|  |
| --- |
| H:\SU_AmblemLOGOtype_4.png |
| **ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ** |

**Mühendislik Tasarımı Dersi**

**Proje Final Raporu**

|  |  |
| --- | --- |
| **HAZIRLAYAN / LAR** |  |
| **Adı Soyadı :**  **Öğrenci Numarası :** | ENES ÇAKIR  203312011 |
|  |  |

|  |  |
| --- | --- |
| **TESLİM TARİHİ :** | 30/05/2025 |
| **PROJE DANIŞMANI :** | DR. YUNUS EMRE ACAR |

**ÖZET**

Bu çalışmada, endüstriyel ve bilimsel uygulamalarda sıklıkla ihtiyaç duyulan yüksek hızlı veri toplama ve depolama işlemlerini gerçekleştiren FPGA tabanlı bir sistem tasarlanmıştır. Sistem, Xilinx Nexys A7 FPGA kartı üzerinde VHDL donanım tanımlama dili kullanılarak gerçeklenmiştir. Tasarım ve geliştirme sürecinde Xilinx Vivado tasarım ortamı kullanılmış, IP Core entegrasyonları ve sistem testleri bu platform üzerinden gerçekleştirilmiştir.

Sistemin temel yapısı, Vivado IP Catalog'dan alınan XADC (Xilinx Analog-to-Digital Converter) IP Core entegrasyonuna dayanmaktadır. XADC modülü ile analog sinyaller 12-bit çözünürlükle dijital ortama aktarılmaktadır. Sistemde gerçek zamanlı durum izlemesi için LED göstergeler kullanılmış ve alarm durumları için uyarı mekanizması uygulanmıştır yapılmıştır.

Mevcut tasarımda, XADC ve verimli veri transferi için FIFO (First-In-First-Out) arabellek yapısı başarıyla entegre edilmiştir. Vivado simülasyon ortamında ve gerçek donanım üzerinde yapılan testler, bu entegrasyonun beklendiği gibi çalıştığını doğrulamaktadır.

Bu çalışma, modern veri toplama ve işleme uygulamalarında kullanılabilecek esnek ve genişletilebilir bir platform sunmaktadır. Özellikle yüksek örnekleme hızı gerektiren ölçüm sistemleri, sensör ağları ve gerçek zamanlı izleme gerektiren alanlarda kullanılabilecek bir altyapı oluşturmaktadır. Sistemin Vivado ortamında geliştirilen modüler yapısı, farklı depolama stratejileri ve veri işleme algoritmaları ile genişletilebilmesine olanak sağlamaktadır.

**Anahtar Kelimeler**: FPGA, XADC, FIFO, VHDL, Yüksek Hızlı Veri Toplama, Asenkron Veri Transferi, Xilinx Vivado, Nexys A7

**Sistem Özellikleri**

* FPGA Kartı: Xilinx Nexys A7
* Tasarım Dili: VHDL
* Ana Bileşenler:
  + XADC (Analog-Dijital Dönüştürücü) IP Core (2-3, 10-11 kanalları aktif)
  + UART(Universal Asynchronous Receiver-Transmitter)
  + FIFO Generator (12-bit genişlik, 1024 derinlik)

**Mevcut Durum**

* Harici analog sinyalleri başarıyla yüksek hızda dijitale çevirmekte.
* ADC ile alınan veriler Binary olarak ledler üzerinde gösterilmektedir.
* UART ile ADC den okunan verileri bilgisayar ortamına aktarılıyor ve kaydediliyor.
* Python ile bilgisayar ortamına aktarılan verinin hangi kanaldan geldiğini gösteriyor ve görselleştiriyoruz.

**1.Giriş**

Modern elektronik sistemlerde, analog sinyallerin yüksek hızda dijital ortama aktarılması ve büyük miktarda verinin depolanması giderek daha fazla önem kazanmaktadır. Özellikle endüstriyel uygulamalar, bilimsel deney sistemleri, IoT sensör ağları ve haberleşme teknolojilerinde, analog sinyallerin yüksek hassasiyetle dijitale dönüştürülmesi ve verimli şekilde saklanması kritik bir gereksinimdir.

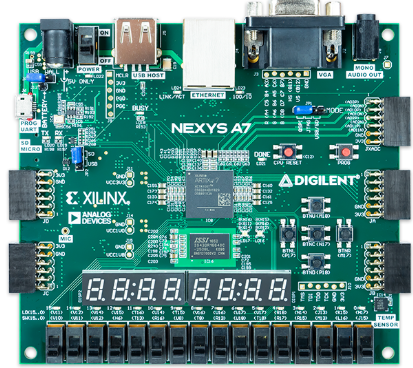
**1.1 Projenin Amacı**

Bu projenin amacı, analog sinyallerin yüksek örnekleme hızında dijital ortama aktarılmasını ve büyük kapasiteli bellek birimlerinde saklanmasını sağlayan FPGA tabanlı bir veri toplama ve depolama sistemi geliştirmektir. Projede aşağıdaki hedefler gerçekleştirilecektir:

* Xilinx Nexys A7 FPGA kartı üzerindeki XADC modülünü kullanarak analog sinyallerin 12-bit çözünürlükle dijitale dönüştürülmesi
* Farklı saat domainleri arasında güvenilir veri transferi için FIFO arabellek kullanımı
* Sistem durumunun gerçek zamanlı olarak LED göstergeler üzerinden izlenmesi
* UART protokolü ile 12-bitlik verinin bilgisayar ortamına aktarılımı ve Python ile verilerin görselleştirilmesi

