

Xilinx Fpga Comport-Usb Port Üzerinden Senkron Haberleşme Yazılımı Tasarımı

Salih Erġin, Murat Aydemir, Korhan Cengiz*, S. Bahadır Karuv
Trakya Üniversitesi
Elektrik-Elektronik Mühendisliđi Bölümü
Edirne
salihergin66@hotmail.com, murataydemir94@gmail.com, bkaruv@trakya.edu.tr

*Trakya Üniversitesi
Elektrik-Elektronik Mühendisliđi Bölümü
Edirne
korhancengiz@trakya.edu.tr

Özet: Bu bildiriye, Xilinx Fpga ile bilgisayar arasında asenkron veri almayı sağlamak için gerçekleştiren yazılım sunulmaktadır. Fpga tarafından alınan veriler, Fpga üzerine gömülü 8 adet yedi-parçalı led görüntüleyicilerde izlenmektedir. Tasarlanan haberleşme arayüzü saniyelik görüntü sayısı (FPS) yazılımı ve 7-parçalı led görüntüleyici için kod çözücü (decoder) yazılımını içermektedir.

Abstract: In this article, a realized software which provides asynchronous data reception between Xilinx FPGA and computer is presented. The data which are received from fpga are monitored on seven-segment led display that is embedded on fpga. The designed communication interface include frame per second (FPS) software and decoder software for seven-segment led display.

1. Giriş

Haberleşme sistemleri veri iletimi açısından eş zamanlı (senkron) ve eş zamanlı olmayan (asenkron) sistemler olarak ikiye ayrılır [1]. Eş zamanlama sistemlerinde veri ve eş zamanlama bilgisinin iletimi için iki ayrı kanal kullanılır: zamanlama kanalı ve veri kanalı. Zamanlama kanalı saat darbelerini alıcıya göndermek için kullanılır. Veri kanalı ise veri iletimi için kullanılır. Saat darbesinin alınması üzerine alıcı, veri kanalını okuyarak o anda kanal üzerinde bulunan bit değerini yakalar. Veri kanalı bir sonraki saat darbesi gelene kadar okunmaz. Verici hem veri hem de zamanlama darbelerini gönderdiğinden, alıcı veri kanalını yalnızca verici tarafından belirtildiğinde okur ve böylelikle eş zamanlama garanti altına alınır.

Eş zamanlı olmayan sistemlerde [1] ayrı bir zamanlama kanalı kullanılmaz. Verici ve alıcı önceden bir veri gönderme hızında anlaşmış olmalıdır. Alıcı içinde bulunan hassas yerel osilatör, vericinin içinde bulunan saat sinyalinin belirli bir yüzdesine eşit dahili bir saat işareti üretir. Veri paketinde, alıcıya, iletimin başladığını bildirmek için 0 (başlangıç biti) biti bulunur. Bu başlangıç biti, verici tarafından alıcıya veri paketi ile birlikte gönderilmektedir. Başlangıç biti alıcı içinde gerekli saat darbelerini üretecek dahili zamanlayıcıyı tetikler. Başlangıç bitini takiben sekiz bitlik mesaj alıcıya gönderilir. Bu gönderilen sekiz bitlik mesaj, ASCII biçimindedir ve hızı sembol iletim hızı (baud rate) mertebesinde dir. Veri paketinde en son alıcıya iletimin bittiğini belirtmek için 1 (sonlandırma biti) biti gönderilerek, iletim sonlandırılır.

Bu çalışmada, Xilinx Fpga [2] ile bilgisayar arasında eş zamanlı olmayan veri alma işlemini gerçekleştirmek için alıcı taraf olan Xilinx Fpga' de kullanılmak üzere bir yazılım tasarlanması amaçlanmıştır. Mevcut seri haberleşme sistemlerinde, iki tarafı 9 pinli VGA kabloları kullanılmaktadır. Bu çalışmada ise, VGA kablosunun bilgisayara bağlanan kısmında comport – usb çevirici kablosu kullanılması amaçlanmıştır. Bu çevirici kablonun kullanılmasıyla, sistemin yeni nesil bilgisayarlar ile uyumlu çalışmasının sağlanması amaçlanmıştır.

Tasarlanan yazılımın yapısı, kullanılan yöntem ve yazılımlar, bölüm 2' de anlatılmaktadır. Bölüm 3' de elde edilen simülasyon sonuçları verilmiş ve bu sonuçlar detaylı bir şekilde analiz edilmiştir.

