EC5612 VLSI LABORATORY

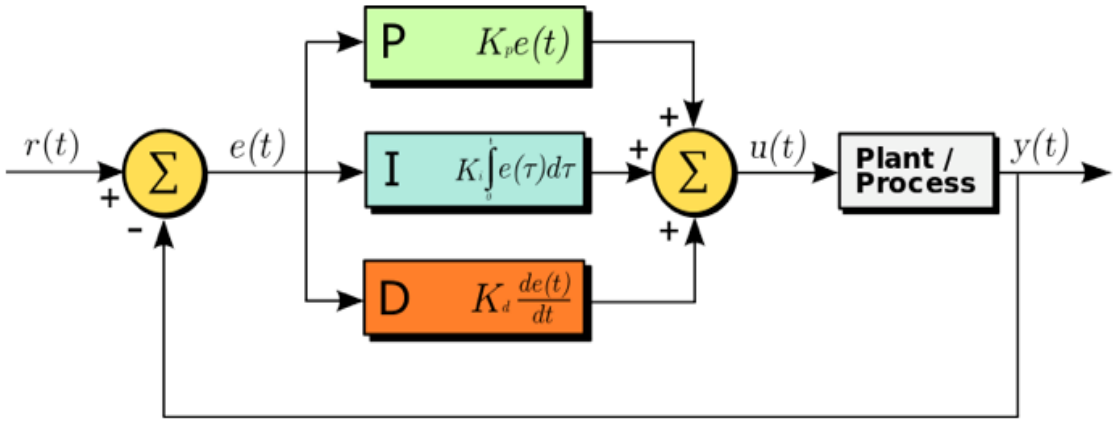
DESIGN AND SIMULATION OF DIGITAL PID CONTROLLER USING VERILOG

Done by,
1) SIVA CHOKKALINGAM S (2019504588)
2) THANISH MALAI P (2019504598)

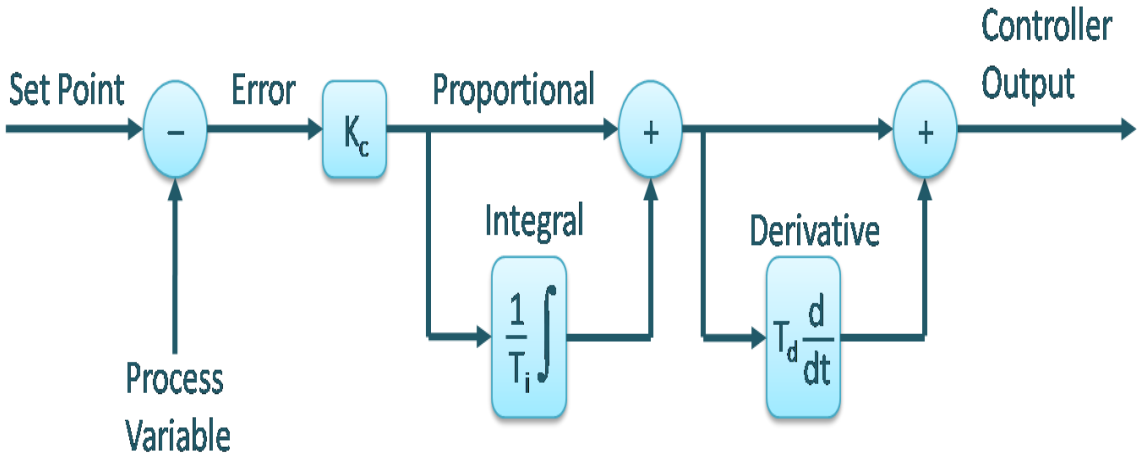
INTRODUCTION:

Proportional Integral Derivative (PID) Controller is one of the feedback controller system that produce desired output. PID controller play an important role in industry because of its excellent properties like simplicity, robustness, good noise tolerance and maintenance requirement. They are not susceptible to environmental noise and very easy to reconfigure.

PID controller gets feedback from the output and adjusts the parameter such that the stabilized output is achieved. By reducing the time delay for arithmetic operations (addition and multiplication), the PID can achieve efficiency. Providing a pipelined architecture helps in reducing the time delay.



