi OSAL (operating system abstraction layer) PSP v eek olarak HWLIB (Hardware Library) katmanlarıdır. Bahsedildiği gibi OSAL, yazılımın işletim sisteminden bağımsız yazılmasını sağlar. Aynı kodun farklı os ler üzerinde koşturulmasını sağlar. NOS3’te linux hedefi için OSAL kullanılır. Bu sayede yazılım, gerçek donanım yerine simüle edilmiş ortamda çalışabilir. PSP(Platform Support Package), donanıma özel ama işletim sistemi bağımsız fonksiyonları ( zamanlayıcı, bellek, clock vs.) içerir. NOS3 içinde PSP, linux PSP’nin modifiye edilmiş bir türüdür. Cfs’in 1HZ ana zamanlayıcısı, NOS engine tarafından sağlanan zamanlayıcı ile eşleştirilir ve bu sayede tüm sistem senkron çalışır.



* 1. **Cfs mimarisi**

HWLIB, ı2c, uart gibi donanıma özgü I/O işlemlerini soyutlar. NOS3 içerisinde bu kütüphane, simüle edilen bileşenlerin uçuş yazılımıyla haberleşmesini sağlar.

Core Flight Executive, cfs’in merkezi yürütme katmanıdır. Yani tüm uçuş yazılımı sisteminin kalbinde yer alır ve diğer uygulamalrın çalışması, mesajlaşması, zamanlanması gibi temel hizmetleri sağlar.

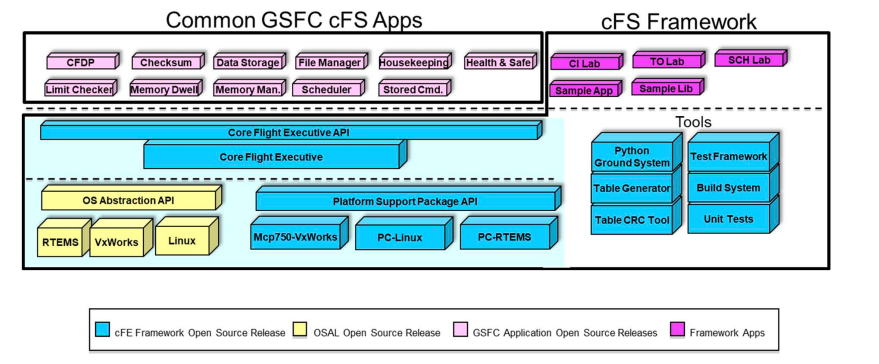
CFE, mesajlaşma hizmeti sunar. Bu hizmetle uygulamalar arasında mesaj alışverişi sağlar. Tele metri mesajları, komut mesajları bu bus üzerinden gider. Mesajı yayınla ve dinle modeli ile çalışır (SB). Görev zamanı ve sistem zamanı takibini sağlar. Uygulamalarla senkronize zaman damgası ekler. Zamanlayıcı uygulamalara zaman sağlar (TIME). Uygulamalardan gelen olay mesajlarını yönetir. Loglama, hata bildirimi, debug çıktısı sağlar (EVS). Tablolar dosya sisteminden yüklenebilir veya görev içinde güncellenebilir (TBL). Uygulamaların başlatılması, durdurulması, sıfırlanması gibi işlemleri yönetir. Sistem başlangıcında hangi uygulamaların yükleneceğini belirler. Uygulama durumlarını izler ve control eder(ES). Cfe yapısı, OSAL üzerinden işletim sistemi çağrılarıı kullanır ve PSP ile donanım erişimini soyutlar.

İlk güç verildiğinden itibaren işlemci boot zincirine girer sistem ayağa kalkar. ES Cfe çekirdeğini kurar, çekirdek servislerini ayağa kaldırır, ardından Cfs Applerini ve görev uygulamalarını ayağa kaldırır.

