Main Purpose of the Lab

The purpose of this lab is to write Verilog code to implement the CNU(Check Node Updating Units) and VNU(Variable Node Updating Units) and generate the schematic of it. Then generate the layout to implement the CNU and VNU and plot the entire schematic of the LDPC controller.

Introduction:

A Low-density parity-check (LDPC) code is a linear error-correcting code, a method of transmitting a message over a noisy transmission channel. CNU and VNU are very important modules in LDPC code. Let's take an example to understand it.

Suppose we have an input binary code S=(s1 s2 s3), and for example, S= (1, 1, 0)
And we have a generator matrix G= $\begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}$

Then we calculate the codeword C(s)=SG = (c1 c2 c3 c4 c5 c6). Since the code is binary, we need to do a mod 2 operation in the end (i.e. do an XOR).

For example, c4=(s1 s2 s3)*$(1\ 1\ 0)^T$ (which is column 4)=1*1+1*1+0*0=2 $\equiv 0$ (mod 2)
Thus, we have $c4 = s1 \oplus s2$, $c5 = s2 \oplus s3$, $c6 = s1 \oplus s2 \oplus s3$
So for the first part, given an input code S and the generator matrix G, you need to generate a module VNU to calculate C(s) as the output.

On the other hand, we can switch the equation into another form:
$c4 \oplus s1 \oplus s2 = 0$, $c5 \oplus s2 \oplus s3 = 0$, $c6 \oplus s1 \oplus s2 \oplus s3 = 0$
This means we can switch the equations to

$$\begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} c1 \\ c2 \\ c3 \\ c4 \\ c5 \\ c6 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

And we name the first matrix as the parity check matrix H.

Thus, theoretically, we have $HC^T(s) = 0$, and if the result is not zero, we can yield that we didn't transfer the correct code.

Verilog Code for CNU:

```
`timescale 1ns / 1ps
module CNU(
  input [5:0] Cs,
  output [2:0] SO,
  output err
); // error bit, 1 if SO 0, 0 if SO = 0. The default value is 0

  assign SO[0] = Cs[1] ^ Cs[3] ^ Cs[4];
  assign SO[1] = Cs[1] ^ Cs[3] ^ Cs[4];
  assign SO[2] = Cs[1] ^ Cs[2] ^ Cs[5];

  assign err = |SO;

endmodule
```

Test Cases Results:

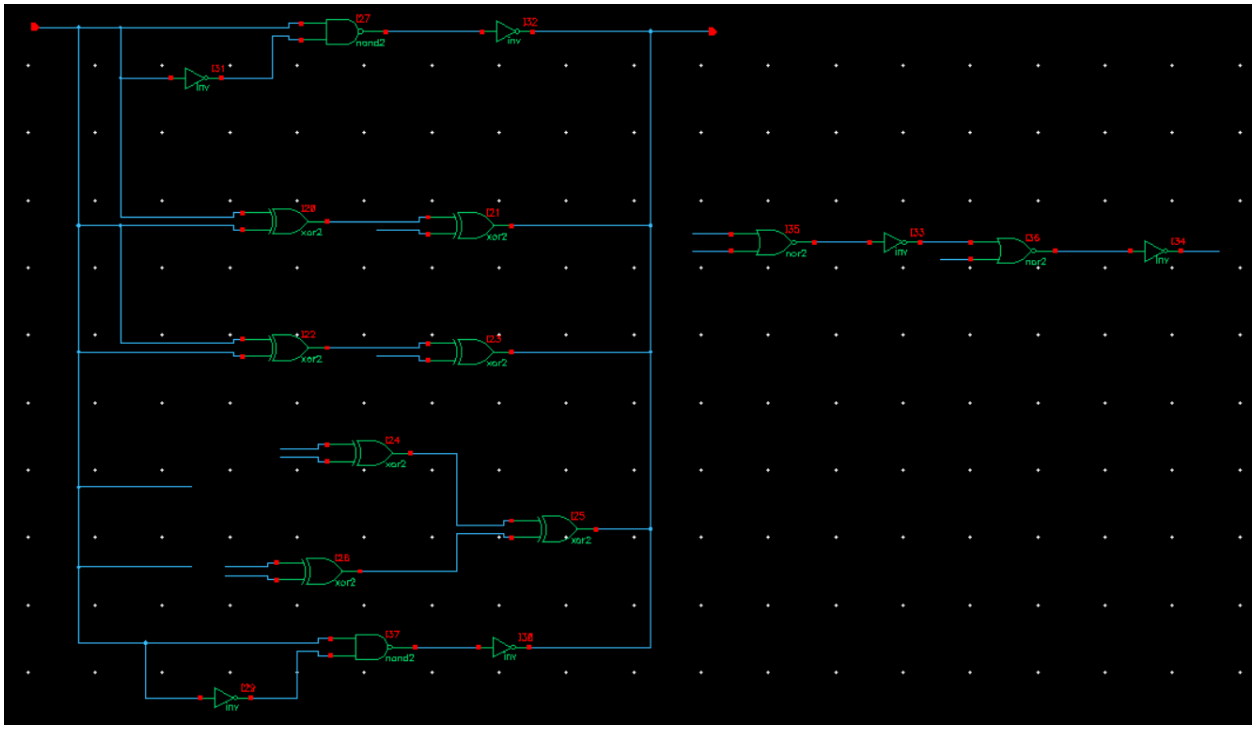
Log | Share

```
[2025-10-23 00:02:57 UTC] iverilog '-Wall' design.sv testbench.sv  && unbuffer vvp a.out
VCD info: dumpfile CNU.vcd opened for output.
     CNU Input 0 passed
     CNU Input 1 passed
     CNU Input 2 passed
     CNU Input 3 passed
All tests passed
Finding VCD file...
./CNU.vcd
[2025-10-23 00:02:57 UTC] Opening EPWave...
Done
```

