HDL Generator Documentation

Overview

Author

Wyatt Gronnemose

Contents

List of Figures

1 Project Introduction

1.1 Purpose

The purpose of this project is to be able to easily describe a Moore state machine and turn it into a hardware description language (HDL). The HDL can then be used with FPGA's.

1.2 System Architecture

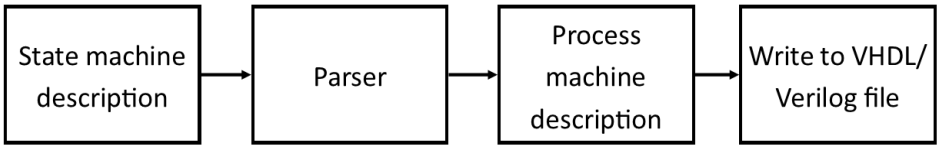

Figure 1.1: System Architecture

2 Language Structure

This section defines how you express a state machine or system of state machines. A Moore machine can be defined using a set of six items. These are:

- Finite set of states (S)
- Initial state (S_0)
- Finite input alphabet (Σ)
- Finite ouitput alphabet (Λ)
- Transition function ($T : S \times \Sigma \rightarrow S$)
- Output function ($G : S \rightarrow \Lambda$)

Graphically this is shown below in Figure 2.1.

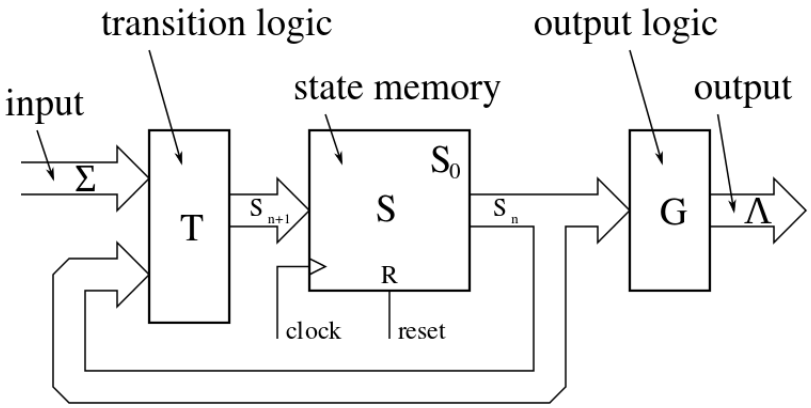

Figure 2.1: Moore Machine
Taken from::://commons.wikimedia.org/wiki/File:Moore-Automat-en.svg

2.1 State Machine Syntax

The structure of a state machine is shown in the code below. It describes the six parts of a Moore machine described in Section 2 Language Structure.

```
Machine <NAME> {
   InputAlphabet {
      <TYPE> <NAME>;
      ...
      <TYPE> <NAME>;
   }

   OutputAlphabet {
      <TYPE> <NAME> = <DEFAULT>;
      ...
      <TYPE> <NAME> = <DEFAULT>;
   }

   States {
      Names = <Comma separated list of names>;
      InitialState = <One name from above>;
      Reset = <async OR sync OR none>;
      Clock = <Name of clock signal for HDL>;
   }

   TransitionLogic {
      <StateName1>:
         if <CONDITION>:
            NextState = <STATE>;
```

```
            else if <CONDITION>:
               NextState = <STATE>;
            ...
            else:
               NextState = <STATE>;
         <StateName2>:
            if <CONDITION>:
               NextState = <STATE>;
            else if <CONDITION>:
               NextState = <STATE>;
            ...
            else:
               NextState = <STATE>;
         ...
      }

      OutputLogic {
         <StateName1>:
            <OutputName> = <VALUE>;
            ...
         <StateName2>:
            <OutputName> = <VALUE>;
            ...
         ...
      }
}
```

2.2 System Syntax