

ECED 4260 IC Design and Fabrication

Assignment #2 reference solution

<http://myweb.dal.ca/~jgu/4260/assignments.html>

Assignment #2 contains the following problems:

- 1) Which of the following are valid VHDL basic identifiers? Which are reserved words?

Not_bad	not good	1word	srl	_score
This_is_one	#5_isnot	How_about_this_one	Last_one	

Valid identifiers: Not_bad This_is_one How_about_this_one Last_one
Reserved word: srl

- 2) Write a while loop that sums the positive integers from 1 to 1000.

```
Sum := 0;
N:=1;
While n < 1000 loop
Sum=Sum+N;
N :=N+1;
End loop;
```

- 3) Write an array type declaration for an array of 10 integers, and a variable declaration for a variable of the type. Write a for loop to calculate the product of the array elements.

```
Type num_vector is array (1 to 10) of integer;
Variable numbers : num_vector;
```

```
...
prod :=1;
for I in numbers 'range loop
prod := prod*numbers(i);
end loop;
```

- 4) Write entities and architectures, which describe the circuits shown below. For the first one, just use Boolean algebra. For the second one, you are required to construct a two input and gate and use component instantiation to build the three input and gate.

For the first one:

```
Entity gates3 is  
    Port(a, b, c : in bit;  
          D: out bit);  
End gates3;
```

```
Architecture behavior1 of gates3 is  
Begin  
D<= a and b and c;  
End behavior1;
```

For the second one, same entity

```
Entity gates3 is  
    Port(a, b, c : in bit;  
          D: out bit);  
End gates3;
```

```
Architecture behavior2 of gates3 is  
Signal e : bit;
```

```
Component and2 –  
Port (i1, i2 : in bit; o1 : out bit);  
End component;
```

```
Begin
```

```
First_gate : and2
```

```
    Port map ( i1 => a, i2 => b, o1 => e);
```

```
second_gate : and2
```

```
    Port map ( i1 => e, i2 => c, o1 => d);
```

```
End behavior2;
```

For the and 2 gate, the entity and architecture are following:

```
Entity and2 is  
    Port(i1, i2: in bit;  
          O1: out bit);
```

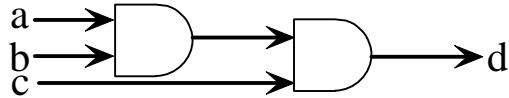
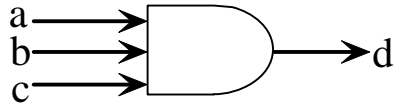
```
End and2;
```

```
Architecture behavior3 of and2 is
```

```
Begin
```

```
O1<= i1 and i2;
```

```
End behavior3;
```



5) A finite state machine is shown below, and the output table is provided. Write a architecture of this state machine.

Architecture state_machine of ass_2 is

Type statetype is (idle,decision,read,write);

Signal present_state,next_state : statetype;

Begin

Start_logic: process(present_satte, r_w, rdy) **begin**

Case present_state is

When idle => oe <='0'; we <='0';

If (rdy='1') **then**

Next_state <=decision;

Else

Next_satte <= idle;

End if;

When decision => oe <='0'; we <='0';

If (r_w='1') **then**

Next_state <=read;

Else

Next_satte <= write;

End if;

When read => oe <='1'; we <='0';

If (rdy='1') **then**

Next_state <=idle;

Else

Next_satte <= read;

End if;

When write => oe <='0'; we <='1';

If (rdy='1') **then**

Next_state <=idle;

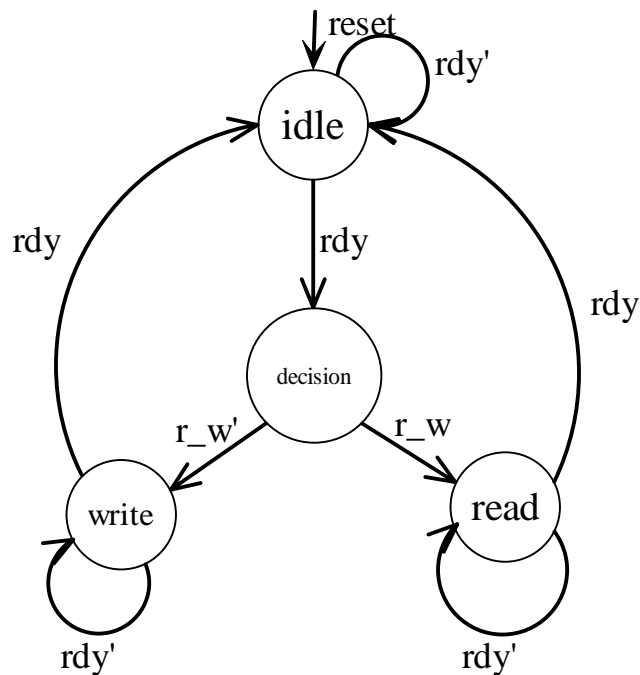
Else

Next_satte <= write;

End if;

End case;
End process start_logic;

State	Outputs	
	Oe	We
Idle	0	0
Decision	0	0
Write	0	1
Read	1	0



- 6) Write two functions, one to find the maximum value in an array of integers and the other to find the minimum value

Type num_vector is array (natural range <>) of integer;

Variable numbers : num_vector;

...

function max (numbers : num_vector) **return** integer **is**

begin

 max_number := numbers(1);

for I in numbers **range loop**

if numbers(i) > max_number **then**

 max_number = numbers(i);

end loop;

return max_number;

end function max;

```

function min (numbers : num_vector) return integer is
begin
    min_number := numbers(1);
    for I in numbers 'range loop
        if numbers(i) < min_number then
            min_number = numbers(i);
        end loop;

    return min_number;
end function max;

```

7) Draw a timing diagram which illustrate the difference between the following VHDL statements

Signal a, b, c, d: bit;
 B <= a after 5 ns;
 C <= transport a after 5 ns;
 D <= reject 3ns inertial a after 5 ns;