**1.2 Kullanılan Donanım**

* **FPGA Kartı**: Xilinx Nexys A7 Geliştirme Kartı
  + FPGA: xc7a100tcsg324-1
  + Çalışma Frekansı: 100 Mhz



**1.3. Kullanılan Yazılım ve IP Core'lar**

* Xilinx Vivado Design Suite 2024.2
* XADC Wizard IP Core
* FIFO IP Core
* PyCharm

**2. IP CATALOG**

IP Catalog (Intellectual Property Catalog), Xilinx Vivado tasarım ortamının önemli bir bileşenidir ve FPGA tasarımlarında kullanılmak üzere hazır IP çekirdekleri sunar. Bu katalog, temel dijital işleme bloklarından karmaşık sistem bileşenlerine kadar geniş bir yelpazede önceden tasarlanmış ve test edilmiş modüller içerir.

Katalogda bulunan IP'ler, bellek arayüzleri, DSP blokları, haberleşme protokolleri, video işleme modülleri ve mikroişlemci sistemleri gibi çeşitli kategorilere ayrılmıştır. Her IP, özelleştirilebilir parametrelere sahiptir ve projenin gereksinimlerine göre yapılandırılabilir. Bu IP'ler, tasarım sürecini hızlandırır ve güvenilirliği artırır çünkü her biri kapsamlı şekilde test edilmiş ve doğrulanmıştır.

IP Catalog'un en önemli avantajlarından biri, kullanıcıların sıfırdan tasarım yapmak yerine hazır ve optimize edilmiş blokları kullanarak karmaşık sistemler oluşturabilmesidir. Ayrıca, bu IP'lerin çoğu otomatik olarak oluşturulan dokümantasyon, simülasyon modelleri ve örnek tasarımlarla birlikte gelir. Bu özellikler, FPGA tasarım sürecini daha verimli ve erişilebilir hale getirir.

**2.1. XADC Wizard**

XADC Wizard, Xilinx FPGA'lerde bulunan Analog-To-Dijital dönüştürücü (ADC) modülünü konfigüre etmek için kullanılan önemli bir araçtır. Bu araç, analog sinyalleri dijital verilere dönüştürme, sıcaklık izleme, voltaj izleme ve çoklu kanal örnekleme gibi temel fonksiyonları yerine getirir.

XADC, 12-bit çözünürlük ve 1 MSPS'ye kadar örnekleme hızı sunar. Ayrıca 16 yardımcı analog giriş (Vaux) kanalına sahiptir. Dahili sıcaklık ve voltaj sensörleri bulunur ve DRP (Dynamic Reconfiguration Port) arayüzü ile donatılmıştır. Otomatik kalibrasyon özelliği sayesinde hassas ölçümler yapılabilir.

Bu modül, sistem sağlığı izleme, sıcaklık kontrol sistemleri, voltaj izleme ve düzenleme, analog sensör veri toplama ve endüstriyel otomasyon gibi çeşitli alanlarda kullanılır. XADC Wizard, tüm bu özelliklerin kolayca konfigüre edilmesini sağlayan kullanıcı dostu bir grafiksel arayüz sunar ve FPGA tasarımlarında analog-dijital dönüşüm işlemlerini yönetmeyi önemli ölçüde kolaylaştırır.

**2.1.1. XADC IP Core Konfigürasyon**

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

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.**2.1.1.1.Basic Konfigürasyonu**

**1. Arayüz Seçenekleri (Interface Options)**

XADC IP Core'un arayüz konfigürasyonunda, DRP (Dynamic Reconfiguration Port - Dinamik Yeniden Yapılandırma Portu) seçilmiştir. Zamanlama için Sürekli Mod (Continuous Mode) aktif edilmiştir. Bu seçim sayesinde ADC, herhangi bir tetikleme olayına ihtiyaç duymadan sürekli olarak örnekleme işlemini gerçekleştirebilmektedir.

**DRP (Dynamic Reconfiguration Port) Nedir?**

DRP (Dynamic Reconfiguration Port), FPGA'larda çalışma zamanında belirli bileşenlerin yeniden yapılandırılmasını sağlayan önemli bir arayüzdür. Bu arayüz sayesinde FPGA'lar çalışma esnasında PLL, SerDes, bellek veya I/O standartları gibi dahili bileşenlerin parametrelerini değiştirilebilir, böylece sistem durdurulmadan ve tam yeniden programlama gerektirmeden adaptif davranış sergileyebilir.

**2. Timing Mode Seçenekleri**

**Continuous Mode özelliği aktif edilmiştir. Bu özellik sürekli olarak çalışacağı anlamına gelmektedir.**

**3. DRP Zamanlama Seçenekleri**

DRP zamanlama seçenekleri aşağıdaki şekilde ayarlanmıştır:

* DCLK (Digital Clock) aktif durumdadır
* DCLK Frekansı 100 MHz olarak belirlenmiştir
* ADC Dönüştürme Hızı (Conversion Rate) 1000 KSPS olarak ayarlanmıştır
* Sistemin Gerçek Dönüştürme Hızı 961.54 KSPS'dir
* Clock bölücü değeri 4 olarak belirlenmiştir
* ADC Saat Frekansı 25.00 MHz olarak hesaplanmıştır

**4. AXI4STREAM Seçenekleri**

AXI4STREAM özelliği devre dışı bırakılmıştır.

