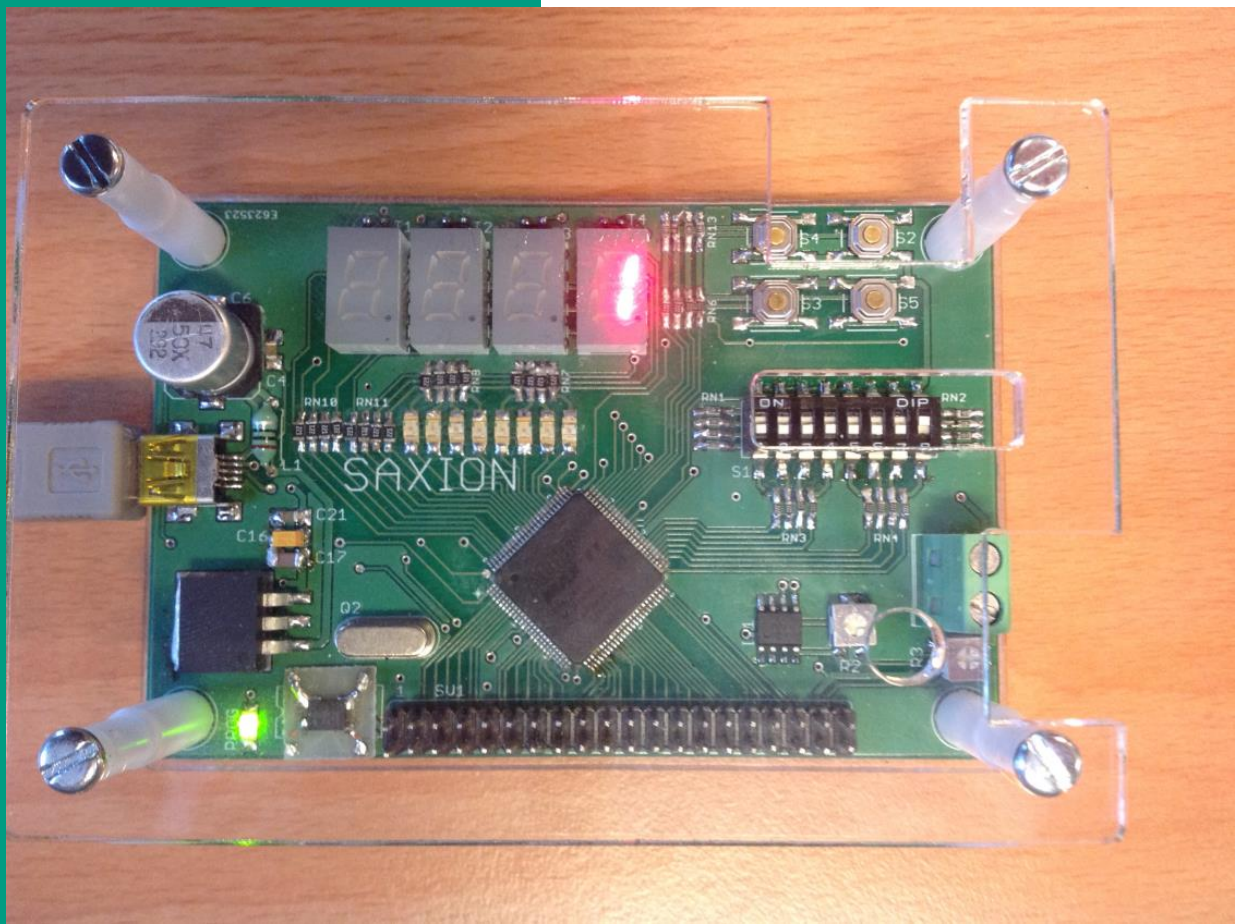
Very High Speed Integrated Circuits Hardware Description Language