Yer sistemi operatöründen, yerden komut yollanır. Yollanan kod blokları uzay aracındaki haberleşme donanımı tarafından alınır ve haberleşme uygulamasına iletilir.

Bu uygulama donanıma özgü frame/protokol soyup kod bloklarını yazılım veriyoluna (SB) aktarılmak üzere command ingest (CI) uygulamasına iletir. CI bu blokları komut paketine çevirir, doğrular ve ilgili uygulamalara dağıtır.

Uygulama komutu SB üzerinden alır. Hedef uygulama, daha önceden SB üzerinde ilgili MsgID’ye abonedir. Abone olduğu pipe’tan komutu çeker. Komut koduna göre uygulama işlevini yürütür; bu sırada task, queue, senkronizasyon, zamanlayıcı gibi ihtiyaçlar varsa OSAL üzerinden çözer. OSAL tek bir api verir; altta hangi os olursa olsun çağrıları yerel os sistem çağrılarına çevirir. PSP de aynı şekilde donanım erişimi için tek bir api verir. Bu PSP gerekli boarda/donanıma göre kütüphaneleri çeker. Bu sayede uygulama kodu farklı os ve donanımlara bağımlı olmadan çalışır ve bu os ve donanım bağımlılıkları OSAL ve PSP katmanlarında kalır. Hedef uygulama komutu alıp iş mantığını çalıştırdıktan sonar ihtiyaca göre TBL servisi üzerinden konfigürasyon veya parametreleri alır. Zaman gereksinimi varsa TIME servisi üzerinden komutları aldığında, uygulama başladığında zaman damgalarını oluşturur. Duruma göre EVS olay üretir ürettiği olaylar uygulama başarıyla başladıysa başarı, uyarı, hata gibi mesajlar üretir, bu mesajlar loglanır ve sb üzerinden yayılır. Uygulama telemetriyi hazırlar, time damgasını basar ve sb’ye gönderir. TO, SB’den telemetrileri alır ve aşağıya yer istasyonuna yollar.



* 1. **Cfs organizasyon şeması**

Cfs, görev kontrolünü sağlayan uygulamalar içerir. Bu uygulamalar, telemetri, dosya yönetimi, veri kaydı ve yönelim kontrolü gibi uygulamalardır. Görev özel uygulamalar görsel 1.8 de görüldüğü gibi üst aplikasyon kısmında geliştirilir. Bu uygulamalar:

* **Cf**, uzay araçlarıyla dosya transferi yapmak için kullanılır. Örneğin, veri logları veya görev verilerini yer istasyonuna göndermek. CCDS File Delivery Protocol ve UDP kullanır.
* **Ci** ise yer istasyonundan gelen komutları almak ve yönlendirmekle görevlidir. Komutun parse edilmesi ve ilgili uygulamaya iletilmesi buradan yapılır. CCSDS ve UDP kullanır.
* **Ci lab**, test/simülasyon amaçlı CI uygulamasının laboratuvar sürümüdür. Yerel ortamda kolay test için kullanılır.
* **Ds**, gelen telemetri verilerini dosyalara kaydeder. Örneğin GPS, IMU verilerini loglar. Loglama politikaları DS filter table ile yapılandırılır.
* **Fm**, uydunun dosya sisteminde dosya kopyalama, slime, taşıma gibi işlemleri sağlar. Yerden komutla tetiklenebilir.
* **Ic**, gelen telemetri verilerini izler, limit dışı durumları otomatik olarak tespit eder ve uyarı üretir.
* **Sc**, zamanlanmış komut dizileri tanımlanmasını sağlar. 10 saniye sonar veri kaydını başlat komutu gönderir.
* **Sch**, belirli zamanlarda belirli uygulamaları periyodik olarak çalıştırır. Her saniyede bir sensor beri oku örnek olarak verilebilir.
* **To**, uçuş bilgisayarından çıkan telemetri verilerini yer istasyonuna gönderir.
* **To\_lab**, test ortamı için özel olarak yapılandırılmış TO uygulamasıdır. Gerçek TO’nun yerine laboratuvar testlerinde kullanılır.

