Composite Types

- A composite type represents a collection of values. There are two composite types: an array type and a record type.
- An array type represents a collection of values all belonging to a single type.
- Record type represents a collection of values that belong to different types.

Array Types

- Examples of array type declarations are-

- type ADDRESS_WORD is array (0 to 63) of BIT;
- type DATA_WORD is array (7 downto 0) of MVL;

- There are two predefined array types in the language, STRING and BIT_VECTOR. STRING is an array of characters while BIT_VECTOR is an array of bits.

Record Types

- An example of a record type declaration is

```
type MODULE is
    record
        SIZE: INTEGER range 20 to 200;
        CRITICAL_DLY: TIME;
        NO_INPUTS: PIN_TYPE:
        NO_OUTPUTS: PIN_TYPE;
end record;
```

Access Types

- Values belonging to an access type are pointers to object of some other type.

- **type FIFO is array (0 to 63, 0 to 7) of BIT;**
- **type FIFO_PTR is access FIFO;**

File Types

- Objects of file types represent files in the host environment. They provide a mechanism by which a VHDL design communicates with the host environment. The syntax of a file type declaration is

- **type *file-type-name Is file of type-name,***
- The *type-name is the type of values contained in the file. Here are two examples.*
- **type VECTORS is file of BIT_VECTOR;**
- **type NAMES is file of STRING;**

File Types cont..

- A file is declared using a file declaration.The syntax of a file declaration is:
- **file *file-name: file-type-name is mode string-expression ',***
- The *string-expression is interpreted by the host environment as the physical name of the file. The mode of a file, in or out, specifies whether it is an input or an output file, respectively. Input files can only be read while output files can only be*

File Types cont..

- Here are two examples of declaring files.
- **file VEC_FILE: VECTORS is in "/usr/home/jb/uart/div.vec";**
- **file OUTPUT: NAMES is out "stdout";**

Types of Delays or Delay Models

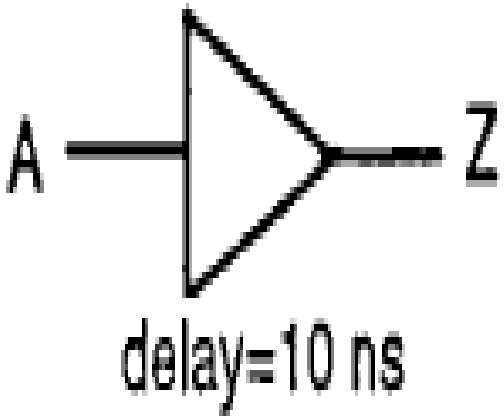
- There are two types of delay that can be applied when assigning a time/value of a signal.
  - Inertial Delay
  - Transport Delay

