OpenROAD Improvement in Area and performance for RISCV32i Processor implemented using ASAP7 Technology

Vanshika Tanwar
Electronics and Communication Engineering
Dronacharya Group Of Institutions, Greater Noida
vanshikatanwar30@gmail.com

***Abstract*— OpenROAD and OpenROAD Flow Scripts are combinely and separately both are an open-source tool that is used for generating the RTL to GDS Flow without any inhuman involvement in between the flow. This paper describes the suggested different solutions by reducing the runtime, improving area or power performance, and also, trying for making the docs for the RISCV32i processor which is already implemented using ASAP7 using OpenROAD Flow Scripts and OpenROAD.**

Keywords*— *OpenROAD Flow Scripts, RTL-to-GDSII flow, open-source tools, automated design, no-human-in-the-loop, Verilog file, ASAP7,etc..

I. INTRODUCTION

. OpenROAD is an efficient open-source tool that is used for generating the RTL-to-GDSII flow which provides an implementation of the layout for any digital system design.

So, here I have taken the RISCV32i processor in which I am going to improve its area basically optimizing the area, and frequency and also going to reduce the running time of the whole circuit for getting the fastest run time . Also, try to improve or optimize its power performance and also try to improve the documentation of OpenROAD documentation.

II. IMPROVEMENT

The suggested improvement in the Riscv32i will be going to do changes when trying the solution in reducing the timing parameters, run time improvement, its die area and area, by doing changes in it's configuration file of the already existing Riscv32i processor and if possible then will also try to improve the power performance by doing some changes in its scripts and also in it's python file and make file.

The configuration file of riscv32i is shown below and the marking of blue line shows where I am going to do changes for area improvement of riscv32i:

```
export DESIGN_NICKNAME = riscv32i
export DESIGN_NAME = riscv_top
export PLATFORM    = asap7

export VERILOG_FILES = $(sort $(wildcard ./designs/src/$(DESIGN_NICKNAME)/*.v))
export SDC_FILE      = ./designs/$(PLATFORM)/$(DESIGN_NICKNAME)/constraint.sdc
export ADDITIONAL_LEFS = ./platforms/$(PLATFORM)/lef/fakeram7_256x32.lef
export ADDITIONAL_LIBS = ./platforms/$(PLATFORM)/lib/fakeram7_256x32.lib

export DIE_AREA = 0 0 120 120
export CORE_AREA = 5 5 110 110

export PLACE_DENSITY = 0.65

export SYNTH_HIERARCHICAL = 1
export RTLMP_FLOW = True
# RTL_MP Settings
export RTLMP_MAX_INST = 10000
export RTLMP_MIN_INST = 5000
export RTLMP_MAX_MACRO = 4
export RTLMP_MIN_MACRO = 1
export HAS_IO_CONSTRAINTS = 1
export PLACE_PINS_ARGS = -exclude left:* -exclude right:* -exclude top:*
```

III. CONCLUSIONS AND RECOMMENDATIONS

The conclusion of this is that we are going to reduce the area and power performance and also try to reduce the run time and power of the RISCV32i processor which is already implemented in the OpenROAD Flow Scripts using ASAP7 technology. So, the main focus of this paper is that going to improve power performance and reduce the Area. And, if possible in the end, then, we are going to also improve the runtime and the docs of the OpenROAD Flow Script tool.

IV. REFERENCES

[1] OPEN ROAD- Key Milestones on the ROAD towards Good PPA
https://theopenroadproject.org/openroad-key-milestones-on-the-road-towards-good-ppa/

[2] OPEN-ROAD – An Opensource Autonomous RTL -GDSII Flow For VLSI Designs by Indira Iyer Almeida and Matt Liberty
https://ospo.ucsc.edu/project/osre23/ucsd/openroad/

[3] AE-AVI Encoder implementation: Using OpenROAD to achieve Real-time Throughput. By Tulio Pereira Bitencourt
https://theopenroadproject.org/ae-av1-encoder-implementation-using-openroad-to-achieve-real-time-throughput/