

ΘΕΜΑ 4

```
Entity counter_4bit is
Port(clk: in std_logic;
      rst: in std_logic;
      count: out unsigned(3 downto 0);
);
end counter;

architecture beh of counter is
    signal counter:unsigned(3 downto 0);
begin
    counter<=count;--εκχώρηση περιεχομενου μετρητη στη θυρα σηματος

process(clk,rst)
begin
    if rst<='1' then--ασυγχρονο reset
        counter<=(others=>'0');--μηδενισμος μετρητη
    elsif clk'event and clk='1' then--ανοδικη ακμη ρολογιου
        if counter="1111"then --αν ο μετρητης ειναι 15, μηδενιζει
            counter<=(others=>'0');--αν ο μετρητης είναι στο 5 παει στο 10
            και συνεχιζει
        elsif counter<="1010";
            counter<=std_logic_vector(unsigned(count)+1);--αυξανει τον
            μετρητη κατα 1
        end if;
    end if;
end process;
end beh;
```

ΘΕΜΑ 3

```
Entity shiftreg4 is
Port(clk: in bit;
      Sin: in bit;
      Dout: out bit_vector(3 downto 0)
);
End entity shiftreg4;

Architecture beh of shiftreg4 is
    Signal reg4: bit_vector (3 downto 0);--ενδιαμεσο σημα 4bit
Begin
```

```

        Dout<=reg4;--εκχώρηση περιεχομένου καταχωρητή στη θύρα σήματος
Process(clk)--διεργασία ρολογιού
Begin
    If clk'event and clk='1' then—εναλλαγή ρολογιού, ανοδική ακμή
        For i in 1 to 3 loop
            Reg4(i)<=reg4(i+1);--μετακίνηση όλων των bit δεξιά
        End loop;
        Reg(3)<=sin;--εισαγωγή σήματος στο σημαντικότερο bit
    End if;
End process;
End beh;

```

Θεμα 1

```

Entity alarmsystem is
Port(sensors: in std_logic_vector(0 to 3);
     Zone: out std_logic_vector(0 to 1);
     Alarm: out std_logic)
);

```

Architecture behavioral of alarmsystem is

```

Begin
    --το κάθε zone παίρνει την τιμή του σύμφωνα με την είσοδο του αισθητήρα που
του αντιστοιχεί
    Zone<="00" when sensors(0)='1' else
    Zone<="01" when sensors(1)='1' else
    Zone<="10" when sensors(2)='1' else
    Zone<="11" when sensors(3)='1' else
    "00";
    Alarm<=sensors(0) or sensors(1) or sensors(2) or sensors(3);--το σύστημα
ενεργοποιείται αν ενεργοποιηθεί εστω και ένα αισθητήρας
End behavioral;

```

Θεμα 5

```

Entity alu is
Port(a: in signed(31 downto 0); -- register 1
     b: in signed(31 downto 0); -- register 2
     Function: in std_logic_vector(1 downto 0); -- είσοδος η οποία επιλέγει την πράξη
που θα γίνει
     Y: out signed(31 downto 0) -- καταχωρητής εξόδου ο οποίος θα κρατάει το
αποτέλεσμα
);
End alu;

```

Architecture beh of alu is

```

Begin
Process(a,b,function)--διεργασία

```

```

Begin
  --περιπτώσεις
  Case function is
    When "00"=> y<=a and b; -- 1η περίπτωση and μεταξύ των
    καταχωρητών A,B
    When "01"=> y<=a or b; -- 2η περίπτωση or μεταξύ των καταχωρητών
    A,B
    When "10"=> y<=a xor b; -- 3η περίπτωση xor μεταξύ των καταχωρητών
    A,B
    When "11"=> o_alu<= a xnor b; -- 4η περίπτωση xor μεταξύ των
    καταχωρητών A,B

    When others=> null;
  End case;
End process;
End beh;

```