Design and Coding Rules

1 Design Rules

Use the well-known the design principles, design tools and design cycle.

- Design principles

Modularity	Decompose into components with well-defined interfaces
Hierarchy	Recursively apply modularity principle
Encapsulation	Hide implementation details from interfaces
Regularity	Leverage structure at various levels of abstraction
Extensibility	Include mechanisms/hooks to simplify future changes

- Design cycle
  1. Thinking: Explore design space for a new system
  2. Designing: Design and model baseline system
  3. Testing: Developing a test strategy or experiments
  4. Analyzing: results and draw conclusions (go back to 1. or 2. if not satisfied with the results)
- Design tools
  - Block-diagrams for data-paths
  - State transition graphs (STGs) or control signal tables (CSTs) for control-paths

2 Coding Rules

- Each file should begin with a header comment that contains
  - filename, author(s), date, task description, revision number with short description of changes
- For better readability
  - Keep lines in your source code to less than 80 characters
  - Use spaces and never insert any real tab characters
  - Use two spaces for each level of indentation
- Comment
  - shortly each function with it's behavior and parameters
  - only the neuralgic / critical points (the points you have taken longer to think about or you needed several attempts to get the code right)
  - each line in assembler programs
- Naming
  - File names and top-level component names should be the same
  - Use descriptive names; prefer longer descriptive names, unless some common abbreviations
  - Avoid mixing *Underscore_Naming* and *CamelCaseNaming*, use only one style
  - Constants and Generics should be in *CAPITAL LETTER*