VHDL
Modeling Digital Systems

EL 310

Erkay Savaş
Sabancı University

Simulation and Synthesis

- VHDL programs are unlike programs written in Pascal, C, C++.
- VHDL is for describing a digital system for mainly two purposes
  - Simulation
  - Synthesis
- The idea is to come up with a description that behaves like the real system with a desired level of accuracy

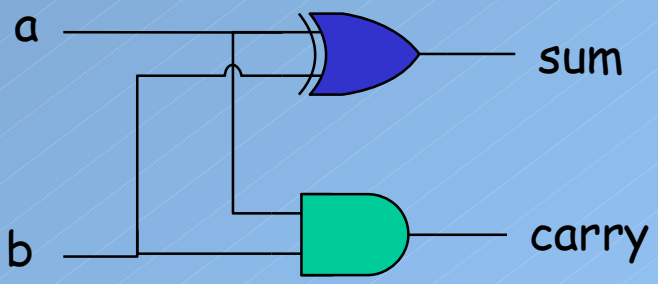
Describing System

- A system is an assemblage of objects united by some form of regular interaction or interdependence
  - Could be anything from single chips to large supercomputers
- What aspects of a digital system we want to describe
  - Interface
  - Function: Structural or Behavioral
  - Structural and behavioral descriptions are complementary of each other and are expected to be supported by any HDL.

A digital system

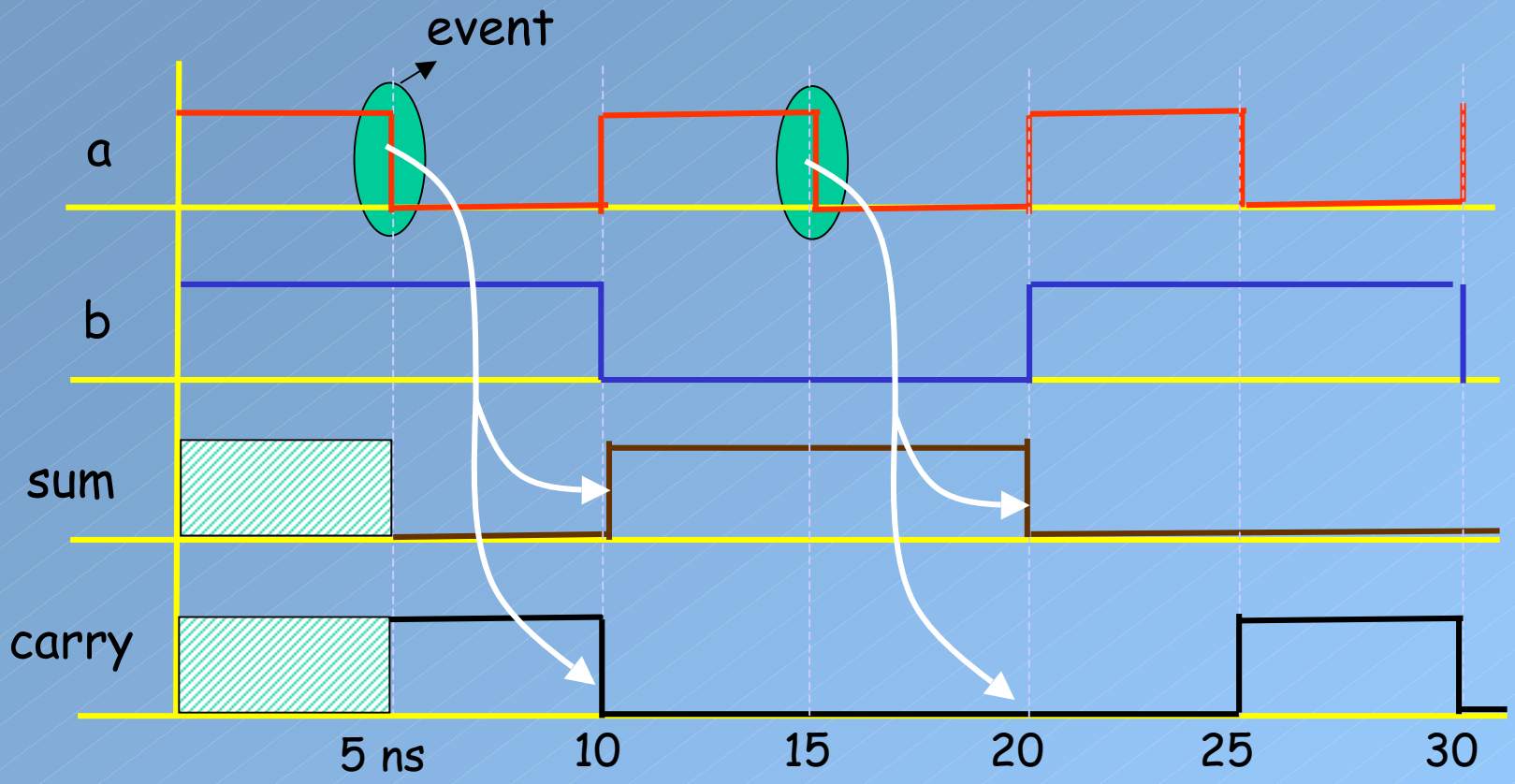
- Consists of
  - Binary signals that may take values of 1 or 0.
  - Components
    - Gates, flip-flops, counters, etc.
    - Transforms input signals to output signals
  - Wires that connect components.