**5. Kontrol/Durum Portları**

Kontrol ve durum portları konfigürasyonunda:

* Reset\_in portu aktif edilmiştir
* Temp Bus özelliği devre dışıdır
* JTAG Arbiter devre dışı bırakılmıştır

**6. Analog Simülasyon Dosya Seçenekleri**

Simülasyon ayarları aşağıdaki şekilde yapılandırılmıştır:

* Simülasyon Dosyası: Default
* Dalga Formu Tipi: CONSTANT
* Frekans: 1.0 KHz
* Dalga Sayısı: 1

Bu konfigürasyon, tek kanallı ve sürekli örnekleme yapabilen, 1000 KSPS hızında çalışan bir ADC sistemi oluşturmaktadır. DRP arayüzünün kullanılması, sistem çalışırken parametrelerin dinamik olarak değiştirilebilmesine olanak sağlamaktadır.

**2.1.1.2.ADC Setup Konfigürasyonu**

**metin, ekran görüntüsü, yazılım, sayı, numara içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.**

**1. Sıralayıcı ve Ortalama Ayarları**

**Sequencer Mode (Sıralayıcı Modu)**

* Sequencer Mode "Off" olarak ayarlanmıştır.
* Bu ayar, belirlenen kanalların sürekli ve sıralı olarak taranmasını sağlar.
* Kapalı olması, otomatik kanal taramasının devre dışı olduğunu gösterir.

**Channel Averaging (Kanal Ortalaması)**

* "None" olarak ayarlanmıştır.
* Bu seçim, örneklenen değerlerin ortalamasının alınmayacağını gösterir.
* Anlık ölçüm değerleri doğrudan kullanılacaktır.

**2. Kalibrasyon Ayarları**

**ADC Kalibrasyon**

* "ADC Offset and Gain Calibration" seçeneği aktif edilmiştir.
* Bu, ADC'nin hem offset hem de kazanç kalibrasyonunu yapacağını gösterir.
* Sadece offset kalibrasyonu seçeneği devre dışı bırakılmıştır.

**Supply Sensor Calibration (Besleme Sensörü Kalibrasyonu)**

* "Sensor Offset and Gain Calibration" seçeneği aktif edilmiştir.
* Bu ayar, sensör ölçümlerinde hem offset hem de kazanç kalibrasyonunun yapılacağını belirtir.

**CALIBRATION Averaging**

* "Enable CALIBRATION Averaging" seçeneği aktif edilmiştir.
* Bu özellik, kalibrasyon işlemi sırasında birden fazla ölçümün ortalamasının alınmasını sağlar.
* Kalibrasyon doğruluğunu artırmak için kullanılır.

**3. Harici Multiplexer Ayarları**

**External Multiplexer Setup**

* "External Multiplexer" seçeneği devre dışı bırakılmıştır.
* Channel for MUX: "VP VN" olarak seçilmiştir.
* "Enable muxaddr\_out port" seçeneği devre dışıdır.
* Bu ayarlar, harici bir multiplexer kullanılmayacağını gösterir.

**4. Power Down Options (Güç Kesme Seçenekleri)**

* ADCB seçeneği devre dışı
* ADCA seçeneği devre dışı
* Bu ayarlar, ADC'nin güç yönetimi özelliklerinin kullanılmadığını gösterir.

Bu konfigürasyon, kalibrasyon özelliklerinin aktif olduğu, harici multiplexer kullanılmayan ve güç yönetimi özelliklerinin devre dışı bırakıldığı bir ADC kurulumunu tanımlar. Özellikle kalibrasyon ayarlarının aktif edilmesi, ölçüm doğruluğuna önem verildiğini göstermektedir.

**2.1.1.3.Alarms Konfigürasyonu**

**metin, yazılım, sayı, numara, bilgisayar simgesi içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.**

**1. Sıcaklık Alarmları**

**Over Temperature Alarm (Aşırı Sıcaklık Alarmı)**

* Ayarlanan değerler: Trigger (Tetikleme): 125.0°C, Reset: 70.0°C

**User Temperature Alarm (Kullanıcı Sıcaklık Alarmı)**

* Ayarlanan değerler: Trigger (Tetikleme): 85.0°C, Reset: 60.0°C

**2. Voltaj Alarmları**

**VCCINT Alarm (İç Çekirdek Voltaj Alarmı)**

* Ayarlanan değerler: Lower (Alt Limit): 0.97V, Upper (Üst Limit): 1.03V

**VCCAUX Alarm (Yardımcı Voltaj Alarmı)**

* Ayarlanan değerler: Lower (Alt Limit): 1.75V, Upper (Üst Limit): 1.89V

**VCCBRAM Alarm (Block RAM Voltaj Alarmı)**

* Bu alarm seçeneği devre dışı bırakılmıştır
* Ayarlanan değerler: Lower (Alt Limit): 0.95V, Upper (Üst Limit): 1.05V
* Sıcaklık alarmları için güvenli çalışma aralıkları belirlenmiştir
* VCCINT için ±3% tolerans aralığı (0.97V - 1.03V)
* VCCAUX için çalışma limitleri (1.75V - 1.89V)
* VCCBRAM için ±5% tolerans aralığı (0.95V - 1.05V)

Bu ayarlar, sistemin kritik parametrelerinin izlenmesi için tanımlanmış olup, gerektiğinde aktif edilebilecek şekilde hazır tutulmaktadır.

