1. Tips to use when starting a project

Here are a few pointers on how to start a project from scratch. I haven't figured out a way to take an existing project and transform it to match this strategy. So, before a better solution is found, it is very important to follow these tips from the beginning of the project.

- When starting a project, create the directories first (IP_repo, src with every sub_directory and the work_dir)
- Deselect the "Create project subdirectory" when creating a new project
- When adding sources to the project, do not add them to "Local to project". Specify the location of your files to be within your hdl directory under the src directory
- Similarly, when creating bd designs, make sure to put their location within the IP_rpeo directory
- For synthesis and implementation runs, they can be run locally as they do not have to be source controlled anyways

2. File Structure

Here is what the file structure should look like if using this strategy. Note that it is really important to maintain the exact name of the folders as their names are part of the code. You could obviously change these names and change the code. You should commit everything in itallic and every files provided (I wrote) are in bold.

- <Your project>
  - work dir
    - project.xpr
    - .cache
    - .sim
    - .runs
    - .hw
    - ip_user_files
  - *src*
    - bd
      - <contains all of your bd_design.tcl files>
      - **block_design_tcl.tcl**
      - **bd_script_changer.tcl**
    - hdl
      - <all of your hdl files
    - sdk
      - <all of your sdk files>
    - sim
      - <all of your simulation sources>
    - constrs
      - <all of your constraints files>
  - IP_repo
    - <all of your bd directories
  - ***build.tcl***
  - ***to_git.tcl***
  - ***to_git.bat***
  - ***from_git.tcl***
  - ***from_git.bat***
  - ***script_changer.py***

3. File Functions

Here is a quick list of every file I wrote and the different functions they have.

Function	Description
to_git.bat	Double click on it and it will simply call the to_git.tcl file
to_git.tcl	Generates a tcl script for every block design in your project and for your project itself
from_git.bat	Deletes the current IP_repo, work_dir and .Xil directories to avoid errors and then calls from_git.tcl
from_git.tcl	Changes the absolute path in every bd_design.tcl to match the current working directory and sources the build.tcl script
script_changer.tcl	Applies the necessary corrections to the automatically generated build.tcl script
bd_script_changer.tcl	Changes the absolute path of every bd_design.tcl created so it matches the current working directory
block_design_tcl.tcl	Generates a tcl script for every block design in your IP_repo directory
project_cleaner.tcl	Eliminates a few useless files generated automatically by Vivado at the end of the process

TABLE 3.1 – Description of the functions created