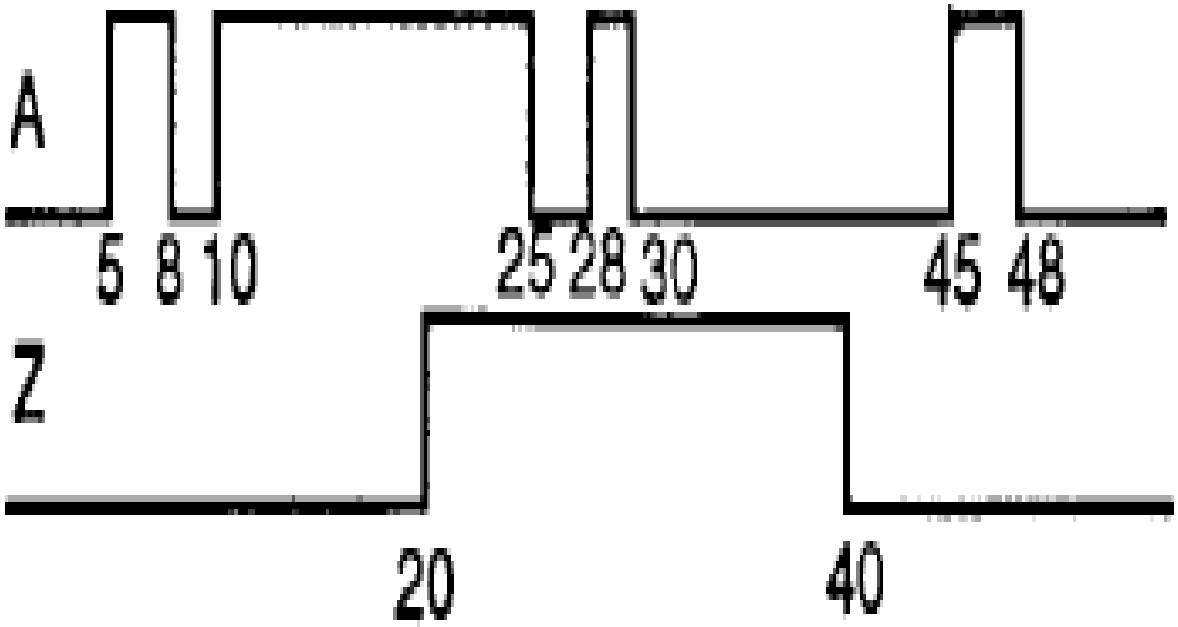
Inertial Delay Model

- *Inertial delay models the delays often found in switching circuits.*
- *It represents the time for which an input value must be stable before the value is allowed to propagate to the output.*
- *In addition, the value appears at the output after the specified delay. If the input is not stable for the specified time, no output change occurs.*

Inertial Delay Model cont..

Z <= A after 10 ns;





Inertial Delay Model cont..

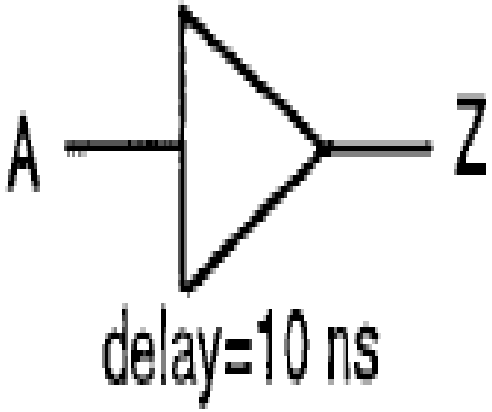
- Since inertial delay is most commonly found in digital circuits, it is the default delay model.
- This delay model is often used to filter out unwanted spikes on signals.

