For:reese
Printed on:Mon, Mar 22, 1999 08:53:05

Document:Test 1
Last saved on:Tue, Feb 3, 1998 15:18:06

Test #1 – EE 4743 – Spring 1998 My Name is: ______________________________

You have **50 minutes** to complete this test. Good luck!

1. (30 pts.) Answer the following questions.

Possible answers: PAL (PLD), CPLD, FPGA, Gate Array, Standard Cell, Full Custom

a. Which implementation technology has the lowest initial cost? That is, which is the cheapest if you planned to buy only one part?

b. Which technology (or technologies) involve the creation of *completely* custom chips (all mask layers designed from scratch)?

c. Which technology has more predictable propagation delay—PAL/PLD or FPGA?

d. Which implementation technologies could you program with a desktop programming device (a programming device small enough to fit on your desk)?

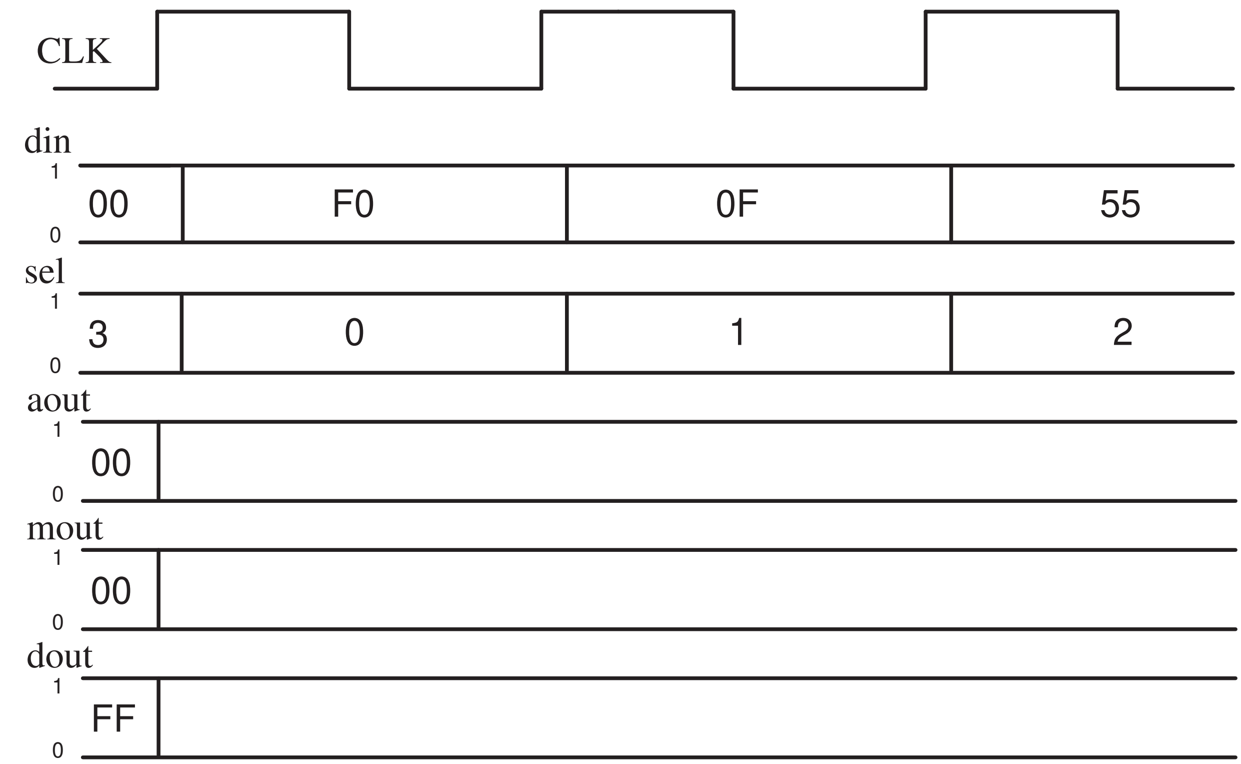
e. Which technology has a RAM–based programming strategy that makes in reprogrammable almost instantly?

2. (30 pts) Consider the following VHDL code.

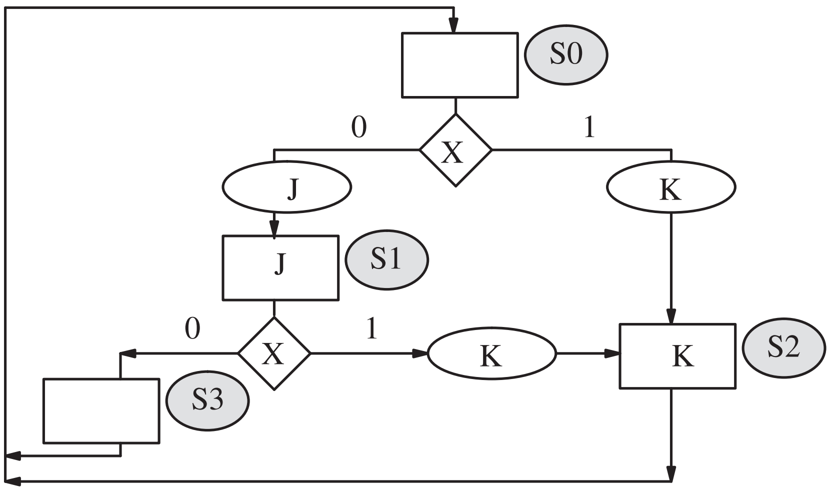
```
library ieee;
use ieee.std_logic_1164.all;

entity regadd is port(
   signal clk: in std_logic;
   signal sel: in std_logic_vector(1 downto 0);
   signal din: in std_logic_vector(3 downto 0);
   signal dout: buffer std_logic_vector(3 downto 0)
   -- remember a “buffer” is both an output and internal input
);
end regadd;

architecture behavior of regadd is

signal mout,aout : std_logic_vector(3 downto 0);

begin

dffs: process(clk)
begin
if (clk'event and clk='1') then
   dout <= mout;
end if;
end process;

bigmux:process(sel,din,dout,aout)
begin
aout <= not dout;
case sel is
   when "00" =>
      mout <= dout;
   when "01" =>
      mout <= aout;
   when others =>
      mout <= din;
end case;
end process bigmux;

end behavior;
```

a. Draw a diagram of logic that implements the VHDL code. (You can use the bottom of page 3 or the back of the last page if you need more space.)

b. Complete the following timing diagram for the code and chart. Assume all flip–flops are rising–edge triggered. Note that numbers are in hex.



3. (40 pts) Consider the following ASM chart.



a. Write a VHDL code that implements it. *Use conventional state encoding.*

```
entity asm is port(
   signal x,clk: in std_logic;
   signal j,k: out std_logic
);
end pchecker;

architecture behavior of asm is

-- put constants here
-- example of syntax: constant S0: std_logic_vector(X downto 0) := B"0...0";





signal pstate,nstate: std_logic_vector(1 downto 0); -- present and next state

begin

dffs:process (                                                          )
begin
-- dff logic here - don't worry about reset







end process;

c_logic:process(                                                        )
begin
```

```
end process c_logic;
end behavior;
```

b. What state assignments would you use to implement one–hot state encoding?

S0: S1: S2: S3:

c. Which of the two state assignments (conventional or one–hot) would you expect to have less output delay?

d. Which assignment would you expect to use more hardware?

e. Is gray–coding of the state assignment possible? Either explain why it is not possible or provide and example of a gray–coded state assignment.

S0: S1: S2: S3: