RTL Simulator User's Manual

Mohammad Bakhshalipour

July, 2014

1 General Information

1.1 Register Transfer Language

In computer science, register transfer language (RTL) is a kind of intermediate representation (IR) that is very close to assembly language, such as that which is used in a compiler. Academic papers and textbooks also often use a form of RTL as an architecture-neutral assembly language.

The idea behind RTL was first described in: Davidson and Fraser; The Design and Application of a Retargetable Peephole Optimizer; ToPLaS v2(2) 191-202 (April 1980). –Wikipedia

2 Installation

2.1 System Requirements

- Windows OS
- .NET Framework 4.5

2.2 How to Install

- Run 'setup.exe'

3 Functionalities

3.1 Simulating RTL Code

By issuing 'Run' button, the simulation will start. Whenever all *RT-Conditions* become *false*, the simulator will be halted.

3.2 Debugging Design

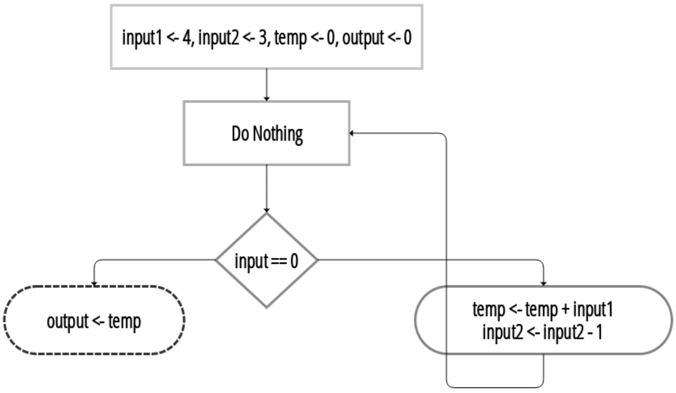
The user can debug her/his design by issuing 'Debug' button (after assembling design). Simulator executes user's code cycle-by-cycle and prints all registers/values.

3.3 Generating Synthesizable Verilog Code

Synthesizable Verilog code can be generated by issuing ‘Generate Verilog’ button. Using output code, registers can be monitored by other simulation softwares like Altera Modelsim.

4 Example

Suppose we want to design a multiplier that takes two inputs and produce the product of them in the output (i.e., $output = input_1 \times input_2$). The ASM-Chart of such system could be similar to the below design:



For designing such system via RTL, in the first step, we should specify registers and their initial values.

Register(s)	R1, R2, R3, R4, R_p, R_q
Initial Value(s)	$4, 3, 0, 0, false, true$

Then we describe the system:

Listing 1: RTL example

```
RTL Text

R_q & !R_p : R1 <- 4, R2 <- 3, R_p <- true, R3 <- 0, R4 <- 0
R_p & (R2 != 0) : R3 <- R3 + R1, R2 <- R2 - 1
R_p & (R2 == 0) : R4 <- R3, R_p <- false , R_q <- false
```

Now, the design can be simulated by the software.

5 FAQ

Q: Which operations can I use in my code?

A: Every operation that a C Compiler supports.

Q: Which syntax rules should I consider?

A: Each time you use a wrong syntax, ‘AssembleCheck’ of simulator will inform you about that and details. But a short list is here:

- Each register name should start with ‘R’ or ‘r’.
- Name of a register cannot be a substring of other(s).
- Value of registers can be 32-bit integer or boolean values (‘true’ or ‘false’).

6 Contact

Feel free to send me your feedbacks through bakhshalipour@ce.sharif.edu