«cocotb oppskrift» 30.01.2023

1: installere WSL eller annet linuxsystem https://techcommunity.microsoft.com/t5/windows-11/how-to-install-the-linux-windows-subsystem-in-windows-11/m-p/2701207

- cmd

wsl --install

- I utgangspunktet skal python være installert med WSL, men hvis ikke må det gjøres
  - `Sudo pip install pytest`
- `pip3 install numpy`
  - ***Numpy trengs for å bruke uint8 int8***

2: I linux: installere ada og Gnats (se evt https://sites.radford.edu/~nokie/classes/320/compileInstall.html)

apt-get install gnat – trengs for å kompilere GHDL.

- Bruk sudo om man får permission error: `sudo apt-get install …`

3: laste ned og pakke ut GHDL (https://github.com/ghdl/ghdl)

- Obs: **Følg guiden**, ikke installer med apt. Versjonen som ligger i ubuntu repository er v1.0.0. V2/v3 må bygges.
- https://github.com/ghdl/ghdl/tree/5726f0eccf874b872ce0729aab42f587d915a3f5
  - *Github lokasjon*
- https://github.com/ghdl/ghdl/archive/5726f0eccf874b872ce0729aab42f587d915a3f5.zip
  - Zip fil med versjon som kan bygges

4: Bygge GHDL (https://ghdl.github.io/ghdl/development/building/index.html)

- Last ned pakken til et sted i WSL filsystemet dersom det er i bruk.
- Pakke ut den nedlastede pakken og navigere inn i den i wsl cmd.
  - (når man lastet ned .tar.gz fil):
    - `tar xf #pakkenavn#`
  - (npr man lastet ned .zip fil):
    - `unzip #pakkenavn#`
- `./configure --prefix=/usr/local`
- `make`
- `sudo make install`
- Bygge GHDL fra patchet versjon (fungerer selv om en annen versjon var lastet ned fra før):
  - Laste ned patched versjon fra github (code → download ZIP)
  - Unzip filen og naviger til den iwsl cmd (resten er lik som før)
  - `./configure --prefix=/usr/local`
  - `make`
  - `sudo make install`
  - `Nå skal ghdl være installert. Kjør følgende kommando for å verifisere installasjon og riktig versjon: GHDL 3.0.0`
  - `ghdl version`

5: installere cocotb

- `pip install cocotb`
  - evt `pip install cocotb[bus]`
    - Feilmelding om gammel pip versjon kan forekomme. 22.2.2 kan virke, men den ønsker 22.3.1
    - Evt. Oppgrader pip først
      - `python -m pip install --upgrade pip`
- cocotb-config -v
  - for å verifisere installasjon og versjon (1.7.2)
-

6: Fixe PATH (om ikke det er gjort fra før) https://opensource.com/article/17/6/set-path-linux:

- Ikke nødvendig på Ubuntu 22
- sjekk med `echo $PATH`
  - er ikke sikker på om alle disse trengs, men de står evt listet underveis og cocotb virker ikke uten.
    - `export PATH=$PATH:/usr/local/bin`
    - `export PATH=$PATH:/usr/bin`
-

7. Installere gtkwave:

- `sudo apt install gtkwave`

For å kjøre Cocotb skriv
`make`
i folderen med makefile

makefila sørger for kompilering og kjøring av cocotb
tb_<modulnavn>.py fila inneholder testene i python

For å starte windows explorer I folder til WSL kjør
explorer.exe .
(ikke glem punktumet..)

```
# Makefile

# defaults
SIM ?= ghdl
TOPLEVEL_LANG ?= vhdl

# VHDL 2008
EXTRA_ARGS +=--std=08

# TOPLEVEL is the name of the toplevel module in your VHDL file
TOPLEVEL = first

VHDL_SOURCES += $(PWD)/../lab1/$(TOPLEVEL).vhd
#VHDL_SOURCES += $(PWD)/../../hdl/*.vhdl

SIM_ARGS +=--wave=$(TOPLEVEL).ghw

# MODULE is the basename of the Python test file
MODULE = tb_$(TOPLEVEL)

# include cocotb's make rules to take care of the simulator setup
include $(shell cocotb-config --makefiles)/Makefile.sim

# removing generated binary of top entity and .o-file on make clean
clean::
      -@rm -f $(TOPLEVEL)
      -@rm -f e~$(TOPLEVEL).o
```

I makefila er det beskrevet at SIM_ARHS +=--wave=$(TOPLEVEL).ghw
Dette sørger for at GTKWave lager en ghw fil som kan åpnes med gtkwave:

- gtkwave <filnavn>

I vinduet til gtkwave: trykk

- høyreklikk på topnivå (eller de signalene du ønsker å ta inn)
    - velg recursive insert, append eller replace (alle valgene vil fungere første gang)
- Zoom ut med knapp eller Time\Zoom\Zoom Best Fit (Shift Alt F)

Only the active "dut" will provide meaningful results to be checked.

- Each "dut" will only report simulation results during the cocotb.test() it is started in.
- The handle to an old dut may live forever to be while newer @coctb.test() methods are run.
    - Holding on to old handles will most likely cause tests to seem
- Simulation wide tests, must be re-instantiated for every cocotb.test()
    - Use completely disjunct tests
    - No test should provide data for the next or previous test