Chapter 8:

Chapter 8. Tasks and Functions

8.5 Exercises

1. Define a function to calculate the factorial of a 4-bit number. The output is a 32-bit value. Invoke the function by using stimulus and check results.

My answer:

```
1   //ex8-1 factorial
2
3   module top;
4
5
6   function automatic integer factorial;
7   input [3:0] oper;
8
9   begin
10    if(oper >=2)
11      factorial=factorial(oper-1)*oper;
12    else
13      factorial=1;
14  end
15  endfunction
16
17  integer result;
18  initial
19  begin
20    result=factorial(10);
21    $display("Factorial of 10 is %d",result);
22  end
23
24  endmodule
```

Factorial of 4 is 3628800

2. Define a function to multiply two 4-bit numbers a and b. The output is an 8-bit value. Invoke the function by using stimulus and check results.

My answer:

```
1   //ex8-2 multiply
2
3   module top;
4
5   function [7:0] product;
6   input [3:0] a,b;
7   begin
8     product=a*b;
9   end
10  endfunction
11
12  reg [3:0] a,b;
13  reg [7:0] result;
14
15  initial
16  begin
17    a=4'd15; b=4'd10;
18    result=product(a,b);
19    $display("a x b= %d",result);
20  end
21
22  endmodule
```

a x b= 150

3. Define a function to design an 8-function ALU that takes two 4-bit numbers a and b and computes a 5-bit result out based on a 3-bit select signal. Ignore overflow or underflow bits.

Select Signal	Function Output
3'b000	a
3'b001	a+b
3'b010	a-b
3'b011	a/b
3'b100	a%b
3'b101	a<<1
3'b110	a>>1
3'b111	(a>b)

My answer:

```
 1  //ex8-3
 2
 3  module top;
 4
 5  function [4:0] out;
 6  input [2:0] s;
 7  input [3:0] a,b;
 8  begin
 9    case(s)
10    3'b000:out=a;
11    3'b001:out=a+b;
12    3'b010:out=a-b;
13    3'b011:out=a/b;
14    3'b100:out=a%b;
15    3'b101:out=a<<1;
16    3'b110:out=a>>1;
17    3'b111:out=(a>b);
18    endcase
19  end
20  endfunction
21
22  reg [3:0] x,y;
23  reg [2:0] s1;
24  reg [4:0] result;
25
26  initial
27  begin
28    x=4'd15; y=4'd10;
29    s1=3'd0;
30  end
31
32  always
33  begin
34    #10 s1=s1+1;
35    result=out(s1,x,y);
36  end
37
38  initial
39    $monitor($time, " s1= %b, result= %d",s1,result);
40
41
42  endmodule
```

```
 0 s1= 000, result=  x
10 s1= 001, result= 25
20 s1= 010, result=  5
30 s1= 011, result=  1
40 s1= 100, result=  5
50 s1= 101, result= 30
60 s1= 110, result=  7
70 s1= 111, result=  1
80 s1= 000, result= 15
90 s1= 001, result= 25
```

4. Define a task to compute the factorial of 4-bit number. The output is a 32-bit value. The result is assigned to the output after a delay of 10 time units.

My answer:

```
1   //ex8-4
2
3   module top;
4
5   parameter delay=10;
6   reg [3:0] n;
7   reg [31:0] n_fac;
8
9   initial
10  begin
11    n=4'd5;
12    #delay factorial(n_fac,n);
13    $display($time, "Factorial of n is %d",n_fac);
14  end
15
16  task automatic factorial;
17  output [31:0] x_fac;
18  input [3:0] x;
19  begin
20  if(x>=2)
21    begin
22      factorial(x_fac,x-1);
23      x_fac=x_fac*x;
24    end
25  else
26   x_fac=1;
27  end
28  endtask
29
30  endmodule
```

```
10Factorial of n is        120
```

5. Define a task to compute even parity of a 16-bit number. The result is a 1-bit value that is assigned to the output after three positive edges of clock. (Hint: Use a repeat loop in the task).

My answer:

```
1   //ex8-5
2
3   module parity;
4
5   reg [15:0] addr;
6   reg par;
7   reg clk;
8
9   initial
10  begin
11    clk=1'b0;
12    forever #10 clk=~clk;
13  end
14
15  initial
16    addr=16'd11;
17    //#30 addr=16'd3;
18
19
20  always @(posedge clk )
21  begin
22    calc_parity(par,addr);
23    $display("Parity calcultaed= %b",par);
24  end
25
26  task calc_parity;
27  output p;
28  input [15:0] a;
29  begin
30    repeat(3) @(posedge clk);
31    p=^a;
32
33  end
34  endtask
35
36  endmodule
```