PID ALGORITHM:



The algorithm first calculates the error between a measured value (PV) and its ideal value (SP), then use the error as an argument to calculate the manipulated value (MV). The MV will adjust the process to minimize the error.

The Kc, Kp and Kd are initially set and the output is obtained. The difference between the desired output and obtained output is fed to the system and the variables are changed such that error becomes zero.

$$u(t) = K_p e(t) + K_i \int_0^t e(t)dt + K_d \frac{de(t)}{dt}$$

Where, e(t) is error signal, u(t) is output of controller, K is gain or proportion gain, Ti is Integration time or rise time, Td is Derivative time.

DISCRETE-TIME PID ALGORITHM

For digital implementation, we are more interested in a Z-transform of the above equation

$$U(z) = \left[K_p + \frac{K_i}{1 - z^{-1}} + K_d (1 - z^{-1}) \right] E(z)$$

Rearranging gives

$$U(z) = \left[\frac{(K_p + K_i + K_d) + (-K_p - 2K_d) z^{-1} + K_d z^{-2}}{1 - z^{-1}} \right] E(z)$$

Define

$$K_1 = K_p + K_i + K_d$$
$$K_2 = -K_p - 2K_d$$
$$K_3 = K_d$$

(7) can then be rewritten as

$$U(z) - z^{-1} U(z) = \left[K_1 + K_2 z^{-1} + K_3 z^{-2} \right] E(z)$$

which then converted back to difference equation as

$$u[k] = u[k-1] + K_1 e[k] + K_2 e[k-1] + K_3 e[k-2]$$

SOFTWARE USED:

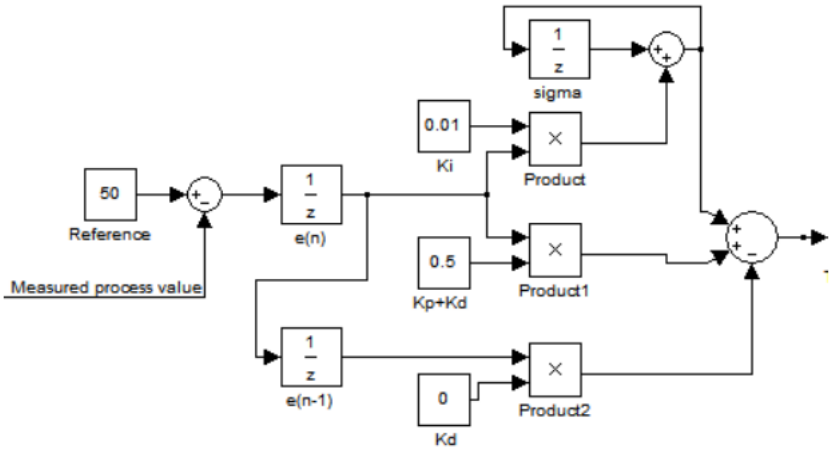
HARDWARE DESCRIPTIVE LANGUAGE-VERILOG

Verilog HDL is a hardware description language that can be used to model a digital system at many levels of abstraction ranging from the algorithmic-level to the gate-level to the switch-level The complexity of the digital system being modelled could vary from that of a simple gate to a complete electronic digital system, or anything in between.

The digital system can be described hierarchically and timing can be explicitly modelled within the same description. The Verilog HDL language has capabilities to describe the behavioral nature of a design, the dataflow nature of a design, a design's structural composition, delays and a waveform generation mechanism including aspects of response monitoring and verification, all modelled using one single language. In addition, the language provides a programming language interface through which the internals of a design can be accessed during simulation including the control of a simulation run.

Capabilities of Verilog HDL Listed below are the major capabilities of the Verilog hardware description: Primitive logic gates, such as and, or and nand, are built-in into the language. Switch-level modelling primitive gates, such as pmos and nmos, are also built- in into the language. High-level programming language constructs such as conditionals, case statements, and loops are available in the language. Powerful file read and write capabilities are provided.