**CCSDS**, uluslararası uzay ajanslarının oluşturduğu bir standart geliştirme organizasyonudurç Uzay görevlerinde veri iletimi, depolama ve işleme süreçleri için ortak standartlar geliştirmektedir. Bu standartlar sayesinde farklı ülkelerin uyduları ve yer istasyonları uyumlu çalışabilir, veriler aynı formatla gönderilip alınabilir ve görevlerin iletişim alt yapısı daha güvenli olur. CCSDS standartları, paketleme ve protokoller (CCSDS Space Packet Protocol, telemetri, telekomut), dosya transferi (CCSDS File Delivery Protocol) , hata düzeltme kodları (Reed-Solomon, LDPC) ve veri arşivleme alanlarını kapsar.

Nos3, gerçek uydularda kullanılan veri protokollerini sümule eden bir test ortamı olduğu için ve CCSDS, nasa ve diğer uzay ajansları uydu-yer istasyonu iletişimi için zaten kullanılan bir standart olduğundan dolayı Nos3 için tercih edilir.

CCDS kullanmak yer istasyonu ve uydu arasındaki veri akışını gerçeğe en yakın şekilde simüle etmeye yarar çünkü CCSDS kullanıldığı zaman uzay görevlerindeki aynı iletişim foırmatını kullanmış oluyoruz. Farklı alt sistemler, görev yazılımı ve yer control yazılımları aynı veri formatını kullanır. CCSDS paket yapısı sayesinde farklı tipte veriler ( telemetri, telekomut, dosya transferi) tek bir çatı altında yönetilir.

CCSDS kapsamında tanımlanan bir önemli protokol ise CFDP (CCSDS File Delivery Protocol) dur. CFDP, büyük boyutlu verilerin dosya formatında güvenilir ve hızlı şekilde transfer edilmesini sağlar. İki model ile çalışır : Class 1 modu UDP (bağlantısız ve hızlı bir iletişim protokolü) benzeri çalışır; hızlıdır ancak veri kaybı yapabilir. Class2 modu ise TCP ( internet protokolleri arasında yer alan, bağlantılı ve güvenilir bir iletişim protokolü) benzeri çalışır; ACK/NAK mekanizması ile eksik parçalar algılanır ve yeniden istenir. Bu şekilde güvenli bir transfer sağlanır. Cfs in tüm ana uygulamaları bu protokolü kullanır.

**Cfs**,yer istasyonundan gelen komutları işleyip telemetri verilerini üretecek şekilde yapılandırıldıktan sonra, bu verilerin yer istasyonu yazılımına aktarılması gerekir. NOS3 ortamında yer istasyonu görevi için COSMOS kullanılmaktadır.

**COSMOS**, (**Command and Telemetry System**), Ball Aerospace tarafından geliştirilmiş, açık kaynaklı bir **yer istasyonu yazılımı**dır.  
Uydu, uzay aracı veya herhangi bir gömülü sistemle **uzaktan komut gönderme** v**e telemetri verisi alma** görevlerini yerine getirir.  
Amaç, **uçuş yazılımı ile yer kontrol merkezi arasında güvenilir ve kolay yönetilebilir bir arayüz** sağlamaktır. NOS3 ortamında ise COSMOS yer istasyonu simülasyonu görevini üstlenir.

**42 Dynamics Simulator**, NASA tarafından açık kaynak olarak sunulan bir dinamik ortam benzetim (simülasyon) framework’üdür. **42**, uydu uzay koşullarını ve ekipman modellerini içermektedir. NASA tarafından geliştirilen bu araç, uzay modelleri kapsamında gök cisimlerinin yörünge hareketlerini, güneş ışınımını, yerçekimi gradyanını, manyetik alan gibi tüm uzay koşullarını gerçek zamanlı olarak modelleyerek çıktılar üretir.

