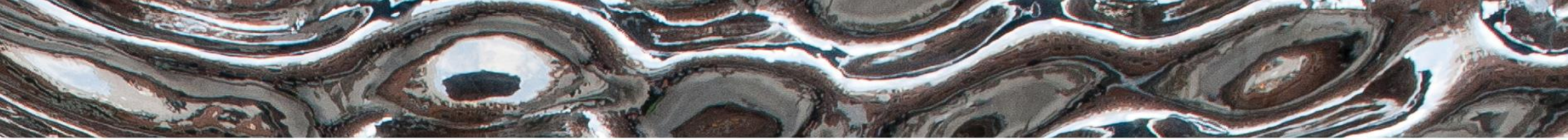


UiO : Institutt for informatikk
Det matematisk-naturvitenskapelige fakultet

IN3160, IN4160

Infrastructure and tool introduction

Yngve Hafting 2022

Overview

- Lab supervision schedule
- Remote access solution
  - login.ifi.uio.no
    - Ssh -Y
    - X-win
  - Vmware / VDI
    - Ifi Digital Electronics
  - Vmware + ssh –Y
    - For accessing lab setups (for programming
- Software tool intro.
  - Questa
  - Vivado
- How to get started…

- Assignments and suggested reading for this week

Lab supervision starting 31.1 ...

- Monday 11-13 : Mojtaba
- Tuesday 11-13 : Georg
- Wednesday 13-15: Jørgen
- Friday 12-14: Sander

Linux setup for general access

- for ssh sessions, *such as* **X-win**
  https://www.mn.uio.no/ifi/tjenester/it/hjelp/it-vakten/laptophjelp/laptophjelp-guide/tilgang-til-uio-hjemmeomrade-og-ifi-linux-terminal/
  - (VDI and Lisp *may* not require this but you do need ssh for programming)
- Edit .bashrc (in your root folder)
  - `>>gedit .bashrc &`
  - Save and exit
  - Start a new session
    - Either close the old and log in again or
    - `>>xterm &`
  - `>> vsim &`
  - Login.ifi.uio.no or VDI IFI-digital electronics
    - Can be used for questa
    - Don't run vivado on login.ifi.uio.no
      - Its hugs too much resources

```
# Vitis Unified 2020.2 64-bit version
if ! [ -x "$(command -v vivado)" ]; then
  source /projects/robin/programs/Vivado/2020.2/settings64.sh
  export PATH=$PATH:/projects/robin/programs/Vivado/2020.2/bin
fi

# License file
export LM_LICENSE_FILE=5370@lisens.ifi.uio.no

# Modelsim library
export MODELSIM=/projects/robin/CADlib/modelsimCADLIB.ini

# Remove duplicants in the PATH variable
PATH=$(printf "%s" "$PATH" | awk -v RS=':' '!a[$1]++ { if (NR > 1)
printf RS; printf $1 }')
```

  - `>> echo $PATH$`
  - `>> echo $MODELSIM$`
  - `>> echo $LM_LICENSE_FILE$`
  - Will give a clue to your succes / lack of

