YILDIZ TEKNİK ÜNİVERSİTESİ
ELEKTRİK-ELEKTRONİK FAKÜLTESİ / ELEKTRONİK VE HABERLEŞME MÜH. BÖLÜMÜ

Öğrencinin Adı Soyadı: Alirıza BİLİR	**Öğrenci No: 18014125**			**İmza: A.B**	
Dersin Adı: Programlanabilir Lojik Devreler ile Prototip Geliştirme	**Tarih/Saat: 5.06.2024 – 10:00**			**Sınav süresi: 120 dak**	
Sınav Türü:	**Vize 1**	**Vize 2**	**Mazeret**	**Final X**	**Bütünleme**
Unvan Ad-Soyad: Prof. Dr. Burcu ERKMEN (Ders Yürütücüsü)					

Mouse PS/2 Protokolü ile Ledleri Yakma Projesi

Proje Tanımı

Bu projede, PS/2 protokolü kullanarak bir fareden veri alıp, bu veriyi işleyerek LED'leri kontrol eden bir sistem tasarladım. Amaç, fare hareketlerini ve buton tıklamalarını algılayarak 16 bit ledleri sağ ve sola doğru hareket ettirmektir. Bu raporda, sistemin nasıl çalıştığını, kullanılan VHDL kodlarının işlevlerini ve veri işleme süreçlerini detaylı bir şekilde açıklayacağız.

PS/2 Protokolü

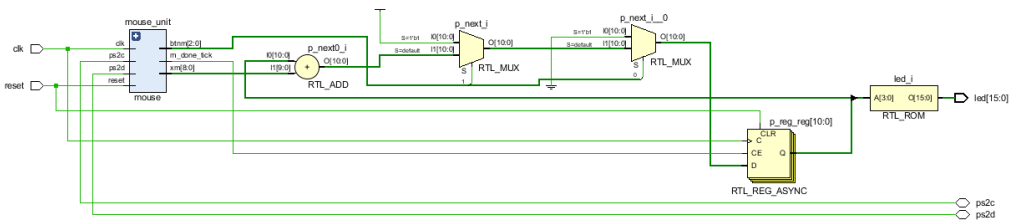
PS/2, eski bilgisayar sistemlerinde yaygın olarak kullanılan bir haberleşme protokolüdür. Klavye ve fare gibi cihazlar için geliştirilmiş olan bu protokol, veri iletimi için seri haberleşme kullanır. Veri, PS/2 veri hattı (ps2d) ve saat hattı (ps2c) üzerinden iletilir. PS/2 fareler, fare hareketlerini ve buton durumlarını içeren paketler halinde veri gönderir. Bu protokolde 3 paket halinde veriler gönderilmektedir. Bu veriler Şekil 1'de görüldüğü üzere 8'er bit artı olarak da her pakete 1 start, 1 parity, 1 stop biti de ayrıca eklenmektedir. PS2 protokolünde FPGA'in verileri okuması için öncelikle Mouse'a F4 kodu göndererek verileri okumaya hazır yani stream moduna geçmesi bildirilir.

byte 1	y_v	x_v	y_8	x_8	1	m	r	l
byte 2	x_7	x_6	x_5	x_4	x_3	x_2	x_1	x_0
byte 3	y_7	y_6	y_5	y_4	y_3	y_2	y_1	y_0

Şekil 1. Mouse Data Paketi Formatı

Vivado Çıktıları ve Açıklamaları

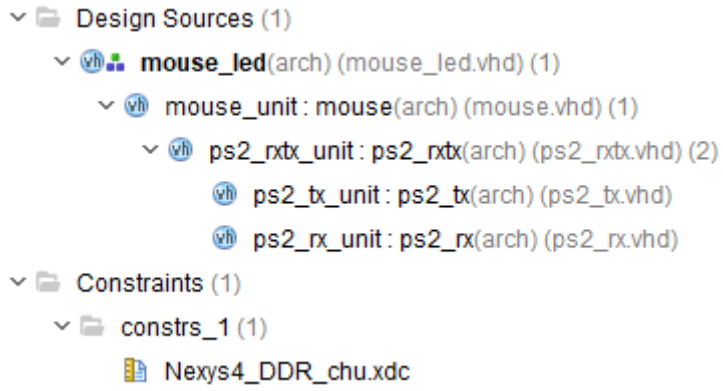
Sistem Şeması



Şekil 2. Devre Şeması

Vhdl kodları ile kurulan şema Şekil 2'de gözükmektedir.

