CSU22022 Computer Architecture I

Third Lecture

Michael Manzke

2023-2024

Trinity College Dublin

Concurrent Signal Assignment Statement (CSA)

- Digital systems operate with concurrent signals
- Signals are assigned values at a specific point in time
- VHDL uses signal assignment statements
  - To specify value and time
- Multiple signal assignment statements are executed concurrently
  - The interface
  - Concurrent Signal Assignment Statement (CSA)

Half-Adder CSA

```vhdl
architecture Behavioral of HalfAdder is

begin

    sum <= x xor y after 5ns;
    carry <= x and y after 5ns;

end Behavioral;
```

- VHDL must specify:
    - Events
    - Delays
    - Concurrency

CSA Statements

- Behavioral
  - Name of architecture that defines the HalfAdder entity
- Signal assignment statements
  - sum <= x xor y after 5ns;
  - carry <= x and y after 5ns;
- "<=" Signal assignment operator
  - Describes how output signals depend on input signals
- An output signal changes if an input signal has changed

Half Adder Operation

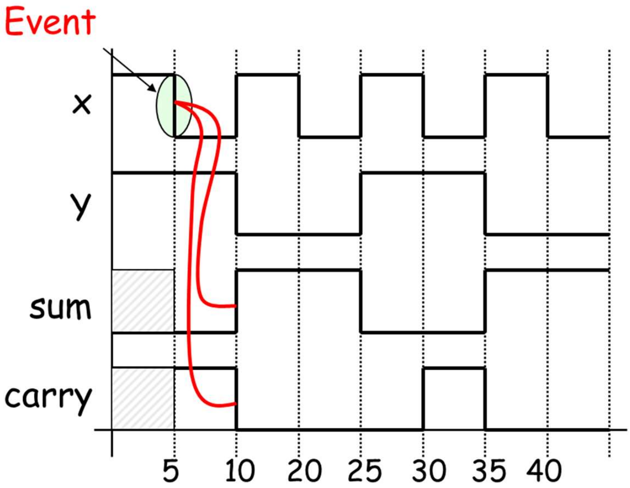

Figure 1: Input and output signals over time [ns]

after Keyword

- Signal propagation of the xor and and gates must be taken into account
  - Both gates require 5 ns propagation delay
- Signal assignment statements define this through the after Keyword
- This keyword specifies when the output signal is set to the result of an evaluation after an input signal transition
  - Event
- The textual order of the assignment statement has no influence on the timing

library and use clauses

```vhdl
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity HalfAdder is
        .
        .
        .
```

- A library contains design entities that can be used
- The library IEEE clause defines the IEEE library
- The library may contain packages
- The above example specifies through the use clause the IEEE.STD_LOGIC_1164.ALL packages
- This package is required for std_logic type declaration

Full-Adder

```vhdl
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity FullAdder is
    Port (in1,in2,c_in : in std_ulogic;
          sum, c_out: out std_ulogic);
end FullAdder;

architecture Behavioral of FullAdder is
    signal s1, s2, s3 : std_ulogic;
    constant gate_delay : Time := 5ns;
begin
    s1 <= in1 xor in2 after gate_delay;
    s2 <= c_in and s1 after gate_delay;
    s3 <= in1 and in2 after gate_delay;
    sum <= s1 xor c_in after gate_delay;
    c_out <= s2 or s3 after gate_delay;
end Behavioral;
```

Listing 1: With signal s1 s2 s3

Full-Adder Schematic

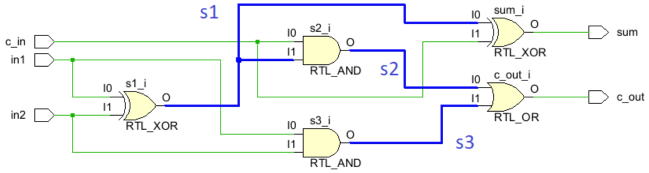

Figure 2: Schematic with signal s1 s2 s3

- s1 <= in1 xor in2 after gate_delay;
- s2 <= c_in and s1 after gate_delay;
- s3 <= in1 and in2 after gate_delay;
- sum <= s1 xor c_in after gate_delay;
- c_out <= s2 or s3 after gate_delay;

Declaration and Body

```vhdl
architecture Behavioral of FullAdder is
    signal s1, s2, s3 : std_ulogic;
    constant gate_delay : Time := 5ns;
begin
```

Listing 2: Architecture Declaration

```vhdl
begin
    s1 <= in1 xor in2 after gate_delay;
    s2 <= c_in and s1 after gate_delay;
    s3 <= in1 and in2 after gate_delay;
    sum <= s1 xor c_in after gate_delay;
    c_out <= s2 or s3 after gate_delay;
end Behavioral;
```

Listing 3: Architecture Body

The Full-Adder Model

- The full-adder simulates the signal transitions at gate-level
- The model has three internal signals
- These signals are not ports to the entity
- The internal signals are declared in the architectural declaration
- The Boolean equations define how each signal is derived as function of:
  - Other signals
  - Propagation delay
- constant can be used to declare a constant of a particular type
  - In this case Time

signal

- A signal is not a variables
    - History of values over time
    - Waveform

signal assignment symbol

```
    signal s1, s2, s3 : std_ulogic := '0';
```

- Signals may use the assignment symbol := followed by an expression
- The value of the expression will be initial value of the signal
- If no initialisation is provided the signal receives a default value
- VHDL signal types:
    - Integer, real, bit_vector...

signals and Time

```
    sum <= s1 xor c_in after 5ns;
```

- In a more general form:
  - signal <= value expression after time expression;
- In the example:
  - if s1 or c_in change its value the sum will be assigned the result of the S1 xor c_in evaluation after 5ns
- The Time-value pair represents the future value of the signal
  - Also called transaction

Multiple signal Transactions

```
    s1 <= x xor y after 5 ns, x or y after 10 ns, not x
    after 10 ns;
```

Listing 4: It is possible to specify multiple signal transactions

- After one of the signals changed all three waveform elements will be evaluated and scheduled according to their after specification
- The simulation keeps an ordered list of all transactions scheduled for a particular signal scheduled transactions are also known as:
    - Projected output waveform