



TOBB ETÜ

Ekonomi ve Teknoloji Üniversitesi

Ad	Büşra
Soyad	Mercan
Bölüm	Elektrik - Elektronik Mühendisliği
Numara	191201005
Tarih	19.12.2022

2022-2023 GÜZ DÖNEMİ

GÖMÜLÜ SİSTEMLER – ELE519 DERSİ DÖNEM PROJESİ RAPORU

AÇIKLAMALAR

ZYNQ veri yazmayı tamamladı: 0xA

FPGA veri okumayı tamamladı: 0xB

FPGA veriyi filtreledi ve yazmayı tamamladı: 0xC

FPGA tarafında işlemler clock rising edge tarafında yapıldı. Bu nedenle 1 clock sonrasında istenen adresten veri okunması ve istenen adrese veri yazılması tamamlandı.

FPGA tarafındaki işlemler ile ilgili:

5 farklı durum belirlendi: bekleme modu, okuma modu, okuma sonu modu, yazma modu, yazma sonu modu.

- Bekleme modu: ZYNQ tarafından görüntü verileri gelmeden önce sürekli veri gelme durumunun kontrol edildiği durumdur. 0. Adresten veri okur ve 0xA değerine eşit olduğunda okuma moduna geçilir.
- Okuma modu: ZYNQ tarafından yazılan veriler okunur ve belirlenen arraye atanır. Okuma tamamlandığında okuma modu sonu moduna geçilir.
- Okuma modu sonu: 0. Adrese veri okunması tamamlandığını belirtmek için 0xB yazılır. Yazma moduna geçilir.
- Yazma modu: Burada arraye atanan veriler filtrelenerek gönderilir. Filtreleme işlemi her elemanın gönderilmesinden önce yapılır. Yazma tamamlandığında yazma sonu moduna geçilir.
- Yazma sonu modu: Filtreleme işlemi bittiğini belirtmek için 0. Adrese 0xC yazılır ve bekleme moduna geçilir.

Kullanılan veri aktarım yöntemleri:

- Kullanıcı bilgisayarı ve Zedboard arasında UART iletişimi.
- Zynq işlemci ile FPGA iletişimi için Block Memory.

NOT: Bu rapordaki adreslemeler byte addressing gösterimindedir.

YÖNTEM

1. Kullanıcının bilgisayarının seri kanalından gönderdiği veri ZYNQ işlemci ile okunur ve block memorye yazılır:

Bu işlemde uart verisi seri kanaldan okunur ve unsigned 8 bitlik arraye yazılır. Arrayden type casting yapılarak brama yazılır. Yazma işlemi bram base address + 4 ten başlar. Bütün veriler yazıldığında base address'e 0xA verilir ve yazma tamamlanır.

2. FPGA tarafından görüntü verileri block memory den okunur ve kaydedilir.

FPGA filtreleme işlemi başlamadan önce bekleme modundadır ve sürekli 0. Adresi okumaktadır. Eğer buradan okunan değer ZYNQ tarafından veri yazımı bittiğini gösteren değere eşitse FPGA okuma moduna geçer.

Okuma modu:

0'dan başlayan bir sayaç tutulur. Her cycle'da 1 veri okunur. Veri okunduktan sonra sayaç 1 arttırılır.

BRAM adres girişine (Sayaç değeri + 1)x4 adresi verilir. BRAM write enable girişine b(0000) verilir. BRAM dout çıkışı görüntü verilerinin okunması için ayrılmış olan arraye atanır. Bütün veriler okunduğunda FPGA okuma sonu moduna geçer. 0. Adrese FPGA tarafından okumanın tammalandığını gösteren 0xB değeri atanır.

3. FPGA tarafından veriler fitrelenerek block memorye yazılır.

Yazılacak veriye sıra geldiğinde filtreleme işlemi yapılır. Verinin indexi i olarak kabul edilirse, filtreleme sonucu:

$$[\text{image}(i-1) + 2*\text{image}(i) + \text{image}(i+)]/4$$

olarak bulunur. Çarpma ve bölme işlemleri kaydırarak yapılmaktadır, bu nedenle hızlıdır.

