

Q: The **ConvertToInteger** function takes a binary vector **BinaryInput** as input and calculates the corresponding integer. The function returns an *integer*. First, write the function. Then, write a design description of the **ConvertToInteger check circuit** (*entity name: ConvertToInteger_check_voting*), i.e., if the corresponding integer of the input *bit_vector* “BinaryInput” > 60, then the *output great_than_60* = TRUE. The circuit uses the function **ConvertToInteger**. The **ConvertToInteger check circuit** has the following entity declaration.

```
library ieee;
use ieee.std_logic_1164.all;
```

```
entity ConvertToInteger_check_voting is
    port(BinaryInput: in bit_vector(15 downto 0);
          great_than_60: out Boolean);
end ConvertToInteger_check_voting;
```

```
architecture behavioral of ConvertToInteger_check_voting is
    function ConvertToInteger (slv : bit_vector) return integer is
```

```
        variable total: integer := 0;
        for i in slv'range loop
            if (slv(i) = '1') then
                total := total + 2**i;
            end if;
        end loop;
        return integer;
```

```
begin
    process (BinaryInput)
    begin
```

```
        great_than_60 <= ConvertToInteger(BinaryInput) > 60;
```

```
    end process ;
```

```
end behavioral;
```