Bu çıktılar, uydu uçuş bilgisayarında çalışan yazılımların **cFS (Core Flight System)** aracılığıyla, gerçek donanım arayüzleri benzetilerek kullanılabilir hâle getirilir. Böylece geliştirilen uçuş yazılımlarının uzay koşullarında nasıl bir performans göstereceği, gerçek zamanlı ve matematiksel doğrulukla değerlendirilebilmektedir. Üretilen veriler üzerinden telemetri, sistem tepkileri ve yazılım algoritmalarının davranışları test edilebilir.

Bu benzetim esnasında, **cFS** ile olan entegrasyonu sayesinde, gerçek donanıma yüklenecek yazılımların testleri aynı ortamda doğrudan ve güvenilir bir şekilde gerçekleştirilebilmektedir. **42** simülasyon ortamına, görev ihtiyaçlarına uygun olarak yeni uzay koşulları ve ekipman modelleri içeren benzetim kütüphaneleri eklenebilir.

**SpaceCube ve Diğer Modüllerin Entegre Edilmesi**

**SpaceCube** ve başka yardımcı modüller (güç ve faydalı yük kartları) entegre edilerek bir CubeSat oluşturulur. Bu CubeSat’ın modüllerinin birbiriyle entegrasyonu için geliştirilmiş bir standart mevcuttur. Bu standart SpaceVPX, endüstriyel ve savunmada kullan VPX backplane standartının uzaya uyumlanmış halidir. Bu kartlar bir araya getirilerek alt sistemler daha sonra da tam sistemi oluşturur. Bu standart, ekstreme koşullarda 3u ile 6u arasındaki formlara sahip sistemler sağlamlaştırılmış ve conduction-cooled özellikleri sağlar. Ayrıca OpenVPX altyapısı, SpaceVPX omurgasını karada prototipleme ve test yapma imkanı sunuyor. Peki OpenVPX nedir?

OpenVPX, VPX ile sistem mimarilerinin birbirleriyle implementasyonudur. VPX sadece mekanik faktör ve temel konnektörleri tanımlar. OpenVPX ise VPX üstüne bir katman getirir. Hangi kart hangi slottan olacak, hangi pin hangi protokol için kullanaılacak ve kartlar arası veri akışı hangi plane üzerinden olacak, hangi pin kangi protokol için kullanılacak ve kartlar arası veri akışınahangi plane üzerinden olacağı belirtilir. OpenVPX, bağlantıları 4 ana katman olarak sınıflandırır:

1. DATA PLANE: Data plane çok yüksek hızlı bağlantılar için olan katmandır. Bu katman, modüller arasında görev ve faydalı yük verisi taşınır.
2. CONTROL PLANE: Control plane, nispeten daha az kapasiteye sahiptir. Tipik olarak konfigürasyon, kurulum, tanı, payload için operasyonel kontrol fonksiyonları ve düşük hız veri aktarımı için kullanılır.
3. UTILITY PLANE: Utility plane, power sequence, low-level diagnostics, clocks ve öbür sistem operasyonları için gerekli olan bazı sinyalleri ve basit modül fonksiyonlarını sağlar.
4. EXPENSION PLANE: Expension plane, modüller arası benzer arayüzlü ama farklı bağlantılar yapmasını sağlayan katmandır. Bu katmanda pinler belirli değildir, kullanıcıya göre değiştirilebilir bir yapıya sahiptir.

metin, ekran görüntüsü, yazı tipi, tasarım içeren bir resim

Yapay zeka tarafından oluşturulmuş içerik yanlış olabilir.

