ECE381(CAD), Lecture 6:

# Timing in VHDL

Mehdi Modarressi

Department of Electrical and Computer Engineering,

University of Tehran

Pictures and examples are taken from the slides of "VHDL: Analysis & Modeling of Digital Systems"

# Timing in VHDL

- Explicit notation of time in VHDL
- To model the behavior of real hardware
  - Delay mechanisms
  - Scheduling
- Readings:
  1. "VHDL: Analysis & Modeling of Digital Systems": Chapter 4.3 and 4.4
  2. "VHDL by Example": pages 20-27
  3. "Designers Guide To VHDL": chapter 5

  → Slides are based on reading 1

# Transactions and events

- Each VHDL expression that assigns a value to a signal is a **driver** fro that signal
- Events of the RHS (right-hand side) cause **evaluation** and **scheduling** a **transaction** on the LHS signal
- A scheduled value may or may not result in an **event** on the LHS, i.e. change the value of the LHS

```
a<= b AND c AFTER10 ns;
x<= NOT a AND d AFTER 20 ns;
```

# Transactions and events

a<= b AND c AFTER10 ns;
x<= NOT a AND d AFTER 20 ns;

- At time *t*: Event on *c* or *b*, the value of a is calculated to be assigned 10ns later, **scheduling** a **transaction** on a for *t+10*
- At time *t+10*: the value is assigned
  - If the new value is different from its previous value, an **event** occurs on *a*
  - Otherwise, the transaction does not end to an event
- At time *t+10*: *x* is activated by the event on *a*, the new value is scheduled for *t+30*
- At time *t+30*: the value is assigned, if is different from its previous value, an event occurs on *x*

# Driver of a signal

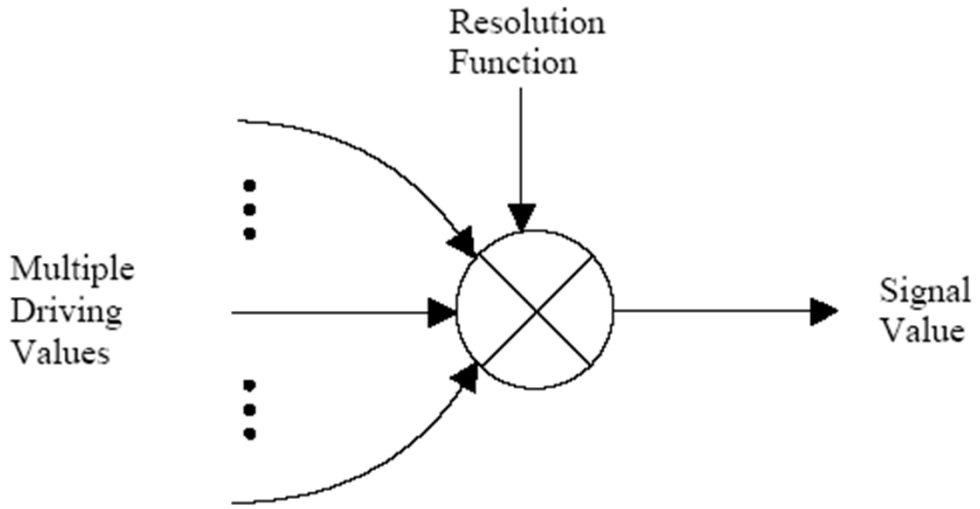
- Each signal has one or multiple drivers
- A driver is a source for a signal in that it provides values to be applied to the signal
- Example: The first concurrent assignment is a driver for signal *w*
- Each driver has its own timing

```
ENTITY figure_5_example IS
  PORT (a, b, c : IN BIT; z : OUT BIT);
END figure_5_example;

ARCHITECTURE  concurrent OF figure_5_example IS
  SIGNAL w, x, y : BIT;
BEGIN
  w <= NOT a AFTER 12 NS;
  x <= a AND b AFTER 12 NS;
  y <= c AND w AFTER 12 NS;
  z <= x OR y AFTER 12 NS;
END concurrent;
```

