Solution to Assignment 6
DRAM Controller

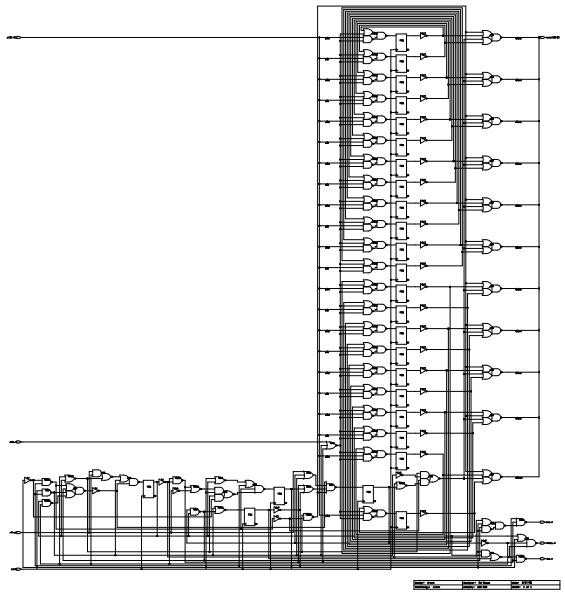
Question 1

The design is a simple state machine that goes through states S1-S4 for a normal read cycle and inserts states F2-F5 after S1 when a refresh cycle is required.

VHDL Code

```
-- Solution to ELEC 379 Assignment 6
-- Simple DRAM controller
-- Ed Casas, 98/3/19

library ieee ;
use ieee.std_logic_1164.all ;

entity dramc is
      port ( clk2, rfrsh, ads_n : in std_logic ;
         a : in std_logic_vector (21 downto 0) ;
         ready_n, ras_n, cas_n : out std_logic ;
         rama : out std_logic_vector (10 downto 0) ) ;
end ;


architecture rtl of dramc is
   -- states
   type statetype is (INIT,S1,R2,R3,R4,F2,F3,F4,F5) ;
   -- latched CPU address
   signal tmpa, nxttmpa : std_logic_vector (21 downto 0) ;
   -- next state
   signal state, nexts : statetype ;
begin

   -- compute next state
   process(state,rfrsh)
   begin
      case state is
         when S1 =>
            if rfrsh = '1' then
               nexts <= F2 ;
            else
               nexts <= R2 ;
            end if ;
         when R2 => nexts <= R3 ;
         when R3 => nexts <= R4 ;
         when R4 => nexts <= S1 ;
         when F2 => nexts <= F3 ;
         when F3 => nexts <= F4 ;
         when F4 => nexts <= F5 ;
         when F5 => nexts <= R2 ;
         when others => nexts <= S1 ;
      end case ;
   end process ;

   -- compute next value of address latch
   nxttmpa <= a when state = S1 and ads_n = '0' else tmpa ;

   -- READY*, CAS*, and RAS* outputs
   process(state)
   begin
      case state is
         when S1 => ready_n <= '0' ; ras_n <= '1' ; cas_n <= '1' ;

         when R2 => ready_n <= '1' ; ras_n <= '0' ; cas_n <= '1' ;
         when R3 => ready_n <= '1' ; ras_n <= '0' ; cas_n <= '0' ;
         when R4 => ready_n <= '0' ; ras_n <= '0' ; cas_n <= '0' ;

         when F2 => ready_n <= '1' ; ras_n <= '1' ; cas_n <= '0' ;
         when F3 => ready_n <= '1' ; ras_n <= '0' ; cas_n <= '0' ;
         when F4 => ready_n <= '1' ; ras_n <= '1' ; cas_n <= '1' ;
         when F5 => ready_n <= '1' ; ras_n <= '1' ; cas_n <= '1' ;
         when others =>
                    ready_n <= '1' ; ras_n <= '1' ; cas_n <= '1' ;
      end case ;
   end process ;

   -- ram address mux
   rama <=
      tmpa(21 downto 11) when state = R2 else
      tmpa(10 downto  0) when ( state = R3 ) or ( state = R4 ) else
      "00000000000" ;

   -- state register
   process(clk2)
   begin
      if clk2'event and clk2='1' then
         state <= nexts ;
      end if ;
   end process ;

   -- CPU address register
   process(clk2)
   begin
      if clk2'event and clk2='1' then
         tmpa <= nxttmpa ;
      end if ;
   end process ;

end rtl ;
```