**2.1.1.4.Channel Sequencer Konfigürasyonu**

**metin, ekran görüntüsü, sayı, numara, paralel içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.**

**1. Sıcaklık Alarmları**

Over Temperature Alarm (Aşırı Sıcaklık Alarmı)  
• Ayarlanan değerler: Trigger (Tetikleme): 125.0°C, Reset: 70.0°C

User Temperature Alarm (Kullanıcı Sıcaklık Alarmı)  
• Bu alarm çıkışı (user\_temp\_alarm\_out) aktif edilmiştir

**2. Voltaj Alarmları**

VCCINT Alarm (İç Çekirdek Voltaj Alarmı)  
• Alarm çıkışı (vccint\_alarm\_out) aktiftir

VCCAUX Alarm (Yardımcı Voltaj Alarmı)  
• Alarm çıkışı (vccaux\_alarm\_out) aktiftir

VCCBRAM Alarm (Block RAM Voltaj Alarmı)  
• Bu alarm seçeneği devre dışı bırakılmıştır

**3. ADC Kanal Seçimi (Channel Sequencer)**

Etkinleştirilen Kanallar:

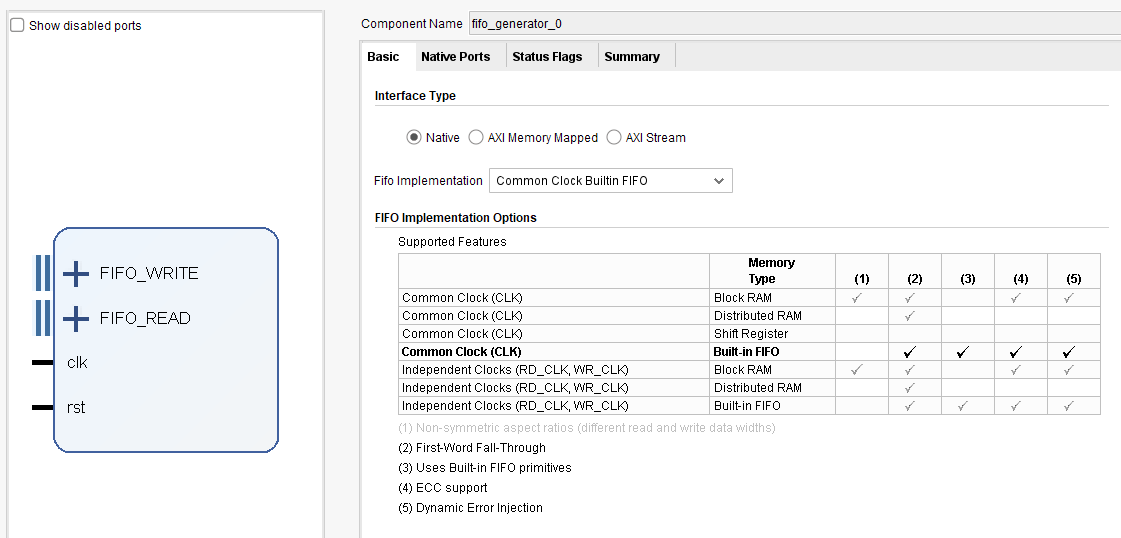
VP/VN Diferansiyel Girişi: Etkinleştirilmiş  
• Bu kanal, harici diferansiyel analog sinyallerin dijital forma çevrilmesi için kullanılır.  
• Kullanıcı harici bir ADC girişi sağladığında genellikle bu kanal kullanılır.

VAUX2 Kanalı: Etkinleştirilmiş  
• VAUX2 kanalı, tek uçlu analog sinyallerin izlenmesi için kullanılır.  
• Kullanıcı, bu kanala analog bir voltaj uygulayarak ölçüm alabilir.

VAUX10 Kanalı: Etkinleştirilmiş  
• VAUX10, sistemdeki başka bir analog sinyali izlemek için kullanılmaktadır.  
• Kullanıcı, belirli bir sensör veya referans gerilim bu kanala bağlıysa izleme yapabilir.

**4. Devre Dışı Bırakılan Kanallar**

Diğer Kanallar (CALIBRATION, TEMPERATURE, VCCINT, VCCAUX, VCCBRAM, VREFP, VREFN, VAUX0–VAUX1, VAUX3–VAUX9, VAUX11–VAUX15): Devre dışı bırakılmıştır  
• Bu kanallar şu anda izlenmemekte olup, ihtiyaç halinde XADC yeniden yapılandırılarak aktif edilebilir.  
• Devre dışı bırakılma, sistem kaynaklarını daha verimli kullanmak amacıyla tercih edilmiştir.

**2.2.1. FIFO Generator IP Core Konfigürasyon**

**2.2.1.1. Genel Ayarlar (Basic)**

**FIFO (First-In-First-Out) Konfigürasyonu**

Xilinx Vivado'daki FIFO Generator IP Core konfigürasyonu gösterilmektedir. Bu FIFO, yüksek hızlı veri toplama sistemimizde XADC (Analog-Dijital Dönüştürücü) ile DDR2 bellek arayüzü arasında bir arabellek olarak kullanılacaktır.

**Temel Arayüz Yapılandırması**

* **Arayüz Tipi**: Native (Yerel) arayüz seçilmiştir. Bu, FIFO'ya doğrudan erişim için basit bir arayüz sağlar.**metin, ekran görüntüsü, yazılım, sayı, numara içeren bir resim

  Yapay zeka tarafından oluşturulan içerik yanlış olabilir.FIFO Implementasyonu**: Common Clock Builtin FIFO seçilmiştir. Bu seçenek, okuma ve yazma işlemlerinin aynı saat (clock) sinyali ile kontrol edildiğini belirtir.