# Multiple drivers for a signal

- For normal signals, only a single driver is allowed
- How to model buses or wired-or signals that may need multiple drivers?
  - Resolving a single value from multiple driving values
- A *resolution function* handles resolving the signal value
  - Will be introduced later



# Delays in VHDL

- Immediate assignment vs. scheduling

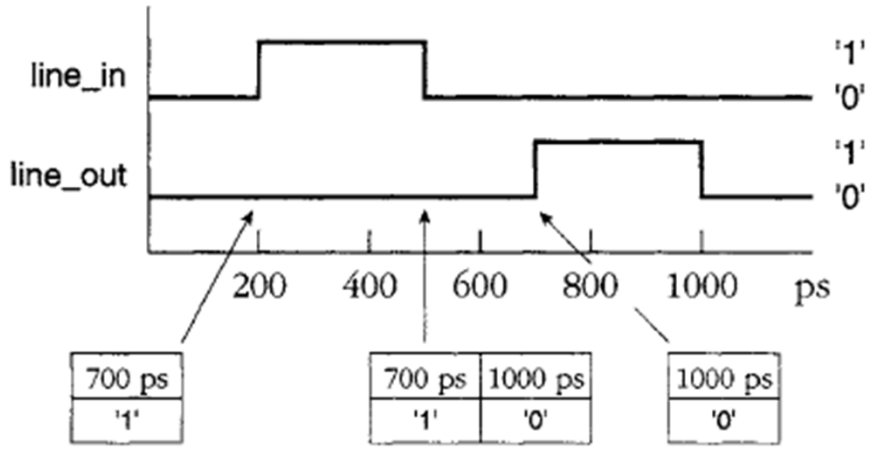
```
a<= b;                  // immediate assignment
a<= b after 10 ns; // scheduled to be done 10 ns later
```

- Two types of delay:
  - Inertial
  - Transport
- Not synthesizable: used to model real hardware behavior
- Delta delay: an internal VHDL delay
  - To correctly implement concurrency

# Transport delay

- To model the latency of the wires
- No rejection, just delays the signal

line_out<=TRANSPORT line_in after 500 ps;



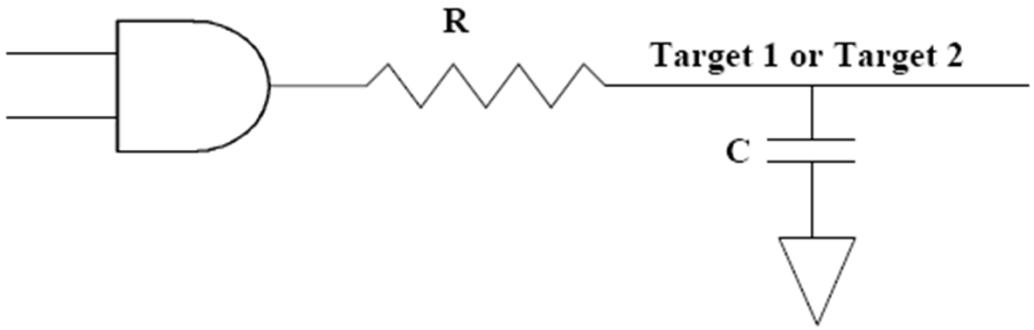
# Transport delay- Transactions rules

- A new transactions are appended to the end of the driver
- A new transaction deletes an older one that is scheduled later
  - This piece of code needs some considerations to work correctly! Just to clarify VHDL event scheduling

```
a<=TRANSPORT b after 800 ps;
a<=TRANSPORT c  after 500 ps;
```

# Inertial delays

- Inertial delay: the delay of changing the output of a gate
  - The *RC* delay is best represented by inertial delay mechanism
  - Gate tends to stay in the same state unless we force it by applying inputs for a sufficiently long duration
  - To charge or discharge the related capacitors