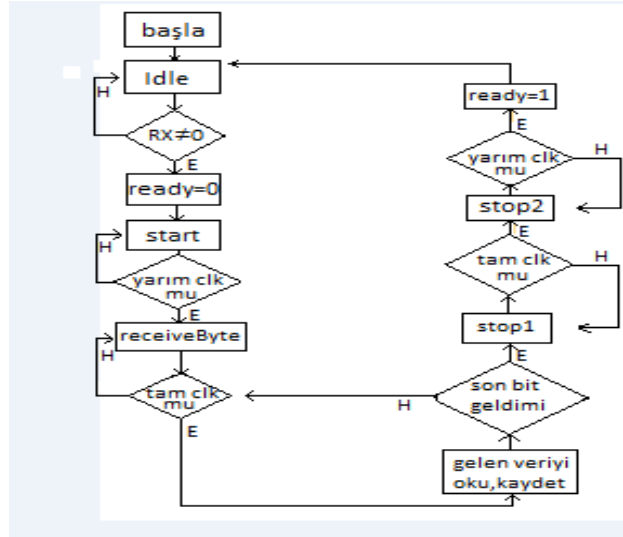
2. Tasarlanan Yazılım

Tasarlanan yazılım, veri alma kodu, saniyelik görüntü sayısını (FPS) belirleyen kod ve yedi-parçalı led görüntüleyicisinin kod çözücü (decoder) kodundan oluşmaktadır.

2.1 Veri Alma Kodu

Veri alma kodu, Xilinx FPGA ile bilgisayar arasında belirli bir veri gönderme hızıyla eş zamanlı olmayan seri haberleşme algoritmasını gerçekleştirmek için tasarlanmıştır. Veri alma kodu, Idle, Start, receiveByte, stop1, ve stop2 parçalarını içeren durum makinasından (state machine) oluşmaktadır.

Veri alma kodunun tasarımında, tam saat sinyali ve yarım saat sinyali tanımlanmıştır. Amaç, haberleşen cihazdan gelen verilerin (start biti, veri bitleri ve stop biti) her birini, saat sinyalinin yükselen kenarı, bitlerin orta noktasında yakalamaktır. Start ve stop2 durumlarında yarım saat sinyali, receiveByte ve stop1 durumlarında tam saat sinyali kullanılmıştır. Bu sayede her bitin orta noktasında saat sinyalinin yükselen kenarının üretilmesi mümkün hale gelmektedir. Tasarlanan kodun akış diyagramı şekil-1' de gösterilmiştir.



Şekil-1 Veri alma kodu akış diyagramı

Idle durumu, hattın veri almayı beklediği "1" durumudur. Xilinx Fpga cihazının veri tabanında veri almak için RX kaydedicisi (register) bulunmaktadır [3]. Haberleşme başladığı anda hat meşgul duruma düşmekte ve RX kaydedicisi gelen veriyi kaydetmektedir. Haberleşme hattından veri geldiğinde hattı meşgul duruma çekmek için kodun içinde tanımlanmış ready biti "0" olarak atanmıştır. Veri geldiği anda bir sonraki "Start" durumuna geçilmektedir.

Start durumunda, veri gönderen cihaz haberleşmenin başladığını alıcı tarafa start biti "0" göndererek bildirmektedir. Alıcı taraf saat sinyalinin yarısı kadar süre beklemektedir. Amaç start bitinin orta noktasında saat sinyalinin yükselen kenarını yakalamaktır. Saat sinyali geldiği anda alıcı cihaz start bitini okur ve bir sonraki "receiveByte" durumuna geçer.

ReceiveByte durumunda, kod içerisinde, gelen 8 bitlik veriyi alması için "data" kaydedicisi ve bit sayısını tutması için "bitCounter" kaydedicisi tanımlanmıştır. ReceiveByte durumunda tam saat sinyalinin yükselen kenarı beklenmektedir. Amaç, okunan her bir veri bitinin orta noktasında saat sinyalinin yükselen kenarını yakalamaktır. Bir önceki start durumunda saat sinyalinin yarısı kadar süre beklenerek bu şart sağlanmıştır. Bir sonraki her bir bit için bu koşul sağlanmaya çalışılır. Bu nedenle tam bir saat sinyali kadar süre beklenir. Saat sinyali geldiğinde, haberleşen cihazdan gelen veriler "data" kaydedicisine kaydedilir. Haberleşen cihazdan gelen 8 bitin hepsini almak için "bitCounter" 8'e kadar saydırılır. Burada, "bitCounter" 8 olduğunda sıfırlanarak, bir sonraki "stop1" durumuna geçilir. Stop1 durumunda, tam bir saat sinyalinin yükselen kenarı kadar beklenir. Burada hedeflenen, haberleşmeyi sonlandıracak "stop" bitinin orta noktasında saat sinyalinin yükselen kenarının

gelmesinin sağlanmasıdır. Haberleşmeyi sonlandıran stop biti "1" geldiğinde bir sonraki "stop2" durumuna geçilir.

Stop2 durumunda, saat sinyalinin yarısı kadar süre beklenerek kodun başında kullanılan yarım saat sinyali süresi tamamlanmış olur. Bu sayede bir sonraki iletimde saat sinyalinin eş zamanlı olarak çalışması sağlanır. Yarım saat sinyalinin yükselen kenarı geldiğinde "ready" biti "1" olarak ayarlanmakta, cihazın haberleşme hattı bir sonraki veri alma işlemine hazır konuma getirilmektedir. Sonra tasarlanan sistem, yeni bir haberleşmenin başlaması için ilk durum olan "Idle" durumuna geri dönmektedir.