CIRCUIT DIAGRAM:

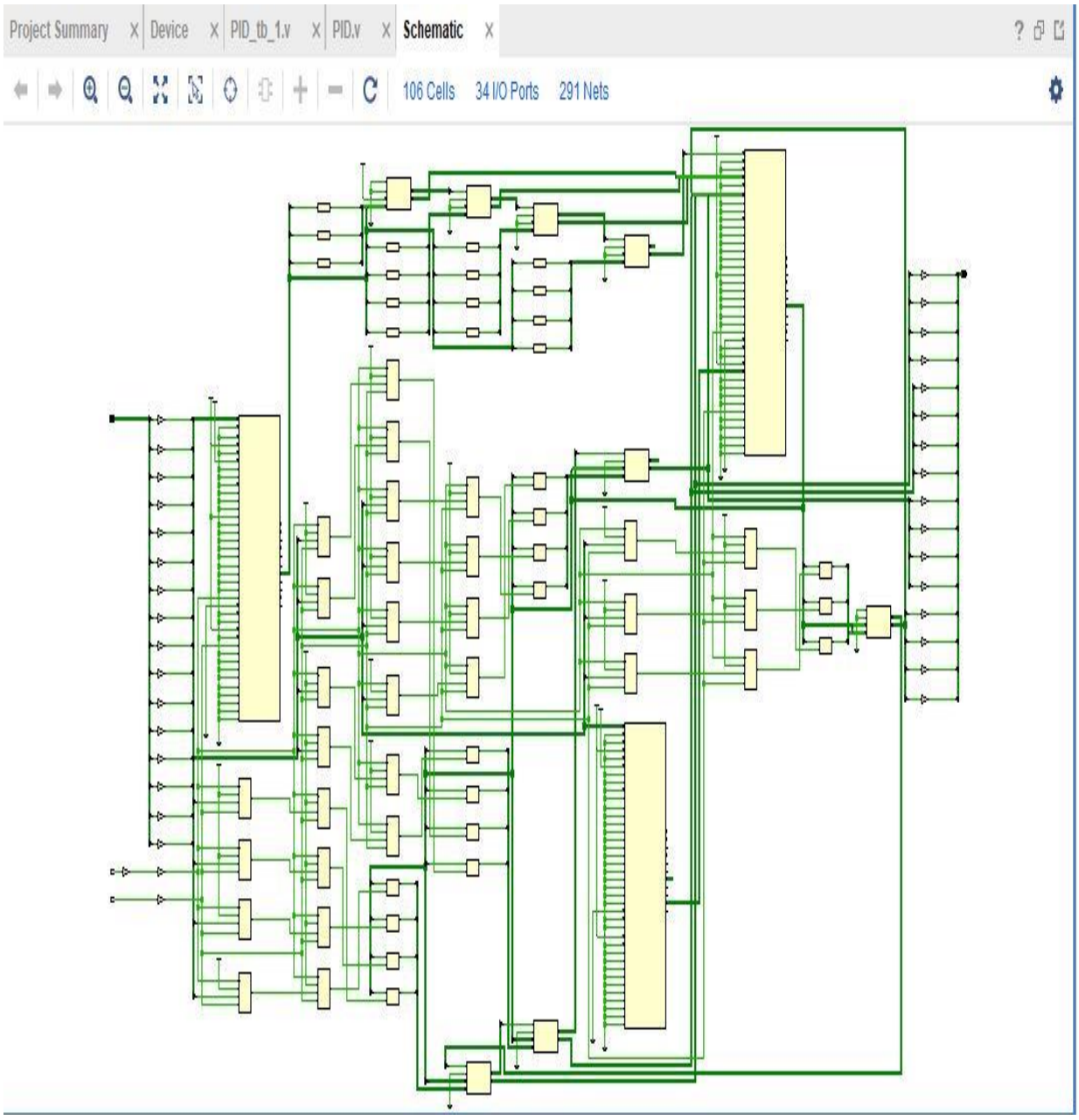


FUNCTION OF CIRCUIT DIAGRAM:

The digital PID controller is the feedback controller. It is divided into two parts main part PID controller and closed loop system. A system in which the control action is somehow depending on the output is called closed loop system. Closed loop system consist of reference input r(t), Feedback signal y(t), Error signal e(t), Control Output u(t). The PID algorithm consists of three basic modes, Proportional, Integral and Derivative modes. While utilizing this algorithm it is necessary to decide which modes are to be used (P, I or D) and then specify the parameters (or settings) for each mode used. Generally, three basic algorithms are used P, PI or PID.