Schematics



Simulation Log

```
Test Input: 0 clk2=1 rfrsh=0 ads_n=1 a=00000000000000000000000
    Output: ready_n=0 ras_n=1 cas_n=1 rama=00000000000
  Required: ready_n=0 ras_n=1 cas_n=1 rama=00000000000
Test Input: 1 clk2=0 rfrsh=0 ads_n=0 a=0001001000110101000101011O
    Output: ready_n=0 ras_n=1 cas_n=1 rama=00000000000
  Required: ready_n=0 ras_n=1 cas_n=1 rama=00000000000
Test Input: 2 clk2=1 rfrsh=0 ads_n=0 a=0001001000110101000101011O
    Output: ready_n=1 ras_n=0 cas_n=1 rama=00100100011
  Required: ready_n=1 ras_n=0 cas_n=1 rama=00100100011
Test Input: 3 clk2=0 rfrsh=0 ads_n=0 a=0001001000110101000101011O
    Output: ready_n=1 ras_n=0 cas_n=1 rama=00100100011
  Required: ready_n=1 ras_n=0 cas_n=1 rama=00100100011
Test Input: 4 clk2=1 rfrsh=0 ads_n=0 a=0001001000110101000101011O
    Output: ready_n=1 ras_n=0 cas_n=0 rama=10001010110
  Required: ready_n=1 ras_n=0 cas_n=0 rama=10001010110
Test Input: 5 clk2=0 rfrsh=0 ads_n=1 a=00000000000000000000000
    Output: ready_n=1 ras_n=0 cas_n=0 rama=10001010110
  Required: ready_n=1 ras_n=0 cas_n=0 rama=10001010110
Test Input: 6 clk2=1 rfrsh=0 ads_n=1 a=00000000000000000000000
    Output: ready_n=0 ras_n=0 cas_n=0 rama=10001010110
  Required: ready_n=0 ras_n=0 cas_n=0 rama=10001010110
Test Input: 7 clk2=0 rfrsh=0 ads_n=1 a=00000000000000000000000
    Output: ready_n=0 ras_n=0 cas_n=0 rama=10001010110
  Required: ready_n=0 ras_n=0 cas_n=0 rama=10001010110
Test Input: 8 clk2=1 rfrsh=0 ads_n=1 a=00000000000000000000000
    Output: ready_n=0 ras_n=1 cas_n=1 rama=00000000000
  Required: ready_n=0 ras_n=1 cas_n=1 rama=00000000000
Test Input: 9 clk2=0 rfrsh=1 ads_n=0 a=0001001000110101000101011O
    Output: ready_n=0 ras_n=1 cas_n=1 rama=00000000000
  Required: ready_n=0 ras_n=1 cas_n=1 rama=00000000000
Test Input: 10 clk2=1 rfrsh=1 ads_n=0 a=0001001000110101000101011O
    Output: ready_n=1 ras_n=1 cas_n=0 rama=00000000000
  Required: ready_n=1 ras_n=1 cas_n=0 rama=00000000000
Test Input: 11 clk2=0 rfrsh=1 ads_n=0 a=0001001000110101000101011O
    Output: ready_n=1 ras_n=1 cas_n=0 rama=00000000000
  Required: ready_n=1 ras_n=1 cas_n=0 rama=00000000000
Test Input: 12 clk2=1 rfrsh=1 ads_n=0 a=0001001000110101000101011O
    Output: ready_n=1 ras_n=0 cas_n=0 rama=00000000000
  Required: ready_n=1 ras_n=0 cas_n=0 rama=00000000000
Test Input: 13 clk2=0 rfrsh=0 ads_n=1 a=00000000000000000000000
    Output: ready_n=1 ras_n=0 cas_n=0 rama=00000000000
  Required: ready_n=1 ras_n=0 cas_n=0 rama=00000000000
Test Input: 14 clk2=1 rfrsh=0 ads_n=1 a=00000000000000000000000
    Output: ready_n=1 ras_n=1 cas_n=1 rama=00000000000
  Required: ready_n=1 ras_n=1 cas_n=1 rama=00000000000
Test Input: 15 clk2=0 rfrsh=0 ads_n=1 a=00000000000000000000000
    Output: ready_n=1 ras_n=1 cas_n=1 rama=00000000000
  Required: ready_n=1 ras_n=1 cas_n=1 rama=00000000000
Test Input: 16 clk2=1 rfrsh=0 ads_n=1 a=00000000000000000000000
    Output: ready_n=1 ras_n=1 cas_n=1 rama=00000000000
  Required: ready_n=1 ras_n=1 cas_n=1 rama=00000000000
Test Input: 17 clk2=0 rfrsh=0 ads_n=1 a=00000000000000000000000
    Output: ready_n=1 ras_n=1 cas_n=1 rama=00000000000
  Required: ready_n=1 ras_n=1 cas_n=1 rama=00000000000
Test Input: 18 clk2=1 rfrsh=0 ads_n=1 a=00000000000000000000000
    Output: ready_n=1 ras_n=0 cas_n=1 rama=00100100011
  Required: ready_n=1 ras_n=0 cas_n=1 rama=00100100011
Test Input: 19 clk2=0 rfrsh=0 ads_n=1 a=00000000000000000000000
    Output: ready_n=1 ras_n=0 cas_n=1 rama=00100100011
  Required: ready_n=1 ras_n=0 cas_n=1 rama=00100100011
Test Input: 20 clk2=1 rfrsh=0 ads_n=1 a=00000000000000000000000
    Output: ready_n=1 ras_n=0 cas_n=0 rama=10001010110
  Required: ready_n=1 ras_n=0 cas_n=0 rama=10001010110
Test Input: 21 clk2=0 rfrsh=0 ads_n=1 a=00000000000000000000000
    Output: ready_n=1 ras_n=0 cas_n=0 rama=10001010110
  Required: ready_n=1 ras_n=0 cas_n=0 rama=10001010110
Test Input: 22 clk2=1 rfrsh=0 ads_n=1 a=00000000000000000000000
    Output: ready_n=0 ras_n=0 cas_n=0 rama=10001010110
  Required: ready_n=0 ras_n=0 cas_n=0 rama=10001010110
Test Input: 23 clk2=0 rfrsh=0 ads_n=1 a=00000000000000000000000
    Output: ready_n=0 ras_n=0 cas_n=0 rama=10001010110
  Required: ready_n=0 ras_n=0 cas_n=0 rama=10001010110
```