*Chapter 1 Introduction*

■ **Synchronization refers to** relationships among events. ■ **Synchronization constraints are** requirements pertaining to the ordering of events, examples include **serialization** (event A before event B) and **mutual exclusion** (event A and B must not happen at the same time). ■ **There are at least two different but equivalent computation models: 1) parallel processor** (multiple processors executing on different sequences of instructions) and **2) multi-threaded** (multiple threads on a single processor executing different sequences of instructions).The models are equivalent because the issue is the same; **the order of instruction execution is unknown**. ■ **Two events are concurrent** if we cannot tell from the program which will happen first. ■ **Concurrent programs are** non-deterministic. Specific order of instruction execution may change, depending on the scheduler. Challenging to debug (Heisenbugs). ■ **The execution path is** the realized sequence of executed instructions during an invocation of a program. It is an instance of all permutations of the instruction execution sequence. ■ **Atomic operations cannot be interrupted.** ■ **A high-level language may hide the atomicity of operations.** For example x = x + 1; is potentially multiple machine instructions, each of which can be interrupted.