* 1. **SpaceVPX BACKPLANE MİMARİSİ**

SpaceVPX, slot profiline ve modül profiline göre plug-in kartlar oluşturan bir standarttır. Böylece farklı üreticilerden gelen kartlar aynı sistemde uyumlu çalışabilir. Bant genişliği kısıtını ortadan kaldırmak, modüler ve fault-tolerant bir uçuş bilgisayarı omurgası sunmak hedeflenmiştir. Klasik VPX’in VMEbus control-plane yerine SpaceWire kullanıyor, ancak kullanıcı tanımlı seri veri düzlemi (data plane) bağlantısının esnekliğini korumuştur. SpaceVPX çok arayüzlü bir bağlantı sağlamaktansa çok çeşitli uygulamaları karşılamaya ve bu uygulamaları destekleyecek bir dizi backplane profili tanımlamaya odaklanır. Bu yaklaşım, SpaceVPX standardını çok karmaşık hale getirir çünkü çok sayıda seçenek içerir ancak aynı zamanda çok esnek ve kullanışlı kılar. SpaceVPXLite(VITA 78.1), SpaceVPX’in boyutunu ve karmaşıklığını azaltmayı amaçlar. Bu standard 3U boyutlu kartlara odaklanır, SpaceVPX’teki olası backplane yapılandırmalarını sınırlandırır ve düzenler, ayrıca utility, control ve data plane desteğine yoğunlaşır. VITA 78.1 standartı şu anda taslak halindedir.

SpaceVPXLite omurgasına ya da rafa takılan dört tip modül vardır:

1. Sistem kontrolcüsü modülü rafa takılan öbür modül/kartlarının operasyonlarını control eder. Kontrolcü modülünün iki tipi vardır, nominal ve yedek tip. Bunlar faydalı yük modüllerine radial veri bağlantısıyla bağlanır ve güç anahtarlama ve güç kaynağı modüllerine I2C ya da öbür utility sinyallerini kullanarak bağlantı sağlar. Radyal teriminin kullanılmasının sebebi point -to-point bağlantılar, system kontrolcüsünden faydalı yük modülüne dışa doğru ışınsal (yayılır gibi) bir şekilde dağılmasını ifade eder . Radyal bağlantı geleneksel bus bağlantısı gibi tek bir ortak veri hattı olmadan, her faydalı yük modülünün kendine ayrı hattıyla system kontrolcüsüne bağlanmaısnı ifade eder.
2. Faydalı yük modülü, ana işleme veya veri işleme fonksiyonların yerine getirir. Maksimum rafta 6 tane faydalı yük modülü olabilir. Faydalı yük modüllerinin yedekliliği uygulama gereksinimlerine göre ayarlanabilir. Örnek olarak, altı tane faydalı yük modülü varsa ve benzer fonksiyonları yerine getiriyolarsa bunlardan beşi operasyonda kalır biri yedek ünit olarak ayarlanır.
3. Güç kaynağı modülü, raflardaki öbür modüllere güç dağıtımının yapılması için güç anahtarlama modülüne güç sağlar. İki tip güç kaynağı modülü vardır. Biri nominal öbürü yedek. Giriş voltajı genel olarak 24 Volt DC’dir.
4. Güç dağıtım modülü, nominal veya yedek güç kaynağından alınan gücü rafta olan öbür modüllere dağıtmakla görevlidir. Sistem kontrolcülerine sağlanan gücün değiştirilmesi bazı dışsal sinyallere ve aktif system kontrolcüsü tarafından aktif görev gerekliliklerine göre seçilen faydalı yük modüllerine göre belirlenir. Güç dağıtım modülü, control devresinin yedeğini barındırır, bu devre hangi system kontrolcüsünün aktif oluğ olmadığını görebilmek için kullanılıyor.