SSH to lab machine for programming FPGA

To be used when you are not able to reach OJD

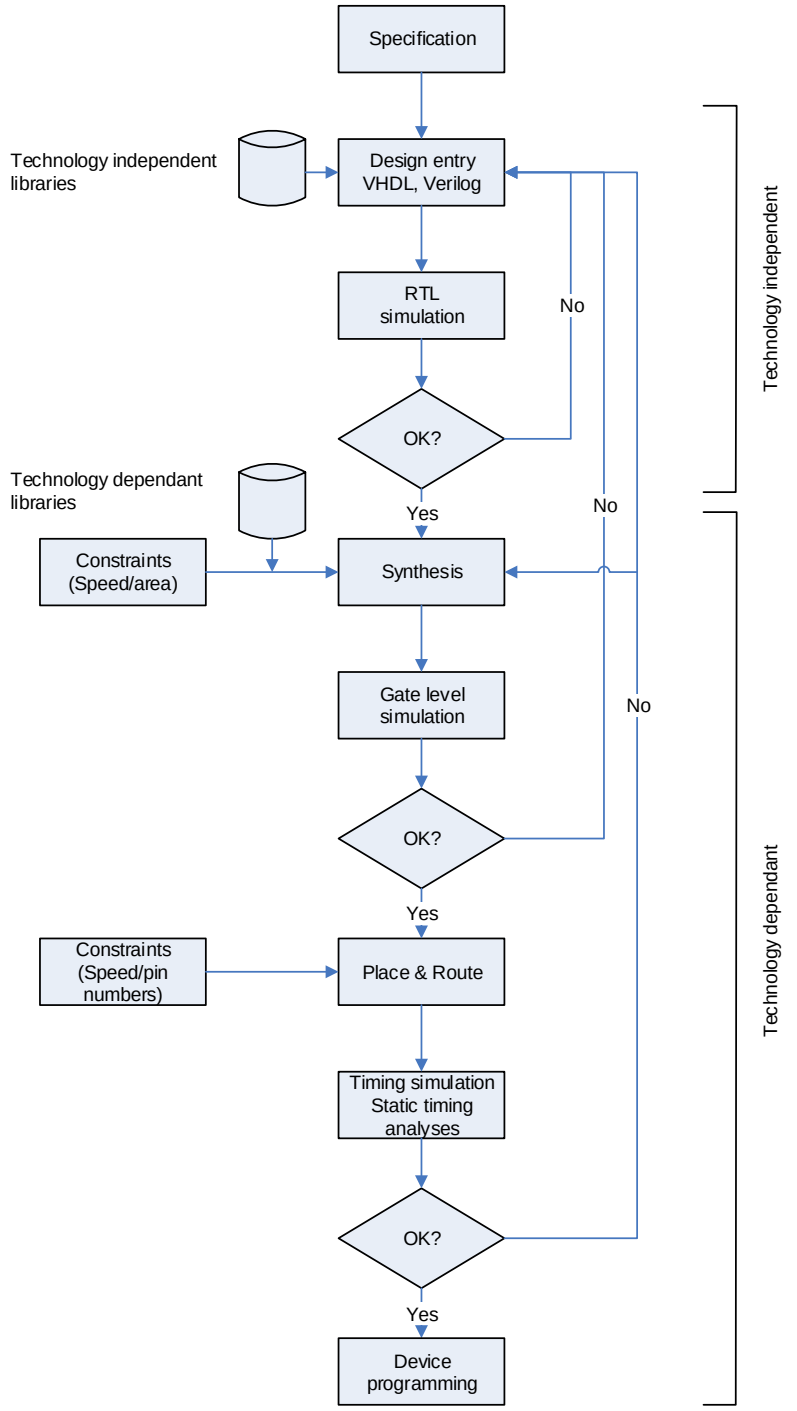
- To be done when already logged onto *login.ifi.uio.no* or *ifi-Digital-Electronics*
  - (no direct access exists)
  - `>> Ssh –Y <computer name>`
  - **Which computers** you are allowed to use **will be posted later**.
- https://robin.wiki.ifi.uio.no/
  - -> Software\FPGA Design\Remote access
    - The procedure for «special setup on LISP» can be used with the machines we allow (TBD name)
    - >> zedboard-webcam
    - >> zedboard-test
    - >> zedboard-wipe
  - Program using vivado or script.
  - Remember to **LOG OUT** properly when done.
    - `>> exit` until logget out from computer
    - Logging out from VDI solution when connected to <computer name > = disconnect
    - Disconnect = denying other students access...

Access through VDI

- https://www.mn.uio.no/ifi/tjenester/it/hjelp/
  - => Linux->Virtuell arbeidsstasjon (VDI)
    - If you don't have the Vmware client already, check
      - https://www.uio.no/tjenester/it/maskin/vdi/hjelp/vdi-installer-og-bruk.html
  - Start Vmware horizon klienten
    - Gå inn på view.uio.no
    - Bruk ifi-Digital-Electronics
      - *Kjør vivado eller vsim fra kommando-linje*
  - Ifi digital electronics kan brukes til alt... *frem til programmering*
    - Vivado, questa, editors..

Digital Design tools…

- Design entry:
  - Use your favourite HDL text editor (Notepad++, Emacs, Vivado or Questa).
- Simulation (RTL, Gate Level, Timing)
  - Here: Typically using Questa (=Modelsim)
- Synthesis, Implementation, Programming
  - Vendor specific tools…
    - Here: Vivado by Xilinx



Simulation and test benches

- Simulation can be run using three different approaches:
  1. Manually setting inputs and specifying time intervals in the GUI or console
     - This way is tedious if much testing is to be done. Normally this is only done initially.
  2. To make scripts (tcl for Questa) in a separate (.do) file.
     - The *script commands will be added to the console during manual use, and can be copied as text into a .do file.*
     - setting up the simulation windows can be done reusing script commands.
  3. Create a test bench in VHDL (possible in combination with running scripts)
     - *This is the preferred method*
     - VHDL can be used to generate code for applying test vectors sequentially to the inputs of an entity for simulating.
     - Test bench code is not synthesizable
     - easy to read and use test data for each particular design,
     - Can be used both prior and post synthesis or implementation