Simulación y Debugging: Xcelium & Simvision

April 23, 2024

1 Ejecutar la Simulación en entorno Cadence

Utilizaremos el comando **xrun** para ejecutar simulaciones. Este requiere de una serie de argumentos para poder compilar y simular el código correctamente. Por simplicidad, guardaremos todos estos argumentos en un archivo llamado `sim.f` y ejecutaremos el simulador de la siguiente manera:

```
xrun -f sim.f
```

El contenido del archivo sim.f es:

```
# General xrun options
# Inicializar con SimVision, sino esta: solo simula
-gui
# Enable line debug (breakpoints, etc)
-linedebug
# Debugging Access
-access +rwc
-nolibcell
# Esperar por la licencia
-licqueue
# SystemVerilog enable
-sv
# Definimos el timescale
-vtimescale 1ns/1ps
# 64bit simulator
-64bit
# Donde guardamos el log
-l ./xrun.log

# Control de Errores, suprimimos algunos
# le damos mas importancia a otros
-xmfatal MACRDF
-xmerror POLICI
-xmerror CUVMPW
-nowarn LIBNOU
-nowarn SPDUSD
-nowarn DSEMEL
-nowarn ZROMCW
-nowarn MEVCON
-nowarn DSSUNC
-nowarn BADPRF
-nowarn DSEM2009
-nowarn COVDNC
-nowarn CLIBNF
-nowarn ENVDEPRREN
-nowarn CLEAN
-nowarn DFSVOL
-nowarn COVFHT
# Seed de la simulacion,
# Fundamental si utilizamos random values
# y queremos replicar resultados
# -> Stable Random
-seed 1
# Probe -> Ejecutamos un archivo tcl inicial donde
# le decimos al simulador que guarde todas las signals
-input probe.tcl
# DUT
../rtl/ssr.v
# Testbench
../tb/ssr_tb.v
```

Listing 1: sim.f Argumentos para XCelium

El contenido del archivo probe.tcl es:

```
# Probing design section
database -open waves.shm -event
probe -create testbench -depth all -tasks -waveform -functions -all -dynamic -memories -database
    waves.shm
run
```

Listing 2: probe.tcl Habilitar Simvision Probe

2 Simvision

2.1 Formas de Ejecución

- Podemos ejecutar SimVision para realizar simulaciones interactivas y debuggear el diseño mediante la consola de linux:
  `xrun -gui &`
  Esto inicia el simulador con SimVision y permite utilizar todas las herramientas de debugging disponibles.
- O bien podemos abrir la base de datos de una simulación ya realizada mediante: `simvision path_waves.db &`. En este caso el usuario ya no tiene mas control de debugging. Solo puede visualizar y utilizar lo existente.
- También es posible ejecutar una simulación con Xcelium y luego abrir SimVision y conectarte a esa simulación en curso.

2.2 Preparando los archivos fuente para Debugging

- `xrun -linedebug`: Permite que podamos utilizar breakpoints en el código fuente durante debugging.
- `xrun -access +rwc`: Los diseños pueden contener múltiples señales internas, esta opción permite hacerlas visibles durante debugging y poder ver su conectividad. El atributo `rwc` habilita: Read Write Connectivity