- Half adder



a	b	sum	carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

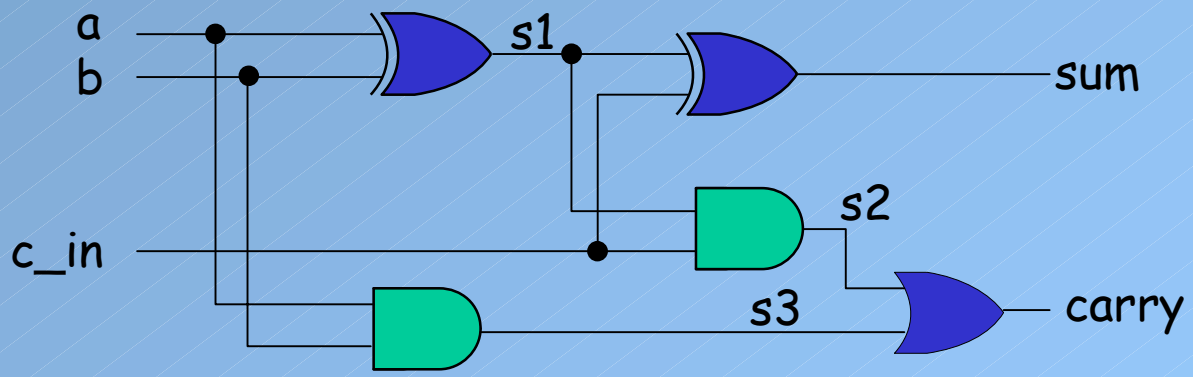
Events and Propagation Delays



- Event: when a signal changes values (1-to-0, 0-to-1)
- Propagation delay: time elapsed before changes at input will produce a change at output (5 ns for this example)

Concurrency of Operations

- From our previous example
  - When a change occurs at one of the inputs, say **a**, the two gates concurrently compute the values of output signals, **sum** and **carry**.
  - New events occur at these signals ➔ trigger more events