The PID controller minimizes the error by changing the controller output, which is given as an input to the system. Proportional is related with present error, Integral on the past one and Derivative means prediction about the future errors. Sum of the each term, multiplying with respective gain gives the output of PID controller.

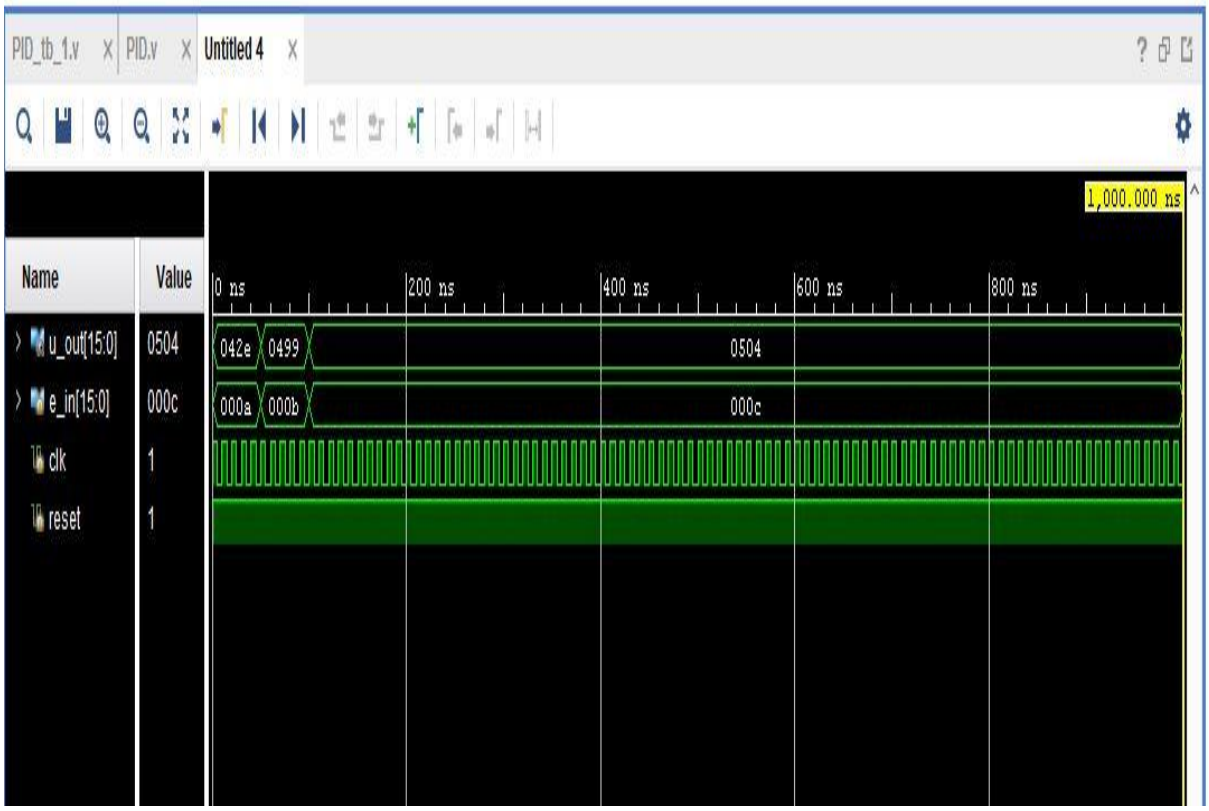