2.3 Debugging Interactivo de Simulaciones

- Design Browser
- Console Windows
- Waveform Viewer Windows
- Schematic Tracer

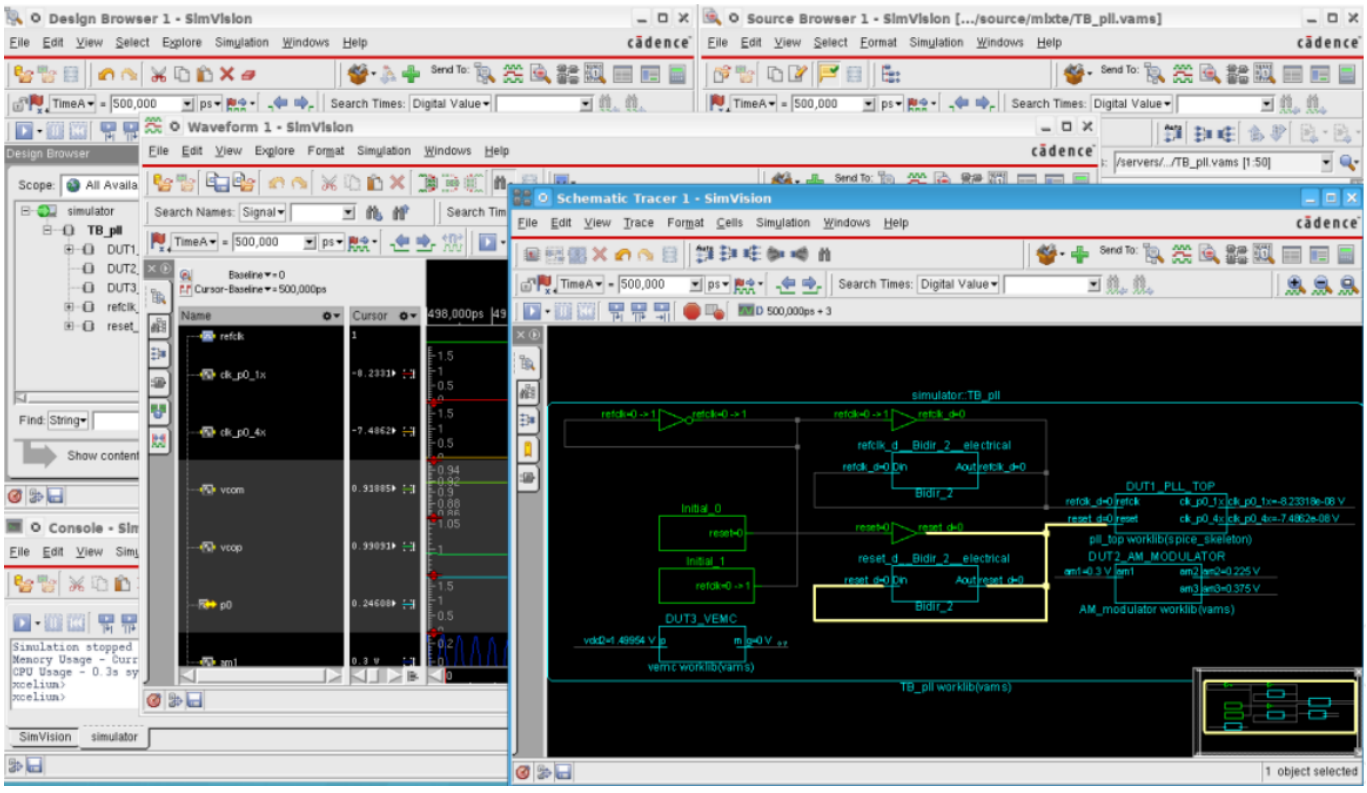

Figure 1: Simvision Tool

2.3.1 Tools for Simulation and Debug

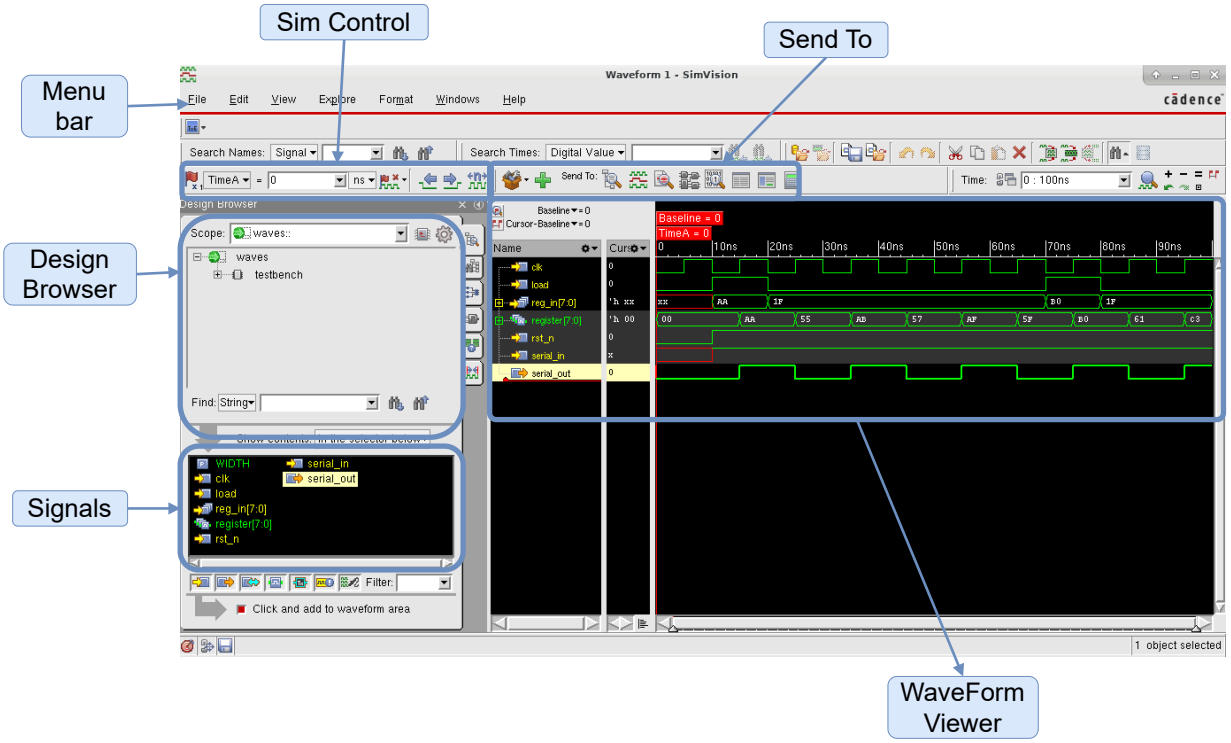

Figure 2: Detalle Herramientas Simvision

2.3.2 Design Browser

- Si ejecutamos `xrun -gui` esta ventana se abre en *startup* y te permite acceder a los objetos del diseño jerarquicamente.
- La jerarquía mas alta la llama *Simulator* y es la simulación actual.