Sistem Hiyerarşisi



Şekil 3. Modül Hiyerarşisi

Sisteme ait kodlar aşağıdaki alanda açıklamaları ile anlatılacaktır. Bu VHDL kodlarının hiyerarşisi Şekil 3'de bulunmaktadır.

Modüller

Mouse_led.vhd (Top Modül)

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;
entity mouse_led is
  port (
    clk, reset: in  std_logic;
    ps2d, ps2c: inout std_logic;
    led: out std_logic_vector(15 downto 0)
  );
end mouse_led;

architecture arch of mouse_led is
  signal p_reg, p_next: unsigned(10 downto
0);
  signal xm: std_logic_vector(8 downto 0);
  signal btnm: std_logic_vector(2 downto 0);
  signal m_done_tick: std_logic;

begin
  -- instantiation
  mouse_unit: entity work.mouse(arch)
    port map(clk=>clk, reset=>reset,
          ps2d=>ps2d, ps2c=>ps2c,
          xm=>xm, ym=>open, btnm=>btnm,
          m_done_tick=>m_done_tick);
  -- register
  process (clk, reset)
  begin
    if reset='1' then
      p_reg <= (others=>'0');
    elsif (clk'event and clk='1') then
      p_reg <= p_next;
    end if;
  end process;
  -- counter
  p_next <= p_reg when m_done_tick='0' else
          "00000000000" when btnm(0)='1' else
--left button
          "11111111111" when btnm(1)='1' else
--right button
          p_reg + unsigned(xm(8) & xm);
    with p_reg(10 downto 7) select
      led <= "1000000000000000" when "0000",
          "0100000000000000" when "0001",
          "0010000000000000" when "0010",
          "0001000000000000" when "0011",
          "0000100000000000" when "0100",
          "0000010000000000" when "0101",
          "0000001000000000" when "0110",
          "0000000100000000" when "0111",
          "0000000010000000" when "1000",
          "0000000001000000" when "1001",
          "0000000000100000" when "1010",
          "0000000000010000" when "1011",
          "0000000000001000" when "1100",
          "0000000000000100" when "1101",
          "0000000000000010" when "1110",
          "0000000000000001" when others;
end arch;
```

Top Modülün İşleyişi

1. **Mouse Ünitesi Entegrasyonu:**
    - mouse_unit adıyla bir fare modülü instantiate edilir ve bu modül, fareden gelen veriyi işler. Bu veriler xm ve btnm sinyallerine iletilir.
    - m_done_tick sinyali, fareden gelen veri paketinin tamamlandığını gösterir.
2. **Sayaç İşlemleri:**
    - p_reg, fare hareketlerine ve buton tıklamalarına göre güncellenir. Eğer sol buton tıklanmışsa sayaç sıfırlanır. Sağ buton tıklanmışsa sayaç maksimum değere ayarlanır. Aksi takdirde, sayaç fare hareket verisine (xm) göre artırılır veya azaltılır.
3. **LED Kontrolü:**
    - p_reg sinyalinin yüksek bitlerine (10 ila 7 bitleri) göre LED'ler kontrol edilir. Bu bitler, LED'lerin belirli bir desenle yanmasını sağlar.

İşleyiş Detayları

- **PS/2 Veri Alma:** Fareden gelen veriler, ps2d ve ps2c hatları üzerinden alınır ve mouse_unit modülü tarafından işlenir.
- **Veri İşleme:** mouse_unit modülü, fare hareketi ve buton durumlarını ayrıştırır ve xm ve btnm sinyallerine iletir.
