

Homework 4

You may discuss the assignments with your classmates but need to write down your solutions independently. Be careful with your handwriting. Unclear solutions will be assumed wrong.

1. (5 pts) What is the difference between a signal and variable assignment?
2. (10 pts) Wait statement is used to temporarily suspend a process. Which one of the wait statement combinations you would use for each one of the following cases:
 - (a) a specified time has passed
 - (b) a specified condition is met
 - (c) an event occurs which affects one or more signals
3. (10 pts) What is the function of the following piece of code?

```
-- D_IN: in BIT;  
-- RESET: in BIT;  
-- CLK: in BIT;  
-- Q_OUT: out BIT;  
  
signal Q1, Q2, Q3 : BIT;  
  
process(CLK, RESET)  
begin  
  if (RESET = '1') then  
    Q1 <= '0';  
    Q2 <= '0';  
    Q3 <= '0';  
  elsif (CLK'event and CLK = '1') then  
    Q1 <= D_IN;  
    Q2 <= Q1;  
    Q3 <= Q2;  
  end if;  
end process;  
  
Q_OUT <= Q1 and Q2 and (not Q3);
```

4. (15 pts) What is the function of the following piece of code?

```

entity unknown is
port(D: in bit_vector(0 to 9);
OUTPUT: out bit);
constant WIDTH: integer := 10;
end unknown;

architecture behavior of unknown is
begin
process(D)
variable tmp: Boolean;
begin
tmp := false;
for i in 0 to D'length - 1 loop
if D(i) = '1' then
tmp := not tmp;
end if;
end loop;
if tmp then
OUTPUT <= '1';
else
OUTPUT <= '0';
end if;
end process;
end behavior;

```

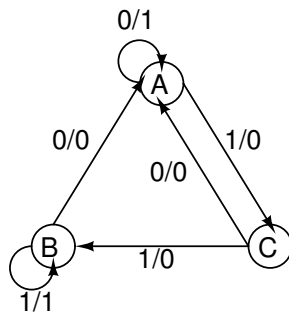
5. (10 pts) What is the function of the following code?

```

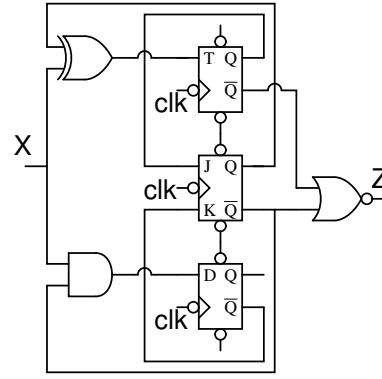
function val (arg1, arg2: integer) return integer is
variable result: integer;
begin
if arg1 > arg2 then
result := arg1;
else
result := arg2;
end if;
return result;
end val;

```

6. (25 pts) Please model the following designs in VHDL. The code does not need to be flawless but the key constructs must be correct.



(a)



(b)

7. (25 pts) Write a VHDL code for a GCD computer. Here is the entity declaration for your reference:

```

entity gcd is
  port (clk: in BIT;
        a: in BIT_VECTOR;
        b: in BIT_VECTOR;
        x: out BIT_VECTOR);
end gcd;
  
```