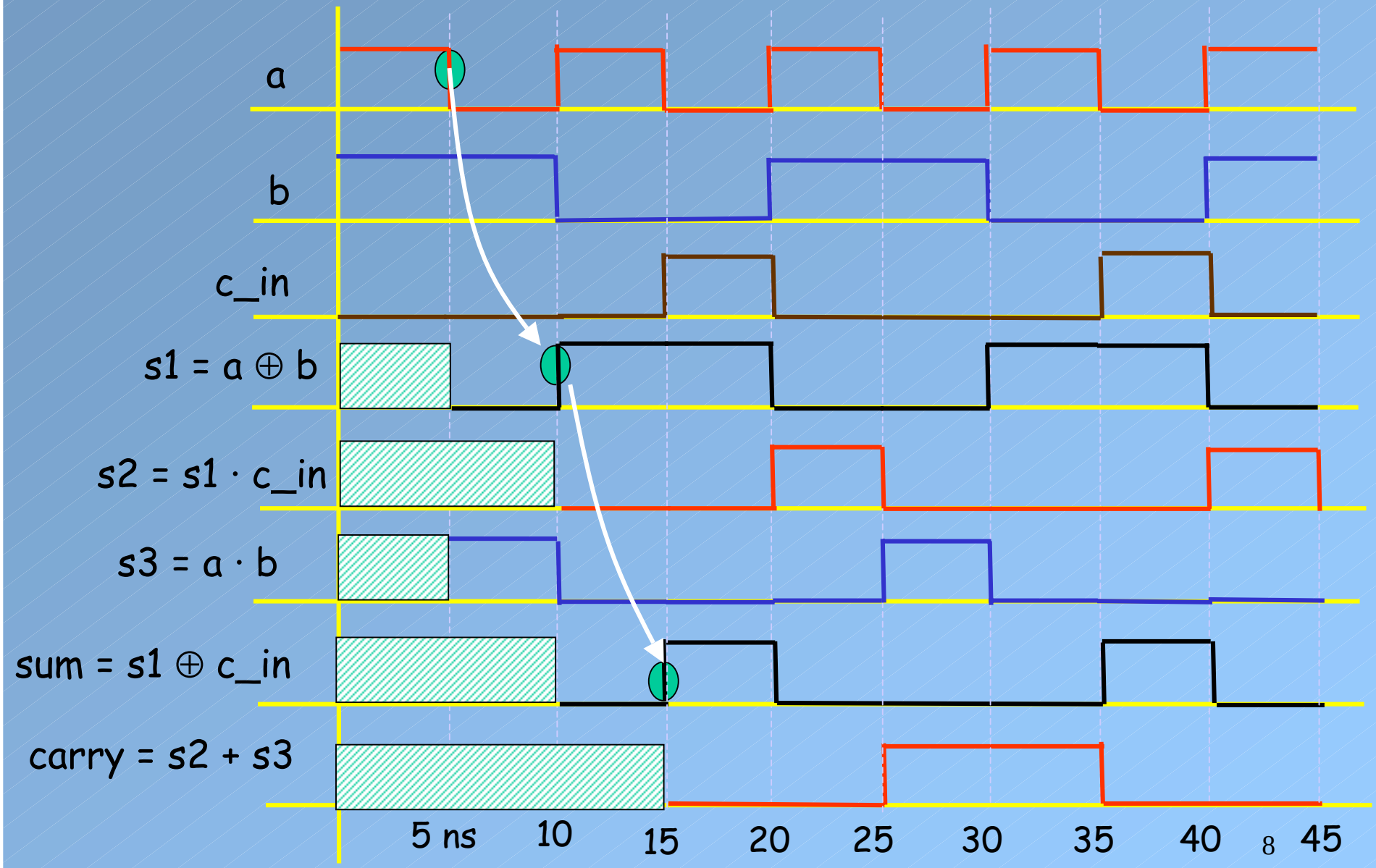
Events on signals lead to computations that may generate events on other signals

Full Adder 1

- Truth table

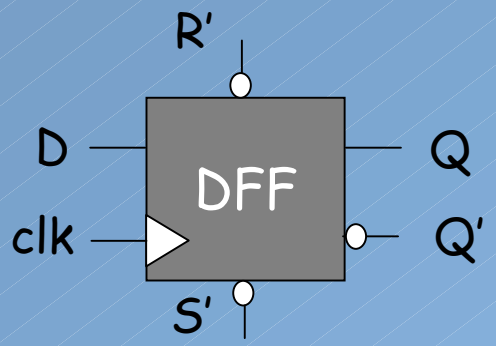
a	b	c_in	sum	carry
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Full Adder 2

