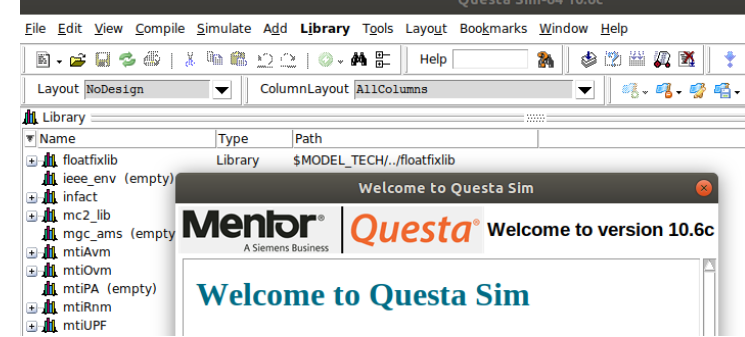


Arquitectura de Computadores

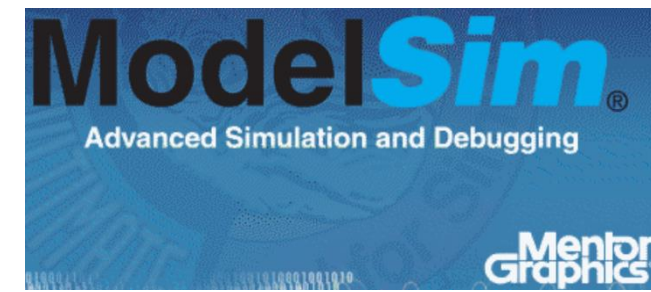
Instalación y uso de Mentor (Siemens EDA) ModelSim/QuestaSim

3º de grado en Ingeniería Informática y
3º de doble grado en Ing. Informática y Matemáticas

Generalidades



- Esta presentación describe el uso básico de QuestaSim/Modelsim
- Se tratan de entornos de simulación HDL (*Hardware Description Languages*) en varios lenguajes (VHDL, Verilog y SystemC) de Mentor Graphics (Ahora Siemens EDA).
- Questa/ModelSim se puede utilizar de forma independiente o junto con otras herramientas de diseño hardware como Intel Quartus, Xilinx ISE o Xilinx Vivado.
- La simulación se realiza mediante la interfaz gráfica (GUI- *Graphical User Inteface*) o automáticamente mediante scripts.