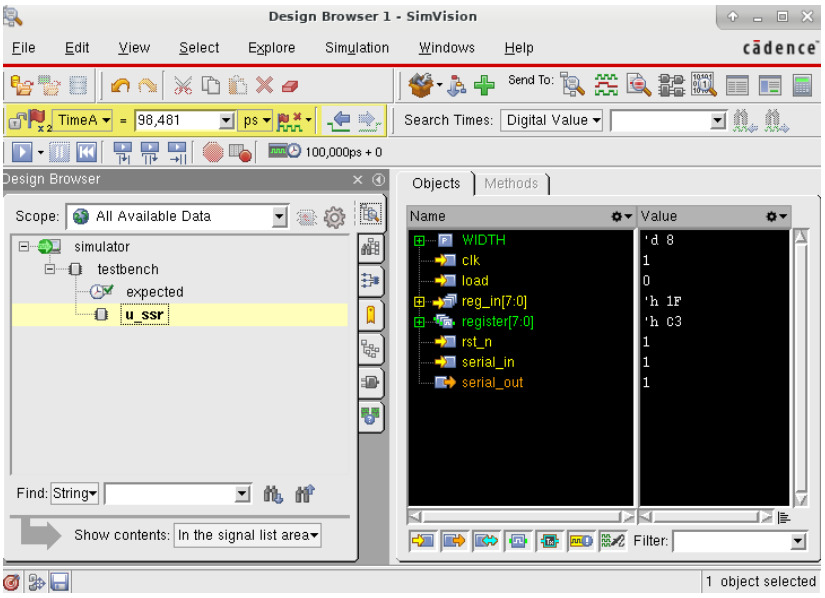

Figure 3: Design Browser

2.3.3 Toolbar Selection Buttons

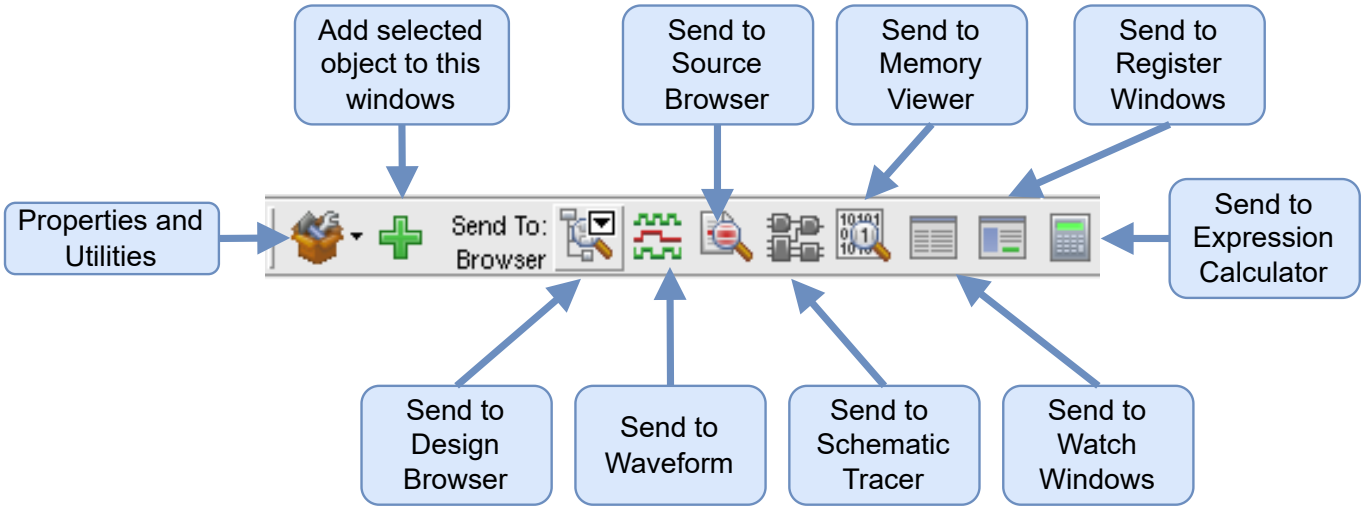

Figure 4: Toolbar: Send To

2.3.4 Console Window

La ventana de Consola arranca en startup si ejecutamos `xrun -gui` y tiene 2 *Tabs*: *SimVision* and *simulator*

- SimVision: permite ingresar comandos especificos de de simvision
- simulator: permite ingresar comandos para correr la simulacion y controlar su ejecución.
- Si se ejecuta SimVision en modo postprocessing, los controles de simulación no estarán presentes ya que estamos viendo un snapshot de una previa ejecución.