**Veri Portu Parametreleri**

* **Yazma Genişliği (Write Width)**: 16 bit olarak ayarlanmıştır. Bu, XADC'nin 12-bit çözünürlükle çalıştığı düşünüldüğünde uygun bir seçimdir.
* **Yazma Derinliği (Write Depth)**: 131072 olarak ayarlanmıştır. Bu, FIFO'nun 1024 adet 12-bitlik veri saklayabileceği anlamına gelir.
* **Okuma Genişliği (Read Width)**: 16 bit olarak ayarlanmıştır, yazma genişliği ile aynıdır.
* **Okuma Derinliği (Read Depth)**: 131072 olarak ayarlanmıştır, yazma derinliği ile aynıdır.

**ECC ve Güç Yönetimi**

* **ECC (Error Correction Code)**: Devre dışı bırakılmıştır.
* **Output Registers**: Embedded Registers seçeneği seçilmiştir, bu FPGA kaynaklarının verimli kullanımını sağlar.

**Başlatma ve Reset Seçenekleri**

* **Reset Pin**: Aktif edilmiştir, FIFO'yu sıfırlamak için bir giriş pini sağlar.
* **Enable Reset Synchronization**: Aktif edilmiştir, reset sinyalinin saat ile senkronize edilmesini sağlar.
* **Reset Type**: Asynchronous Reset seçilmiştir, reset sinyali saate bakılmaksızın anında etki eder.
* **Full Flags Reset Value**: 0 olarak ayarlanmıştır, reset sonrası "dolu" bayrağını devre dışı bırakır.

**metin, ekran görüntüsü, yazı tipi, sayı, numara içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.**

**Handshaking Seçenekleri ve Programlanabilir Bayraklar**

* Handshaking (el sıkışma) özellikleri devre dışı bırakılmıştır.
* **Programmable Full Type**: No Programmable Full Threshold seçilmiştir.
* **Full Threshold Assert/Negate Value**: Sırasıyla 1022 ve 1021 olarak ayarlanmıştır.
* **Programmable Empty Type**: No Programmable Empty Threshold seçilmiştir.
* **Empty Threshold Assert/Negate Value**: Sırasıyla 2 ve 3 olarak ayarlanmıştır.

**3. SİSTEM TASARIMI**

**3.1. Top Level Kod**

**3.1.1 Entity Kod**

Entity: Verilen bir lojik fonksiyon için bütün giriş ve çıkışları tanımlar. Yani lojik fonksiyonun dış dünyayla bağlantısını tanımlar. Her VHDL tasarım mutlaka en az bir entity içerir.

Entity kısmının örneği aşağıdaki gibidir.

metin, ekran görüntüsü, yazı tipi içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

Entity kısmının ne işe yaradığını öğrendiysek bu projemdeki entity kısmına geçebiliriz.

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

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

Projemin Entity Kısmı aşağıdaki gibidir. Peki bunlar ne işe yarar?

**Giriş Birimleri**:

**Clock**: Sistemin zamanlamasını sağlayan dahili saat birimidir. Nexys A7 kartının sağladığı 100 MHz’lik bir saat sinyali kullanılmaktadır. Bu sinyal, sistemdeki tüm zamanlama ve senkronizasyon işlemleri için temel referans olarak görev yapar.

**SW (Switch)**: Nexys A7 kartında bulunan 16 adet anahtardan biri kullanılmıştır. Bu anahtar, hangi ADC kanalından okunan verinin LED’lerde görüntüleneceğini belirler.

* SW = '0' iken Channel 2’den okunan veri LED’lerde gösterilir.
* SW = '1' iken ise **channel 10**'dan okunan veri gösterilir.

**BTNU (Button Up)**: Kart üzerindeki dört butondan biri olan BTNU, sistemin çalışmasını başlatmak için kullanılır. Kullanıcı bu butona bastığında ADC’den veri okuma işlemi başlatılır ve UART üzerinden veri aktarımı gerçekleştirilir.

channel 2 ve channel 10: Bu iki port, ADC’nin sırasıyla 2. ve 10. kanallarından veri okunmasını sağlar. Sistemin, farklı veri kaynaklarından bilgi alabilmesi için bu iki kanal kullanılmıştır

**Çıkış Birimleri**:

T**x\_o**: UART haberleşmesi için kullanılan çıkış portudur. ADC'den okunan veriler, bu port aracılığıyla bit düzeyinde dış bir sisteme seri olarak iletilir.

**LED**: Nexys A7 kartındaki 16 adet LED çıkış olarak tanımlanmıştır. Bu LED'ler, ADC kanallarından okunan verileri görsel olarak kullanıcıya göstermek için kullanılır. Hangi kanalın verisinin gösterileceği, SW anahtarının konumuna göre belirlenir.

**LED16\_B**: Kart üzerinde yer alan RGB LED’lerden biridir. Sistem çalışmaya başladığında bu LED aktif hale gelir. Bu sayede sistemin başlatıldığını kullanıcı görsel olarak anlayabilir.