SpaceVPXLite omurga haberleşmesini üç ana katmanda veya işleve bölüyor. Bu katmanlar :

1. Kontrol katmanı, system kontrolcüleri ve faydalı yük modülleri arasında konfigürasyon, control ve izleme bilgileri aktarımında kullanılıyor. İşlenmek için system kontrolcüsü ve faydalı yük arasında veri aktarımı mümkündür . Kontrol düzlemi genellikle, her sistem kontrolcüsünden her payload modülüne giden bağımsız SpaceWire bağlantıları ile uygulanır. Her system kontrolcüsünden altı bağlantı çıkar, faydalı yükte ise nominal ve yedek olmak üzere iki tane çıkar.

metin, ekran görüntüsü, çizgi, sayı, numara içeren bir resim

Yapay zeka tarafından oluşturulmuş içerik yanlış olabilir.

* 1. SpaceVPXLite Omurga Bağlantıları

1. Veri düzlemi, her faydalı yük modülü arasında bilgi aktarımı için kullanılır. Verigenellikle bir sensörden, bir payload kartı üzerindeki bir veya daha fazla **ön panel konektörü** aracılığıylagelirveihtiyaçlaragöre **işleme, sıkıştırma, depolama** vb. için payload kartları arasında **omurga üzerinden dağıtılır. Veri düzlemi (data plane),** örneğin **Serial RapidIO** gibi birkaç farklı **multi-Gbit/s seri teknolojisini** destekler.Payload modülleri arasındaki bağlantılar için hem **anahtarlama (switch)** hem de **nokta-nokta (point-to-point)** bağlantılar kullanılabilir.
2. Utility düzlemi, SpaceLiteVPX rafındaki çeşitli modüllere güç, saat sinyali ve control/durum sinyallerini iletmek için kullanılır.

Özet olarak, biri nominal diğeri yedek üzere güç anahtarlamasına sahip iki tane güç kaynağı vardır, biri nominal diğeri yedek iki tane sistem kontrolcüsü kartı vardır ve maksimum 6 tane faydalı yük kartı eklenebilir. Bu faydalı yük kartları, işlemciler, yığın bellek modülü vb. modüllerdir. Güç anahtarları, gücü aktif olan kartlara dağıtır. Kontroli veri ve sistem yönetim katmanları faydalı yükleri ve kontrolcülleri birbirine bağlar. Saat sinyali dağılımı gibi utility fonksiyonları, utility sinyalleir sayesinde taşınır.

Kontrol düzlemi, faydalı yük modüllerinin aktif sistem kontrolcüsü tarafından konfigürasyon, kontrol ve izlenmesinden sorumludur. Veri düzlemi ise faydalı yük modülleri arasında yüksek bant genişlikli haberleşme sağlar.

Uzay görev bilgisayarlarında SWaP (Size, Weight and Power) kısıtları, özellikle CubeSat ve SmallSat sınıfındaki küçük uydular için kritik bir tasarım parametresidir. Klasik SpaceVPX standardı, büyük ve karmaşık görev bilgisayarları için geliştirilmiş olup yüksek sayıda modül, fazla güç tüketimi ve geniş kart alanı gerektirir. Bu yapı, küçük uydular için çoğu zaman gereksiz derecede ağır ve kaynak tüketici olabilmektedir.

Bu noktada geliştirilen SpaceVPXLite standardı, tam SpaceVPX’in sağladığı esneklik ve hata toleransını daha kompakt ve sade bir yapıda sunar. SpaceVPXLite özellikle 3U ve 6U form faktörlü küçük uydular için optimize edilmiştir. Standart, destek modüllerinin sayısını azaltarak daha verimli bir yerleşim sağlar; böylece görev bilgisayarında kullanılan kart sayısı, ağırlık ve güç tüketimi önemli ölçüde azalır.

Ayrıca SpaceVPXLite, yalnızca tek nokta hata toleransı için gerekli olan asgari altyapıyı korur. Bu sayede sistem güvenilirliğini düşürmeden, tam SpaceVPX’in gerektirdiği karmaşıklığın büyük kısmını ortadan kaldırır. Sonuç olarak küçük uydularda daha az alan kaplayan, daha az güç tüketen ve daha hafif bir görev bilgisayarı mimarisi elde edilir.