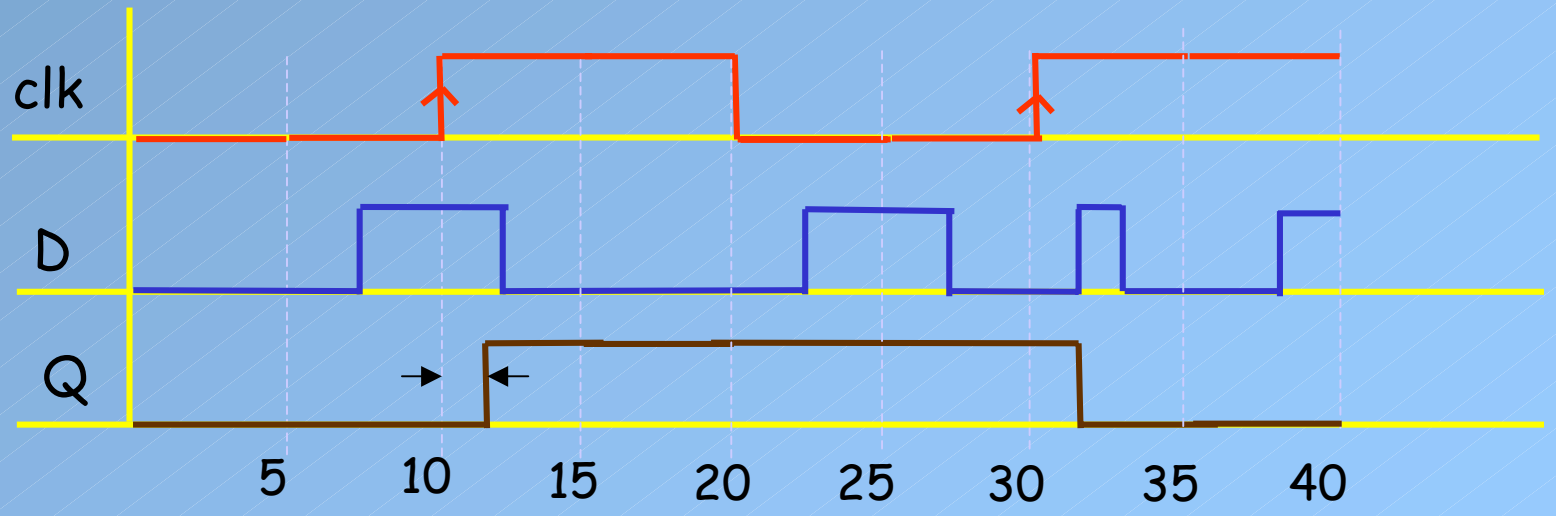

a
b
c_in
s1 = a ⊕ b
s2 = s1 · c_in
s3 = a · b
sum = s1 ⊕ c_in
carry = s2 + s3
5 ns
10
15
20
25
30
35
40
45

Waveforms and Timing

- The sequence of events that occur on a signal produces a waveform on that signal



S'	R'	clk	D	Q	Q'
0	1	X	X	1	0
1	0	X	X	0	1
1	1	R	1	1	0
1	1	R	0	0	1
0	0	X	X	?	?

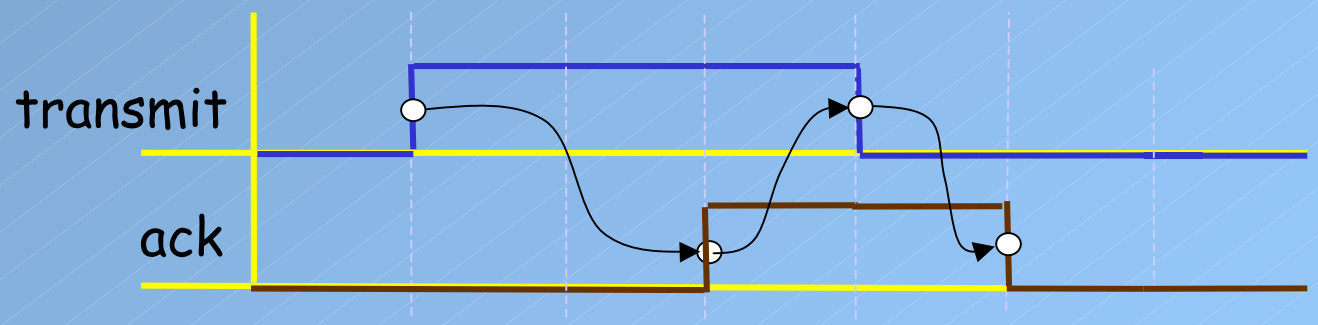


Synchronous Circuit

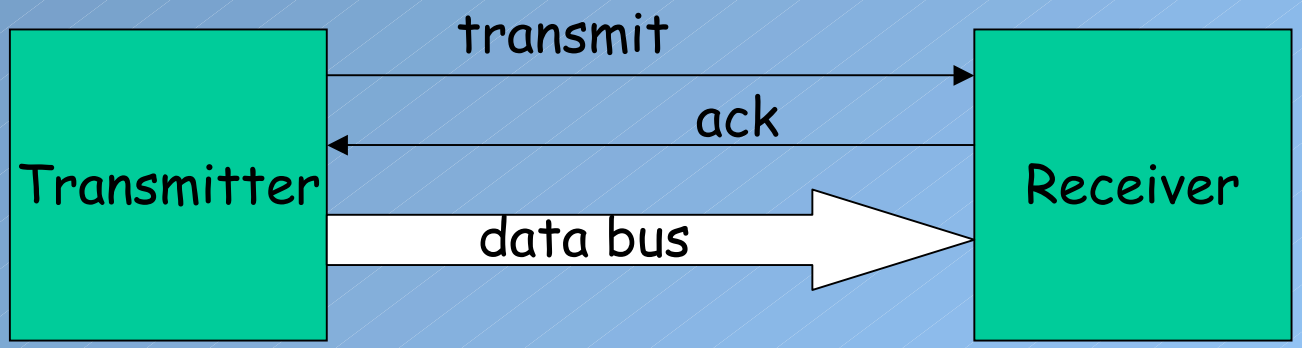
- Synchronous circuits operate with a periodic signal commonly referred to as clock.
  - clock serves as a common time base
  - The transition in clock (rising or falling) determines when the computation of output events is initiated.
  - The input value on signal is sampled at clock transition,
  - if there is an event on input signal before the clock transition occurs, the output may also change.
  - This change, of course, takes some time (propagation time).