**3.1.2 Mimari Yapının Tanımlama Alanı: Architecture – Begin Arası**

VHDL'de bir mimarinin (architecture) tanımında, architecture ve begin anahtar kelimeleri arasında yer alan kısım, devrenin yapısal veya davranışsal özelliklerini belirlemek için kullanılır. Bu bölümde sinyaller (signal), tür tanımları (type), sabitler (constant) ve bileşen tanımları (component) gibi çeşitli öğeler tanımlanır. Bu tanımlar, begin anahtar kelimesinden sonra gelen mimari gövdesinde kullanılacak olan yapı taşlarını oluşturur. Böylece, devrenin iç yapısı daha modüler ve anlaşılır bir şekilde inşa edilebilir. Özellikle bileşen tanımlamaları, alt modüllerin üst modüle nasıl bağlandığını belirtmek için önemlidir.

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

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

**Component Syntax Yapısı:**

metin, ekran görüntüsü, yazı tipi içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

**Signal Syntax Yapısı:**



**Type Syntax Yapısı:**



**Constant Syntax Yapısı:**



Bu kısmın ne işe yaradığını öğrendiysek bu projemdeki Architecture – Begin kısmına geçebiliriz.

**Component Tanımlamaları:**

**XADC**

**metin, ekran görüntüsü, yazı tipi, mor içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.**

**XADC IP Core – Giriş (IN) ve Çıkış (OUT) Portları**

**Girişler (Inputs):**

* **di\_in:** Veri giriş hattıdır. Yazma işlemlerinde kullanılacak olan 16 bitlik veri bu port üzerinden girilir.
* **daddr\_in:** Okuma/yazma yapılacak register adresini belirtir (7 bitlik adres verisi).
* **den\_in:** Veri erişimini etkinleştirir. Yüksek olduğunda, okuma veya yazma işlemi başlatılır.
* **dwe\_in:** Veri yazma işlemi için enable (yazma etkinleştirici) sinyaldir.
* **dclk\_in:** XADC modülünün saat (clock) sinyalidir.
* **reset\_in:** Modülü sıfırlamak için kullanılır. Aktif olduğunda XADC yeniden başlatılır.
* **vp\_in:** Diferansiyel analog girişin pozitif ucudur (ana analog girişlerden biridir).
* **vn\_in:** Diferansiyel analog girişin negatif ucudur.
* **Vauxp2:** Yardımcı analog giriş kanalı 2’nin pozitif ucudur.
* **Vauxn2:** Yardımcı analog giriş kanalı 2’nin negatif ucudur.
* **Vauxp10:** Yardımcı analog giriş kanalı 10’ın pozitif ucudur.
* Vauxn10: Yardımcı analog giriş kanalı 10’ın negatif ucudur.

**Çıkışlar (Outputs):**

* **drdy\_out:** Veri hazır sinyali. ADC dönüşüm sonucu hazır olduğunda aktif olur.
* **do\_out:** ADC dönüşüm sonucu verinin 16 bitlik çıktısıdır.
* **user\_temp\_alarm\_out:** Kullanıcı tanımlı sıcaklık eşiği aşıldığında aktif olan alarm sinyalidir.
* **vccint\_alarm\_out:** İç voltaj (VCCINT) alarm durumu çıktısıdır.
* **vccaux\_alarm\_out:** Yardımcı voltaj (VCCAUX) alarm durumu çıktısıdır.
* **ot\_out:** Aşırı sıcaklık durumu alarm sinyalidir (Over Temperature).
* **channel\_out:** O an dönüşüm yapılan kanalın numarasını veren 5 bitlik çıkıştır.
* **eoc\_out:** ADC dönüşümünün bittiğini (End of Conversion) belirten sinyaldir.
* **alarm\_out:** Genel alarm sinyalidir. Birden fazla alarm tetiklendiğinde aktif olur.
* **eos\_out:** Sıralı tüm ADC kanallarının dönüşümünün tamamlandığını gösterir (End of Sequence).
* **busy\_out:** ADC’nin meşgul olduğunu gösteren durum sinyalidir.

**FIFO**

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

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

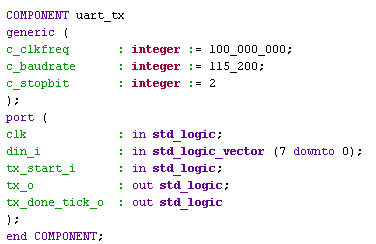
**FIFO IP Core – Giriş ve Çıkışlar**

**Girişler (Inputs):**

* **clk:** FIFO’nun saat (clock) sinyali.
* **Rst:** FIFO modülünü sıfırlamak için kullanılır.
* **din:** FIFO’ya yazılacak olan 16 bitlik veri giriş hattı.
* **wr\_en:** FIFO’ya veri yazma işlemini etkinleştiren sinyal.
* **rd\_en:** FIFO’dan veri okuma işlemini etkinleştiren sinyal.

**Çıkışlar (Outputs):**

* **dout:** FIFO’dan okunan 16 bitlik veri çıkış hattı.
* **full:** FIFO’nun dolu olduğunu belirten çıkış sinyali.
* **empty:** FIFO’nun boş olduğunu belirten çıkış sinyali.
* **UART**



**UART\_TX Modülü – Giriş, Çıkışlar ve Generics**

**Generics (Sabit Parametreler):**

* **c\_clkfreq:** Sistem saat frekansı (varsayılan: 100\_000\_000 Hz).
* **c\_baudrate:** UART baud rate değeri (varsayılan: 115200).
* **c\_stopbit:** Kullanılacak stop bit sayısı (varsayılan: 2)

**Girişler (Inputs):**

* **clk:** UART modülünün saat sinyali.
* **din\_i:** UART üzerinden gönderilecek 8 bitlik veri.
* **tx\_start\_i:** Veri gönderimini başlatan sinyal (start trigger).

