Hardware Modeling [VU] (191.011)
– WS25 –
VHDL Delay Mechanisms

Florian Huemer & Sebastian Wiedemann & Dylan Baumann

WS 2025/26

Modified: 2025-12-16, 16:04 (f8a58e9)

Introduction

- Previously: `after` for delaying the assignments of signals

Introduction

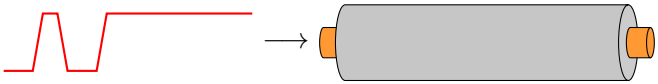
- Previously: `after` for delaying the assignments of signals
  - Why is this needed? $\Rightarrow$ Delays exist in real hardware

Introduction

- Previously: `after` for delaying the assignments of signals
  - Why is this needed? $\Rightarrow$ Delays exist in real hardware
- Two types of delay
  - *Pure* delay: finite signal propagation speed
  - *Inertial* delay: (dis)charging of capacitance

Pure Delay

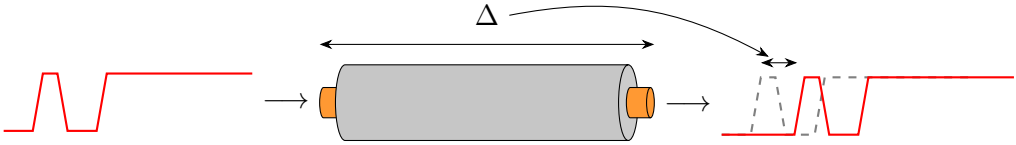
Pure Delay



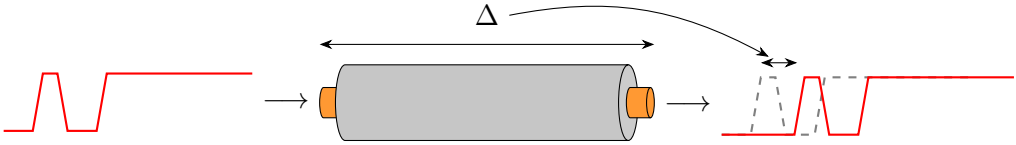
Pure Delay

Pure Delay

Pure Delay

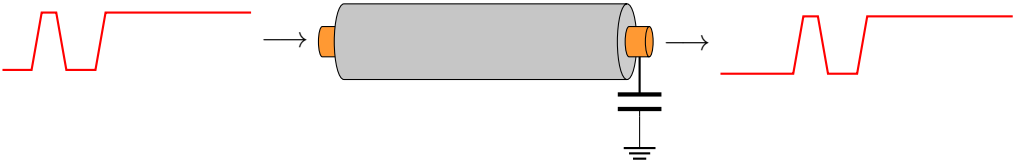


Pure Delay

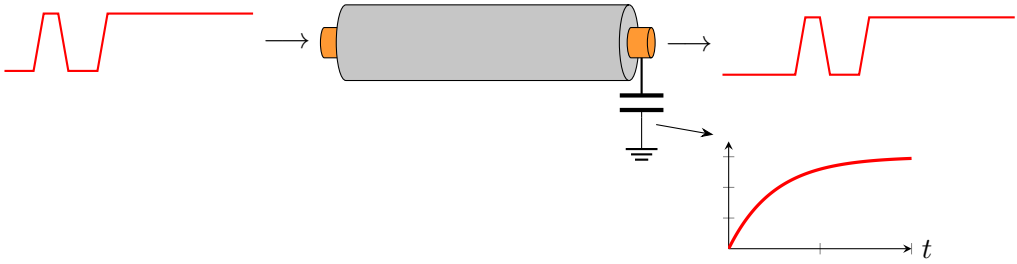


Inertial Delay

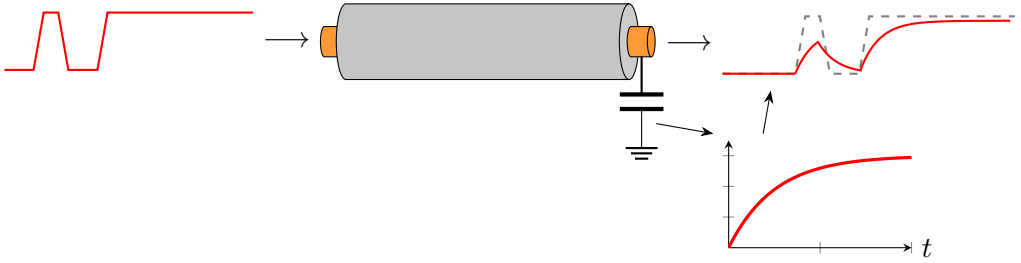
Inertial Delay



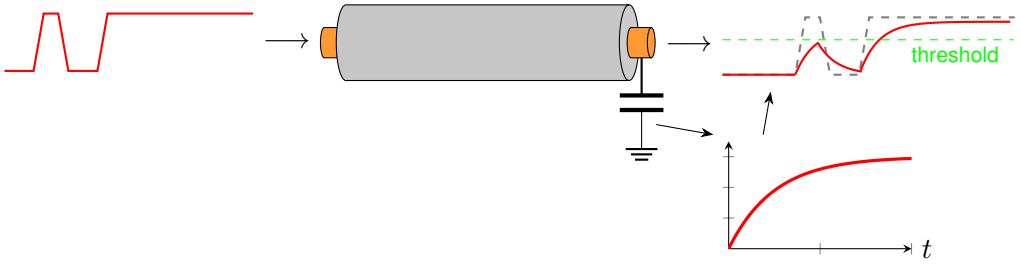
Inertial Delay



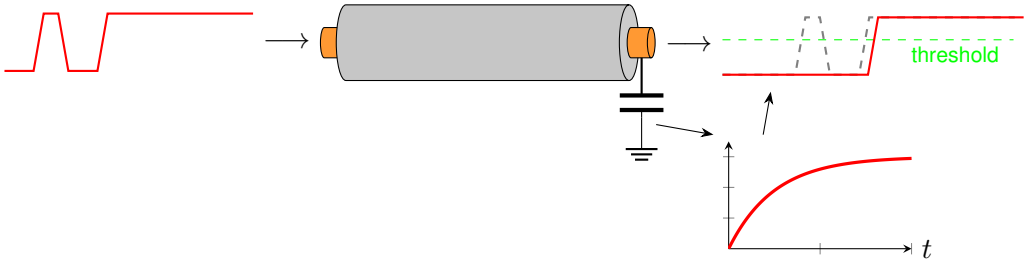