Asynchronous 1

- No global clock
  - Request-acknowledge protocols
  - Or handshaking
  - Need to wait for events on specific signals
  - Example: Asynchronous communication channels



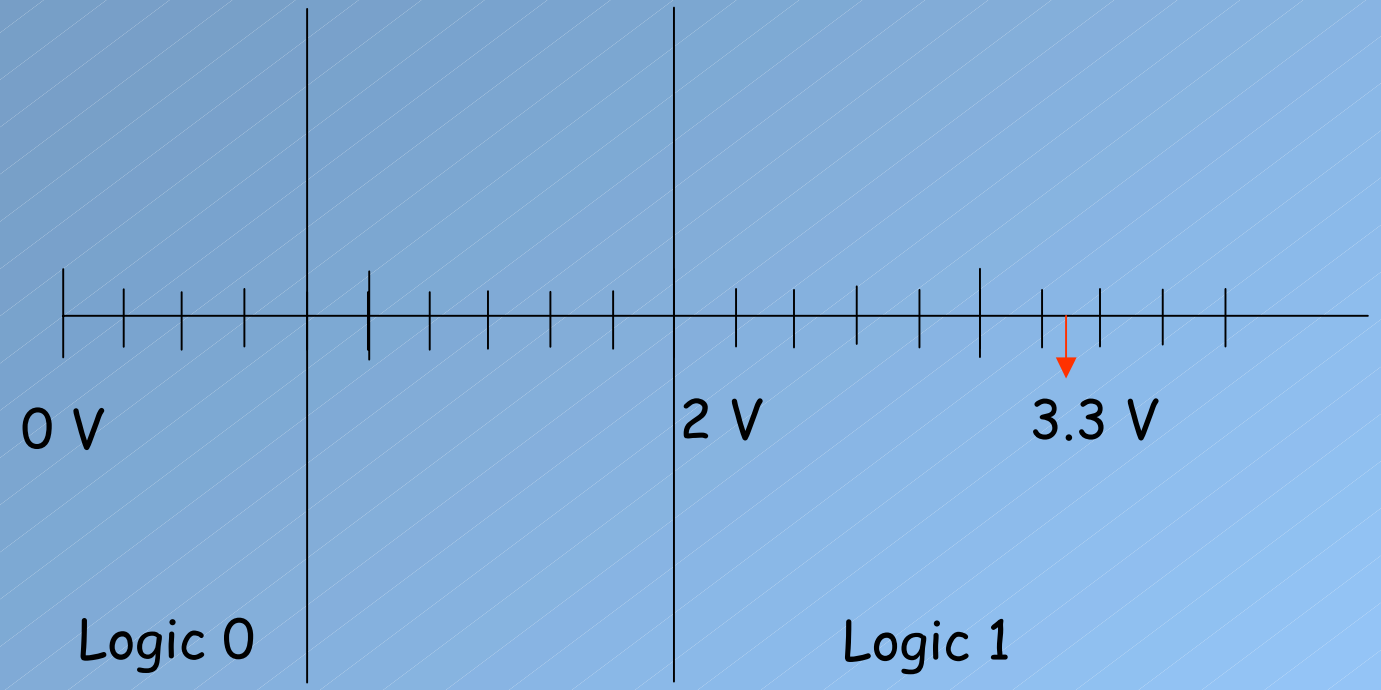
Asynchronous 2



Signal Values

- Binary values
  - 0 or 1
    (driven by a power supply or the output of a gate)
  - Voltage values
  - [0-0.8] V ➔ logic 0
  - [2.0 – 3.3] V ➔ logic 1
  - What if a signal is disconnected? (high-impedance)
  - What if a signal is currently driven to both a 0 and a 1 value? (unknown)
  - What if the initial value of a signal is unknown?
  - How about the signal strength.
- Recall that we want to describe a digital system as accurately as possible.