Generalidades

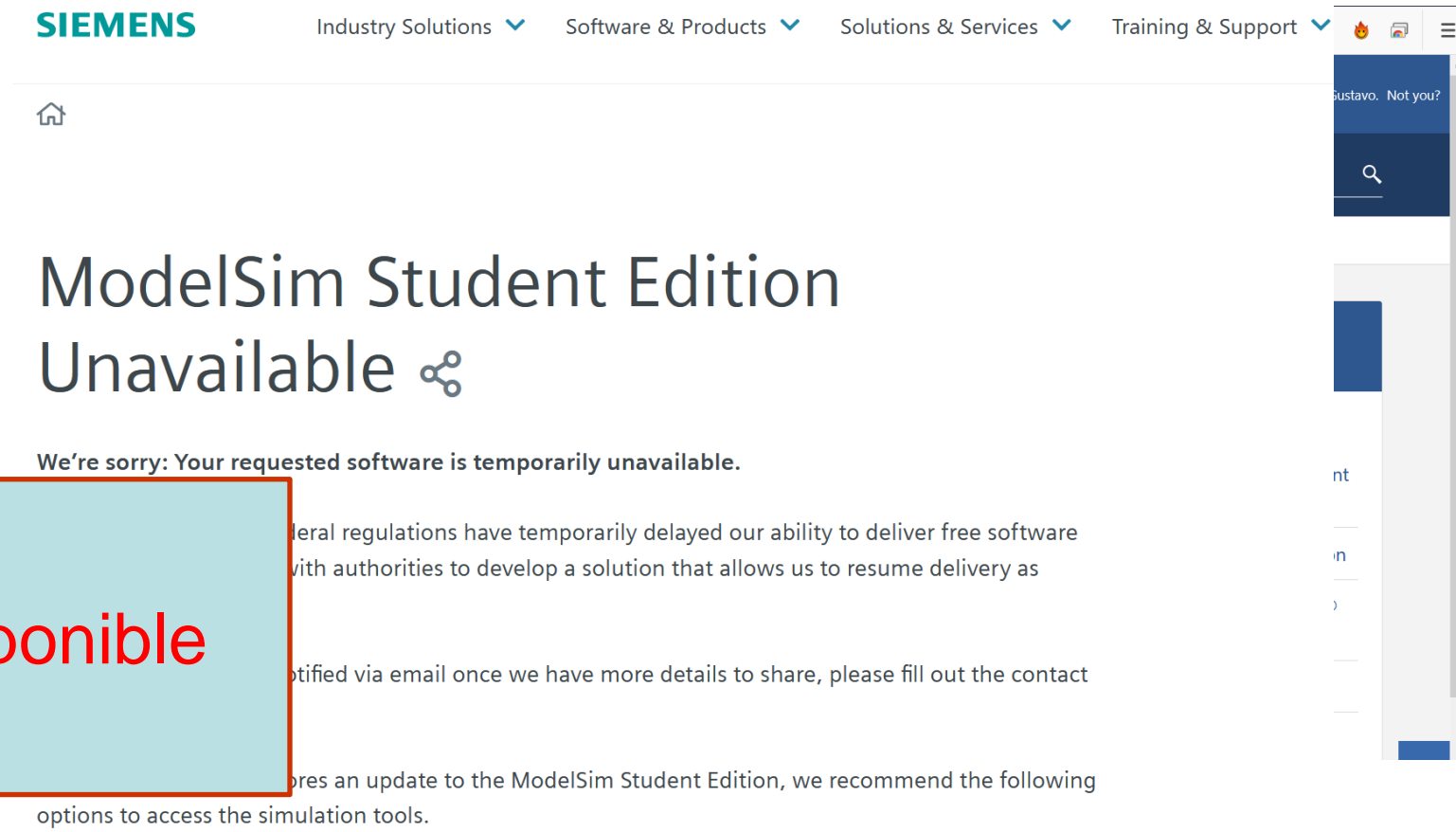
- En la EPS-UAM existen instalaciones de QuestaSim o Modelsim
 - Se trata de versiones completas que utilizan licencias desde un servidor y no pueden usarse fuera del entorno de la universidad
 - En las imágenes Linux de los laboratorio existe QuestaSim (recomendado)
 - En algunas imágenes Windows, existen instalaciones de ModelSim SE.
- Para trabajar desde casa:
 - Versión gratuita (ModelSim PE Student Edition) de Modelsim. **De momento no está disponible.**
 - Versión prueba provista por Intel, Microsemi o Lattice.
 - Máquina Virtual provista por la asignatura. Necesita VPN

Instalación de ModelSim Student Edition

1. Buscar en internet “modelsim student” o seguir el enlace

https://www.mentor.com/company/higher_ed/modelsim-student-edition

Se trata de una versión
solo Windows



De Momento no está disponible

ModelSim Altera/Microsemi/Lattice Edition

- Para usuarios avanzados. Versiones Linux y Windows.
- Licencia de prueba de algunos meses
- Ejemplo Intel (Antigua Altera):

<https://www.intel.es/content/www/es/es/software/programmable/quartus-prime/model-sim.html>

