Signal & Variable

Young Won Lim
02/23/2012

Please send corrections (or suggestions) to youngwlim@hotmail.com.

This document was produced by using OpenOffice and Octave.

Concurrent Statement

- Block Statement
- Process Statement
- Component Statement
- Generate Statement
- Concurrent Signal Assignment
- Concurrent Assertion
- Concurrent Procedure Call

- Architecture Body
- Block Statement
- Generate Statement

- Conditional Signal Assignment
- Selected Signal Assignemnt

Sequential Statement

- Wait Statement
- Assertion Statement
- Report Statement
- Generate Statement
- Signal Assignment
- Variable Assignment
- Procedure Call
- If
- Case
- Loop
- Next
- Exit
- Return
- Null

- Case Statement
- If Statement
- Loop Statement
- Process Statement
- Subprogram Body

- Conditional Signal Assignment
- Selected Signal Assignment

Conditional Signal Assignment

```
Z <=   A or B   [ after 1 ns ]   when S0 = '1'  else
       A or C   [ after 2 ns ]   when S1 = '1' else
       A or D   [ after 3 ns ]   ;
```

```
Z <=   A or B   [ after 1 ns ]   when S0 = '1' else
       A or C   [ after 2 ns ]   ;
```

```
Z <=   A or B   [ after 1 ns ]   when S0 = '1' ;
```

```
Z <=   A or B   [ after 1 ns ]   ;
```

⬅ *simple concurrent statement*

- Concurrent Signal Assignment
  - Conditional Signal Assignment
  - Selected Signal Assignment

Selected Signal Assignment

- Conditional Signal Assignment

```
Z <=    A or B    [ after 1 ns ]   when SEL = "00"  else
        A or C    [ after 2 ns ]   when SEL = "01" else
        A or D    [ after 2 ns ]   when SEL = "10" else
        A or E    [ after 3 ns ]   when SEL = "11" else
        A or F    [ after 4 ns ]   ;
```

- Selected Signal Assignment

```
with SEL select
Z <=    A or B    [ after 1 ns ]   when "00",
        A or C    [ after 2 ns ]   when "01",
        A or D    [ after 3 ns ]   when "10",
        A or E    [ after 4 ns ]   when "11",
        A or F    [ after 5 ns ]   when others;
```

Concurrent vs Sequential

```
with SEL select
Z <=    A or B    [ after 1 ns ]    when "00" ,
                  ● ● ●

Z <=    A or B    [ after 1 ns ]    when SEL = "00"  else
                  ● ● ●

Z <=    A or B    [ after 1 ns ] ;
```

Should be outside process statement

Simple Concurrent signal statement outside process statement

```
process (A, B, C)
begin

    Z <=    A or B    [ after 1 ns ] ;

end
```

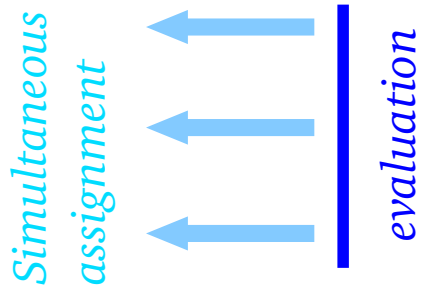
Sequential signal statement inside process statement

- Architecture Body
- Block Statement
- Generate Statement

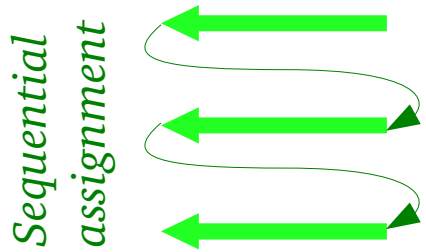
Order

```
architecture arch of entity ent is
begin
    X1 <= A or B ;
    Y1 <= C or D ;
    Z1 <= E or F ;

    process (A, B, C)
    begin
        X2 <= A or B ;
        Y2 <= C or D ;
        Z2 <= E or F ;
    end

end
```

Simulation of parallel activities



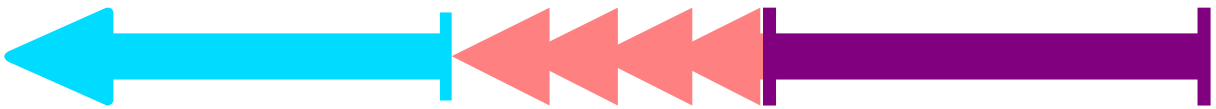
The order of statements is important



Simulation Time

Evaluation

Simulation Time *is assumed to take no time*



Unit: ms, ns, ps, … *Unitless Delta Δ*

Real Delay

$1\,ms = 1000\,ns$

$1\,ns = 1000\,ps$

– used for a simulator to mimic parallel activities simulator

$1\,ps \neq n \cdot \Delta$

no integer n that make n delta equal to 1 ps.

