



Bachelor in Computer Vision

## Digital Electronics

### Labs 1

Cedric Lemaitre  
c.lemaitre58@gmail.com

## 「 VHDL Design 」

We propose in this labs some small basic exercises wich allow to discover VHDL Language.

*NB : all examples should be check using **test-benchs** and simulatuon tools*

### └ Problem 1 ▸

Create a AND3 with :

1. 3 inputs : a,b,c
2. 1 output : s
3. Concurrency assignement

### └ Problem 2 ▸

Create a ADD1<sup>1</sup> with :

1. 3 inputs : a, b, cin
2. 2 output : s, cout
3. Concurrency assignement

### └ Problem 3 ▸

Create a ADD4<sup>2</sup> with :

1. 3 inputs : a, b, cin

---

<sup>1</sup>Add 2 signals of 1 bit

<sup>2</sup>Add 2 signals of 4 bit

2. 2 outputs : s, cout
3. Concurrency assignement
4. use your previous ADD1 as a function

*Please consider **port map** instructions for instantiation process*

#### ▮ Problem 4 ▮

Create a D flipflop with :

1. 2 inputs : D, clk
2. 1 output : q

#### ▮ Problem 5 ▮

Create an asynchronous 8-bit counter with :

1. 2 inputs : reset, clk
2. 1 output : s

#### ▮ Problem 6 ▮

Create an 8-bit shift register :

In digital circuits, a shift register is a cascade of D flip flops, sharing the same clock, in which the output of each flip-flop is connected to the 'data' input of the next flip-flop in the chain, resulting in a circuit that shifts by one position the 'bit array' stored in it, 'shifting in' the data present at its input and 'shifting out' the last bit in the array, at each transition of the clock input.

#### ▮ Notes ▮

A test bench is an environment used to verify the correctness or soundness of a design or model, for example, that of a software product. ISE software allows to create vhdl test bench files as follow :

```
library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.numeric_std.all;

entity test is
end entity;

architecture arc of test is
    signal a, b : unsigned(3 downto 0);

    signal cout1, cout2 : std_logic;

    signal i, j, k      : integer := 0;
    signal error_found1 : integer := 0;
    signal error_found2 : integer := 0;
```

```

signal clk           : std_logic;

component add_struct is
  port
  (
    a, b : in  unsigned(3 downto 0);
    s     : out unsigned(3 downto 0);
    cout  : out std_logic
  );
end component;

component add_comp is
  port (
    a, b : in  unsigned(3 downto 0);
    s     : out unsigned(3 downto 0);
    cout  : out std_logic
  );
end component;

begin

  inst1 : add_struct port map (a, b, s1, cout1);
  inst2 : add_comp  port map (a, b, s2, cout2);


process
begin
  clk <= '0';
  wait for 5 ns;
  clk <= '1';
  wait for 5 ns;
end process;

a <= to_unsigned (i, 4);
b <= to_unsigned (j, 4);
k <= i+j;

process
begin
  wait until clk'event and clk = '1';
  if k /= to_integer(cout1 & s1) then
    error_found1 <= 1;
  end if;
  if k /= to_integer(cout2 & s2) then
    error_found2 <= 1;
  end if;

  if i = 15 then
    i <= 0;
    j <= j + 1;
  else
    i <= i + 1;
  end if;

end process;

end arc;

```

Table 1 provides a complete list of VHDL reserved words.

abs	downto	library	postponed	srl
access	else	linkage	procedure	subtype
after	elsif	literal	process	then
alias	end	loop	pure	to
all	entity	map	range	transport
and	exit	mod	record	type
architecture	file	nand	register	unaffected
array	for	new	reject	units
assert	function	next	rem	until
attribute	generate	nor	report	use
begin	generic	not	return	variable
block	group	null	rol	wait
body	guarded	of	ror	when
buffer	if	on	select	while
bus	impure	open	severity	with
case	in	or	signal	xnor
component	inertial	others	shared	xor
configuration	inout	out	sla	
constant	is	package	sll	
disconnect	label	port	sra	

Table 1: A complete list of VHDL reserved words.