2.2 Test Kodu

Tasarlanan yazılımın Test Kodu kısmında, veri alma kodunu simülasyon ortamında gerçeklemek için, hexadecimal A5 verisi gönderilmiştir. Gönderilen hexadecimal A5 verisi, 1010_0101 formunda bir veridir. Bu verinin Xilinx Fpga tarafından saat frekansının yükselen kenarlarında alınıp alınmadığı gözlemlenmiş ve başarılı bir şekilde verinin alındığı tespit edilmiştir. Ayrıca, test kodunda, ready bitinin ve saat frekansının doğruluğu da kontrol edilmektedir.

2.3 Saniyelik Görüntü Sayısı (FPS) Kodu

Gönderilen verilerin ASCII karşılığının Xilinx Fpga üzerine gömülü 8 adet yedi-parçalı led görüntüleyicide görüntülenmesi için saniyelik görüntü sayısı kodu tasarlanmıştır. Görüntüleme için, FPS oranı 60 olarak belirlenmiştir. FPS kodunda, 3 bitlik "displayCounter" kaydedicisi tanımlanmıştır. Xilinx Fpga çalışma frekansı, FPS oranına ve yedi-parçalı led görüntüleyici sayısına bölünerek yeni bir saat frekansı oluşturulmuştur. Bu saat frekansının yardımıyla da yeni bir saat sinyali üretilmiştir. Saat sinyalinin her bir yükselen kenarında "displayCounter" bir artırılmıştır. "displayCounter" değerine göre, yedi-parçalı led görüntüleyicisinin saniyelik görüntü sayısı oranı ayarlanmış olmaktadır.

2.4 Yedi-Parçalı Led Görüntüleyici Kod Çözücü (Decoder) Kodu

Gönderici taraftan gelen veriler yedi-parçalı led görüntüleyici üzerinde ASCII karşılığı ile görüntülenir. Bu nedenle her bir yedi-parçalı led görüntüleyici için 0-F sayısı kadar karakter için kod çözücü kodu hazırlanmıştır. Gelen her bir veri çözümlendikten sonra Xilinx Fpga veritabanında yedi-parçalı led görüntüleyici için ayrılmış olan "SEG" kaydedicisine gönderilir.

3. Simülasyon Sonuçları

Tasarlanan yazılımın, kullanılan Verilog HDL programında test kodu yazılarak simülasyon ortamında sonuçları gözlemlenmiştir. Bilgisayar ile Xilinx Fpga arasında seri haberleşme arayüzü olan PUTY programı kullanılmıştır. Xilinx Fpga cihazı 100MHz çalışma frekansına sahiptir. Saat frekansının her bir periyodu 10ns olarak hesaplanmıştır [4]. Eş zamanlı olmayan haberleşme yapıldığı için karakter gönderme hızı 9600 baud / sn olarak belirlenmiştir. Kullanılan Verilog HDL programının simülasyon sonuçları incelenmiştir. Test kodunda ise hexadecimal A5 verisinin (1010_0101) gönderilmesi planlanmıştır. Simülasyon sonuçlarında yazılan test kodunun yardımıyla, alınan her bir veri, ready bitinin durumları ve saat sinyalinin yükselen kenarlarının bitlerin orta noktasına gelip gelmediği kontrol edilebilmektedir. Alınan sonuçlarda bu üç yapının eşzamanlı ve başarılı bir şekilde çalıştığı gözlemlenmiştir. Seri haberleşme arayüzü olan PUTY programı sayesinde bilgisayar klavyesindeki karakterler Xilinx Fpga cihazına gönderilmiştir. Gönderilen her bir karakterin ASCII karşılığı cihaz üzerinde bulunan 8 adet yedi-parçalı led görüntüleyicide başarılı bir şekilde görüntülenmiştir. İlk karakter geldikten sonra takip eden karakterlerin gönderilmesinde sorun yaşanmadığı tespit edilmiştir.

5. Kaynaklar

- [1]. 3. E. Curry, "Message-Oriented Middleware", Middleware for Communications, Q.H. Mahmoud, ed., John Wiley &, Sons, 2004
- [2]. M. Straka, J. Tobola ve Z. Kotasek, "Checker Design for On-line Testing of Xilinx FPGA Communication Protocols," Proceedings 22nd IEEE International Symposium on Defect and Fault-Tolerance in VLSI Systems, Rome, 2007, s. 152-160.
- [3]. G. J. Minden et al., "KUAR: A Flexible Software-Defined Radio Development Platform," 2007 2nd IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks, Dublin, 2007, pp. 428-439.
- [4]. E. Lemoine ve D. Merceron, "Run time reconfiguration of FPGA for scanning genomic databases," Proceedings IEEE Symposium on, Napa Valley, CA, 1995, s. 90-98.