Guía Proyecto 2

Andrés Rojas Barboza
Francisco Coto Alcázar

25 de noviembre de 2025

1. Guía de ejecución del ambiente de pruebas

Esta sección describe paso a paso cómo compilar y ejecutar el ambiente de verificación del Proyecto 2 utilizando VCS y UVM, así como la generación de reportes y gráficos a partir de los resultados.

2. Repositorio

El repositorio del proyecto se encuentra en:

```
https://github.com/AndresRB506/Proyecto-2-Veri.git
```

Una vez clonado el repositorio, debe posicionarse en el directorio raíz del proyecto, de forma que la jerarquía sea la siguiente:

```
rtl scripts sva tb uvm
```

3. Pasos para ejecutar el ambiente

3.1. Cargar herramientas de Synopsys (VCS)

Antes de compilar, es necesario cargar el entorno de VCS mediante el siguiente comando:

```
source /mnt/vol_NFS_rh003/estudiantes/archivos_config/synopsys_tools2.sh
```

Esto configura las variables de entorno necesarias para el uso de VCS.

3.2. Compilación del ambiente

Ejecute el siguiente comando para compilar el diseño y el ambiente de verificación:

vcs −sverilog +define+DEBUG −full64 −timescale=1ns/1**ps** −ntb_opts uvm −cm line+cond+tgl+fsm −cm_dir ./cm_out −f rtl/filelist.f −f uvm/filelist.f uvm/agent/mesh_driver.sv uvm/scoreboard/mesh_scoreboard.sv uvm/reporte/metrics_collector.sv uvm/**env**/mesh_env.sv uvm/tests/algn_cov_suite.sv sva/∗.sv tb/tb_top.sv −o simv

Este comando generará el ejecutable `simv`, el cual se utilizará para correr las simulaciones.

3.3. Ejecución de la simulación

Dependiendo del tamaño de la malla, se puede ejecutar el ambiente con distintas configuraciones:

3.3.1. Caso 4x4

```
./simv +UVM_TESTNAME=algn_cov_suite +ROWS=4 +COLUMS=4
    +PCKG_SZ=40 +FIFO_DEPTH=4
+BDCST=255 +TXS_PER_TERM=5 +MIN_GAP=0 +MAX_GAP=2
    +ERR_RATE_PCT=5
+CSV_PATH=cm_out/metrics.csv +UVM_TIMEOUT=100s
    +UVM_VERBOSITY=UVM_HIGH
+define+DISABLE_SVA +ntb_random_seed=auto
-cm line+cond+tgl+fsm -cm_dir ./cm_out/4x4_s_auto
```

3.3.2. Caso 6x6

Caso 6x6

```
./simv +UVM_TESTNAME=algn_cov_suite +ROWS=6 +COLUMS=6
    +PCKG_SZ=40 +FIFO_DEPTH=4
+BDCST=255 +TXS_PER_TERM=5 +MIN_GAP=0 +MAX_GAP=2
    +ERR_RATE_PCT=5
+CSV_PATH=cm_out/metrics.csv +UVM_TIMEOUT=100s
    +UVM_VERBOSITY=UVM_HIGH
+define+DISABLE_SVA +ntb_random_seed=auto
-cm line+cond+tgl+fsm -cm_dir ./cm_out/6x6_s_auto
```

3.3.3. Caso 8x8

```
./simv +UVM_TESTNAME=algn_cov_suite +ROWS=8 +COLUMS=8
    +PCKG_SZ=40 +FIFO_DEPTH=4
+BDCST=255 +TXS_PER_TERM=5 +MIN_GAP=0 +MAX_GAP=2
    +ERR_RATE_PCT=5
+CSV_PATH=cm_out/metrics.csv +UVM_TIMEOUT=100s
    +UVM_VERBOSITY=UVM_HIGH
+define+DISABLE_SVA +ntb_random_seed=auto
-cm line+cond+tgl+fsm -cm_dir ./cm_out/8x8_s_auto
```

4. Generación de reportes

La simulación genera automáticamente un archivo CSV en la ruta:

`cm_out/metrics.csv`

Este archivo contiene las métricas recolectadas durante la simulación, tales como ancho de banda y retrasos en la entrega de paquetes.

5. Generación de gráficas con GNUplot

Para generar las gráficas a partir del archivo CSV, ejecute:

```
gnuplot scripts/bw_plot.gnuplot gnuplot scripts/bw_plot.gnuplot
```

Esto generará dos archivos `.png`, los cuales pueden visualizarse con:

```
xdg-open <archivo.png>
```

6. Archivos de cobertura y cm.log

El archivo `cm.log` se genera porque la simulación se ejecuta con la opción `-cm`, la cual habilita la cobertura en VCS.

Cuando VCS corre con cobertura habilitada:

- Crea un directorio de cobertura (por ejemplo: `-cm_dir ./cm_out/4x4_s_auto`).
- Dentro de este directorio se almacenan:
  - `cm.log`: Log del motor de cobertura, que indica qué se recolectó, así como errores y advertencias.
  - Base de datos de cobertura (por ejemplo, `urg.vdb`), utilizada posteriormente para análisis con la herramienta `urg`.

El archivo `cm.log` es útil para:

- Verificar que la cobertura se habilitó correctamente.
- Detectar problemas como módulos sin instrumentación o fallos en la recolección de métricas.

Con estos pasos, el ambiente de pruebas queda correctamente configurado, ejecutado y listo para el análisis de resultados y cobertura.