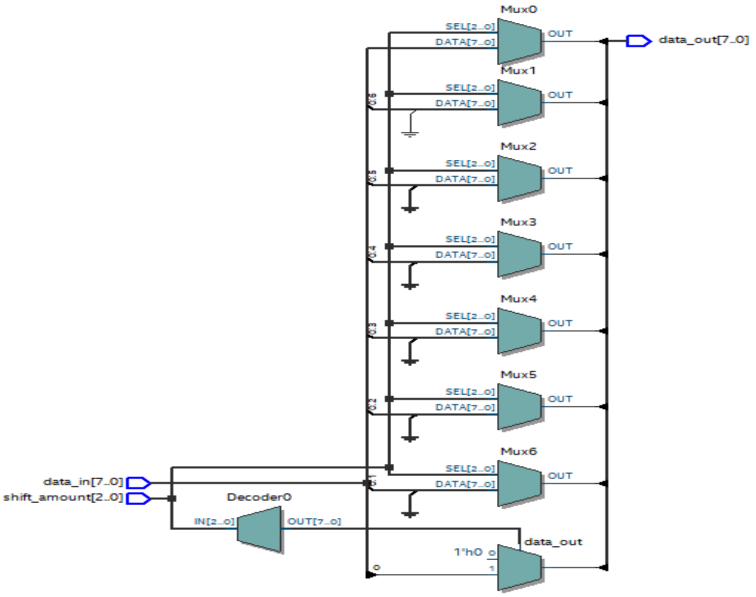
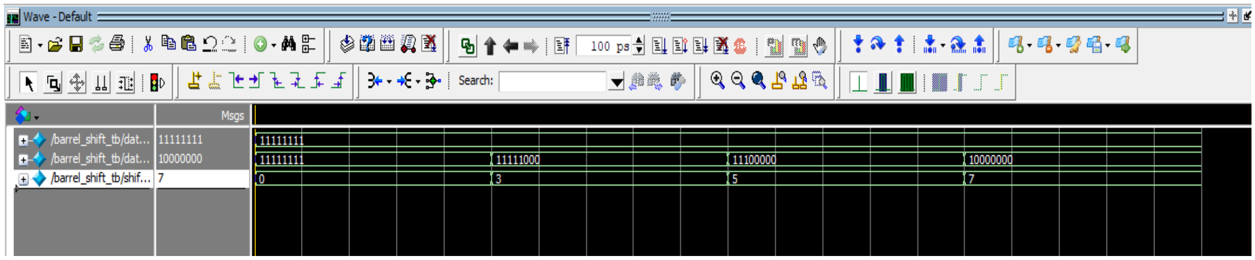
6. Barrel Shifter without using Shift Operator

Design:



- ❖ Designed a module named “barrel_shift” for an assumed ‘Left Shift Operation’, having ports “data_in” (8 bits) and “shift_amount” (3 bits) as inputs and “data_out” (8 bits) as output.
- ❖ Inside an ‘always_comb’ block, wrote all possible shift_amount values that can be done in 3 bits using case block.
- ❖ For every shift_amount case, wrote the corresponding extent of left shift to be done in data_in.

Verification:


Wave - Default
100 ps
Search:
Msgs
/barrel_shift_tb/dat... 11111111
/barrel_shift_tb/dat... 10000000
/barrel_shift_tb/shif... 7
11111111
11111111
11111000
11100000
10000000
0
3
5
7

```
# Applying input data of all 1's
# shift_amount = 0, data_out = 11111111, expected = 11111111
# PASS
# shift_amount = 3, data_out = 11111000, expected = 11111000
# PASS
# shift_amount = 5, data_out = 11100000, expected = 11100000
# PASS
# shift_amount = 7, data_out = 10000000, expected = 10000000
# PASS
```

- Designed testbench to validate the barrel_shift module by applying left shifts to a supposed 8-bit input of all 1’s.
- It uses a function block to compute the expected result using built-in shift logic (<<) and compares it with the output.
- Few cases of shift amount from 0 to 7 are tested to confirm correct bit movement and zero-padding behavior.