RTL Simulator
USER'S MANUAL

Mohammad Bakhshalipour

Sharif University of Technology, Department of Computer Engineering

July, 2014

1.0 GENERAL INFORMATION

1.0 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. It is basically an operation which is used to transfer information from one place to another. Academic papers and textbooks also often use a form of RTL as an architecture-neutral assembly language. --Wikipedia

1.2 System Requirements

You should have .NET Framework (at least 3.0) in your system.

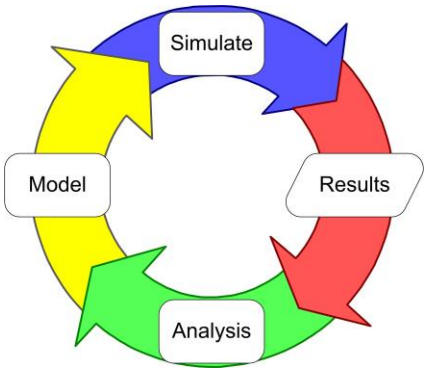
1.3 Points of Contact

For additional information, Designer can be contacted through m.bakhshalipour@gmail.com.

2.0 SOFTWARE SUMMARY

2.0 SOFTWARE SUMMARY

2.1 Software Configuration



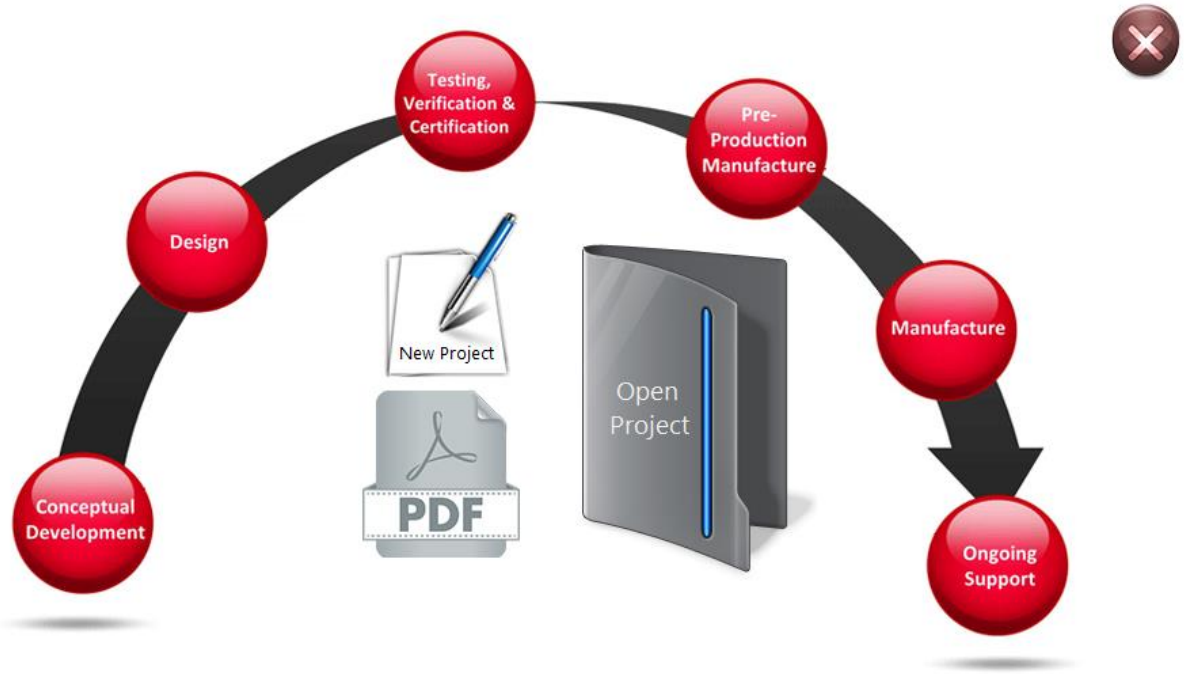
This software simulates a system described in RTL code by user. This generates a text file in programmer-defined form.

2.2 Data Flows

Simulator takes the names and initial values of registers of system and definition in RTL form and simulates till system goes to halt-state.

3.0 MAIN FORM

3.0 MAIN FORM



3.1 New Project

By clicking the 'New Project' button, current form is hided and 'New RTL File' is shown.

3.2 Open Project

By clicking the 'Open Project' button, you should choose a text file, generated by this software then flow of 'New Project' will be done.

