Chapter 3-4: MoC and Programming

Soo-Ik Chae

Reactive systems

- Reactive systems
  - Computing systems which continuously interact with a given physical environment
  - when this environment is unable to synchronize logically with the system (for instance it cannot wait)
- Transformational systems
  - Classical programs whose data are available at their beginning, and which provide results when terminating.
- Interactive systems
  - Which interact continuously with environment that possess synchronization capabilities (for instance operating systems)

Synchronous languages

- The focus of synchronous languages is thus to allow modeling and programming of systems where cycle (computation step) precision is needed.
- The objective is to provide domain-specific structured language for their description, and to study matching techniques for efficient design, including compilation/synthesis, optimization, and analysis/verification.
- The strong condition insuring the feasibility of these design activities is the synchronous hypothesis.

Synchronous languages

- Imperative: Esterel, SyncCharts
  - Provide constructs to shape control-dominated programs as hierarchical synchronous automata.
- Declarative: Lustre, Signal
  - Shape applications based on intensive data computation and data-flow organization, with the control flow operating under the form of (internally generated) activation clocks.

Synchronous hypothesis

- Is really a collection of assumptions of a common nature, sometimes adapted to the framework considered.
- Instants and reactions: In each instant, input signals possible occur (for instance by being sampled), internal computation take place, and control and data are propagated until output values are computed and a new global system state is reached.
  - This execution cycle is called reaction of the system to the input signals. Reactions converge and computations are entirely performed before the current execution instant ends and a new one begins.
  - This empowers the obvious conceptual abstraction that computations are infinitely fast (instantaneous, zero-time), and take place only at discrete points in (physical) time. With no duration.

Synchronous hypothesis

- Signals: broadcast signals are used to propagate information.
  - At each execution instant, a signal can either be present or absent. A signal must be consistent for all read operations during any given instant.
- Causality: an important part of the theoretical body behind the Synchronous Hypothesis.
  - The presence status and value of a signal should be defined before they are read (and tested).
  - “before” refers to here to causal dependency in the computation of the instant, and not to physical or even logical time between successive instant.
- The Synchronous Hypothesis ensures that all possible schedules of operations amounts to the same result (convergence).

Synchronous languages

- A change in the state of one module is simultaneous with receipt of inputs.
- Outputs from a module are simultaneous with changes in state.
- Communication between modules is synchronous and instantaneous.
- Output behavior of the modules is entirely determined by the interleaving of input signals.