**Çıkışlar (Outputs):**

* **tx\_o:** Seri iletişim hattına gönderilen veri çıkışı.
* **tx\_done\_tick\_o:** Veri gönderiminin tamamlandığını gösteren bit.
* **BUTTON DEBOUNCE**

metin, ekran görüntüsü, yazı tipi içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

**Debounce Modülü – Giriş, Çıkışlar ve Generics**

**Generics (Sabit Parametreler):**

* **c\_clkfreq:**Sistem saat frekansı (varsayılan: 100\_000\_000 Hz).
* **c\_debtime:**Zıplamayı (bounce) engellemek için kullanılan debounce süresi (varsayılan: 1000 clock cycle).
* **c\_initval:**Başlangıç değeri (varsayılan: '0').

**Girişler (Inputs):**

* **clk:**Saat sinyali.
* **sigmal\_i:**Giriş sinyali (buton vb.).

**Çıkışlar (Outputs):**

* **sigmal\_o:**Debounce işleminden geçmiş, kararlı çıkış sinyali.

Sinyal ve Type Tanımlamaları:

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

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

metin, ekran görüntüsü, yazı tipi içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

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

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

metin, ekran görüntüsü, yazı tipi, doküman, belge içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

**3.2.1.Componentlerin Implemention’ı ve Process’ler**

**metin, ekran görüntüsü, yazı tipi, doküman, belge içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.XADC PORT MAP:**

metin, ekran görüntüsü, yazı tipi içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.**FIFO PORT MAP:**

metin, ekran görüntüsü, yazı tipi içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.**UART PORT MAP:**

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

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.**

**BUTTON PORT MAP:**

**CHANNEL\_PROCESS:**

Öncelikle aşağıdaki sonlu durum makinesinde kanallar arasında geçiş yapabilmemiz için 2 adet durum oluşturuyoruz. Bu sonlu durum makinesiyle kanallar arasında geçişi gerçekleştiriyoruz.

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

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

metin, ekran görüntüsü, yazı tipi içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

.

Sistemimizde durumlar arasındaki geçişi data\_process ile gerçekleştiriyoruz.

**DATA\_PROCESS:**

Data\_process de UART ile her adımda 8 bit gönderdiğimiz için bir kanaldan okuduğumuz verinin tamamını ancak 4 adımda gönderebiliyoruz. Bunun için aşağıdaki gibi bir sonlu durum makinesi oluşturuyoruz.

Okunan 16 bitlik verinin önce **MSB (Most Significant Bits)** kısmı do\_out1\_first durumunda UART üzerinden gönderilir. Ardından do\_out1\_last durumuna geçilir ve verinin **LSB (Least Significant Bits)** kısmı gönderilir. ADC tarafından sağlanan veri yalnızca 12 bit olduğundan, LSB’nin son 4 biti "0000" ile doldurulur. Bu işlem, UART protokolüne uygun 8 bitlik veri gönderimi sağlamak için yapılır. do\_out1\_last\_done\_tick sinyali '1' yapılarak channel2’den channel10’a geçişi başlatacak sürecin tetiklenmesi sağlanır. Aynı mantık do\_out2\_first ve do\_out2\_last durumları için de geçerlidir; burada channel10’dan alınan verinin MSB ve LSB kısımları sırayla UART ile gönderilir. LSB’nin düşük 4 biti yine "0000" ile tamamlanır ve do\_out2\_last\_done\_tick sinyali '1' yapılarak FSM’in tekrar channel2'ye geçmesi sağlanır. Bu şekilde, channel2 ve channel10 arasında dönüşümlü bir veri aktarım döngüsü sağlanır ve bu geçişler ilgili **done\_tick** sinyalleriyle kontrol edilir. Done\_tick sinyalleri channel\_process kısmında tetiklenerek kanallar arası geçiş sağlanır.

metin, yazı tipi, ekran görüntüsü içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

metin, ekran görüntüsü, yazı tipi içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

metin, ekran görüntüsü, yazı tipi, doküman, belge içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

**Switch ve Led Process’leri:**

Bu kısımda ise anahtarın konumuna göre konumlar arasında geçiş yapıyor ve ledler ile gösteriyoruz.

metin, ekran görüntüsü, yazı tipi içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

**Button Process:**

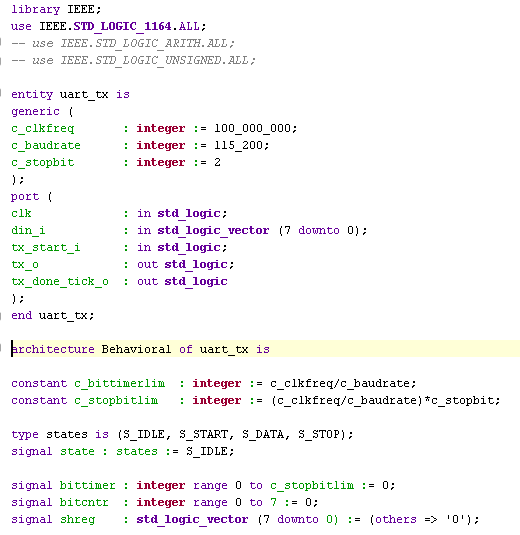
BTNU ile Sistemi başlatıyoruz.

