Signal Selection in VHDL (for FPGAs)

- Signal selection in VHDL for some FPGA targets could be different from that used for ASICs, as shown below:

General signal selection in VHDL

```
if (sel ='0') then  output <= signal_0;
else
                    output <= signal_1;
end if;
```

Signal selection is implemented here using muxes

In some FPGAs, muxes can be expensive, and tri-state buffers are cheaper

Preferred signal selection for FPGAs using tri-state busses

```
output <= signal_0 when (EnA = '1') else 'Z';
output <= signal_1 when (EnB = '1') else 'Z';
```

Memory Design for in VHDL FPGAs

- The memory address decode should be implemented with tri-state busses for some FPGA families, *and*
- The RAM itself should instantiate library cells provided, e.g., RAM16X1 in Xilinx XACT library, as shown below:

```
Architecture tech_depend_ram of scratch_pad is
…
component ram16x1
port (D, A3, A2, A1, A0, WE: in std_logic; O : out std_logic);
end;
begin
for I in 0 to width -1 generate
cell_ram16x1: ram16x1 port map ( D => value_in(I),
A3 => addr(3), A2 => addr(2), A1 => addr(1), A0 => addr(0),
WE => write, O => value_out (I);
end generate;
end tech_depend_ram;
```

Memory Design for in VHDL FPGAs

```
architecture rtl of single_port_ramtest is
type mem_type is array (7 downto 0) of std_logic_vector (3 downto 0);
signal mem : mem_type;
        begin
        q <= mem(conv_integer(addr)) when (En = ‘1’) else (others => 'Z’);
        process (clk)
                begin
                if (rising_edge (clk)) then
                        if (we = ‘1’) and (En=‘1’) then mem(conv_integer(addr)) <= d;
                        end if;
[…]
```

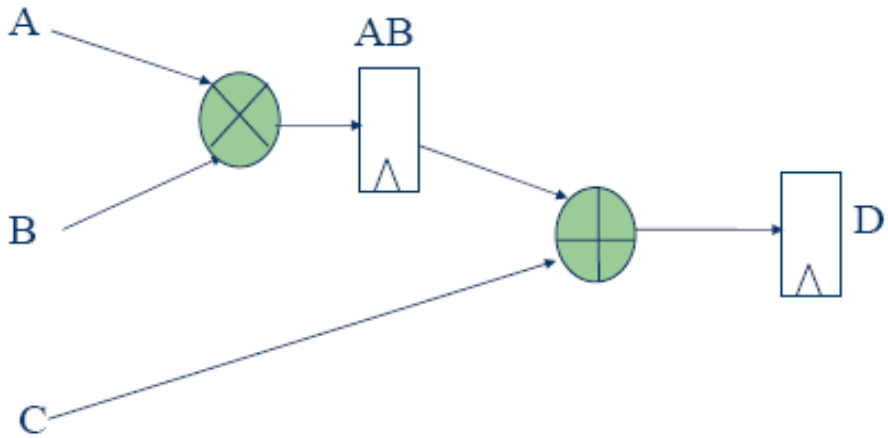
```
architecture rtl of dual_port_ramtest is
type mem_type is array (7 downto 0) of std_logic_vector (3 downto 0);
signal mem : mem_type;
begin
        q <= mem(conv_integer(addr_out)) when (En = ‘1’) else (others => 'Z’);
        process (clk,)
                begin
                if (rising_edge (clk)) then
                        if (we = ‘1’) and (En=‘1’) then mem(conv_integer(addr_in)) <= d;
[…]
```

Memory Design for in VHDL FPGAs

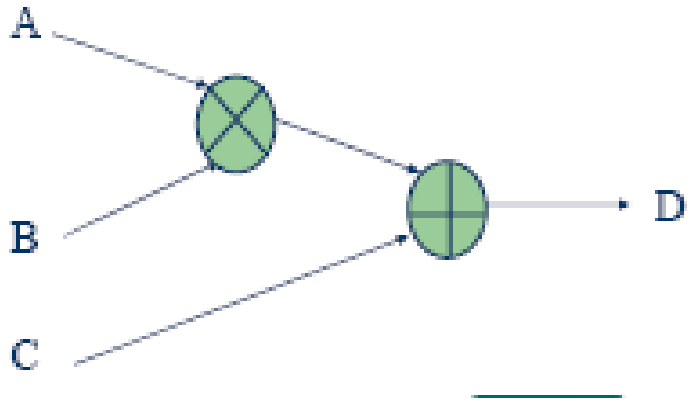
```
architecture rtl of dual_port_ramtest is
Component single_port_ramtest is
port  ( clk,wr,en : in Std_Logic; addr: in Std_Logic_vector(2 downto 0); d: in Std_Logic_vector(3
downto 0); q: out Std_Logic_vector(3 downto 0));
end  component ;
Signal clk,wr,en1,en2 :Std_Logic;
Signal data_in: Std_Logic_vector(3 downto 0);
Signal data_out: Std_Logic_vector(3 downto 0);
Signal addr: Std_Logic_vector(3 downto 0);
begin
        ram1: single_port_ramtest port map ( clk, wr,en1,addr(2 downto 0), data_in,data_out);
        ram2: single_port_ramtest port map ( clk, wr,en2,addr(2 downto 0), data_in,data_out);
        en1<=addr(3);
        en2<=not(addr(3));

[…]
```

Pipeline

```
--operazione in pipeline
signal ab : std_logic_vector(7 downto 0);
begin
process(clk)
begin
if(clk'event and clk = '1') then
ab <= a * b;
d <= ab + c;
end if;
end process;
```




```
--operazione combinatoria
a,b,c : in std_logic_vector(3 downto 0);
d : out std_logic_vector(7 downto 0);
d <= a*b + c;
```



Pipeline systems

- Il pipeline consente di aumentare la frequenza di funzionamento del sistema
- La **latenza** di un sistema in pipeline è il numero di colpi di clock da quando vengono introdotti gli input a quando sono disponibili gli output
- Il **Throughput** è il numero di operazioni che si riescono ad effettuare in un colpo di clock