

MSc Individual Research Project

Initial written report

**Shun Wan**

**01538313**

**Department of Electrical Engineering and Electronic**

**10 March 2019**

# Background and motivation

Fuzz testing, fuzzing is a kind of software test technique. Providing invalid, unexpected or random data as input to a C compiler, for example, GCC and LLVM. In this fuzzing netlist, the fuzz test objective is focusing on low level hardware synthesis software for example, ABC and SIS. The random netlist description files will be entered to two hardware synthesis software and if the synthesised output circuits are not equivalent, it indicates that might be a bug existing in these softwares.

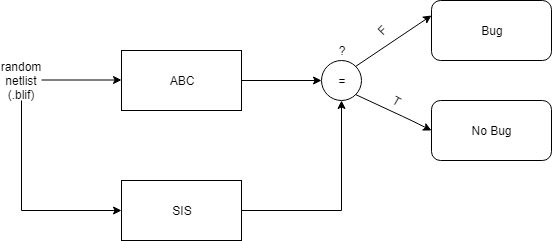
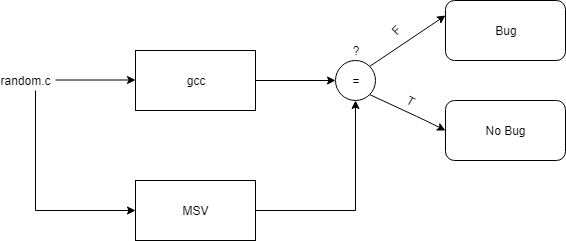


Figure : Fuzzing test block diagram

The motivation of this project is providing fuzzing test algorithm in hardware synthesis software such as ABC and SIS (even VIVADO) rather in software compiler. According to research, fuzz test has rarely been applied in the hardware area. The existing fuzzer Yosys, VirlogHammer only support the test of Verilog file which means the test range is limited. The purpose of this project is to realize the This project is also an improvement of the fuzzing test, because it broadened its application field.

# Literature review

As Xuejun Yang, Yang Chen mentioned in their report, they designed a fuzzer called Csmith. The design of this fuzzer considered most of the C language and reported more than 325 bugs. However, some important aspect such as strings and dynamic memory allocation are not covered in the C smith. The algorithm of randomly generating program is the core part of this fuzzer and could be referred to the generating of the random netlist.

The BLIF is an abbreviation of the Berkeley Logic Interchange Format which traditionally used by SIS, VIS, and MVISS to represent logic networks (logic-level hierarchical circuit in textual form). In 1992 July 28's report presented by University of California Berkeley described the specific definition of BLIF, the circuit was regarded as a directed graph of combinational logic nodes and sequential logic elements. Models, Logical gates and Latches were clearly described in this document. The BLIF is the input of the netlist synthesis software.

ABC is similar to SIS/MVSIS, (Multi-Valued Logic Synthesis) which processes the design by applying a sequence of transformation to the current network created by the design specification form file (BLIF). The vision of the ABC used in this project is Windows version abc10216. The latest version of the abc was provided on Github. The Programmer’s Manual wrote in December 25, 2006 introduces the ABC internal data representation and illustrate the difference between the SIS/MVSIS. Netlist is an original network representation. In another word, the netlist is an one-to-one correspondence with the deigan specification form file (BLIF). It should be noticed that the ABC only support the BDDs(Binary decision diagram) format of the netlist. Besides, each net has uniqueness and the output of it is single. The standard of ABC will always automatically transform the netlist into logic network.

Yosys is a current existing recursive Verilog test software to verify the front circuit correctness of the Verilog synthesis circuit (Quartus).

# Milestones

The core objective of this project is generating simple combinational logic circuit (No clock signal) without high level elements (flip-flop, register, ROM). The extension objective is constructing more complex circuit (NOT guarantee supported by the ABC). Optimize the generated circuit with circuit folding method. Using sequential (Synchronous, Asynchronous more than one clock signal) flip-flop, register, adder, mux and other logical elements. The input/output can be realized in tri-state. (1,0, x, z). The following table shows the weekly plan of this project.

|  |  |
| --- | --- |
| Week 6 (11 Feb) | * 1. Analyse the ABC software (Instruction code)   2. Generate self-programmed aig/blif file |
| Week 7(18 Feb) | * 1. Analyse the random file generation   2. VlogHammer open source analysis |
| Week 8(25 Feb) | * 1. Understand the standard of the random netlist structure. |
| Week 9(4 March) | * 1. Further research or literature review (Due to the coming report submission) |
| Week 10(11 March) | \*Need submit a 1-2 pages report.   1. Finish the writing of the short report |
| Week 11(18 March) | * 1. If above going well, synthesis the code to finish the first part. |
| Week 12(25 March) | * 1. Considering another synthesis tool using same input aig/blif |
| Week 13(1 April) | * 1. How to compare the results of two different netlist synthesis tool. |
| Week 14(8 April) | Easter |
| After Exam | Start extension objectives |
| July | Finish the whole project |
| August | Finish the final thesis writing |