$n \cdot \Delta = 0\,ps = 0\,ns \cdots$

Zero Delay

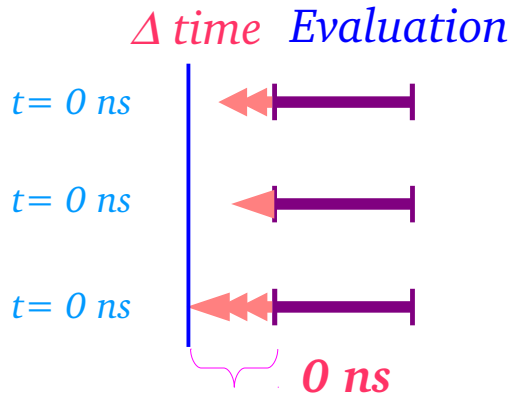
Zero Delay Assignment

```
X1 <= A or B ;
```

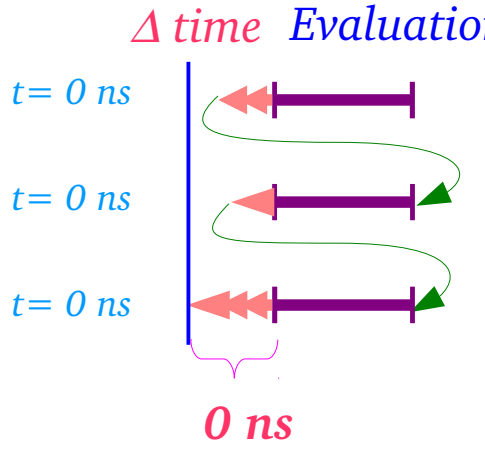
```
X1 <= A or B after 0 ns;
```

Zero Delay Assignment

```
architecture arch of entity ent is
begin
    X1 <= A or B ;
    Y1 <= C or D ;
    Z1 <= E or F ;

    process (A, B, C)
    begin
        X2 <= A or B ;
        Y2 <= C or D ;
        Z2 <= E or F ;
    end

end
```

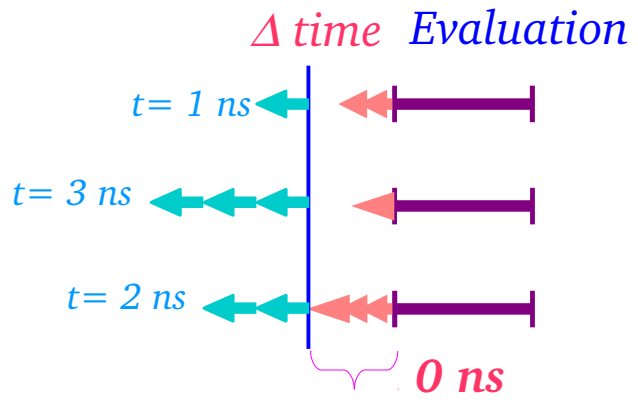


The exact no of delta is determined by the simulator and the context

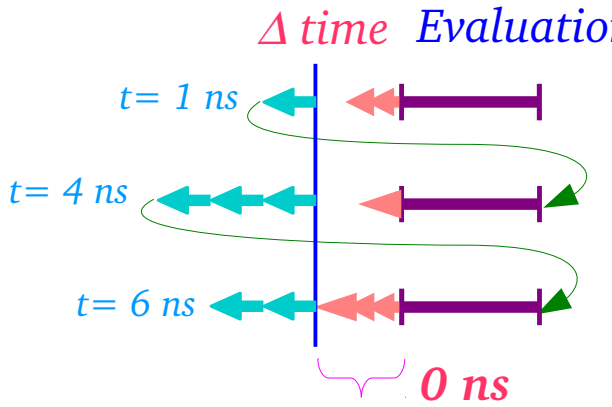


Non-Zero Delay Assignment

```
architecture arch of entity ent is
begin
    X1 <= A or B after 1 ns;
    Y1 <= C or D after 3 ns;
    Z1 <= E or F after 2 ns;

    process (A, B, C)
    begin
        X2 <= A or B after 1 ns;
        Y2 <= C or D after 3 ns;
        Z2 <= E or F after 2 ns;
    end

end
```



The exact no of delta is determined by the simulator and the context



References

[1] http://en.wikipedia.org/
[2]