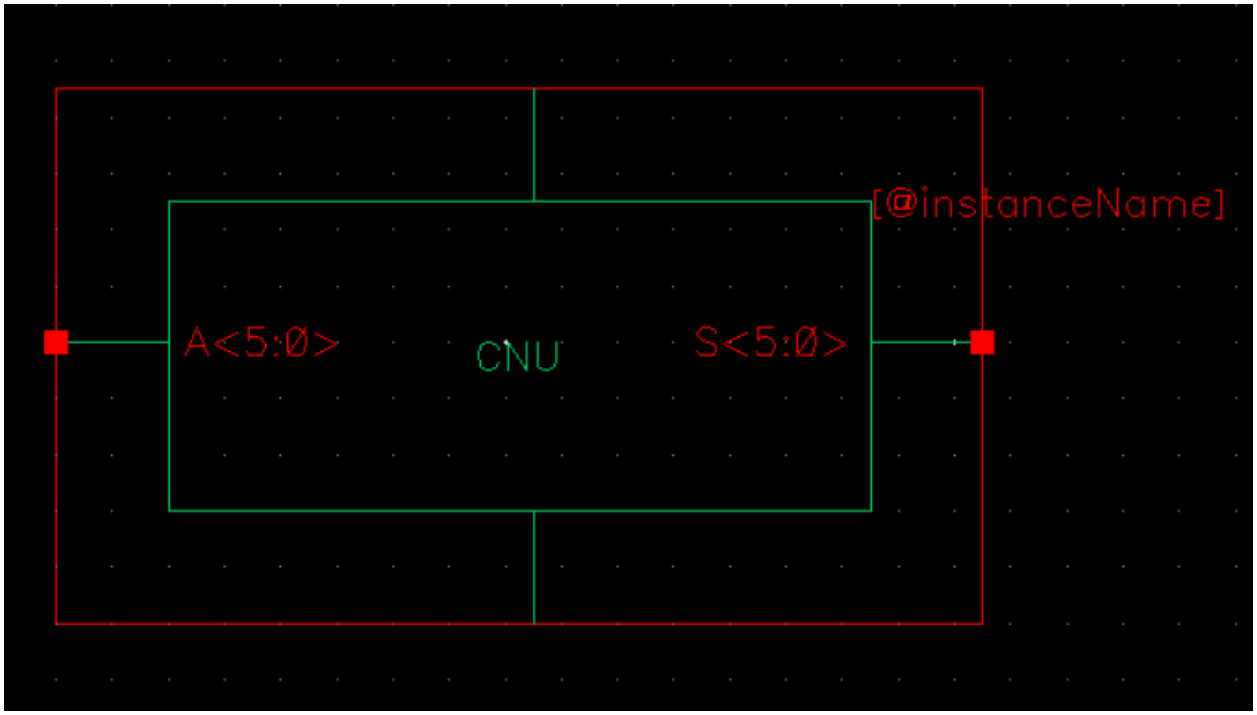
CNU Waveform:

Note: To revert to EPWave opening in a new browser window, set that option on your profile page.