Array olarak tutulan veri 50x50 matris olarak düşünüldüğünde sağ veya sol taraftan renk gelmiyorsa 0 olarak kabul edilmelidir. Bu kontrol i%50 değerine bakarak anlaşılır.

Pixelin değeri bu şekilde bulunur ve BRAM din çıkışına yazılır. Tüm verilerin yazımı tamamlandığında yazma sonu moduna geçilir. Yazma işleminin tamamlandığını ZYNQ tarafından anlayabilmek için 0. Adrese 0xC yazılır. FPGA tarafında filtreleme işlemi bitmiş olur ve tekrar bekleme moduna geçer. Böylece yeni veri gönderildiğinde filtreleme işlemi baştan başlar.

4. Filtrelenen resim ZYNQ tarafından okunur ve seri kanaldan gönderilir.

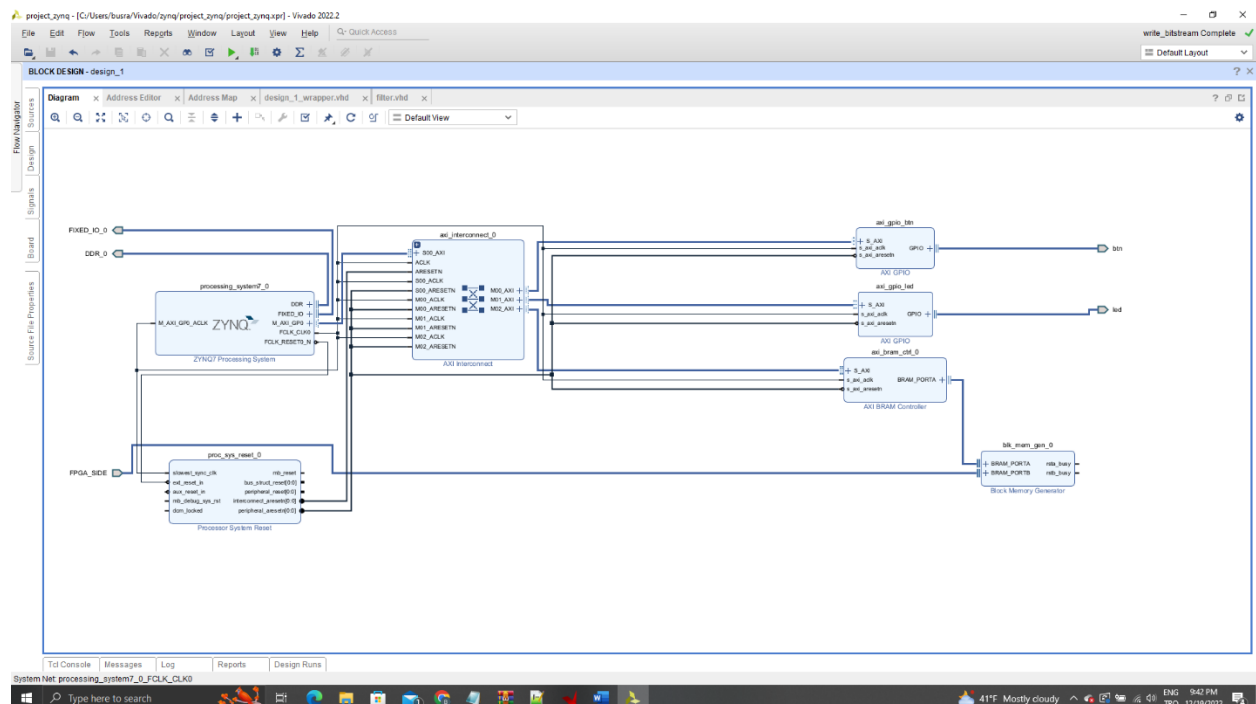
BRAM base adresindeki değerin 0xC olduğu ZYNQ tarafından gözlemlenir ve block memory'den uart ile kullanıcıya gönderilir. Burada ara değişken kullanmadan her seferinde 1 byte veri göndererek işlem yapıldı.