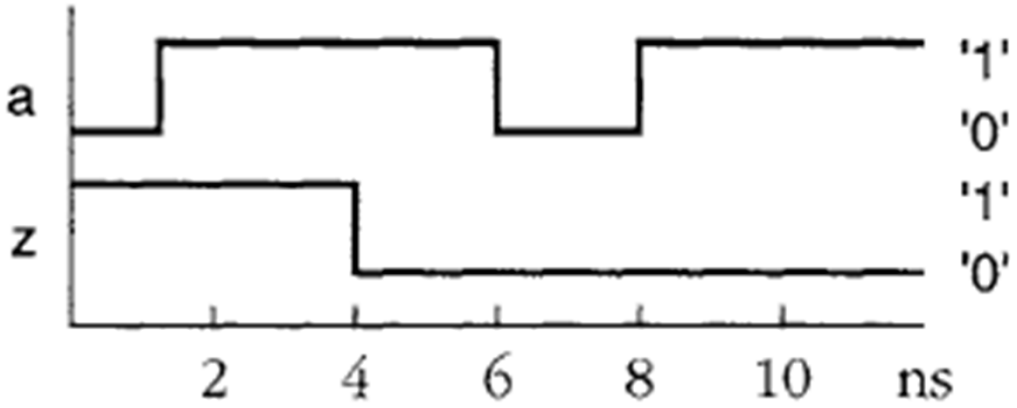
# Inertial delay

y <= INERTIAL NOT a AFTER 3 ns ≈ y <= NOT a AFTER 3 ns

- Inertial is the default delay: when the delay type is not specified, it is considered as inertial
- Inertial delay key point: if a signal assignment would produce an output pulse shorter than the propagation delay, then the output pulse does not happen
- This is what really happens in real hardware
  - If the signal duration is short, it cannot charge or discharge the capacitors to generate new value

# Inertial delay

Z<= NOT a AFTER 3 ns;



# Inertial delay

- INERTIAL and REJECT keywords
- INERTIAL: rejects anything less than its delay
- REJECT: rejects anything less than or equal to its reject
  - The pulse rejection limit specified must be between 0 and the delay specified in the signal assignment
  - If REJECT is not specified, it is equal to the inertial delay

```
a<= b AND c AFTER 10 ns;
b<= INERTIAL b  AND c  AFTER 10 ns;
c<= REJECT  5 ns INERTIAL b  AND c  AFTER 10 ns;
```