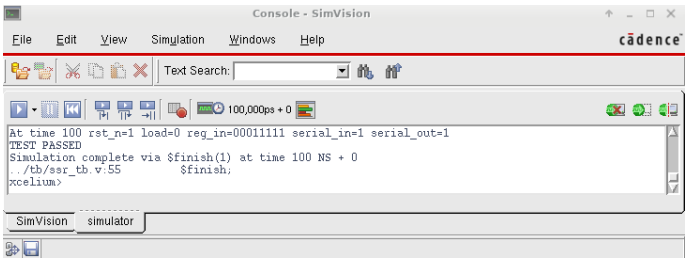

Figure 5: Console Window

2.3.5 Source Browser

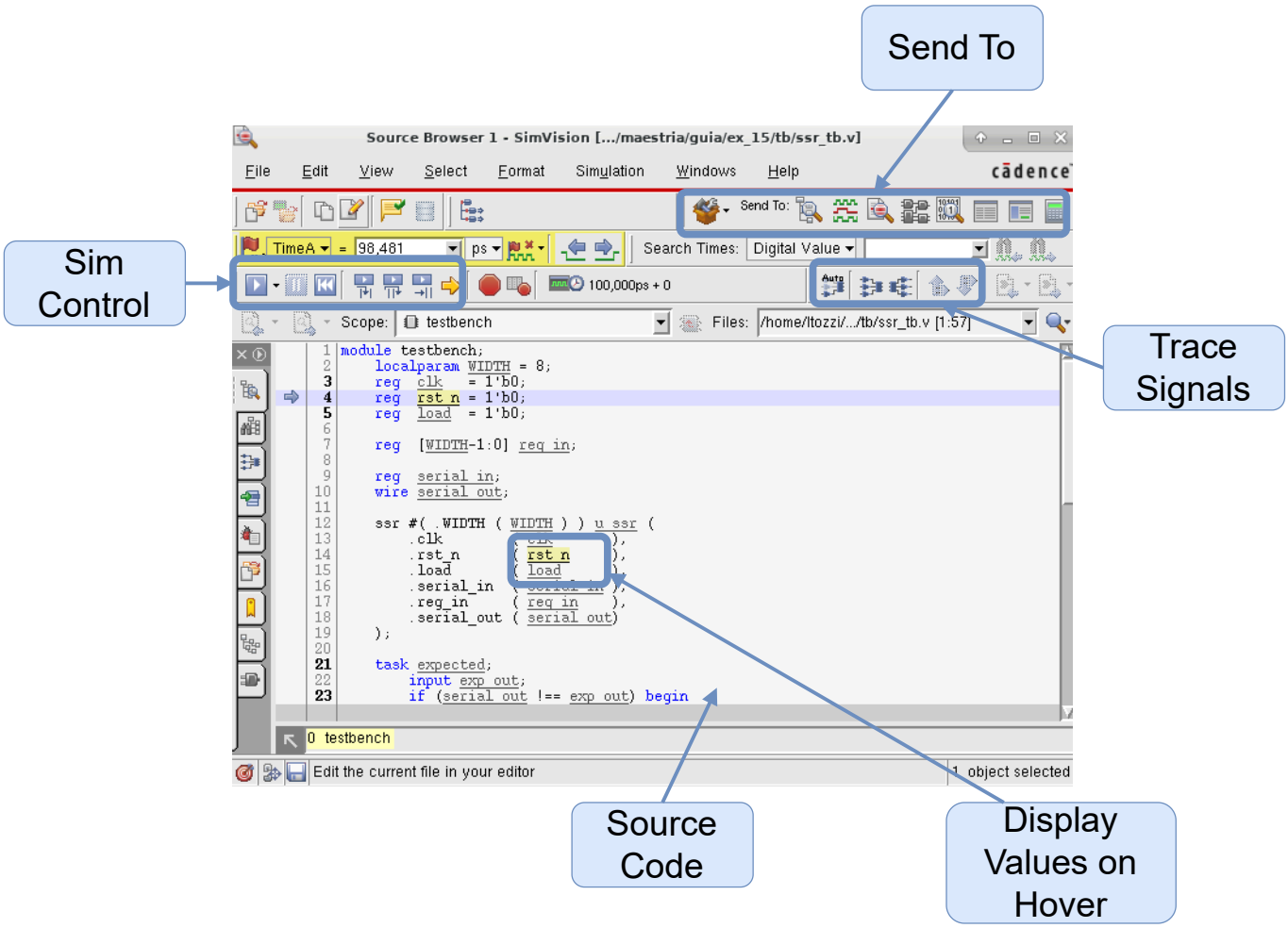

Figure 6: Source Browser

2.3.6 Tracing Signal Path

- Por default, si hacemos doble click en la señal, el source browser nos direcciona al driver más probable de la señal.
- Podemos utilizar los botones Trace Driver and Trace Loads para recorrer el diseño.

Figure 7: Trace Tool

2.3.7 BreakPoints

Los breakpoints se utilizan para controlar cuando parar y volver a ejecutar una simulacion en puntos determinados para poder examinar el estado del diseño en ese punto. Existen distintos tipos de breakpoints:

- **Time Breakpoint**: Tiempo determinado o en un intervalo
- **Signal Breakpoint**: Basado en el valor de una señal.
- **Condition Breakpoint**: Condicion especifica. Expresión TCL

- **Line Breakpoint**: Cuando se alcanza cierta línea en el código
- **Power Domain Breakpoint**: No veremos este tipo de simulaciones en esta materia pero la simulación se para cuando se alcanza un power domain específico.

