

Barrel Shifter (4-bit)

- This is a combinational ckt (not sequential)
- It shifts the i/p by 0-3 positions (4-bit) either left or Right based on direction (dir)
- Uses multiplexing logic (case statement) to instantly rotate/shift data - no clk, no seq. logic
- Code explained :-

⇒ Module ports

1. `data_in [3:0]` ⇒ 4-bit i/p word
2. `shift_amt [1:0]` ⇒ shift 0, 1, 2, 3
3. `dir` ⇒ shift directⁿ

⇒ 0 = logical left shift

⇒ 1 = logical right shift

4. `data_out [3:0]` ⇒ shifted o/p.

⇒ core logic :- (always @(*) pure combinational)

A nested case block selects shift directⁿ first then select the shift amount.

⇒ left logical shift (dir = 0)

<u>shift_amt</u>	<u>opⁿ</u>	<u>o/p pattern</u>
0	no-shift	<code>data_in</code>
1	shift by 1 (left)	<code>{ data_in [2:0], 1'b0 }</code>
2	shift left by 2	<code>{ data_in [1:0], 2'b00 }</code>
3	shift left by 3	<code>{ data_in [0], 3'b000 }</code>

* MSB move towards left, 0 fill at right

⇒ Right logical shift (dir = 1)

shift_amt

opⁿ

o/p pattern

0

no shift

data-in

1

shift right by 1

{1'b0, data-in[3:1]}

2

shift right by 2

{2'b00, data-in[3:2]}

3

shift right by 3

{3'b000, data-in[3]}

* LSBs moves towards the right, zeros fill on the left

eg:- input = 01101

① shift_amt = 2, dir = 0 (left)

1101 = 1010 = [0100] = 4
 ← ←

② shift_amt = 3, dir = 1 (right)

1101 = 0110 = 0010 = [0001] = 1
 → → →

- Hence ckt implements pure logical shifting (no rotation, no arithmetic sign extend).