RTL SCHEMATIC



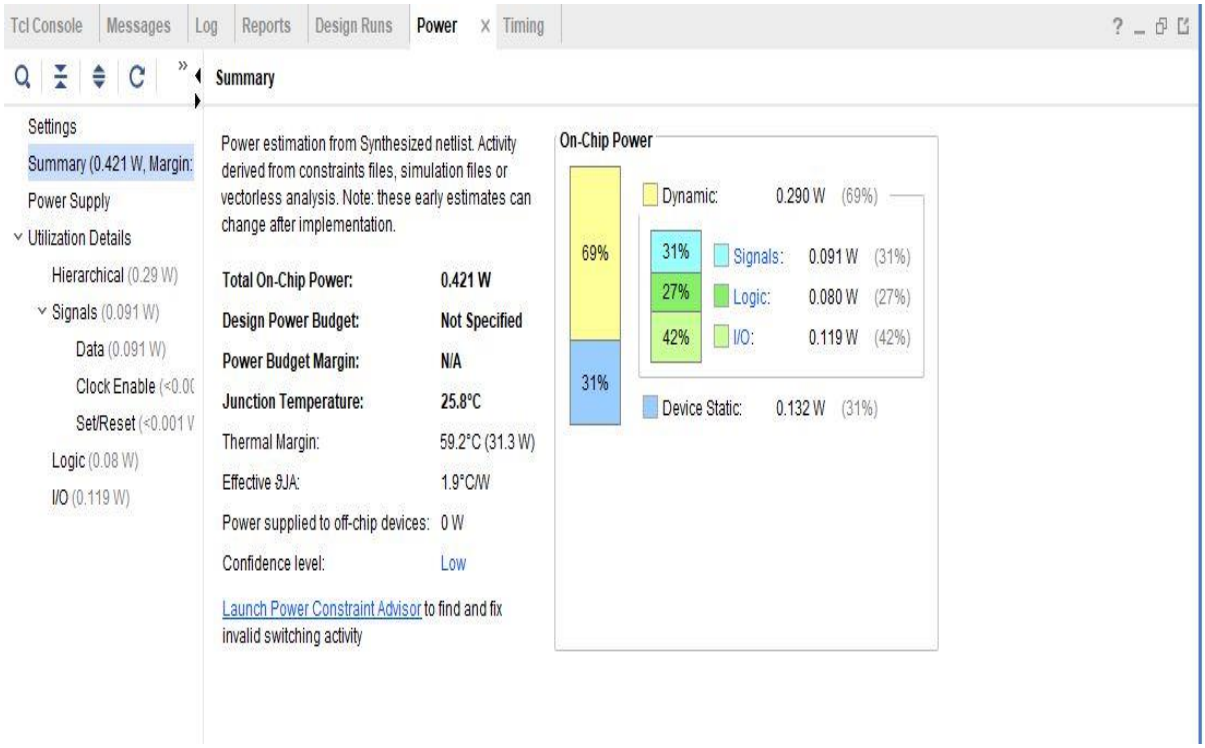
RESULT

We have given the simulated result of digital PID controller obtained from Verilog code. In this simulation, test bench is observing in form of waveforms according to which output waveform is generated. The following figures show RTL view and simulated result of digital PID controller.

