

Verificación de circuitos digitales con software libre

Eduardo Vázquez Díaz
lalohao@gmail.com

CONTENIDO

I. Introducción	1
I-A. Virtualización	1
I-B. Requisitos	1
I-B1. Verilator y GTKwave .	1

Referencias 1

Resumen—Se creó una maquina virtual con *Ubuntu Desktop 16.04.1 LTS* en un contenedor utilizando el software de virtualización *Qemu*, donde posteriormente se instaló *verilator* y *gtkwave*; a partir de este sistema se exponen algunas técnicas para simular circuitos con *verilog/C++*, además de visualizar las ondas generadas de manera gráfica.

I. INTRODUCCIÓN

La importancia de probar los circuitos antes de ser llevados al silicio puede representar millones de dolares, sin contar el tiempo invertido en el diseño, y el que se necesitará volver a invertir para arreglarlo.

En 1990 el lenguaje de descripción de hardware mas usado era VHDL, a pesar de que solo tenia constructores básicos para probar los circuitos (TestBench). Los diseños empezaban a crecer y nuevo software comercial se creaba para compensar, algunas empresas invertían horas de trabajo para crear su propio sistema y no pagar los miles de dolares en licencias, una de ellas llevó a la creación de Accelera que fue la base de SystemVerilog [4].

De la misma manera surgió Verilator, un simulador potente de Verilog HDL que además es software libre, este compila el código y lo optimiza para ser simulado rápidamente [1], en algunos casos es incluso mas veloz que los simuladores comerciales [2].

I-A. Virtualización

La maquina virtual permite encapsular nuestro sistema de verificación en un contenedor que no será afectado (y que no afectará) la maquina utilizada, esto elimina errores que podrian ser causados al tener instalado software que utilice configuraciones globales (PATHS) como ocurre con HSPICE y Questa SIM por ejemplo.

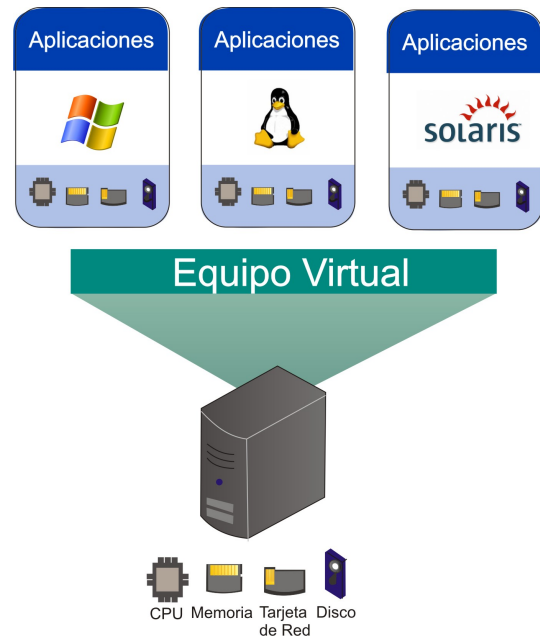


Figura 1. Las maquinas virtuales pueden o no conectarse entre ellas o hacia la red externa y pueden ser de diferentes arquitecturas y sistemas operativos independientemente del sistema anfitrión.

Se le dice anfitrión a la maquina donde se encuentran los contenedores virtuales, en este caso la anfitriona usa *Arch Linux*, pero esto no afecta a los contenedores ya que estan aislados, esto aplica de igual manera para Windows o Mac.

I-B. Requisitos

En la maquina virtual (Ubuntu) se instala el software necesario para simular, al igual que el editor de texto de su preferencia para modificar los archivos.

I-B1. Verilator y GTKwave: Se pueden instalar desde la terminal con el siguiente comando [3]:

```
sudo apt-get install git make autoconf \
g++ flex bison verilator gtkwave
```

REFERENCIAS

- [1] <https://www.veripool.org/wiki/verilator> Last accessed Sat Jan 28 12:01:25 2017.
- [2] https://www.veripool.org/wiki/veripool/Verilog_Simulator_Benchmarks Last accessed Wed Feb 1 20:41:50 2017.
- [3] <https://www.veripool.org/projects/verilator/wiki/Installing> Last accessed Sat Jan 28 12:25:07 2017.
- [4] Chris Spear. *SystemVerilog for verification a guide to learning the testbench language features*. Springer, 2008.