Inertial Delay



Inertial Delay

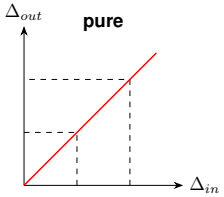


Inertial Delay



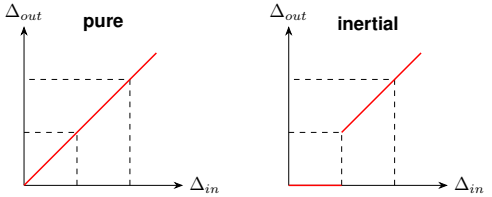
Pure vs. Inertial Delay

- Pure delay
  - Output wave is input wave “shifted backward” in time
  - Arbitrary small pulses are propagated
  - Use where capacitance negligible



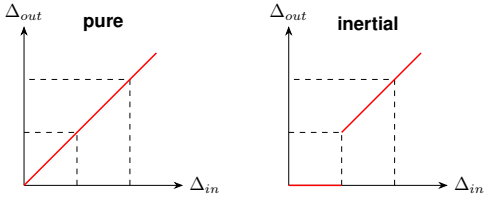
Pure vs. Inertial Delay

- Pure delay
  - Output wave is input wave “shifted backward” in time
  - Arbitrary small pulses are propagated
  - Use where capacitance negligible
- Inertial delay
  - Output wave is input wave “shifted backward” in time plus pulse-width filter
  - Only pulses above a certain width propagate
  - Use where capacitance relevant



Pure vs. Inertial Delay

- Pure delay
  - Output wave is input wave “shifted backward” in time
  - Arbitrary small pulses are propagated
  - Use where capacitance negligible
- Inertial delay
  - Output wave is input wave “shifted backward” in time plus pulse-width filter
  - Only pulses above a certain width propagate
  - Use where capacitance relevant
- Further, more specialized, models exist



