

Finite State Machine in VHDL

Basically, FSM serves to describe the workings of a sequential circuit. Therefore, the VHDL code of an FSM is not much different from the VHDL code of an ordinary sequential circuit, which uses a process statement (behavioral model). There are many methods that can be used to create an FSM using VHDL. However, we will only learn the most basic method. The method requires a minimum of two processes, commonly called Synchronous Process and Combinatorial Process.

- Synchronous Process in the context of a Finite State Machine (FSM) is responsible for arranging state transitions based on the current state and received inputs. This process is similar to how a D flip-flop works, controlling state transitions by utilizing clock signals and memory to store the current state. Its main functions include controlling state transitions, input management, and integration with the clock and memory, ensuring that state transitions take place in a synchronized and orderly manner.
- The Combinatorial Process is responsible for generating outputs based on the current state and the inputs received. This process is similar to how logic gates work, generating outputs based on the inputs received. Its main functions include output management, integration with inputs, and decision-making based on the current state, ensuring that outputs are generated in a synchronized and orderly manner.

Finite State Machine
Image not loaded. File type unknown

Revision #2

Created 2 November 2024 08:04:24 by NS

Updated 2 November 2024 08:10:07 by NS