- **Sayaç ve LED Güncelleme:**
    - m_done_tick sinyali, fareden yeni bir veri paketi geldiğini gösterir ve sayaç (p_reg) güncellenir.
    - Sol buton tıklanmışsa sayaç sıfırlanır, sağ buton tıklanmışsa sayaç maksimum değere ayarlanır.
    - Sayaç değeri, fare hareket verisi (xm) ile güncellenir.
    - Sayaç değeri (özellikle yüksek bitler) kullanılarak LED'ler belirli bir desende yanar.
    -

Bu modül, fare hareketlerini ve buton tıklamalarını kullanarak LED'lerin durumunu dinamik olarak kontrol eder. Fare hareketlerine göre sayaç değeri değiştirilir ve bu değer kullanılarak LED'ler belirli bir desenle yanar.

mouse.vhd modülün İşleyişi

1. **PS/2 İletişimi:**
    - ps2_rxtx_unit modülü, PS/2 fare ile iletişim kurar. Bu modül, fareden gelen veriyi alır ve işlemler için gerekli sinyalleri (rx_done_tick, tx_done_tick, rx_data) sağlar.
    - wr_ps2 sinyali, fareye veri gönderme işlemi için kullanılır.
2. **Durum ve Veri Kayıtları:**
    - state_reg ve state_next: FSM durumu ve bir sonraki durumu. FSM, veri paketlerini almak ve işlemek için farklı durumlarda çalışır.
    - x_reg, y_reg, btn_reg: x ve y ekseni hareket verileri ve buton durumları.
    - x_next, y_next, btn_next: Bir sonraki durumda kullanılacak x ve y ekseni verileri ve buton durumları.
3. **FSM İşleyişi:**
    - **init1:** PS/2 veri yazma işlemini başlatır (wr_ps2 <= '1').
    - **init2:** Veri gönderimi tamamlanana kadar bekler (tx_done_tick sinyali kontrol edilir).
    - **init3:** Acknowledge paketinin alınmasını bekler (rx_done_tick sinyali kontrol edilir).
    - **pack1:** İlk veri paketinin alınmasını bekler. Alındığında x ve y ekseni verileri ve buton durumları güncellenir (rx_data kullanılarak).
    - **pack2:** İkinci veri paketinin alınmasını bekler. Alındığında x ekseni verileri güncellenir.

- **pack3:** Üçüncü veri paketinin alınmasını bekler. Alındığında y ekseni verileri güncellenir.
- **done:** Veri alım işleminin tamamlandığını belirtir (m_done_tick <= '1') ve tekrar pack1 durumuna geçer.

ps2_rxtx.vhd modülün İşleyişi

1. **PS/2 TX (Gönderim) Birimi:**
   - ps2_tx_unit isimli iç modül, PS/2 veri hattı üzerinden veri göndermekle sorumludur.
   - Bu birim, clk, reset, wr_ps2, din, ps2d, ps2c sinyallerini kullanarak veri gönderir.
   - tx_idle sinyali, TX biriminin boşta olup olmadığını belirler.
   - tx_done_tick sinyali, veri gönderiminin tamamlandığını belirtir.
2. **PS/2 RX (Alım) Birimi:**
   - ps2_rx_unit isimli iç modül, PS/2 veri hattı üzerinden veri almakla sorumludur.
   - Bu birim, clk, reset, tx_idle, ps2d, ps2c sinyallerini kullanarak veri alır.
   - rx_done_tick sinyali, veri alımının tamamlandığını belirtir.
   - dout sinyali, alınan veriyi taşır.

Algoritma

1. **Veri Gönderimi:**
   - wr_ps2 sinyali aktif hale getirildiğinde, ps2_tx_unit modülü, din girişindeki veriyi ps2d ve ps2c hatları üzerinden gönderir.