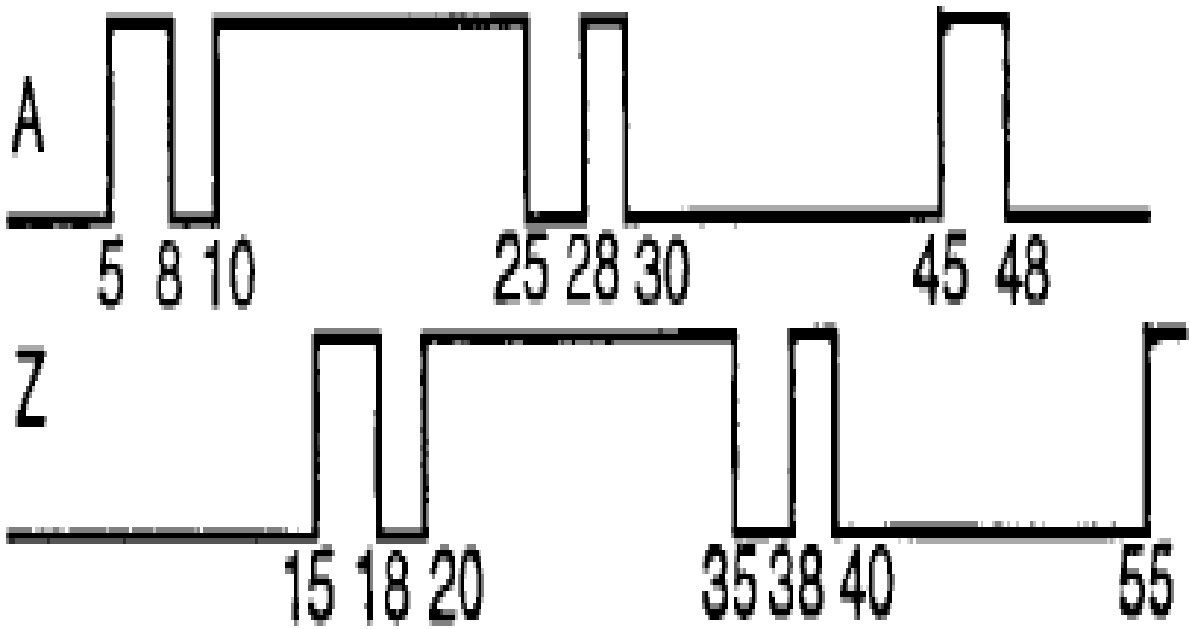
Transport Delay Model

- *Transport delay models the delays in hardware that do not exhibit any inertial delay.*
- *This delay represents pure propagation delay, that is, any changes on an input is transported to the output, no matter how small, after the specified delay.*
- *To use a transport delay model, the keyword transport must be used in a signal assignment statement.*

Transport Delay Model cont..

Z <= transport A after 10 ns;





Transport Delay Model cont..

- Ideal delay modeling can be obtained by using this delay model.
- In this case, spikes would be propagated through instead of being ignored as in the inertial delay case.

Delta Delay

- A *delta delay is a very small delay (infinitesimally small). It does not correspond to any real delay and actual simulation time does not advance.*
- *This delay models hardware where a minimal amount of time is needed for a change to occur, for example, in performing zero delay simulation.*

Delta Delay cont..

- When a signal assignment statement is executed, the value of the expression is computed and this value is scheduled to be assigned to the signal after the specified delay.
- It is important to note that the expression is evaluated at the time the statement is executed (which is the current simulation time) and not after the specified delay.
- If no after clause is specified, the delay is assumed to be a default delta delay.

- Z <= (AO and A1) or (BO and B1) or (CO and C1) ;

Creating Signal Waveforms

- It is possible to assign multiple values to a signal, each with a different delay value. For example,:
- PHASE1 <= '0', '1' **after 8 ns, '0' after 13 ns, '1' after 50 ns;**

- When this signal assignment statement is executed, say at time T, it causes four values to be scheduled for signal PHASEI, the value '0' is scheduled to be assigned at time T+A, 1' at T+8 ns, '0' at T+13 ns, and 1' at T+50 ns.

Signal waveform cont..

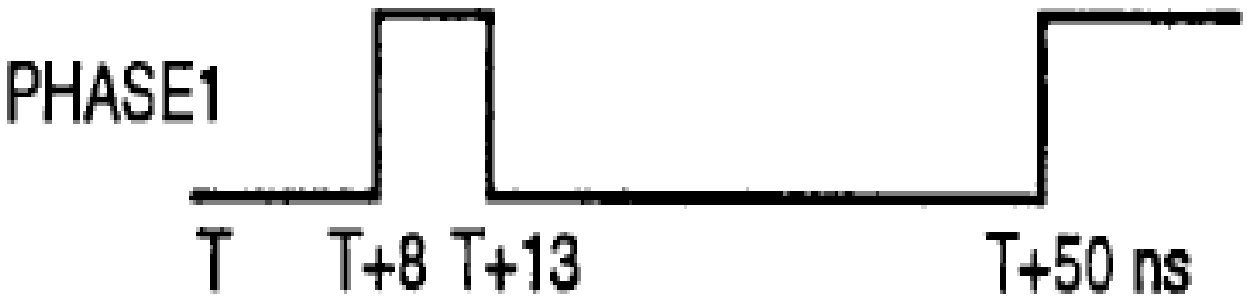


Figure 4.5 A signal waveform.