**Kintex UltraScale Yerine RT PolarFire SoC (MPFS250TS-1FCG1152I) Kullanımının Değerlendirilmesi**

Microchip’in Polarfire FPGA ailesi, düşük güç tüketimi, yüksek güvenilirlik ve güvenlik odaklı tasarlanmış, flash-tabanlı(non-volatile) FPGA mimarisine sahip cihazlardan oluşutç Geleneksel SRAM tabanlı FPGA’lardan farklı olarak, konfigürasyon bilgisini flash hücrelerde sakladığı için enerji kesintisinde kaybolmaz ve radrasyon kaynaklı single event upset riskine karşı çok daha dayanıklıdır.

MPFS250TS-1FCG1152I FPGA, Microchip’in Radiation-Tolerant (RT) ailesine ait olan bir cihazdır. düşük güç tüketimi, güvenilir flash tabanlı konfigürasyon belleği ve entegre RISC-V işlemci alt sistemi ile öne çıkar. MPFS250TS-1FCG1152I modeli, 250K mantık hücresi, 7.6 Mb LSRAM/uSRAM, 784 DSP blok ve 16 adet SerDes (12.7 Gbps’a kadar) kapasitesi sunar. Flash tabanlı konfigürasyon yapısı sayesinde, SRAM tabanlı FPGA’larda olduğu gibi harici scrubbing veya ikinci bir FPGA’ya ihtiyaç duymaz.

PolarFire SoC içindeki blok bellekler (uSRAM ve LSRAM), yüksek hızlı ve düşük gecikmeli bellek birimleridir. Toplam 7.6 Mb (yaklaşık 0.95 MB) kapasite, farklı genişlik ve derinlik kombinasyonlarında yapılandırılabilir. Bu bloklar FIFO, buffer, önbellek ve geçici veri depolama için kullanılır. Ayrıca, dahili ECC destekli eSRAM ve eNVM (non-volatile memory), sistem parametreleri ve yazılım saklama için güvenilir depolama sunar. Harici DDR4/LPDDR4 bellek kontrolcüsü de ECC ile korunmuş çalışma belleği sağlar.

Bu cihazların SoC versiyonu olan Polarfire SoC FPGA ailesi endüstrinin ilk RISC-V işlemci bazlı SoC FPGA’lerini ortaya sunmuştur. Bu aile, SiFive U54-MC çekirdek ailesi üzerine kurulu, toplamda beş çekirdekli 64-bit RISC-V mikroişlemci alt sistemi (MSS) ile PolarFire FPGA fabric’ini tek bir cihaz üzerinde birleştirir. Bu hibrit yapı sayesinde cihazlar FPGA’ların sunduğu ölçeklenebilir ve esnek donanım kaynaklaırnı, ASIC benzeri yüksek performans ve düşük güç tüketimini aynı anda kullanıma açar. İşlemci alt sistem şemasını görsel 1.11 de bulabilirsiniz.

metin, diyagram, plan, teknik çizim içeren bir resim

Yapay zeka tarafından oluşturulmuş içerik yanlış olabilir.

**1.11)** **PolarFire SoC işemci alt system şeması**

U54 uygulama çekirdekleri, üzerinde parallel olarak OS çalıştırılabilmesine olanak sağlar, yüksek performanslı uygulamalar için kullanılır. Bu çekirdekşer asıl iş gücünü yüklenir. Dört çekirdek parallel bir şekilde çalışarak yükse performanslı uygulamaları yürütür.

E51 monitor çekirdeği ise bu sistemin yönetici çekirdeği gibidir. Cihaz ilk açıldığında E51 çalışır ve tüm sistemi ayağa kaldırır. Boot işlemleri buradan yönetilir, gerekli donanım ayarlarını yapar. Sistemin güvenliğini denetler. Belleklerde veya FPGA fabric içinde oluşabilecek ECC/parity hatalarını terpit eder ve raporlar. Uygulama çekirdeklerini başlatır, gerekirse durdurur veya resetler, görev dağıtımınmı yapar.