Delay Mechanisms in VHDL

- VHDL contains support for delays

```
target <= [ delay_mechanism ] waveform;
```

Delay Mechanisms in VHDL

- VHDL contains support for delays

```
target <= [ delay_mechanism ] waveform;
```

- Both presented delay models are supported

```
delay_mechanism ::=
    transport
  | [ reject time_expression ] inertial
```

Delay Mechanisms in VHDL

- VHDL contains support for delays
  `target <= [ delay_mechanism ] waveform;`
- Both presented delay models are supported
  - Supports pure delay via `transport`

```
delay_mechanism ::=
    transport
  | [ reject time_expression ] inertial
```

Delay Mechanisms in VHDL

- VHDL contains support for delays

```
target <= [ delay_mechanism ] waveform;
```

- Both presented delay models are supported
  - Supports pure delay via `transport`
  - Supports inertial delay via `reject` and `inertial`

```
delay_mechanism ::=
    transport
  | [ reject time_expression ] inertial
```

Pure Delay in VHDL

- Keyword `transport` before the `waveform` in an assignment

```
target <= transport waveform
```

Pure Delay in VHDL

- Keyword `transport` before the `waveform` in an assignment
  `target <= transport waveform`
- Signal assignment is simply delayed by some time

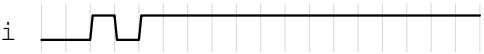
Pure Delay in VHDL

- Keyword `transport` before the `waveform` in an assignment
  `target <= transport waveform`
- Signal assignment is simply delayed by some time
  - Time given by `waveform`'s `after` clause
  - Per default `0 ns`

Pure Delay in VHDL

- Keyword `transport` before the `waveform` in an assignment
  `target <= transport waveform`
- Signal assignment is simply delayed by some time
  - Time given by `waveform`'s `after` clause
  - Per default `0 ns`
- Example

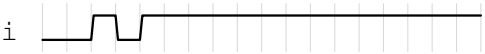
```
i <= '1' after 2 ns, '0' after 3 ns,
     '1' after 4 ns;
```



Pure Delay in VHDL

- Keyword `transport` before the `waveform` in an assignment
  `target <= transport waveform`
- Signal assignment is simply delayed by some time
  - Time given by `waveform`'s `after` clause
  - Per default `0 ns`
- Example

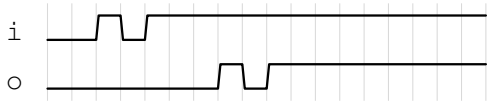
```
i <= '1' after 2 ns, '0' after 3 ns,
     '1' after 4 ns;
[...]
o <= transport i after 5 ns;
```



Pure Delay in VHDL

- Keyword `transport` before the `waveform` in an assignment
  `target <= transport waveform`
- Signal assignment is simply delayed by some time
  - Time given by `waveform`'s `after` clause
  - Per default `0 ns`
- Example

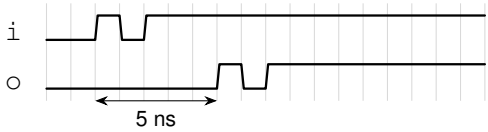
```
i <= '1' after 2 ns, '0' after 3 ns,
     '1' after 4 ns;
[...]
o <= transport i after 5 ns;
```



Pure Delay in VHDL

- Keyword `transport` before the `waveform` in an assignment
  `target <= transport waveform`
- Signal assignment is simply delayed by some time
  - Time given by `waveform`'s `after` clause
  - Per default `0 ns`
- Example

```
i <= '1' after 2 ns, '0' after 3 ns,
     '1' after 4 ns;
[...]
o <= transport i after 5 ns;
```



Inertial Delay in VHDL

- Default mechanism, explicit via `inertial` in an assignment

```
target <= [reject time_expression] inertial waveform
```

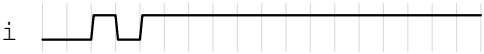
Inertial Delay in VHDL

- Default mechanism, explicit via `inertial` in an assignment
  `target <= [reject time_expression] inertial waveform`
- Signal assignment is delayed and too short pulses are filtered out
  - Minimum pulse width optionally defined by reject clause
  - Per default time expression associated with first element of `waveform`