Para setear un breakpoint podemos ir a Menu: ***Simulation - Set Breakpoint - tipo de breakpoint***

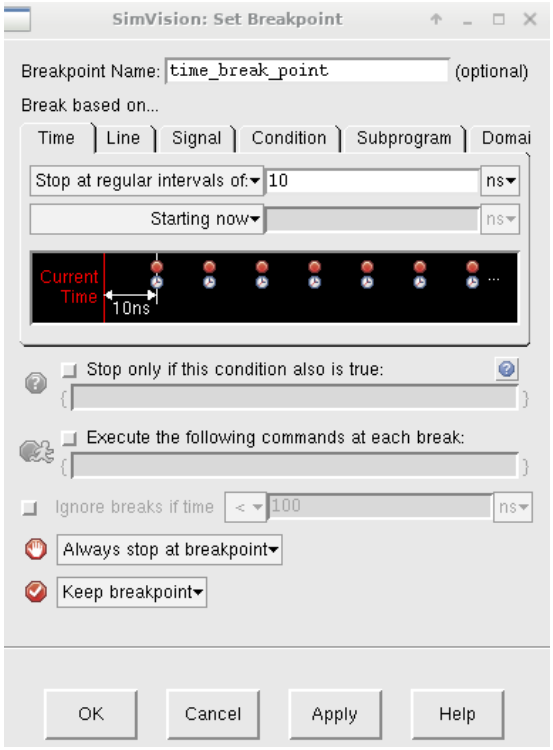

Figure 8: Ejemplo Breakpoint de tiempo

Podemos visualizar y controlar los distintos breakpoints del diseño mediante Menu: ***Simulation - Breakpoints*** Donde podemos crear nuevos breakpoints, borrar existentes, habilitarlos y deshabilitarlos.

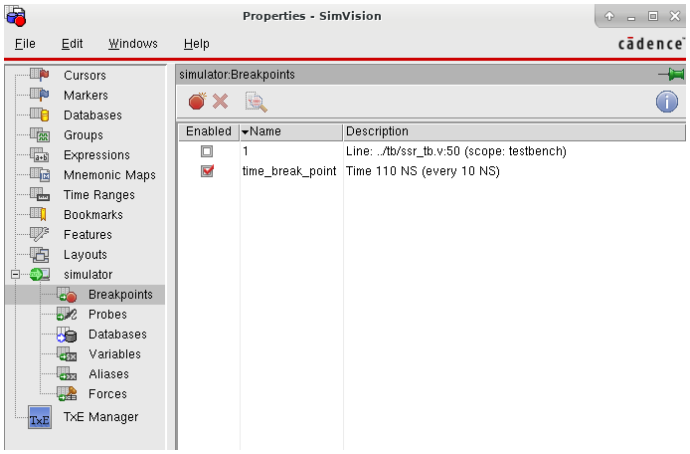

Figure 9: Breakpoint summary

2.3.8 Visualización Delta Cycles

- Podemos acceder a esta herramienta mediante Menu: ***Windows - Tools - Simulation Cycle Debug***
- Esta herramienta permite simular desde un ***delta cycle*** hasta el siguiente.

- En cada momento, muestra las tareas programadas para ejecutarse durante el delta cycle actual y las tareas programadas para ejecutarse al final del tiempo en 2 subventanas diferentes.
- Un diagrama de estado presente debajo de la barra de herramientas de control de simulación muestra la lista de statements que se ha ejecutado durante el delta cycle actual.