CNU Cadence Schematic:



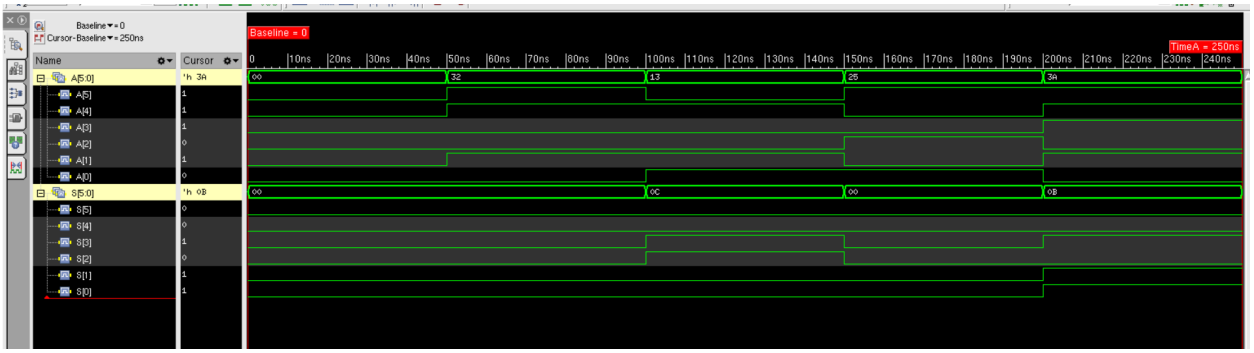
CNU Cadence Symbol:



CNU NC-Verilog Simulation:

```
Relinquished control to SimVision...
ncsim>
ncsim> source /opt/coe/cadence/INCISIVE152/tools/inca/files/ncsimrc
ncsim> database -open shmWave -shm -default -into shm.db
Created default SHM database shmWave
ncsim> probe -create -shm test  -all   -depth 1
Created probe 1
ncsim> run
                   0 A=000000, S=000000
                  50 A=110010, S=000000
                 100 A=010011, S=001100
                 150 A=100101, S=000000
                 200 A=111010, S=001011
Simulation complete via $finish(1) at time 250 NS + 0
./testfixture.verilog:15   #50 $finish;
ncsim>
```

CNU NC-Verilog Simulation Waveform:



VNU Verilog Code:

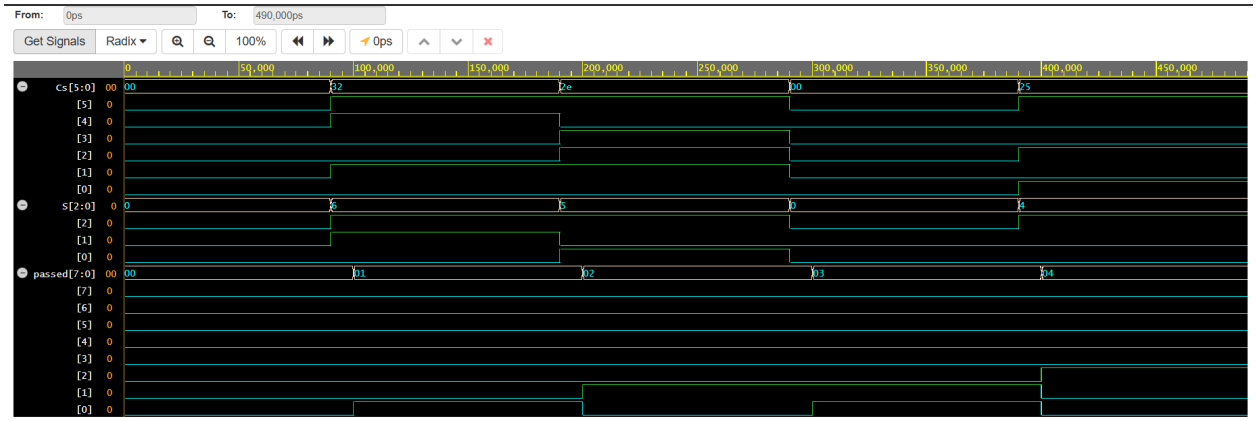
```
`timescale 1ns / 1ps
module VNU(
  output [5:0] Cs,
  input [2:0] S
);

  assign Cs[5] = S[2];
  assign Cs[4] = S[1];
  assign Cs[3] = S[0];
  assign Cs[2] = S[2] ^ S[1];
  assign Cs[1] = S[1] ^ S[0];
  assign Cs[0] = S[2] ^ S[1] ^ S[0];
endmodule
```

VNU Test Cases Results:

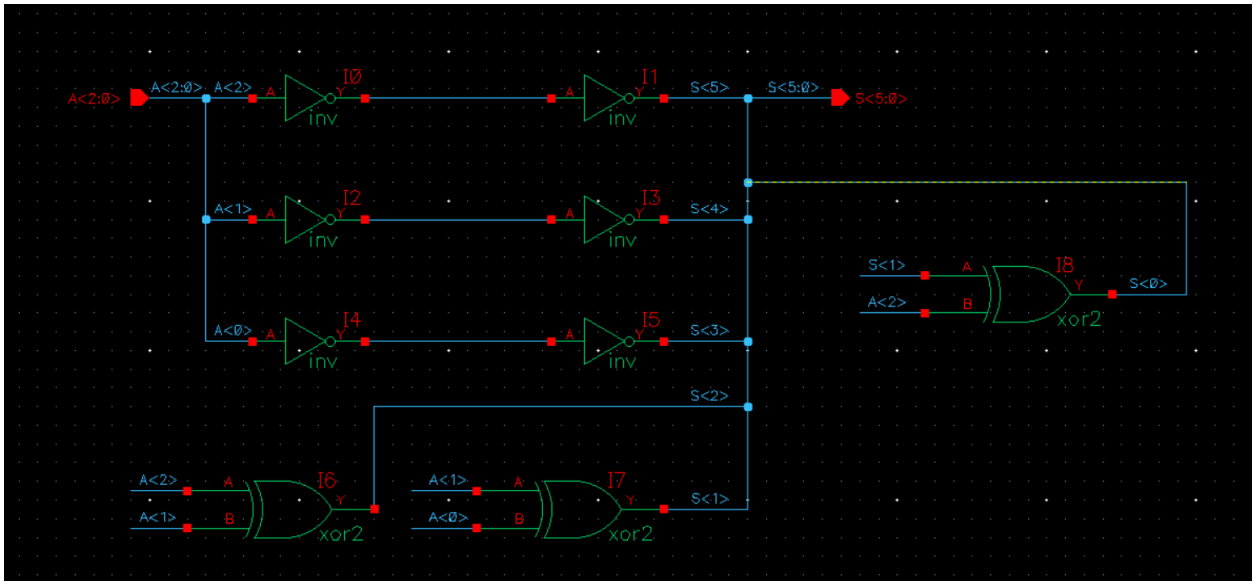
Log | Share

```
[2025-10-22 23:58:12 UTC] iverilog '-Wall' design.sv testbench.sv  && unbuffer vvp a.out
VCD info: dumpfile VNU.vcd opened for output.
     VNU Input 0 passed
     VNU Input 1 passed
     VNU Input 2 passed
     VNU Input 3 passed
All tests passed
Finding VCD file...
./VNU.vcd
[2025-10-22 23:58:13 UTC] Opening EPWave...
Done
```

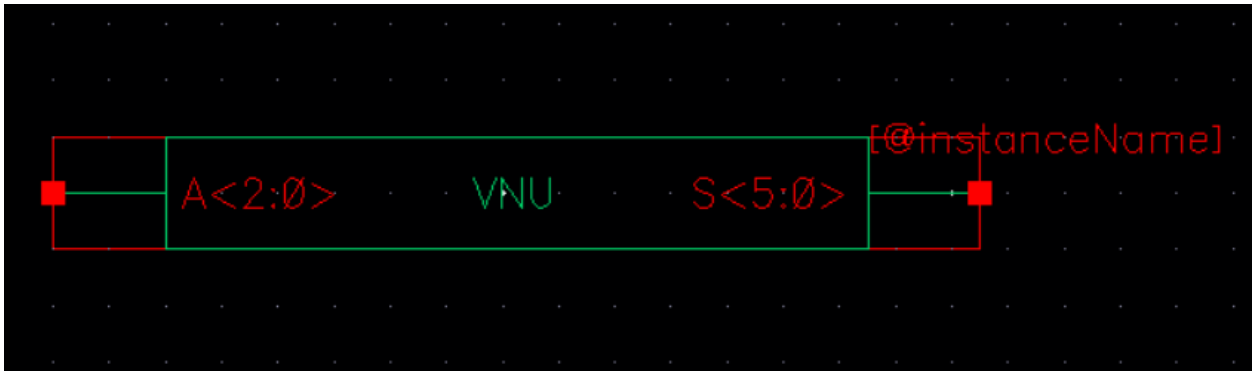
VNU Waveform:



VNU Cadence Schematic:



VNU Cadence Symbol:



VNU NC-Verilog Simulation:

```
-------------------------------------
Relinquished control to SimVision...
ncsim>
ncsim> source /opt/coe/cadence/INCISIVE152/tools/inca/files/ncsimrc
ncsim> database -open shmWave -shm -default -into shm.db
Created default SHM database shmWave
ncsim> probe -create -shm test  -all   -depth 1
Created probe 1
ncsim> run
                   0 A=000, S=000000
                  50 A=110, S=110010
                 100 A=101, S=101110
                 150 A=000, S=000000
                 200 A=100, S=100101
Simulation complete via $finish(1) at time 250 NS + 0
./testfixture.verilog:15   #50 $finish;
ncsim>
```

VNU NC-Verilog Simulation Waveform: