FACULTY OF
ENGINEERING
CAIRO UNIVERSITY

LAB 4
INTRODUCTION TO VHDL

CMP301A: COMPUTER ARCHITECTURE COURSE

EXERTED FROM DR. ADNAN SHAOUT- *THE UNIVERSITY OF MICHIGAN-DEARBORN*

OBJECTIVES

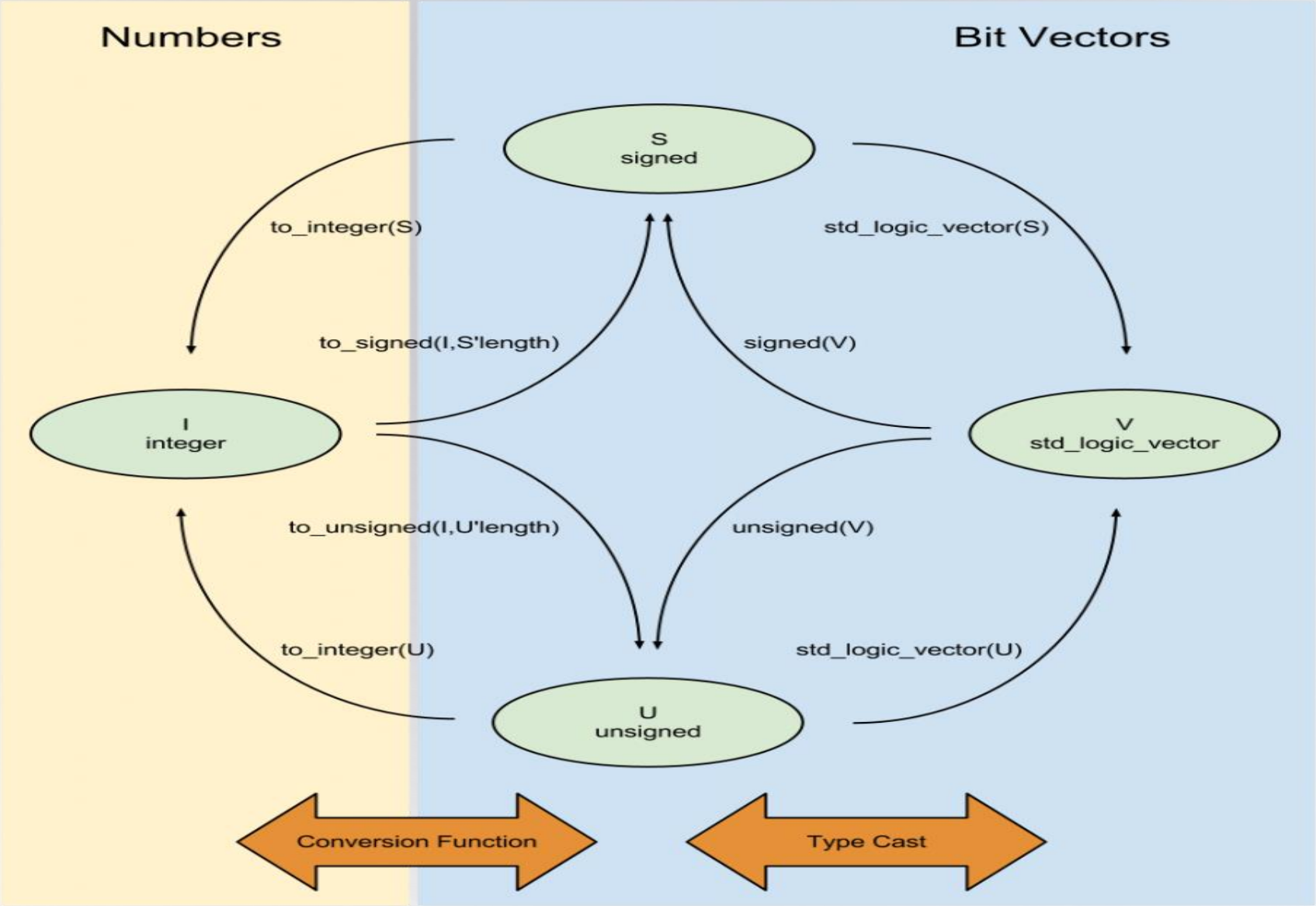
- Review Memory
- Signals vs Variables vs Constant
- Demo:
  - Dealing With Memory
  - Testbench with Quartus
- Extras

Memory …

MEMORY

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.numeric_std.all;
ENTITY ram IS
 PORT (clk : IN std_logic;
we : IN std_logic;
address : IN std_logic_vector(5 DOWNTO 0);
datain   : IN std_logic_vector(7 DOWNTO 0);
dataout : OUT std_logic_vector(7 DOWNTO 0)
);
END ENTITY ram;
```

```
ARCHITECTURE sync_ram_a OF ram IS
  TYPE ram_type IS ARRAY(0 TO 63) of
std_logic_vector(7 DOWNTO 0);
     SIGNAL ram : ram_type ;
BEGIN
PROCESS(clk) IS
BEGIN
    IF rising_edge(clk) THEN
          IF we = '1' THEN
ram(to_integer(unsigned((address))) <= datain;
          END IF;
   END IF;
END PROCESS;
      dataout <= ram(to_integer(unsigned((address)));
END sync_ram_a;
```



HTTP://WWW.BITWEENIE.COM/LISTINGS/VHDL-TYPE-CONVERSION/

VARIABLE, SIGNAL, CONSTANT

CONSTANT DECLARATION

- *A constant can have a single value of a given type.*
- *A constant's value cannot be changed during the simulation.*
- *Constants declared at the start of an architecture can be used anywhere in the architecture.*
- *Constants declared in a process can only be used inside the specific process.*

```
CONSTANT  constant_name  :  type_name  := value;

CONSTANT  rise_fall_time  :  TIME  := 2 ns;
CONSTANT  data_bus  :  INTEGER  := 16;
```

VARIABLE DECLARATION

- *Variables are used for local storage of data.*
- *Variables are generally not available to multiple components or processes.*
- *All variable assignments take place immediately.*
- *Variables are more convenient than signals for the storage of (temporary) data.*

```
VARIABLE variable_name : type_name [:=value];

VARIABLE opcode : BIT_VECTOR(3 DOWNTO 0) := "0000";
VARIABLE freq : INTEGER;
```

SIGNAL DECLARATION

- *Signals are used for communication between components.*
- *Signals are declared outside the process.*
- *Signals can be seen as real, physical signals.*
- *Some delay must be incurred in a signal assignment.*

```
SIGNAL signal_name : type_name [:=value];

SIGNAL brdy : BIT;
SIGNAL output : INTEGER := 2;
```

SIGNAL VS VARIABLE

CAN BE USED ANYWHERE

X <= Y

CHANGES CAN'T BE SEEN UNLESS AFTER A DELAY

```
SIGNAL  s : bit :='0';
....
PROCESS(x)
BEGIN
    s < =  '1';
   a <= s;
END PROCESS;
```

a = '0'

ONLY USED INSIDE A PROCESS

X : = Y

CHANGES CAN BE SEEN IMMEDIATELY

```
PROCESS(x)
   VARIABLE  s : bit :='0';
BEGIN
    s : =  '1';
   a <= s;
END PROCESS;
```

a = '1'

```
ENTITY testbench IS
END testbench;

ARCHITECTURE testbench_a OF testbench IS
COMPONENT and2 IS
PORT ( a,b : IN std_logic; z : OUT std_logic);
END COMPONENT;
SIGNAL test: std_logic (2 downto 0);
BEGIN
PROCESS
BEGIN
test(2) <= '0';
test(1) <= '0';
WAIT FOR 10 ns;
ASSERT(test(0) = '0') REPORT "z is not 0 for 00"
SEVERITY ERROR;
test(2) <= '0';
test(1) <= '1';
WAIT FOR 10 ns;
ASSERT(test(0) = '0') REPORT "z is not 0 for 01"
SEVERITY ERROR;
test(2) <= '1';
test(1) <= '0';
WAIT FOR 10 ns;
ASSERT(test(0) = '0') REPORT "z is not 0 for 10"
SEVERITY ERROR;
test(2) <= '1';
test(1) <= '1';
WAIT FOR 10 ns;
ASSERT(test(0) = '1') REPORT "z is not 1 for 11"
SEVERITY ERROR;
WAIT;
END PROCESS;

uut: and2 PORT MAP (a => test(2), b => test(1),  z =>test(0));

END testbench_a;
```

Can’t We just use a loop instead of writing cases ?

```
PROCESS
        CONSTANT outCases : std_logic_vector(0 to 3) :="0001";
BEGIN
        For i in 0 to 3 Loop
                test(2 downto 1) <= std_logic_vector(to_unsigned(i,2));

                WAIT FOR 10 ns;
                ASSERT(test(0) = outCases(i))     REPORT  " Z is not "
                  &std_logic'image(outCases(i))&" zero for "&integer'image(i)
                SEVERITY ERROR;
        end loop;
        WAIT;
END PROCESS;
```

DEMO

https://youtu.be/oMF2EFMhf_s