3.3 PDF

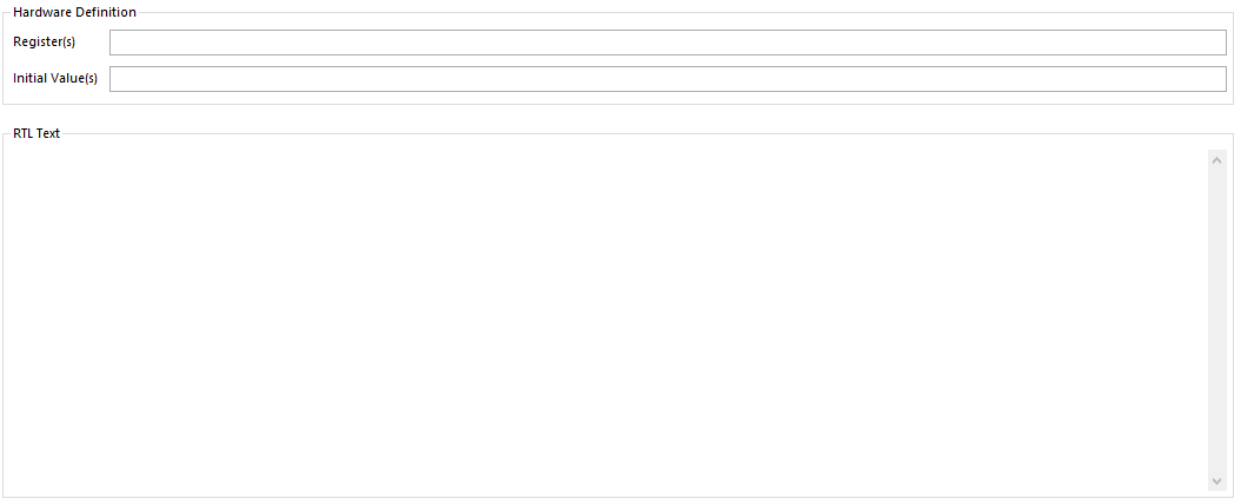
Shows the User's Manual of software (this).

3.4 Exit System

Closes the application and ends the process.

4.0 New RTL File Form

4.0 NEW RTL FILE FORM



4.1 Registers Field

You should enter the registers name in this field. First letter of a register name should be ‘r’ or ‘R’. Names should be separated by comma (,).

4.2 Initial Values Field

You should enter the initial values of registers in this field. Values should be integer for computational registers and should be ‘true’ or ‘false’ for Boolean flags. Values should be separated by comma (,).

4.3 RTL Text

Let’s have an example. We design a hardware for calculation of Multiplication of two integer values using Add-and-Subtract algorithm. Assume the Registers field is ‘r1, r2, r3, r4, r_p, r_q’ and the Initial Values field is ‘4, 3, 0, 0, false, true’. My RTL code is:

```
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
```

The syntax and priority of operations is completely C-like.

4.4 Save File

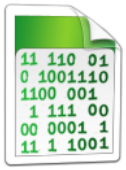
By clicking this button, your design will be saved into a text file. You should save your file every time you change your hardware description.

4.5 Assemble

After you saved your design, this button will be enabled. You should assemble your file before running or debugging that. If your design has any error, you'll see 'Error' window and cause of error.

4.6 Run

When your design has been assemble successfully, you can run your hardware by clicking this button. Running of a simulation shouldn't be more than 50 clocks of cycles. When all the conditions of your RTL code evaluated as false, simulation will be stopped. You could find final value of registers if your design has been run successfully.



```
> Running ...
> Simulation has been ended at clock cycle: 5
> Final value of registers:
(r1, 4)
(r2, 0)
(r3, 12)
(r4, 12)
(r_p, false)
(r_q, false)
> End of result
```

4.7 Debug

You can debug your design by clicking this button. When you clicked on this, a new form named 'Debug' will be shown. Every time you clicked on the 'Execute One Cycle' button, you'll see the result of simulation for one period of clock.

```
(r1, 4)
(r2, 3)
(r3, 0)
(r4, 0)
(r_p, true)
(r_q, true)

Value of registers at the end of clock cycle 2
(r1, 4)
(r2, 2)
(r3, 4)
(r4, 0)
(r_p, true)
(r_q, true)

Value of registers at the end of clock cycle 3
(r1, 4)
(r2, 1)
(r3, 8)
(r4, 0)
(r_p, true)
(r_q, true)
```

4.8 Clear All and Return

By these buttons you can clear all fields and return to Main window.