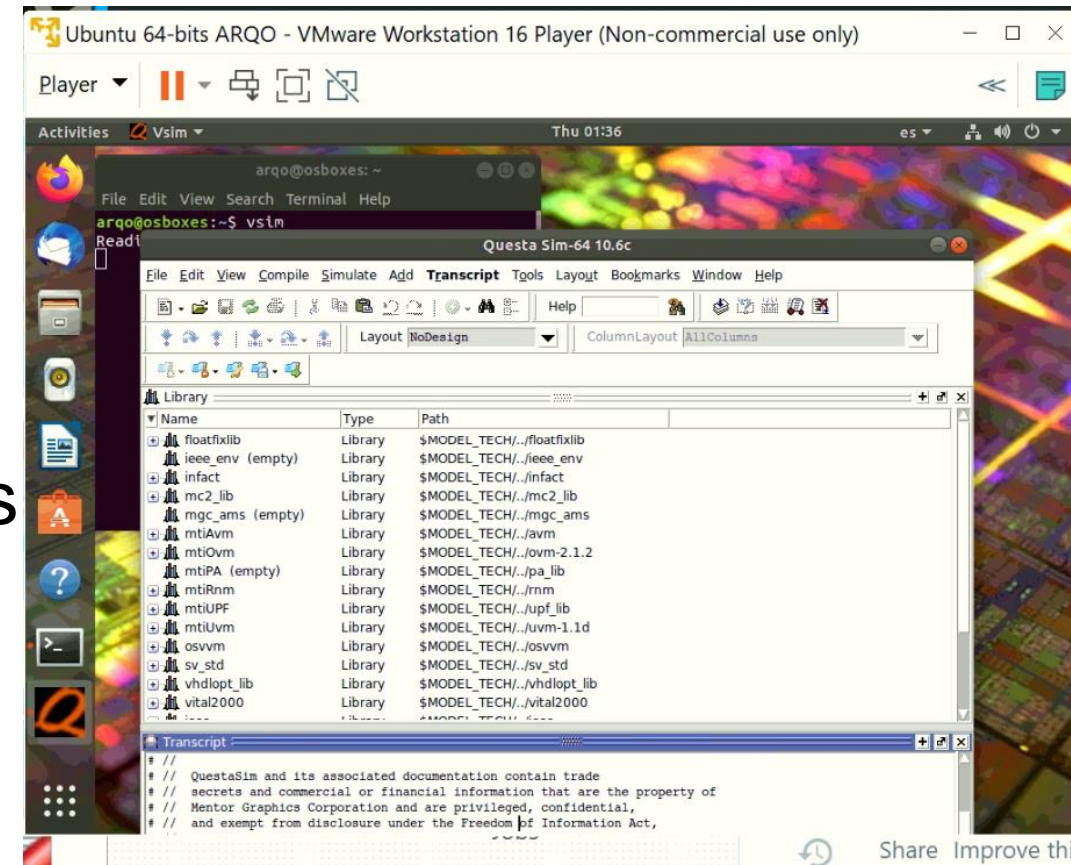
ModelSim - Intel FPGA Edition (includes Starter Edition) Size: 176.2 MB MD5: 76FA490F4E4013F13C32E4401D9C170A	↓
ModelSim - Intel FPGA Edition (includes Starter Edition) Part 2 Size: 7.6 GB MD5: 3BDA2FF7B1630A372D072069D4B66B3E	↓
Questa - Intel FPGA Edition (includes Starter Edition) Size: 461.9 MB MD5: 21913DECC7E81CF15121D4704504253D	↓
Questa - Intel FPGA Edition (includes Starter Edition) Part 2 Size: 4.3 GB MD5: 3FC5E17D7ED03991BE6A5D1FCEF57AA0	↓

The screenshot shows the Intel website's page for downloading ModelSim*-Intel® FPGA Edition Software. The page has a blue header with the Intel logo and navigation links. Below the header, there's a section titled "ModelSim*-Intel® FPGA Edition Software" with links for Benefits, FAQ, and Support. A large blue button labeled "Download ModelSim*-Intel® FPGA edition software" is prominent. Below this, there's a description of the software and a comparison table between the full edition and the starter edition.

ModelSim*-Intel® FPGA Edition Software	ModelSim*-Intel® FPGA Starter Edition Software
<ul style="list-style-type: none">• Recommended for simulating all Intel® FPGA designs (Intel® Arria® FPGA, Intel® Cyclone® FPGA, and Intel® Stratix® FPGA designs, and Intel® MAX® CPLDs)• 33 percent faster simulation performance than ModelSim*-Intel® FPGA starter edition software• No line limitations	<ul style="list-style-type: none">• Support for simulating small Intel® FPGA designs• 10,000 executable line limitations• Free no license required• Mixed language support