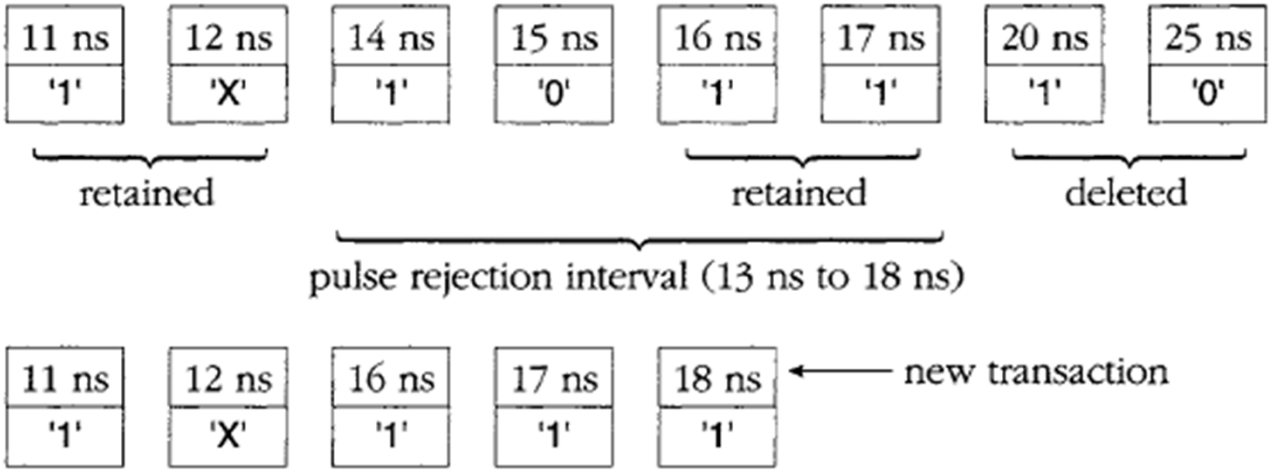
# Pulse rejection rules

- Suppose a signal assignment schedules a new transaction for time *n*, with a pulse rejection limit of *m*
- First, any pending transactions scheduled for a time later than or equal to *n* are deleted
- Second, any pending transactions scheduled in the interval *n-m* are examined
  - The consecutive transactions immediately preceding the new transaction with the same value (if any) are kept
  - All other transactions in the interval are deleted

# Pulse rejection- example

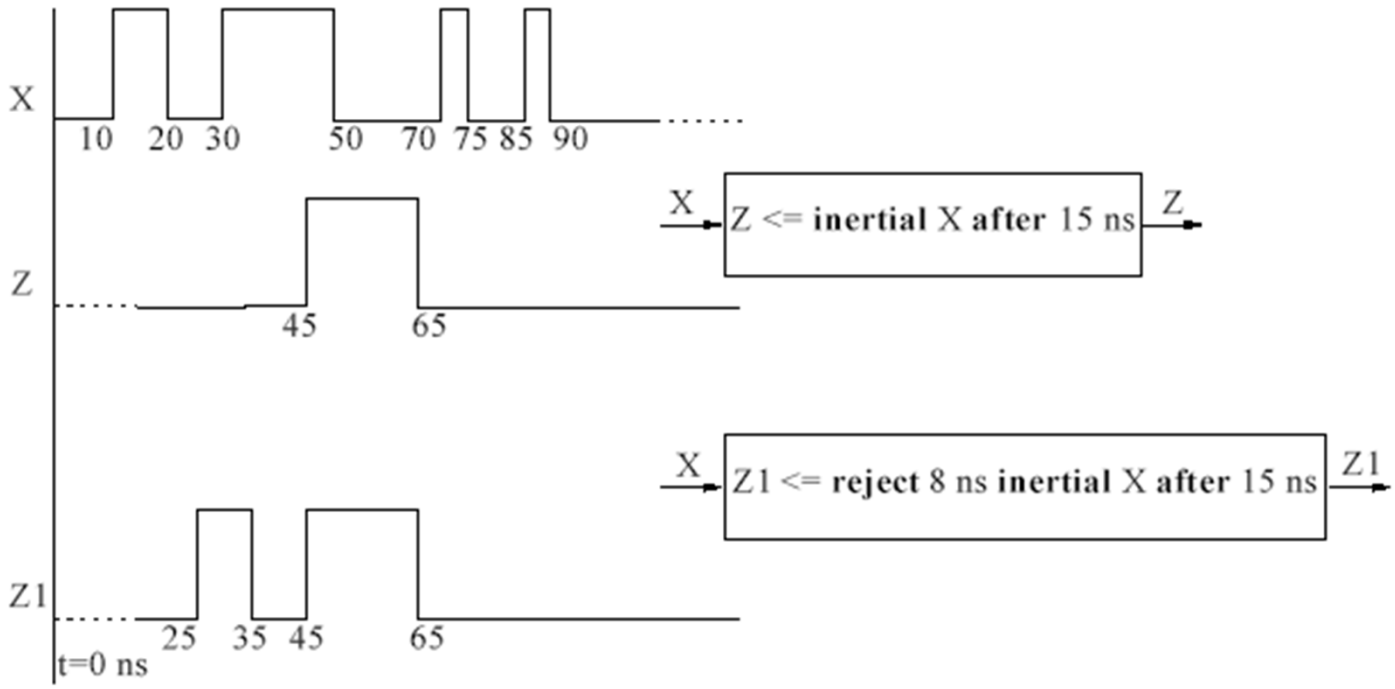
- What if the following signal assignment statement executes at time 10 ns, while the following transactions have already been scheduled?