NOTLAR

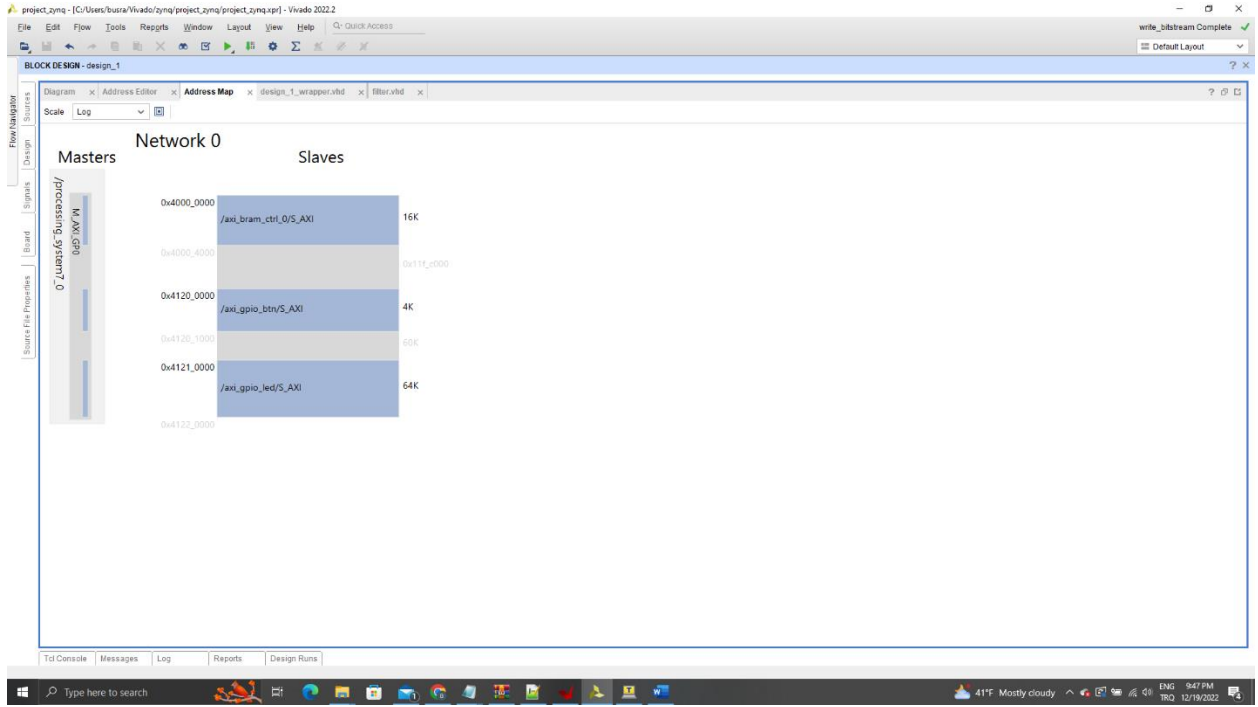
Verileri seri kanaldan direk block ram'e yazma işlemi pointer uyumsuzluğundan dolayı yapılamadı. Uart fonksiyonları çok sayıda veri okumak için aldıkları pointer üzerinde u8 pointer gibi işlem yapmakta, bu yüzden bram için yazılacak adres verilerek 1'er 1'er veri okunabilmekte ancak 2500 veri için bu işlem yapılamadı. Bu yüzden direk block rame yazmak yerine u8 tipinde array kullanıldı, bu tasarımın yavaşlamasına neden oldu. Bu yöntem yerine 8 bitlik verileri 32 bitlik block ram parçasına 4'erli yazma ve okuma yapılabilirdi, tasarımı çok hızlandıırırdı.

EKLER

Block dizayn:



Adres haritası:



BRAM hafızası 10K'dan büyük olmalı çünkü 2500 adet 4 baytlık veri okunup yazılmakta, 10K + 1 bayt hafıza gerekmektedir.