Uso Máquina Virtual con QuestaSim

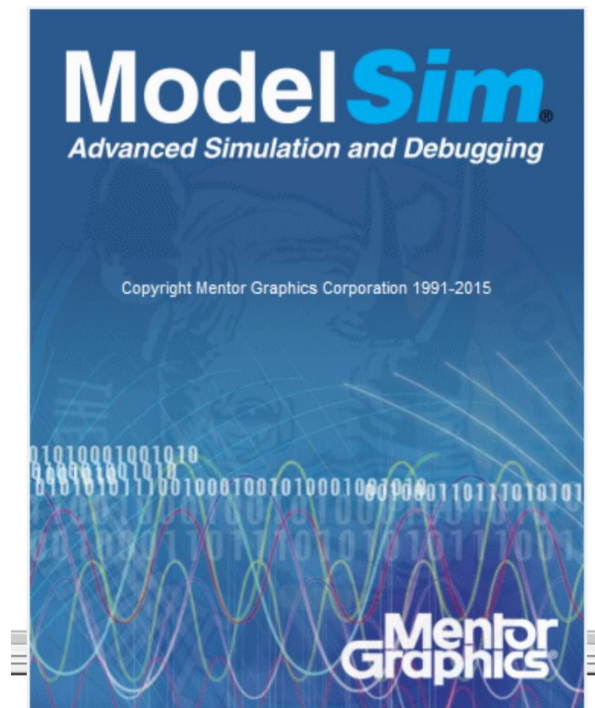
- Se provee un máquina virtual VMWare
 - Debes tener instalado VMWare Player
- La MV es un Ubuntu 18.04 con QuestaSim instalado
 - Usuario Arqo, passw: arq2021
 - Es necesario usar la VPN
 - Se lanza desde línea de comandos



Recordatorio uso ModelSim o QuestaSim

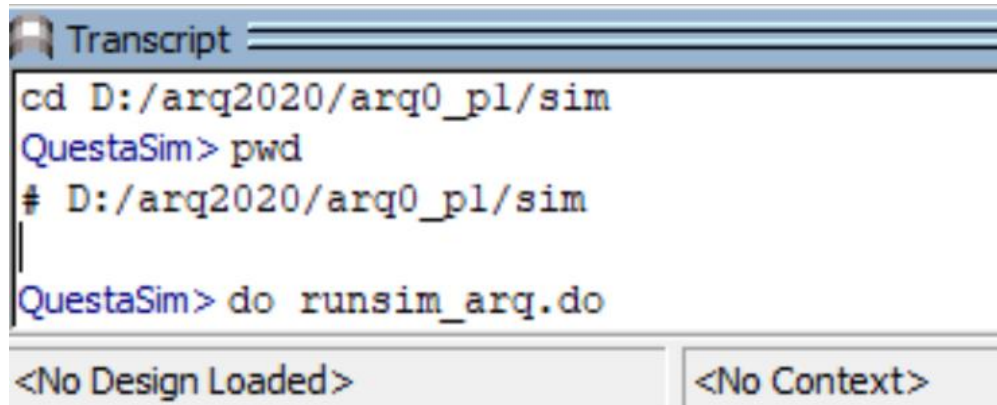
- El flujo de trabajo es igual en ambos (al menos lo que se utilizará en arquitectura de ordenadores)
- Se puede crear un proyecto, o usar script (recomendado en arqo)
- Para iniciar Model/Questa
 - ***vsim*** en Linux Shell

```
arqo@osboxes: ~  
File Edit View Search Terminal Help  
arqo@osboxes:~$ vsim  
Reading pref.tcl
```




Ejecutar script

- Desde la consola de modelSim/Questa sim
- Ir a la carpeta donde está el script (extensión .do)
 - cd ruta_a_arq2021/arq_ejerc_xx/sim (o usar file-> change directory ...)
 - Asegurarse estar donde queremos (pwd)
- Ejecutar el script (comando “do”)
 - **do** run_sim_arq.do



```
Transcript
cd D:/arq2020/arq0_pl/sim
QuestaSim> pwd
# D:/arq2020/arq0_pl/sim
QuestaSim> do runsim_arq.do
```

<No Design Loaded> <No Context>



```
Transcript
// 10 U.S.C. SECTION 1909.
# //
QuestaSim> pwd
# /home/arqo
pwd
# /home/arqo
QuestaSim> cd /home/arqo/arqo2021/arq0_pl/sim
QuestaSim> pwd
# /home/arqo/arqo2021/arq0_pl/sim
QuestaSim> do runsim_arq.do
```

<No Design Loaded> \$MODEL_TECH/../../vhdl_opt_lib

¿Que hace el script?

```
-----  
# Script ModelSim para la simulacion del procesador Arq2019-2020  
-----
```

```
# Crear library, borrando cualquier compilacion previa:
```

```
if [file exists work] {vdel -lib work -all }  
vlib work
```

```
# Compilar RTL:
```

```
vcom -work work -2002 -explicit -check_synthesis ../rtl/reg_bank.vhd  
vcom -work work -2002 -explicit -check_synthesis ../rtl/alu.vhd  
vcom -work work -2002 -explicit -check_synthesis ../rtl/alu_control.vhd  
vcom -work work -2002 -explicit -check_synthesis ../rtl/control_unit.vhd  
vcom -work work -2002 -explicit -check_synthesis ../rtl/processor.vhd
```

```
# Compilar testbench:
```

```
vcom -work work -2002 -explicit memory.vhd  
vcom -work work -2002 -explicit processor_tb.vhd
```

```
---
```

```
set StdArithNoWarnings 0
```

```
# Lanzar la simulacion, hast  
run -all  
-----
```

- Borra compilación Previa
- Crea biblioteca work
- Compila los fuentes del procesador (vcom)
- Compila los fuentes del simulación (vcom)
- Elabora el diseño (lanza simulación) usando processor_tb como top
 - -g para los generics del toplevel
- Abre las formas de onda del archivo wave_Arq.do
- Simula hasta el final (run -all)

¿Que hace el script?

```
-----  
# Script ModelSim para la simulacion del procesador Arq2019-2020  
-----
```

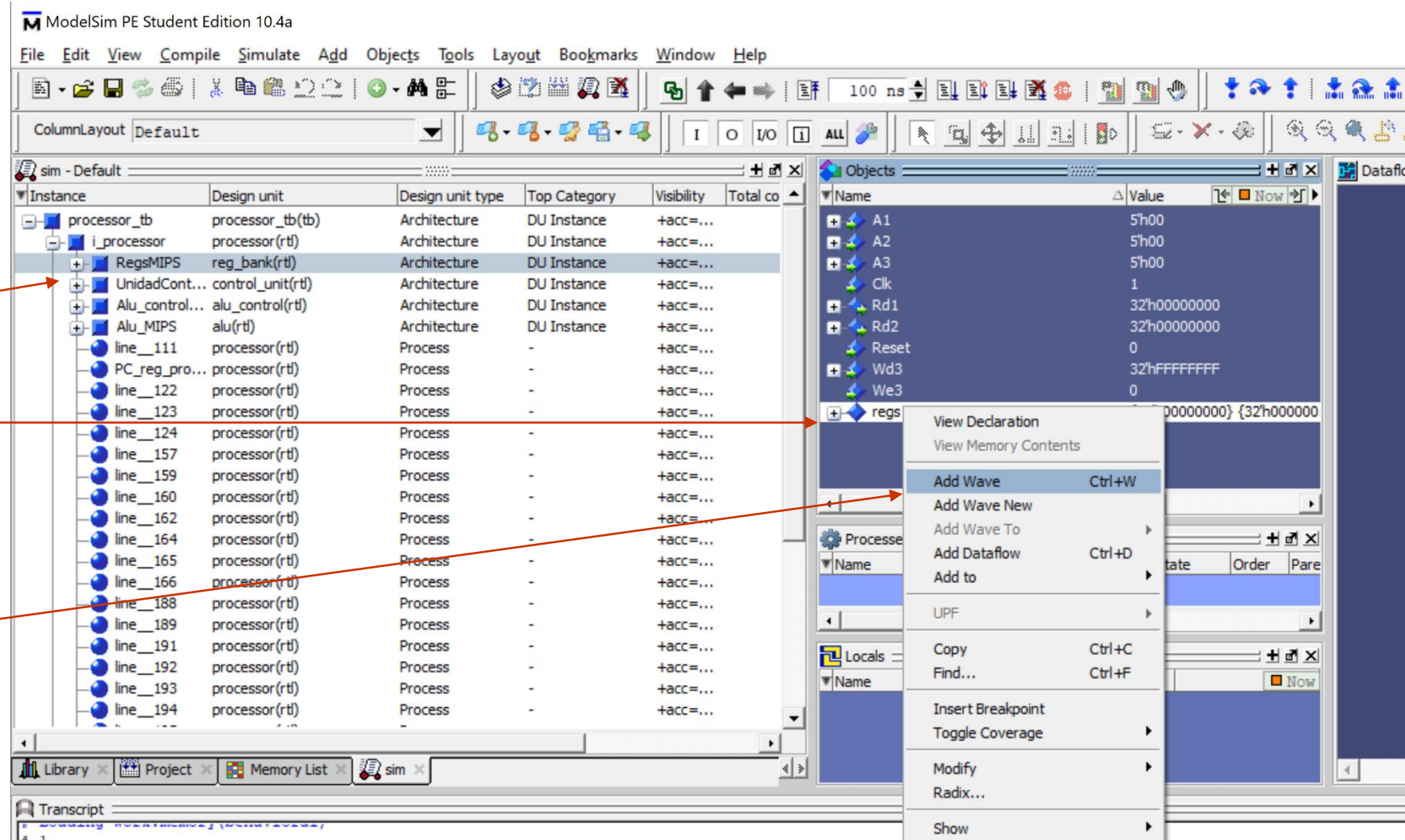
```
# Crear library, borrando cualquier compilacion previa:
```