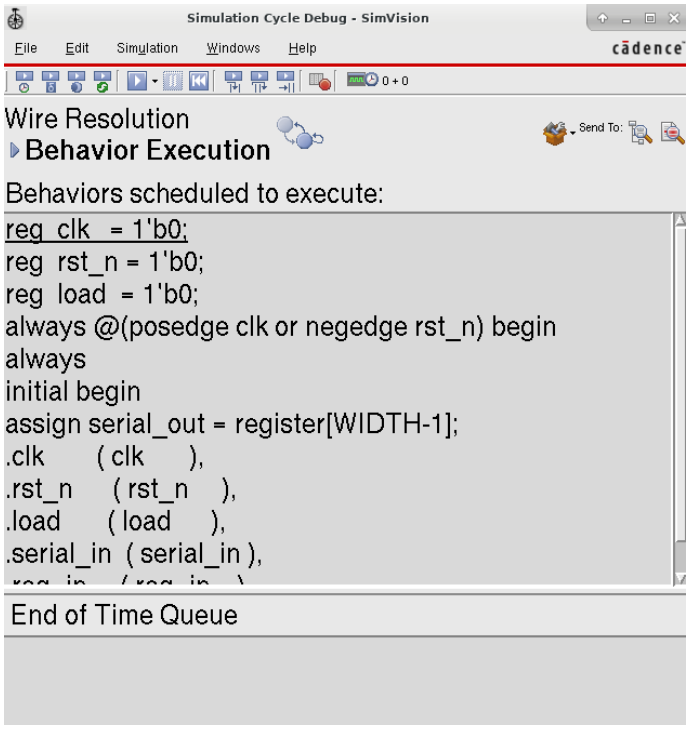

Figure 10: Delta Cycles

2.4 WaveForm Viewer

Para visualizar el tiempo de la simulación podemos utilizar cursores, el cursor baseline (único, solo tenemos uno) y marcadores. Los controles de Zoom son muy importantes para navegar por las formas de ondas y tenemos disponibles:

- Zoom in/out
- Zoom to fit x/y axes (Muy útil)
- Zoom Between cursors
- Toggle between set time ranges

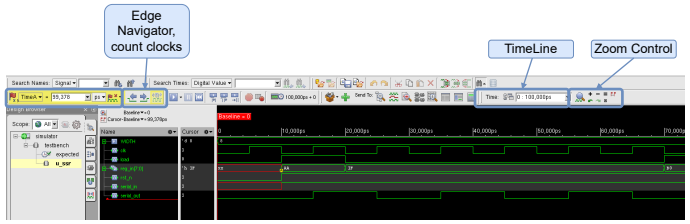

Figure 11: WaveForm Viewer

2.4.1 Cómo agrupar y Formatear señales

Un grupo le da un nombre a una colección de señales que pueden manipularse como una sola entidad. El orden y la base de un grupo de señales se pueden cambiar aplicándole formatos. En el waveform viewer sobre la señal de interés hacemos: ***boton derecho - Create - Seleccionamos la opcion***

Al Crear un ***bus*** en un grupo de señales digitales las combinará en una única forma de onda.
Podemos cambiar la base numérica de los valores utilizando `Format - Radix/Mnemonic` Se puede invertir el orden de un bus digital eligiendo `Format { Reverse Bus`

Figure 12: Group Singals

2.5 Schematic Tracer

Herramienta de SimVision que construye el diagrama de bloques de un diseño con sus interconexiones. Permite ver el ***signal path*** a través del diseño.
Proporciona una completa visualización de los modelos abstractos, gatelevel y diseños a nivel de transistor en un esquemático.

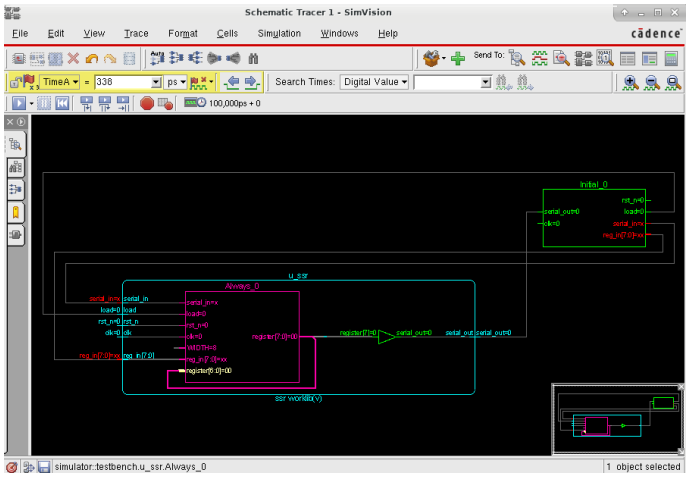

Figure 13: Schematic Viewer