`always` Block Variations

- System Verilog has three specialized always blocks.
- These blocks reduce the ambiguity when modeling hardware.
- `always_comb` indicates intent to model combinatorial logic.
- `always_ff` indicates intent to model sequential logic.
- `always_latch` indicates intent to model latch-based logic.
- These specialized procedural blocks act just like any always block, but they enforce synthesis rules so that the desired logic is created.
- These blocks clearly indicate design intent.

`always` Block Variations

- `always_comb`
  - Indicates intent is to model combinatorial logic.

```
always_comb
  if(!mode)
    y = a + b;
  else
    y = a - b;
```

  - No sensitivity list is required. It is automatically inferred.
  - Variables on LHS of assignments cannot be assigned to by other procedural blocks.
  - `always_comb` always triggers once just after all initial and always blocks have run to initalize logic correctly.

`always` Block Variations

- `always_latch`
  - Indicates intent is to model latch-based logic.

```
always_comb
  if(!mode)
    y <= a + b;
```

  - No sensitivity list is required. It is automatically inferred.
  - Variables on LHS of assignments cannot be assigned to by other procedural blocks.
  - `always_latch` always triggers once just after all initial and always blocks have run to initalize logic correctly.

always Block Variations

- always_ff
  - Indicates intent to model sequential flip-flop based logic.

```
always_ff @(posedge clk, negedge reset_n)
  if(!reset_n)
    q <= 1'b0;
  else
    q <= d;
```

  - Sensitivity list is required. Each signal must be specified with either posedge or negedge.
  - Variables on LHS of assignments cannot be assigned to by other procedural blocks.
  - always_ff always triggers once just after all initial and always blocks have run to initalize logic correctly.