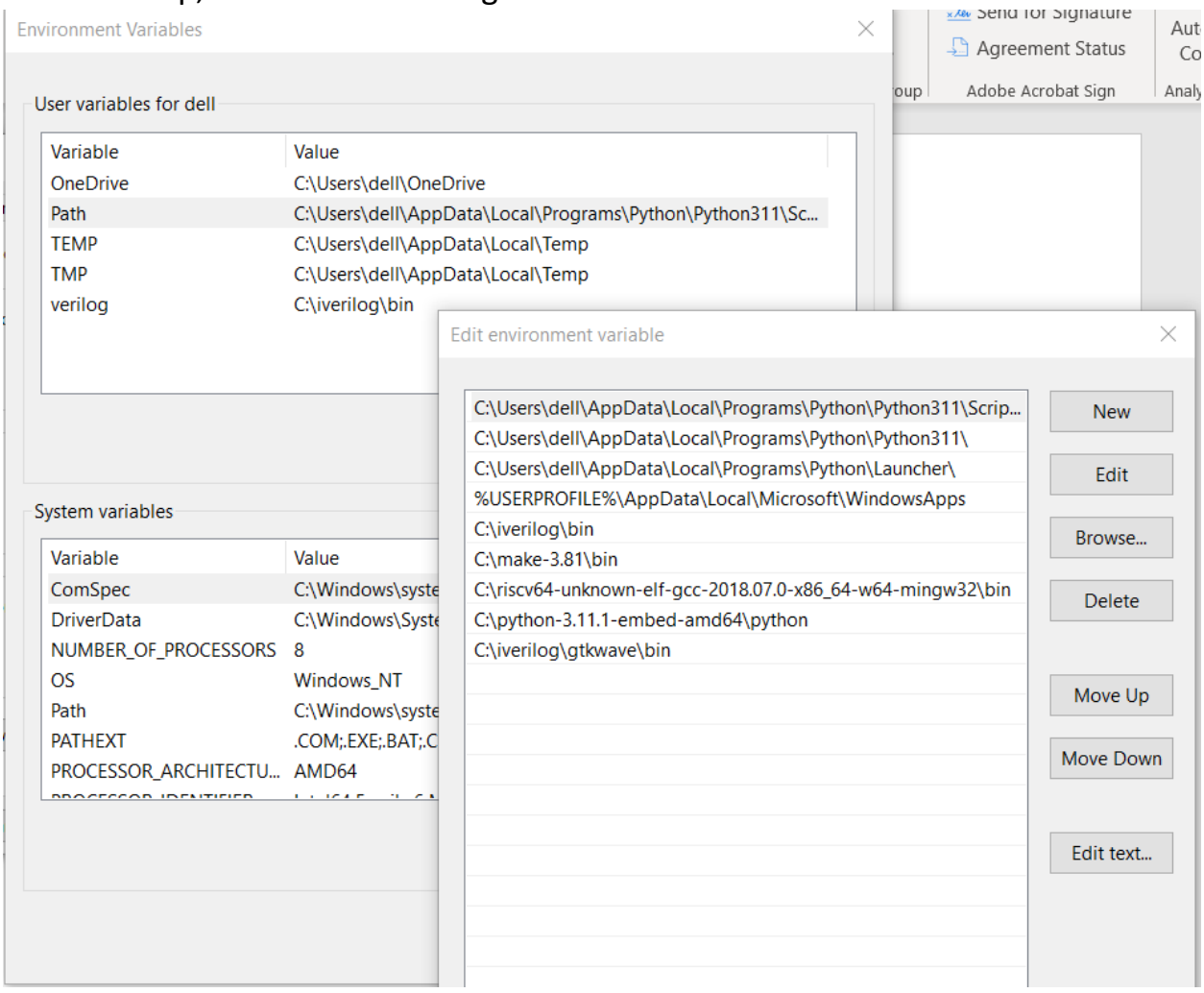
- **Picorv32_v1** is the folder that contains the original picorv32 code with no ecc integration and no modifications.
- **Picorv32HammingandResidue** is the folder that contains the picorv32 code with *Residue error detection in ALU and Hamming Codes in Main State Machine*. So Picorv32 code and the testbenches are modified in this folder. All details about the modifications done are given in this pdf.
- **Picorv32_v1_ecc** is the folder that contains the *RRNS error correction algorithm in ALU*. All details about the modifications done are given in this pdf.

- For the setup, install the following software:



Details about the software to be installed(RISCV toolchain box, Iverilog, Python) are given in the readme document of picorv32 GitHub.

Link: https://github.com/YosysHQ/picorv32

- I have performed the simulations in Icarus Verilog
- I have also uploaded the make-3.81 folder.
- For a better understanding of the code, always take the help of the waveform.
- Also, you may refer to this link also:

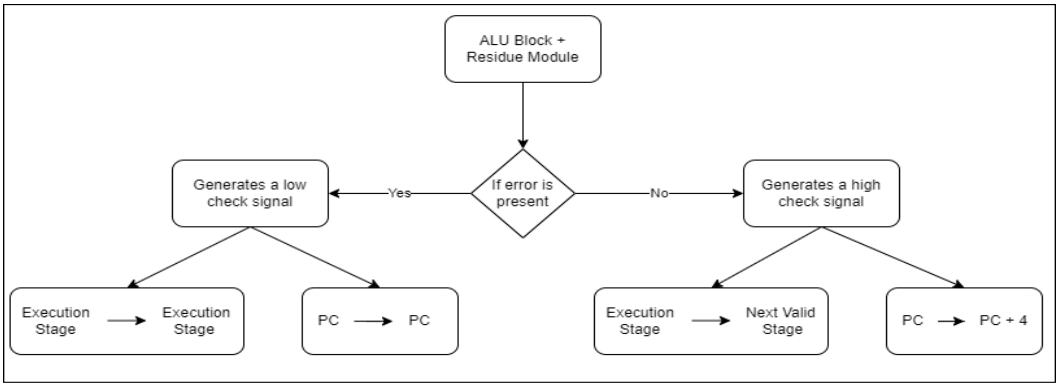
https://hackmd.io/@30vhEV7FQECcWeCF1eAN5A/S1ybboVnK

PicoRV32 Modifications in the picorv32.v file in picorv32HammingandResidue folder:

At line 92 error signal as input is given.