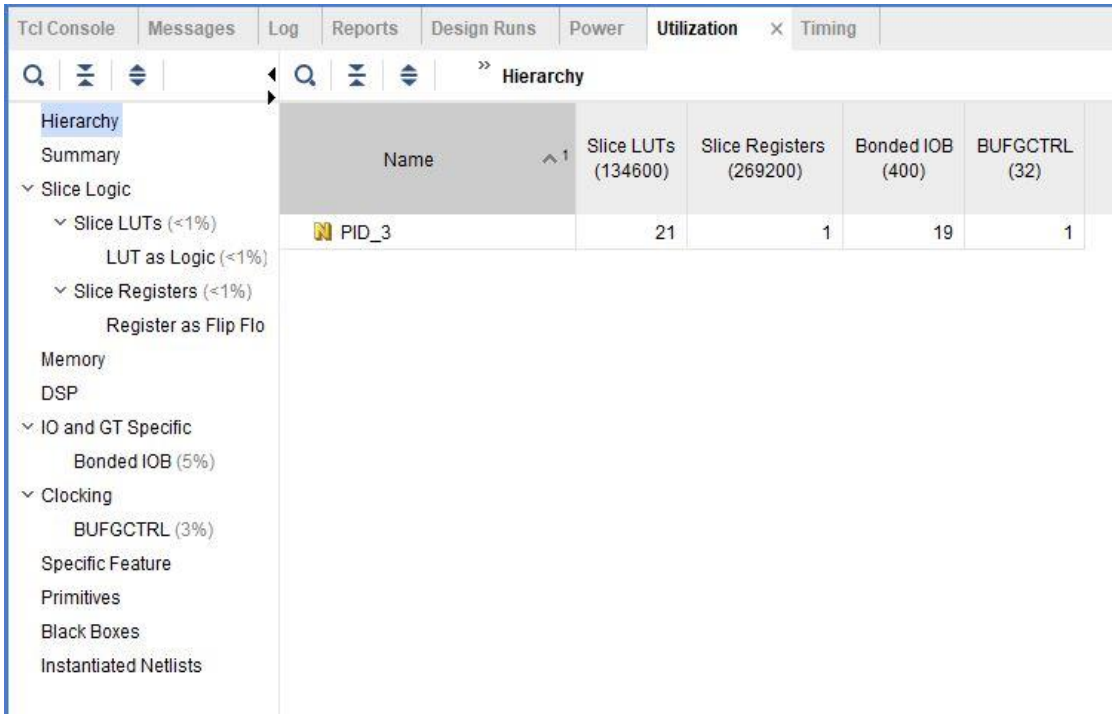
SIMULATION WAVEFORM:



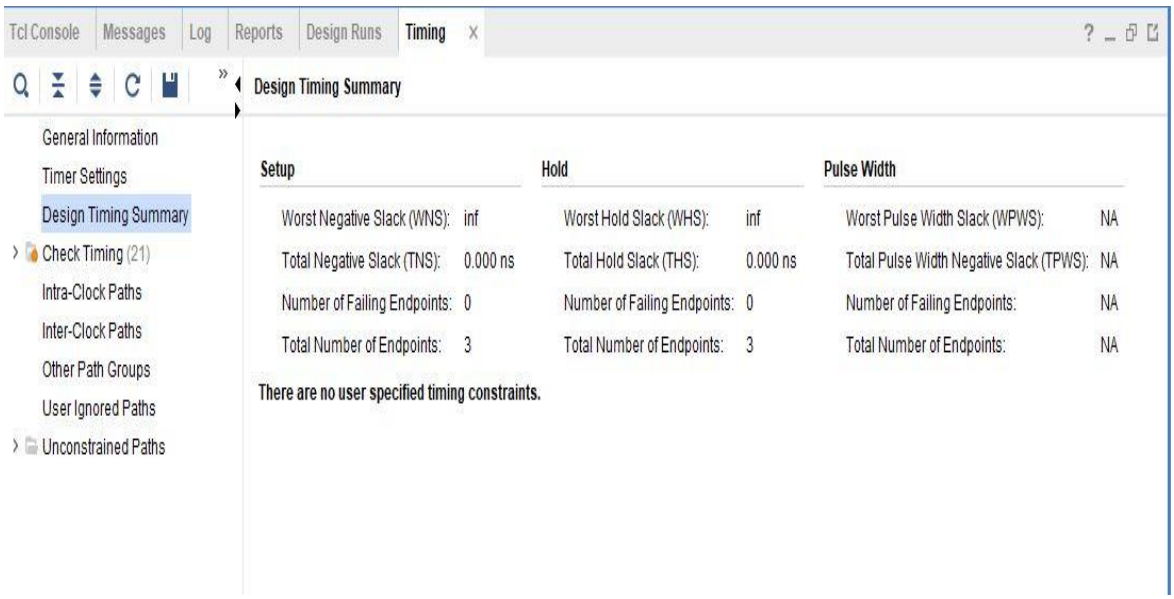
POWER UTILIZATION



REPORT SUMMARY



TIMING SUMMARY



CONCLUSION

We have successfully design digital PID controller on Xilinx 17.4 software with Verilog preferred language.