- Borra compilación Previa
- Crea biblioteca work
- Compila los fuentes del procesador (vcom)
- Compila los fuentes de la simulación (vcom)
- Elabora el diseño (lanza simulación) usando processor_tb como top
 - -g para los generics del toplevel
- Abre las formas de onda del archivo wave_arq.do
- Simula hasta el final (run -all)

```
all }  
  
# Elaboracion:  
vsim -novopt -gINIT_FILENAME_INST="instrucciones"  
-gINIT_FILENAME_DATA="datos" -gN_CYCLES=100 processor_tb  
  
# Opcion para guardar todas las ondas:  
log -r /*  
  
# Mostrar las ondas:  
do wave_arq.do  
  
# Opcion del simulador para evitar warnings tipicos en tiempo 0 :  
set StdArithNoWarnings 1  
run 0 ns  
set StdArithNoWarnings 0  
  
# Lanzar la simulacion, hasta que pare sola:  
run -all  
-----
```

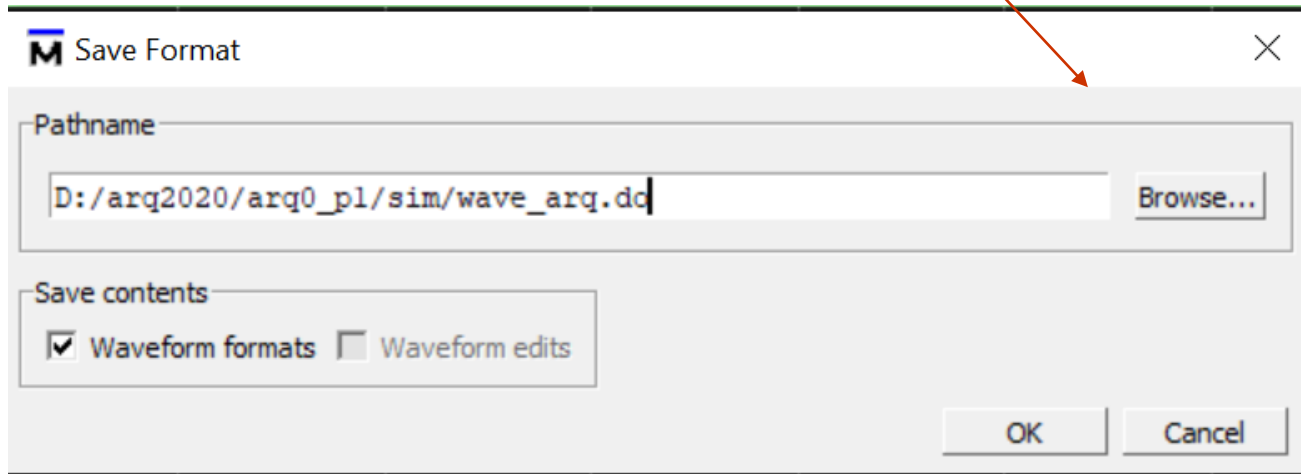
Visualizar más formas de ondas

- Seleccionar el elemento en la jerarquía de componentes, luego la(s) señales, botón derecho “add wave”



Configurar vista y guardar para futuras simulaciones

- Mover, cambiar base (radix), formato (color), agregar separadores, etc.
- Guardar la forma de ondas
 - File -> “save format...” (ctrl+s)



- La próxima vez que se simule, se visualizarán los cambios en el formato