s <= reject 5 ns inertial '1' after 8 ns;



# Pulse rejection- example 2

- A buffer with 15 ns inertial delay




Source: Petru Eles, IDA, LiTH

# Sequential placement of transactions

- **Quiz:**
  - Sketch the waveform of *waveform*, *target1*, *target2*, and *target3*

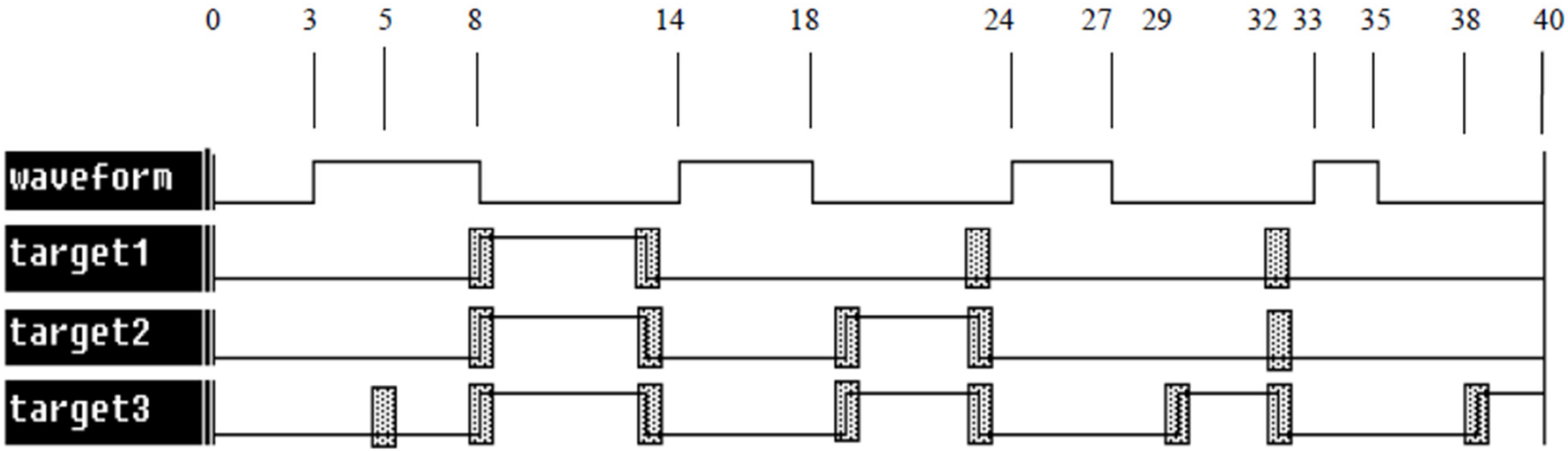
```
ENTITY example IS END ENTITY;
--
ARCHITECTURE delay OF example IS
  SIGNAL waveform : BIT;
  SIGNAL target1, target2, target3 : BIT;
BEGIN

  -- Signal assignments
  target1 <= waveform AFTER 5 NS;
  target2 <= REJECT 3 NS INERTIAL waveform AFTER 5 NS;
  target3 <= TRANSPORT waveform AFTER 5 NS;

  -- Creating waveform
  waveform <=
        '1' AFTER 03 NS, '0' AFTER 08 NS, '1' AFTER 14 NS, '0' AFTER 18 NS,
        '1' AFTER 24 NS, '0' AFTER 27 NS, '1' AFTER 33 NS, '0' AFTER 35 NS;
END delay;
```

# Waveform

- **Solution:**




- See more examples in Dr Navabi's book, Chapter 4