   - Gönderim işlemi tamamlandığında, tx_done_tick sinyali aktif hale gelir ve tx_idle sinyali boşta olduğunu belirtir.
2. **Veri Alımı:**
   - tx_idle sinyali aktif hale geldiğinde, ps2_rx_unit modülü, ps2d ve ps2c hatları üzerinden veri almaya başlar.
   - Alım işlemi tamamlandığında, rx_done_tick sinyali aktif hale gelir ve alınan veri dout sinyali üzerinden dışarıya iletilir.

ps2_tx.vhd algoritması ve İşleyişi

1. **Filtreleme ve Düşen Kenar Üretimi:**
   - filter_reg ve f_ps2c_reg sinyalleri, PS/2 saat sinyalinin düşen kenarını tespit etmek için kullanılır.
   - fall_edge sinyali, PS/2 saat sinyalinin düşen kenarını belirtir.
2. **Finite State Machine (FSM):**
   - idle Durumu:
     - Modül boşta bekler.
     - wr_ps2 sinyali aktif hale geldiğinde, gönderilecek veri (din) ve parite biti (par) b_next sinyaline yüklenir.
     - Durum rts durumuna geçer.
   - rts Durumu:
     - PS/2 saat sinyali (ps2c_out) sıfırlanır ve gönderim isteği sinyali (tri_c) aktif hale gelir.
     - Sayaç (c_reg) sıfıra ulaştığında, durum start durumuna geçer.
   - start Durumu:
     - Başlangıç biti (ps2d_out) sıfırlanır ve veri hattı (tri_d) aktif hale gelir.
     - PS/2 saat sinyalinin düşen kenarı tespit edildiğinde, durum data durumuna geçer.

- data Durumu:
  - Veri bitleri ve parite biti sırasıyla gönderilir.
  - Her düşen kenarda, b_reg sinyali kaydırılarak veri gönderimi gerçekleştirilir.
  - Tüm bitler gönderildiğinde, durum stop durumuna geçer.
- stop Durumu:
  - PS/2 veri hattı yüksek duruma getirilir.
  - PS/2 saat sinyalinin düşen kenarı tespit edildiğinde, durum idle durumuna geçer ve tx_done_tick sinyali aktif hale gelir.

3. **Üç Durumlu Tamponlar:**
   - ps2c ve ps2d sinyalleri, ps2c_out ve ps2d_out sinyalleri ile kontrol edilir.
   - tri_c ve tri_d sinyalleri aktif olduğunda, ps2c ve ps2d sinyalleri çıkış olarak ayarlanır, aksi takdirde yüksek empedans durumunda kalır ('Z').

ps2_rx.vhd Algoritması ve İşleyişi

1. **Filtreleme ve Düşen Kenar Üretimi:**
   - filter_reg ve f_ps2c_reg sinyalleri, PS/2 saat sinyalinin düşen kenarını tespit etmek için kullanılır.
   - fall_edge sinyali, PS/2 saat sinyalinin düşen kenarını belirtir.
2. **Finite State Machine (FSM):**
   - **idle Durumu:**
     - Modül boşta bekler.
     - PS/2 saat sinyalinin düşen kenarı ve rx_en sinyali aktif olduğunda, başlangıç biti b_next sinyaline kaydırılır ve bit sayısı (n_next) 9 olarak ayarlanır.
     - Durum dps durumuna geçer.
   - **dps Durumu:**
     - 8 veri biti, 1 parite biti ve 1 durdurma biti okunur.
     - Her düşen kenarda ps2d sinyali b_reg sinyaline kaydırılır.
     - Tüm bitler okunduğunda, durum load durumuna geçer.
   - **load Durumu:**
     - Veriyi yüklemek için ekstra bir saat sinyali beklenir.
     - Durum idle durumuna geri döner ve rx_done_tick sinyali aktif hale gelir.

Kodlar rar şeklinde rapora eklenerek teslim edilmiştir.

Alirıza BİLİR
18014125