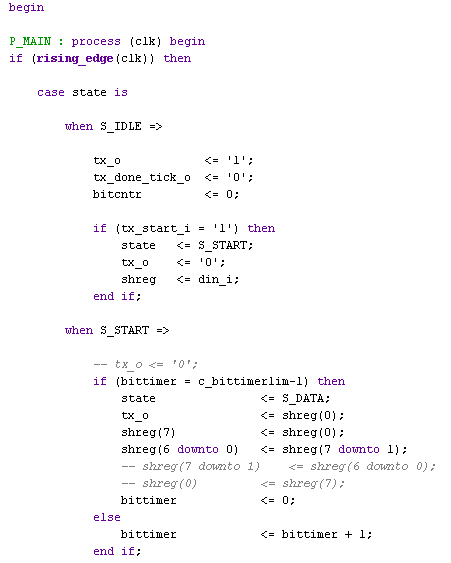
metin, ekran görüntüsü, yazı tipi içeren bir resim

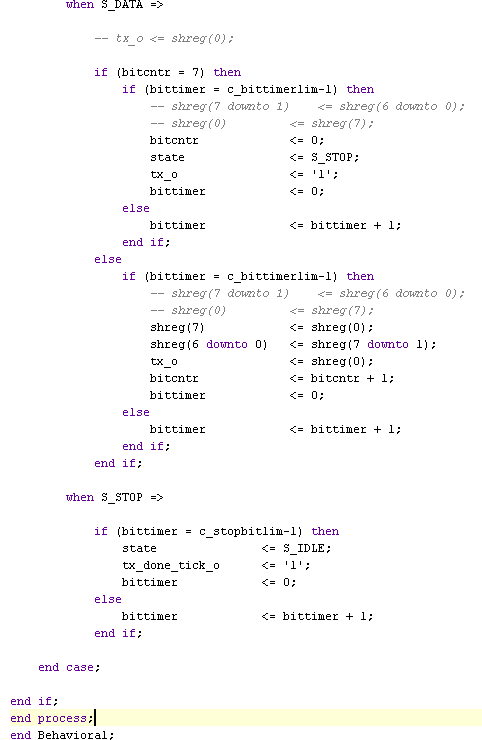
Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

**3.3.VHDL Modelleri**

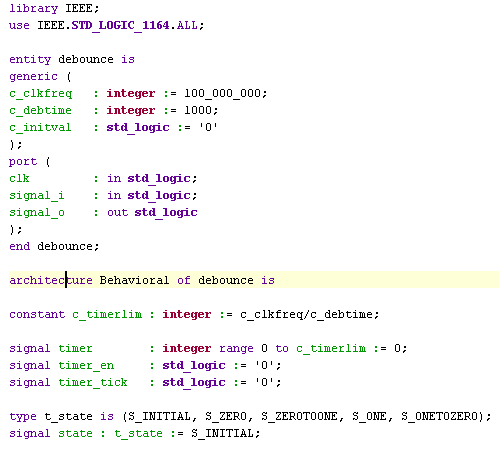
**3.3.1 UART VHDL kod**

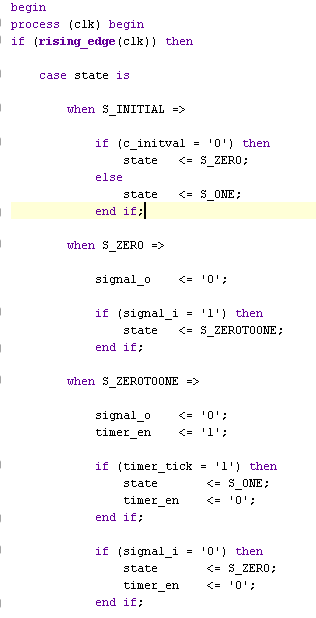
****

****



**3.3.2 Debounce VHDL kod**





metin, ekran görüntüsü, yazı tipi, doküman, belge içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

**3.3 Constraint kodları**

****

**metin, ekran görüntüsü, dikdörtgen, kalıp, desen, düzen içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.**

**4.1.0 HTERM ile Verilerin Toplanması**

**metin, elektronik donanım, ekran görüntüsü, ekran, görüntüleme içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.kalıp, desen, düzen, metin, kumaş, doku içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.**

Buradaki veriler ilk 2 veri channel2 sonraki 2 veri channel10’dan gelecek şekilde döngü olarak ADC’yi çalıştırdığımız sürece devam ediyor. Buradaki Verileri topladıktan sonra kaydedip PyCharm ortamında işlemlere devam ediyoruz.

**4.1.1 Pyton ile Grafik Oluşturma ve Kanalları Ayırma**

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

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

metin, ekran görüntüsü, yazılım içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

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

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

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

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

HTERM’de topladığımız verileri kaydettiğimiz dosyanın ismini koddaki dosya ismi değişkenine yazıyoruz

ekran görüntüsü, metin, yazılım, bilgisayar içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

metin, yazılım, multimedya yazılımı, grafik yazılımı içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.metin, yazılım, multimedya yazılımı, grafik yazılımı içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.Kodu çalıştırdığımızda verileri ve kanalları ayırıp grafik haline getiriyor.

metin, ekran görüntüsü, çizgi, diyagram içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

metin, ekran görüntüsü, öykü gelişim çizgisi; kumpas; grafiğini çıkarma, çizgi içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.