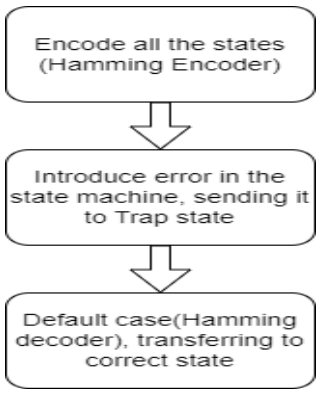
For Error Detection in ALU:

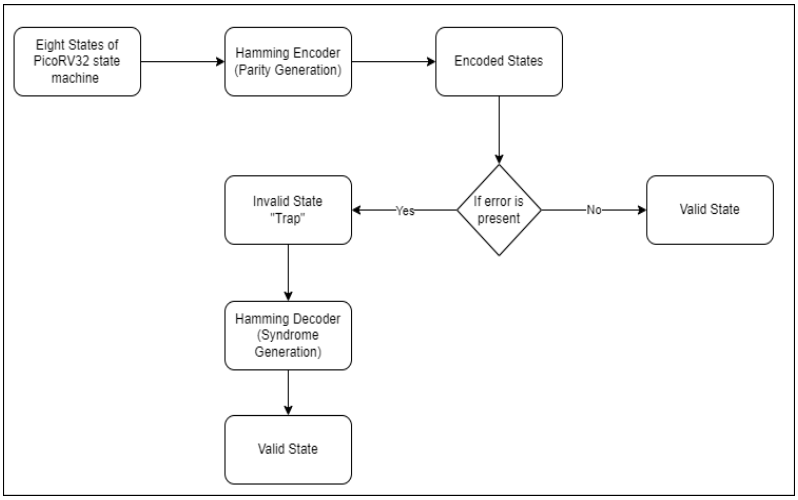
- At line 1271, an error condition is applied to the addition and subtraction operations in ALU.
- From line 1289 to 1299 Residue modules are called for error detection, also at line 1297 a check condition for overflow is given and at line 1299 a check signal is generated, based upon the residue algorithm for subtraction and addition operation, and it is high for all other operations, but if the check signal is low that means error has occurred.
- Now if the check is low, that implies an error is detected and so PC is rollbacked, for this from line 1615 to line 1618 Program Counter's next value is modified based on the check signal.
- Also, if the check is low, the Execution state will remain on hold so that the incorrect result is not passed on, for this the modifications are done from line 1888 to 1895.
- At lines 585 and 591 error=0 condition is also applied so that mem_rdata which depends on mem_valid signal does not get change
- From line 2324 to 2380 modulo7n code is defined.



Hamming Codes Integration in Main State Machine:

- From lines 1201 to 1208, hamming encoder module is called for all the states of the main state machine so that the respective encoded states can be generated.
- At line 1784 error condition is applied here to generate some invalid state
- From line 1987 to 2009, In the default section of the case statement (begins at line 1539), hamming decoder algorithm is written to generate the valid state from the invalid one (error correction).
- From lines 2274 to 2290, hamming encoder module is defined.





Modifications in the testbench of picorv32:

error signal as input is given to the wrapper module in the testbench.v as it calls the picorv32_axi module which is then called the core module of picorv32 (In the core module of PicoRV32, all the ECC Algorithms are defined). (see lines 2734 and 2868 of the picorv32.v file or at lines 115 and 223 of testbench.v of picorv32)

In the picorv32 directory run the command: make test

Modifications in the Dhrystone testbench of picorv32:

Error is injected at several time instants.

In the dhrystone directory run the command: make test

Modifications in the testbench_ez of picorv32:

Random test instructions are given.

In the picorv32 directory run the command: make test_ez_vcd

- The above algorithms (Residue error detection and Hamming code) are functionality correct for all the testbenches.

PicoRV32 Modifications in the picorv32.v file in picorv32_v1_ecc folder:

- At line 91 error input signal is applied (in_err)
- At lines 585 and 591, the error=0 condition is also applied so that mem_rdata which depends on the mem_valid signal does not get changed.
- From lines 1226 to 1231, and at line 1248, the RRNS error correction algorithm is being applied to ALU.
- If there is a single residue error, then the error correction algorithm will do the correction but if there is a multiple residue error, then error detection has to be done. For this at line 1231, an “err” signal is generated which turns high if there is multiple residue error otherwise it is zero.
- So based on the “err”, the PC value is determined as can be seen from line no. 1568 to 1572.
- Also in the above case, the execution state will remain on hold as can be seen from line no. 1838 to 1845.
- From line 2212, the rrns ecc modules are defined.

- The modifications in the testbench folders are the same as shown above in the picorv32Hamming andResidue folder.

- The above algorithm is working correctly for testbench_ez.v but for dhrystone testbench the simulation crashes.