Microcontroller in eSim - Documentation

1. Overview –

Currently, ATTINY 85 has been targeted to be implemented in eSim. The component outline including in / out ports (a completed skeleton of the microcontroller) is developed in a main VHDL file – "attiny85_nghdl.vhdl" and the microcontroller is logically simulated by a C file – "tiny85_c.c".
Please note that this isn't a cycle accurate simulation but a logically accurate one.

2. Process –

In order to simulate a microcontroller, we need to simulate a C file and a VHDL file together during runtime. And to facilitate communication of variables and functions between the two files, we first need to make object files of them individually and link them. This step is done before simulating the project. The following steps need to be followed before simulation.

Please note that these steps have to be performed only once before simulating a project.

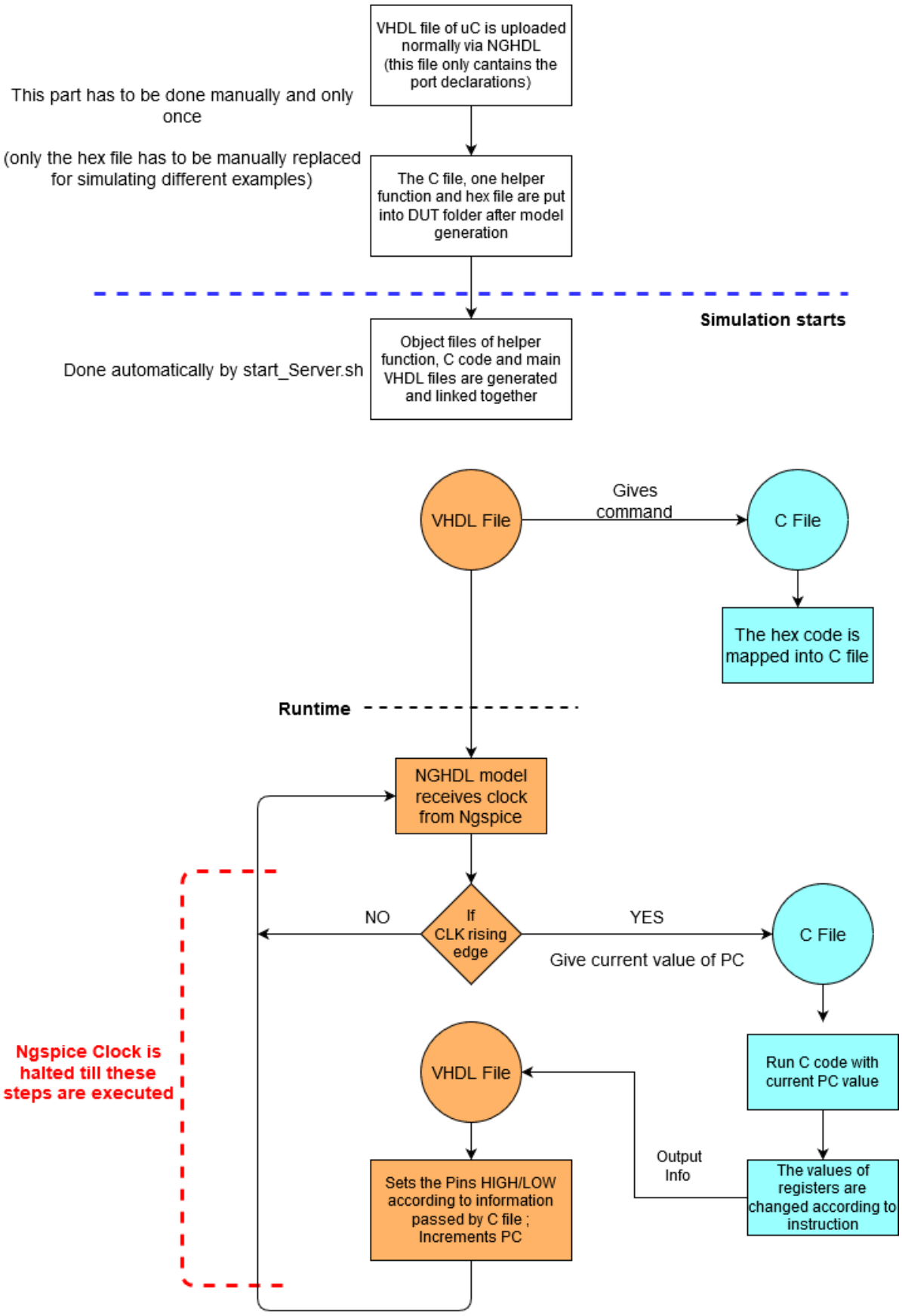
2.1 Before simulation –

i) The microcontroller model needs to be created in NGHDL via the VHDL file for microcontroller ("attiny85_nghdl.vhdl" here).
ii) Then the helper ghdl function, modified start_server.sh file, C file for microcontroller and hex file for the example have to be moved into the DUTghdl folder of the microcontroller.

2.2 When simulation starts –

i) The object file for microcontroller C file is created.
ii) The object files for helper function and main VHDL file are created.
iii) The object files of C file and main VHDL file are linked.
iv) The hex file is mapped into a structure in C file.
(Steps 1-4 happen only once after simulation starts)

v) Following steps are repeated till simulation ends –
vi) If CLK is rising edge, C file is given the current value of PC.
vii) C file runs once, setting required I/O pins HIGH or LOW depending on the instruction and changing the PC value as per the current instruction.


VHDL file of uC is uploaded normally via NGHDL (this file only cantains the port declarations)
This part has to be done manually and only once
(only the hex file has to be manually replaced for simulating different examples)
The C file, one helper function and hex file are put into DUT folder after model generation
Simulation starts
Done automatically by start_Server.sh
Object files of helper function, C code and main VHDL files are generated and linked together
VHDL File
Gives command
C File
The hex code is mapped into C file
Runtime
NGHDL model receives clock from Ngspice
NO
If CLK rising edge
YES
Give current value of PC
C File
Ngspice Clock is halted till these steps are executed
VHDL File
Run C code with current PC value
Output Info
Sets the Pins HIGH/LOW according to information passed by C file ; Increments PC
The values of registers are changed according to instruction