

## Verification Guide

# SystemVerilog Shallow Copy

## Shallow Copy

### Table of Contents



#### 1. Shallow Copy

##### 1.1. Shallow copy example

###### 1.1.1. Difference between Assignment and Shallow Copy

An object will be created only after doing new to a class handle,

```
packet    pkt_1;
pkt_1    = new();
```

```
packet    pkt_2;
pkt_2    = new pkt_1;
```

In the last statement, pkt\_2 is created and class properties were copied from pkt\_1 to pkt\_2, this is called as “shallow copy”.

Shallow copy allocates the memory, copies the variable values and returns the memory handle.

In shallow copy, All of the variables are copied across: integers, strings, instance handles, etc.

**Note::**

Objects will not be copied, only their handles will be copied.

to perform the full or deep copy, the custom method can be added.

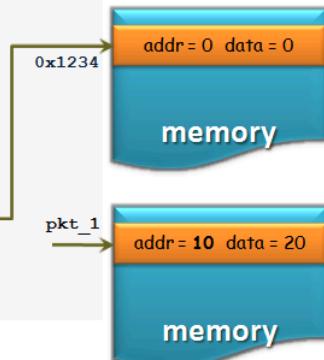
**Shallow Copy:**

```
//declare the handle's
```

```
packet pkt_1;
packet pkt_2;
```

```
//construct the object
```

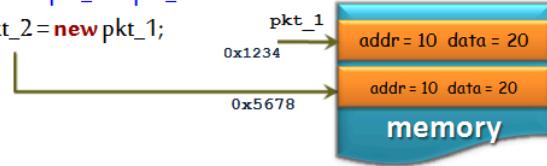
```
pkt_1=new();
pkt_1.addr=10;
pkt_1.data=20;
```



```
//Shallow copy,
```

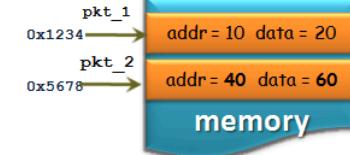
```
//allocates the memory and copies the variable values
//from pkt_1 to pkt_2
```

```
pkt_2=new pkt_1;
```



```
pkt_2.addr= 40;
```

```
pkt_2.data= 60;
```



[www.verificationguide.com](http://www.verificationguide.com)

SystemVerilog Class Shallow Copy

### Limitation of Shallow Copy:

```

class address_range;           //declare the handle's
    bit[31:0] s_addr;
    bit[31:0] e_addr;
endclass

//class declaration
class packet;
    //class properties
    bit[31:0] addr;
    bit[31:0] data;
    address_range ad_r;
    //constructor
    function new();
        //creating object
        ad_r=new();
    endfunction
endclass

```

On Shallow copy, allocates the memory for `pkt_2` and copies the variable values from `pkt_1` to `pkt_2`. but the internal objects **shares** the same memory.  
i.e, `pkt_1.ad_r` and `pkt_2.ad_r` shares the same memory.

[www.verificationguide.com](http://www.verificationguide.com)

*SystemVerilog Shallow Copy Limitation*

## Shallow copy example

In the below example,

packet class has the properties of bit type and object type (address\_range). after the shallow copy addr, data and handle to ar were copied. As it is shallow copy any changes on `pkt_2.ar` will reflect in `pkt_1.ar` (because `pkt_2.ar` and `pkt_1.ar` will point to the same object).

```

///-- class ---
class address_range;
    bit [31:0] start_address;
    bit [31:0] end_address ;
    function new();    start_address = 10;
                      end_address   = 50;
    endfunction
endclass

///-- class ---

```

```

class packet;
    //class properties
    bit [31:0] addr;
    bit [31:0] data;
    address_range ar; //class handle

    //constructor
    function new();
        addr = 32'h10;
        data = 32'hFF;
        ar = new(); //creating object
    endfunction
    //method to display class properties
    function void display();
        $display("-----");
        $display("\t addr = %0h",addr);
        $display("\t data = %0h",data);
        $display("\t start_address = %0d",ar.start_address);
        $display("\t end_address = %0d",ar.end_address);
        $display("-----");
    endfunction
endclass

// -- module ---
module class_assignment;
    packet pkt_1;
    packet pkt_2;

    initial begin
        pkt_1 = new(); //creating pkt_1 object
        $display("\t**** calling pkt_1 display ****");
        pkt_1.display();

        pkt_2 = new pkt_1; //creating pkt_2 object and copying pkt_1 to
        pkt_2
        $display("\t**** calling pkt_2 display ****");
        pkt_2.display();

        //changing values with pkt_2 handle
        pkt_2.addr = 32'h68;
        pkt_2.ar.start_address = 60;
        pkt_2.ar.end_address = 80;
        $display("\t**** calling pkt_1 display after changing pkt_2
properties ****");

        //changes made to pkt_2.ar properties reflected on pkt_1.ar, so only
        //handle of the object get copied, this is called shallow copy
    end
endmodule

```

```
pkt_1.display();
$display("\t**** calling pkt_2 display after changing pkt_2
properties ****");
pkt_2.display(); //
end
endmodule
```

## Simulator Output

```
**** calling pkt_1 display ****
-----
addr = 10
data = ff
start_address = 10
end_address = 50
-----
**** calling pkt_2 display ****
-----
addr = 10
data = ff
start_address = 10
end_address = 50
-----
**** calling pkt_1 display after changing pkt_2 properties ****
-----
addr = 10
data = ff
start_address = 60
end_address = 80
-----
**** calling pkt_2 display after changing pkt_2 properties ****
-----
addr = 68
data = ff
start_address = 60
end_address = 80
-----
```

Click to execute on EDA playground

## Difference between Assignment and Shallow Copy

| Class Assignment                                                                                                                                                                          | V/S                                                                                                                                                                                                                                                                                                                                                                                                                 | Shallow Copy                                                                                                                                                                                                                                                                                                                                                                                         |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>//class declaration class packet; //class properties bit [31:0] addr; bit [31:0] data; //constructor function new (bit [31:0] a,b ); addr = a; data = b; endfunction  endclass</pre> | <p>//declare the handle<br/>packet pkt_1;<br/>packet pkt_2;</p> <p>//construct the object<br/>pkt_1=new(10,20);<br/>pkt_2=pkt_1;//assignment</p> <ul style="list-style-type: none"> <li>With the assignment, both the objects will point to the same memory.</li> <li>Changes made by using pkt_2 will reflect on pkt_1, because both share the same memory.</li> </ul> <p>pkt_2.addr=30;      pkt_1      pkt_2</p> | <p>//declare the handle<br/>packet pkt_1;<br/>packet pkt_2;</p> <p>//construct the object<br/>pkt_1=new(10,20);<br/>pkt_2=new pkt_1;//shallow copy</p> <ul style="list-style-type: none"> <li>Shallow Copy, Allocates the memory, copies the variable and returns the memory handle.</li> <li>Changes made by using pkt_2 will not reflect on pkt_1, because both share different memory.</li> </ul> |

[www.verificationguide.com](http://www.verificationguide.com)

assignment vs shallow copy

[◀ Previous](#)
[Next ▶](#)

Verification Guide / Proudly powered by WordPress