Report for OctalConverter

File 1: OctalConverter_Pkg.vhd

- **Package Declaration: Defines a package OctalConverter_Pkg that includes a record type RecordType.**
- **RecordType: Contains three fields:**
  - **as_integer: An integer.**
  - **as_octal: An integer representing the octal value.**
  - **as_std_logic_vec: A 9-bit std_logic_vector.**

```
5  package OctalConverter_Pkg is
6      -- Define a record type to hold integer, octal, and std_logic_vector values
7      type RecordType is record
8          as_integer      : integer;
9          as_octal        : integer;
10         as_std_logic_vec: std_logic_vector(8 downto 0);
11     end record;
12 end OctalConverter_Pkg;
```

File 2: OctalConverter.vhd

- **Entity Declaration: Defines the entity OctalConverter with:**
  - **Inputs: A (integer ranging from 0 to 511) and B (9-bit std_logic_vector).**
  - **Outputs: X and Y (both of type RecordType).**
- **Architecture:**
  - **Signals: Internal signals octalA, octalB, hundreds, tens, and unit for intermediate conversion results.**
  - **Conversion Logic:**
    - **Converts integer input A to its octal representation and stores it in octalA.**
    - **Calculates each octal digit for B using conditional statements and stores them in hundreds, tens, and unit.**
    - **Combines the octal digits into a single integer for octalB.**
  - **Record Population:**
    - **Populates record X with the integer input A, its octal representation octalA, and its std_logic_vector representation.**
    - **Populates record Y with the integer representation of the std_logic_vector input B, its octal representation octalB, and the std_logic_vector itself.**

```
6  entity OctalConverter is
7      Port (
8          A : in integer range 0 to 511;  -- Integer input
9          B : in std_logic_vector(8 downto 0);  -- std_logic_vector input
10         X : out RecordType;  -- Record for integer input A
11         Y : out RecordType   -- Record for std_logic_vector input B
12     );
13 end OctalConverter;
```

File 3: OctalConverter_tb.vhd

- **Entity Declaration: Defines the test bench entity OctalConverter_tb with no ports.**
- **Architecture:**
  - **Component Declaration: Declares the OctalConverter component.**
  - **Signal Declarations: Signals for the inputs and outputs of the OctalConverter.**
  - **Instantiation: Instantiates the OctalConverter component and maps its ports to the signals.**
  - **Stimulus Process:**
    - **Test Cases:**
      - **Test case 1: Inputs A = 123 and B = "001111010" (decimal 123).**
      - **Test case 2: Inputs A = 256 and B = "100000000" (decimal 256).**
      - **Test case 3: Inputs A = 511 and B = "111111111" (decimal 511).**
      - **Test case 4: Inputs A = 0 and B = "000000000" (decimal 0).**
    - **Assertions: Uses assert statements to check if the outputs match the expected values. If not, an error message is reported.**

```
39      -- Stimulus process
40      process
41      begin
42          -- Test case 1
43          A <= 123;  -- Integer input
44          B <= "001111010";  -- Equivalent of decimal 123
45          wait for 10 ns;
46          assert (X.as_integer = 123) and (X.as_octal = 173) and (X.as_std_logic_vec = std_logic_vector(to_unsigned(123, 9)))
47              report "Test Case 1 Failed: X values are incorrect" severity error;
48          assert (Y.as_integer = 123) and (Y.as_octal = 173) and (Y.as_std_logic_vec = B)
49              report "Test Case 1 Failed: Y values are incorrect" severity error;
```

Summary

The provided VHDL files implement an octal converter that converts integer and std_logic_vector inputs to their octal representations and outputs them in a record type. The test bench verifies the functionality of the converter by applying various test cases and checking the outputs using assertions. The report summarizes the key components and functionality of each file, providing a concise overview of the implementation and testing process