Geleneksel SpaceCube yaklaşımında, SRAM tabanlı yüksek-performans Xilinx kintex ultrascale kullanılır. Bu cihazlar çok hızlı olmasına rağmen volatile bir bellek yapısı kullandıkları için SEU’ya dayanıksız olurlar. Radrasyona dayanıklaştırılmış bir fpga daha olması gerekir. Bu FPGA ana işlemci görevini üstlenen FPGA’i izler. Bunun için ise PROASIC RTG4 FPGA’I kullanılır. Bu FPGA’in görevi, konfigürasyon bozulup bozulmadığını izlemek, gerektiğinde yeniden yükleme ve konfigürasyon dosyalarını güncellemek ve hata olduğu zaman blind, controlled scrubbing yapıp SEU’ları düzeltme görevini üstlenir. Buna karşı olarak PolarFire SoC FPGA, flash tabanlı bellek hapısına sahiptir. Flash hücreler, SRAM gibi SEU’ya karşı hassas değildir ve güç kesilse bile konfigürasyon korunur. Bu nedenle, harici scrubbing mekanizmasona ya da ikinci bir FPGA’ye gerk yoktur, çünkü bu görevi işlemci içerisinde olan E51 çekirdeği sahipleniyor ve fazladan bir FPGA kullanma ihtiyacı ortadan kalkıyor.

Güç tüketimi açısında kintex UltraScale FPGA’lerin güç tüketimi yüksektir, SpaceCube gibi düşük bütçesi sınırlı uzay görevlerinde PolarFire SoC daha avantajlı bir rol oynuyor.

Ayrıca Kintex UltraScale harici bir işlemciye gerek duyar ya da MicroBlaze soft-core işlemci kullanılabilir, bu da kaynak tüketimini arttırır ve daha az güvenilir olur. PolarFire Soc yukarıda da bahsedildiği gibi içerisinde bir RISC-V işlemci alt sistemi içerir.

PolarFire SoC, SpaceCube mimarisinde sistemi basitleştirerek Kintex UltraScale FPGA’lere göre daha uygun bir çözüm sunuyor.

**Referanslar**

* <https://www.microchip.com/en-us/product/mpfs250ts#Design%20Resources>
* <https://www.microchip.com/en-us/development-tool/mpfs250-video-kit>
* <https://digitalcommons.usu.edu/smallsat/2019/all2019/158/>
* <https://github.com/nasa/nos3>
* <https://github.com/nasa/cFS>
* <https://ballaerospace.github.io/cosmos-website/>
* <https://www.star-dundee.com/wp-content/star_uploads/conference_papers/spacewire/2018_SPW_SpaceVPX-RTG4_Board_SpaceWire_SpaceFibre_Backplane.pdf>
* <https://www.te.com/en/industries/aerospace/insights/spacevpx-and-the-world-of-interconnect.html>
* L. Berthoud, M. Swartwout, J. Cutler ve D. Klumpar, “University CubeSat Project Management for Success,” presented at the 33rd Annual AIAA/USU Conference on Small Satellites (SSC), SSC19-WKIII-07, 2019.
* A. Geist et al., “SpaceCube v3.0 NASA Next-Generation High-Performance Processor for Science Applications,” presented at the Small Satellite Conference, Utah State University, Logan, UT, 2019.
* R. Diez, F. Jorge, K. Cahoy ve P. Collier, “SpaceVPX/FMC for Electronics Standardization and Modularity in High-Performance SmallSat Architectures,” presented at the Small Satellite Conference, Utah State University, Logan, UT, 2020.
* SpaceVPX Interoperability Assessment, NASA Technical Reports Server, 2022.
* https://technology.nasa.gov/patent/GSC-TOPS-35