

Uso de la herramienta ISE mediante Makefile

Laboratorio de Electrónica Digital I
Departamento de Ingeniería Eléctrica y Electrónica
Facultad de Ingeniería
Universidad Nacional de Colombia
29 de agosto de 2017

1. En primera instancia, debe crear un directorio con el nombre del proyecto (GPU es un ejemplo)

```
$ mkdir GPU
```

2. Coloque todos los archivos de código Verilog en la carpeta creada

```
$ cp *.v GPU
```

 (comando para copiar todos los archivos .v a la ubicación GPU)

3. Coloque el archivo Makefile (Ver siguiente página) en la carpeta del proyecto (GPU). Recuerde que debe especificar el archivo.v de mayor jerarquía dentro de este archivo Makefile:

```
DESIGN= main
```

4. Cree el archivo.ucf en la carpeta del proyecto. Este archivo debe tener el mismo nombre del proyecto. En este caso GPU.ucf
5. Por favor instalar ICARUS VERILOG y GTKWAVE.

```
$ sudo apt-get update
```

```
$ sudo apt-get install gtkwave
```

```
$ sudo apt-get install iverilog
```

6. Modifique sus archivos de código con el editor de texto que prefiera, por ejemplo:

```
$ gedit vga_controller.v
```

7. Tenga en cuenta los siguientes comandos de uso general del Makefile:

- **make clean:** Eliminar todos los archivos temporales.
- **make cleanall:** Elimina todos los archivos generados después de ejecutar make all.
- **make all:** Realiza la síntesis y crea el archivo .bit.
- **make remake:** Realiza clean&all
- **make config:** Programar la FPGA con el archivo .bit generado después de ejecutar make all

8. En el Makefile, donde dice DEVICE, debe cerciorarse de cual dispositivo tiene (revisar en el archivo rm de cada nexys)

- XC3S500E-FG320-4 (nexys2)
- XC6SLX16-3CSG324 (nexys3)
- XC7A100T-3CSG324 (nexys4)
- XC7A100T-3CSG324 (nexys4 DDR)

Lo siguiente deben copiarlo en un editor de texto plano, y guardarlo con el nombre Makefile.

#-----ARCHIVO Makefile-----

DESIGN = counter#Aquí va el nombre del proyecto

PINS = \$(DESIGN).ucf

DEVICE = xc3s500e-fg320-4 #Esto depende de cada tarjeta

BOARD = Nexys2 #Nombre de la tarjeta

BGFLAGS = -g TdoPin:PULLNONE -g DonePin:PULLUP \ -g CRC:enable -g StartUpClk:CCLK

XILINXCADROOT = /opt/Xilinx/14.4/ISE_DS/ISE/

SIM_COMP_SCRIPT = simulation/\$(DESIGN)_TB.do

SIMGEN_OPTIONS = -p \$(FPGA_ARCH) -lang \$(LANGUAGE)

IVERILOG = iverilog

SRC = \$(DESIGN).v

SIM_SRC = \$(DESIGN)_TB.v \ #Archivo testbench, para simulación verilog

all: bits

remake: clean-build all

clean:

rm -f *~ */*~ a.out *.log *.key *.edf *.ps trace.dat

rm -f *.bit

cleanall: clean

rm -rf build simulation/work simulation/transcript simulation/vsim.wlf simulation/\$(DESIGN)_TB.vvp
simulation/\$(DESIGN)_TB.vcd

bits: \$(DESIGN).bit

Synthesis

build/project.src:

@[-d build] || mkdir build

@[-d simulation] || mkdir simulation

@rm -f \$@

for i in \$(SRC); do echo verilog work ../\$i >> \$@; done

```
for i in $(SRC_HDL); do echo VHDL work ../$i >> $@; done

build/project.xst: build/project.src

    echo "run" > $@

    echo "-top $(DESIGN) " >> $@

    echo "-p $(DEVICE)" >> $@

    echo "-opt_mode Area" >> $@

    echo "-opt_level 1" >> $@

    echo "-ifn project.src" >> $@

    echo "-ifmt mixed" >> $@

    echo "-ofn project.ngc" >> $@

    echo "-ofmt NGC" >> $@

    echo "-rtlview yes" >> $@

build/project.ngc: build/project.xst $(SRC)

    cd build && xst -ifn project.xst -ofn project.log

build/project.ngd: build/project.ngc $(PINS)

    cd build && ngdbuild -p $(DEVICE) project.ngc -uc ../$(PINS)

build/project.ncd: build/project.ngd

    cd build && map -pr b -p $(DEVICE) project

build/project_r.ncd: build/project.ncd

    cd build && par -w project project_r.ncd

build/project_r.twr: build/project_r.ncd

    cd build && trce -v 25 project_r.ncd project.pcf

$(DESIGN).bit: build/project_r.ncd build/project_r.twr

    cd build && bitgen project_r.ncd -l -w $(BGFLAGS)

    @mv -f build/project_r.bit $@

build/project_r.v: build/project_r.ncd

    cd build && ngd2ver project.ngd -w project.v
```

postsim: build/project.ngc

```
cd build && netgen -sim -ofmt verilog -w project.ngc
```

```
cd build && iverilog -Wall \
```

```
-y $(XILINXCADROOT)/verilog/src/unisims \
```

```
-y $(XILINXCADROOT)/verilog/src/XilinxCoreLib \
```

```
project.v ../$(DESIGN)_TB.v -o $(DESIGN).bin
```

```
cd build && vvp $(DESIGN).bin
```

```
gtkwave build/$(DESIGN)_TB.vcd&
```

iversim: build/project.src

```
$(IVERILOG) -o simulation/$(DESIGN)_TB.vvp $(VINCDIR) $(SRC) $(SIM_SRC) -s $(DESIGN)_TB
```

```
vvp simulation/$(DESIGN)_TB.vvp; mv $(DESIGN)_TB.vcd simulation/
```

```
gtkwave simulation/$(DESIGN)_TB.vcd&
```

simcontrol:

```
$(IVERILOG) -o simulation/control_TB.vvp control.v control_TB.v -s control_TB
```

```
vvp simulation/control_TB.vvp; mv control_TB.vcd simulation/
```

```
gtkwave simulation/control_TB.vcd&
```

config: \$(DESIGN).bit

```
djtgcfg init -d $(BOARD)
```

```
djtgcfg prog -d $(BOARD) --index 0 --file $(DESIGN).bit
```