1.Localparam’s cannot be modified during module instantiation parameters where as a param can be.

2.For asynchronous design, wires and registers can be used in a similar fashion. However, in a synchronous design registers can be used to store values with a proper clock signal; wires cannot do this as they are simply connections. Initialing wires and registers is different with the “reg” and “wire” keyword. Also wires can be assigned values whereas registers cannot, registers need a “always @” signal to be given a value. There is no advantage to using registers over wires in combinational logic.

3. Registers can be used in sequential spots and wires can be used in combinational logic. For example, registers can be used to hold the current and next state as they update at every clock cycle. On the other hand, you would use a wire to send control signals to the state machine to determine what the next state should be.

4.Blocking assignments are shown by “=” symbol and evaluate one after the other, whereas, non-blocking “=>” statements evaluate all parallel statements at once. Somewhat like sequential vs combinational.