Lesson 3
VHDL
Registers

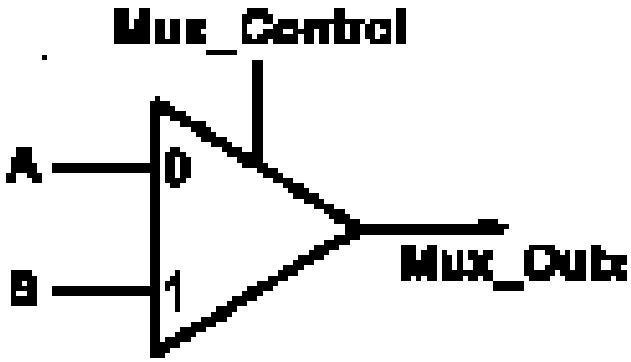


SAXION
UNIVERSITY OF APPLIED SCIENCES

Lesson 3 outline

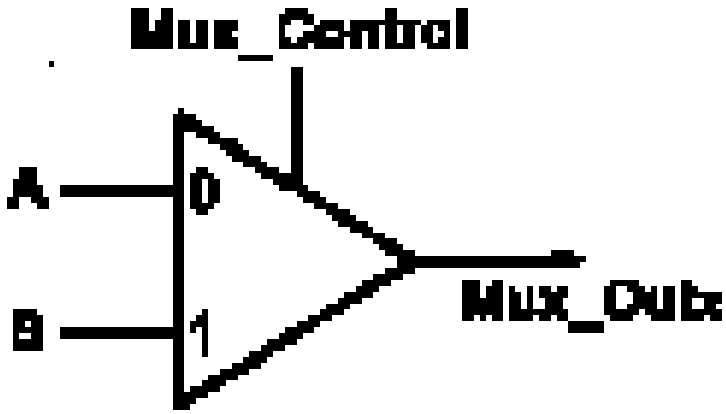
- Multiplex and demultiplex (Chapter 6)
- Tri state (Chapter 7)
- Flip-flops and registers (Chapter 8)

Multiplexer “when else”



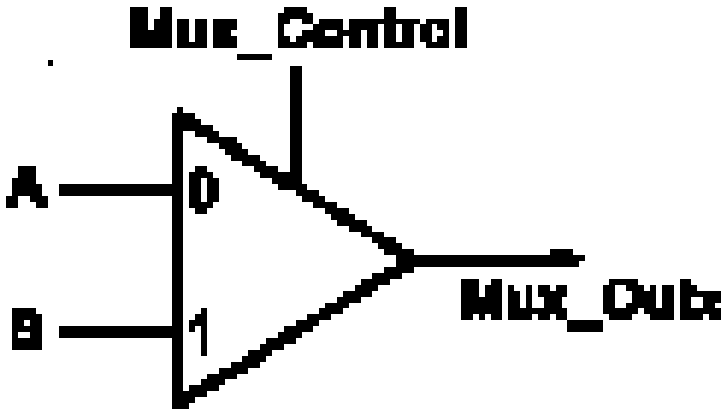

```
Mux_out <=A WHEN Mux_Control =‘0’
ELSE  Mux_out <=B;
```

Multiplexer “if”



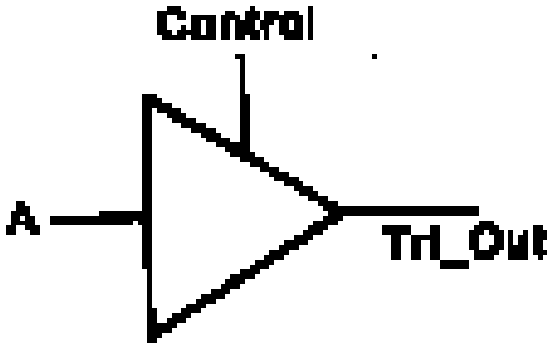

```
Process(A,B,Mux_control)
begin
 if Mux_control ='0' then Mux_out <=A
    else Mux_out <= B; end if;
end process;
```

Multiplexer “case”




```
Process(A,B,Mux_control)
Begin
 case Mux_control is
  when ‘0’ =>
    mux_out <= A;
  when ‘1’=>
    mux_out <= B;
  when others =>
    mux_out <= A;
 end case;
end process;
end
```

Tri-State




```
ENTITY tristate IS
   PORT(A,Control : IN STD_LOGIC;
    Tri_out              :INOUT STD_LOGIC);
END tristate;
```

Tri-State 2

```
ARCHITECTURE behaviour OF tristate IS
BEGIN
Tri_out <= A WHEN Control ='0' ELSE 'Z';
END behaviour;
```

FLIP FLOPS

In VHDL you have many possibilities to implement flipflops.

The different ff as jk and rs can also be implemented.

However, most of the time when using FPGA, d-ff are already implemented.

Therefore we always use d-ff.

D- FF

```
-- Positive edge triggered D flip-flop
-- If WAIT is used no sensitivity list is used
PROCESS
BEGIN
        WAIT UNTIL (Clock'EVENT AND
Clock='1');
        Q1 <= D;
END PROCESS;
```

Gated D- FF

```
PROCESS (Reset,Clock)
  BEGIN
        IF reset = '1' THEN
              Q4 <= '0';
        ELSIF (clock'EVENT AND clock='1') THEN
              IF Enable = '1'
                    THEN Q4 <= D;
              END IF;
        END IF;
  END PROCESS;
```

Gated D-FF registered

```
PROCESS (Reset,Clock)
        BEGIN
                IF reset = '1' THEN
                        Q4 <= '00000000';
                ELSIF (clock'EVENT AND clock='1') THEN
                        IF Enable = '1'
                                THEN Q4 <= D;
                         END IF;
                END IF;
        END PROCESS;
```

Case statement

```
process(a)
 begin
        case a is
                when "00" => b <= "1000";
                when "01" => b <= "0100";
                when "10" => b <= "0010";
                when "11" => b <= "0001";
                when others => report
               "unreachable" severity    failure;
        end case;
end process
```

Assignments

- **Do self test chapter 7,8**
- **Do the exercises chapter 6,7,8**