Inertial Delay in VHDL

- Default mechanism, explicit via `inertial` in an assignment
  `target <= [reject time_expression] inertial waveform`
- Signal assignment is delayed and too short pulses are filtered out
  - Minimum pulse width optionally defined by reject clause
  - Per default time expression associated with first element of `waveform`
- Example

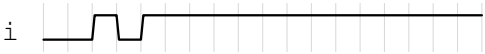
```
i <= '1' after 2 ns, '0' after 3 ns,
     '1' after 4 ns;
```



Inertial Delay in VHDL

- Default mechanism, explicit via `inertial` in an assignment
  `target <= [reject time_expression] inertial waveform`
- Signal assignment is delayed and too short pulses are filtered out
  - Minimum pulse width optionally defined by reject clause
  - Per default time expression associated with first element of `waveform`
- Example

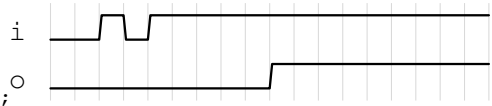
```
i <= '1' after 2 ns, '0' after 3 ns,
     '1' after 4 ns;
[...]
o <= reject 1 ns inertial i after 5 ns
```



Inertial Delay in VHDL

- Default mechanism, explicit via `inertial` in an assignment
  `target <= [reject time_expression] inertial waveform`
- Signal assignment is delayed and too short pulses are filtered out
  - Minimum pulse width optionally defined by reject clause
  - Per default time expression associated with first element of `waveform`
- Example

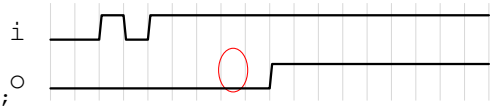
```
i <= '1' after 2 ns, '0' after 3 ns,
     '1' after 4 ns;
[...]
o <= reject 1 ns inertial i after 5 ns;
```



Inertial Delay in VHDL

- Default mechanism, explicit via `inertial` in an assignment
  `target <= [reject time_expression] inertial waveform`
- Signal assignment is delayed and too short pulses are filtered out
  - Minimum pulse width optionally defined by reject clause
  - Per default time expression associated with first element of `waveform`
- Example

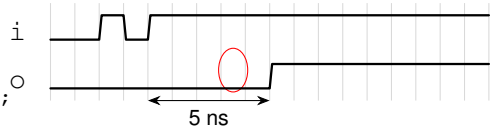
```
i <= '1' after 2 ns, '0' after 3 ns,
     '1' after 4 ns;
[...]
o <= reject 1 ns inertial i after 5 ns;
```



Inertial Delay in VHDL

- Default mechanism, explicit via `inertial` in an assignment
  `target <= [reject time_expression] inertial waveform`
- Signal assignment is delayed and too short pulses are filtered out
  - Minimum pulse width optionally defined by reject clause
  - Per default time expression associated with first element of `waveform`
- Example

```
i <= '1' after 2 ns, '0' after 3 ns,
     '1' after 4 ns;
[...]
o <= reject 1 ns inertial i after 5 ns;
```



Remarks

- Default is inertial delay!

Remarks

- Default is inertial delay!
- Pulse rejection time positive and smaller or equal the `after` time

Remarks

- Default is inertial delay!
- Pulse rejection time positive and smaller or equal the `after` time
- All equivalent

```
o <= i after 1 ns;
o <= inertial i after 1 ns;
o <= reject 1 ns inertial i after 1 ns;
```

Remarks

- Default is inertial delay!
- Pulse rejection time positive and smaller or equal the `after` time
- All equivalent

```
o <= i after 1 ns;
o <= inertial i after 1 ns;
o <= reject 1 ns inertial i after 1 ns;
```

- *Any* pulse shorter than the limit is rejected

Remarks

- Default is inertial delay!
- Pulse rejection time positive and smaller or equal the `after` time
- All equivalent

```
o <= i after 1 ns;
o <= inertial i after 1 ns;
o <= reject 1 ns inertial i after 1 ns;
```

- *Any* pulse shorter than the limit is rejected
- Pure delay: can convert to equivalent `inertial` delay

```
o <= transport i after 10 ns;
o <= reject 0 ns inertial i after 10 ns;
```

Lecture Complete!