Signal Values



- Question: Why do we want to represent a situation which apparently corresponds to a design error?
  - we may want to observe the effects of a design error.
  - how is it propagated?

Signal Values: IEEE 1164

Value	Interpretation
U	Uninitialized
X	Forcing Unknown
0	Forcing 0
1	Forcing 1
Z	High impedance
W	Weak unknown
L	Weak 0
H	Weak 1
-	Don't care

- This nine valued system is not a part of VHDL language; but rather a standard definition that vendors are motivated to support for portability.

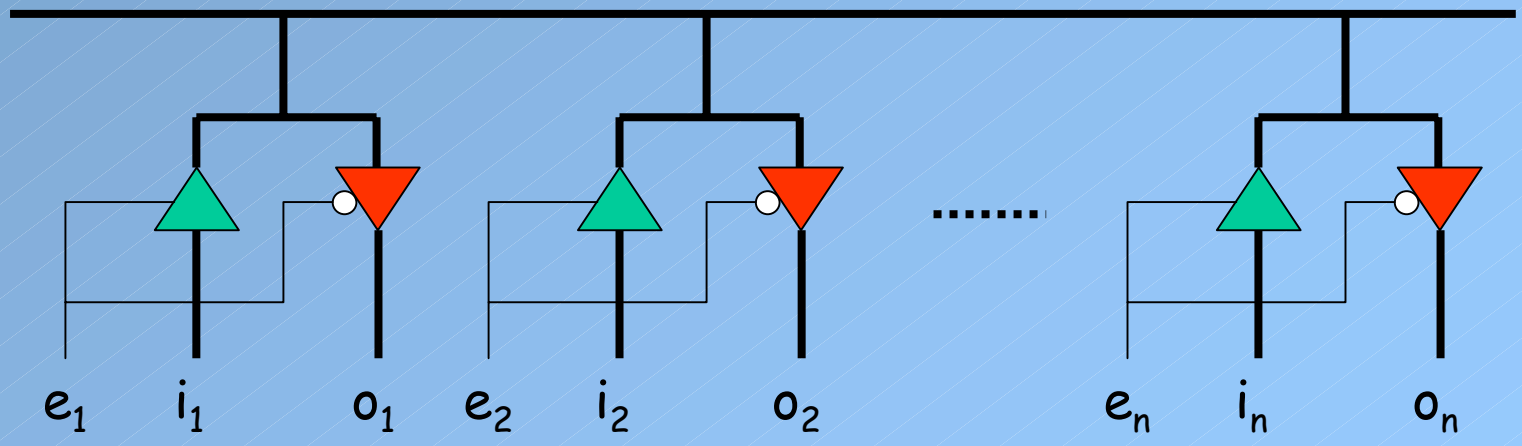
- Use the package that contains this definition.
- Not all these values are synthesizeable.

Signal Values: IEEE 1164

- IEEE 1164 package defines multivalue logic values
- logic operations for multivalue logic
- and some conversion routines from this logic to other types that are part of language
  - **function** TO_BIT(S: STD_ULOGIC, XMAP:BIT:='0')
    return BIT;
  - **function** TO_BITVECTOR(S: STD_ULOGIC_VECTOR,
    XMAP:BIT:='0')
    return BIT_VECTOR;
  - **function** TO_STDULOGIC(B:BIT)
    return STD_ULOGIC;

Shared Signals

- Multiple drivers for input and multiple destinations for output signals
  - Bus construction
  - Drivers (transmitters): tri-state buffers can be controlled by a decoder to ensure that only one source is driving the bus at a time
  - There may be multiple receivers listening the bus



Multiple Drivers

- Wired logic
  - Buses permit one transmitter at a time
  - However, certain forms of switching circuits are designed based on wired logic and permit multiple transmitters.
  - Wires in these circuits produce AND and OR Boolean functions.
  - wired-AND logic: if at least one device driving the signal to a 0, the value of the signal will be 0.
  - HDL must be expressive enough to describe such circuits for accurate simulation.
  - See resolved signals for this.
  - Synthesis compilers produce logic that arbitrates among accesses to shared signals. For example, when only one source is permitted then a decoder must be synthesized to control multiple drivers.

Overview

- We seek to describe attributes of digital systems common to multiple levels of abstraction
  - Events, propagation delays, concurrency
  - Waveforms and timing (synchronous vs. asynchronous)
  - Shared signals
- Hardware description languages must provide constructs for naturally describing these attributes of a specific design
  - Simulators use such descriptions for mimicking the physical system
  - Synthesis compilers use such descriptions for synthesizing hardware