NMK20603

Computer Architecture

Testing dreg

MY1

Practice:

⇒ task>dreg_write

```
// modified from dff_wr
// assume wr_enb drive signal dwen
// using parameter BITS
reg dwen;
reg[BITS-1:0] ddat;
task dreg_write ;
  input integer dval;
  begin
    ddat = dval; // setup data for write
    dwen = 1'b1; // enable write for 1 clk period
    $display("-- [%05g] WR-dreg:'%b' ",$time,ddat);
    #(CLKP/2); // setup time
    #(CLKP/2); // hold time
    dwen = 1'b0; // disable write
  end
endtask
```

MY1

Practice:

⇒ 'monitor' code for dreg?

```
// similar to that in dff_tb
// monitor reg changes (NOT output)
always @(dut.tdat) begin
  $write("## [%05g] dregQ=%b\n",$time,dut.tdat);
end
```

Practice:

⇒ task>dreg_read

```
// assume rd_enb drive signal dren
// assume output net is mdat
// assume read on +ve
// call on -ve clk edge, mutual exclusive to dreg_wr
reg dren;
wire[BITS-1:0] mdat;
task dreg_read ;
  begin
    dren = 1'b1; // enable read for 1 clk period
    #(CLKP/2); // setup time
    $display("@@ [%05g] RD-dreg:'%b' ",$time,mdat);
    #(CLKP/2); // hold time
    dren = 1'b0; // disable read
  end
endtask
```

Practice:

⇒ reset condition?

```
dwen = 1'b0;
dren = 1'b0;
```

Practice:

dreg_tb

```
module dreg_tb ();
parameter BITS=4;
// clock generator
// task:dreg_write
// dreg output monitor
// task:dreg_read
// the rest...
initial begin
  dclk = 1'b1; //ddat = 4'h5;
  dwen = 1'b0; dren = 1'b0;
  // start test
  dreg_read();
  #(CLKP/2); // wait for -ve edge
  dreg_write(4'b1010);
  #CLKP;
  dreg_read();
  dreg_write(12);
  #CLKP;
  dreg_read();
  $stop;
end
defparam dut.BITS = BITS;
dreg dut (dclk,dwen,dren,ddat,mdat);
endmodule
```

Something

'extra'

MY1

Revisiting System Tasks

Code Examples

```
module systask1_tb ();
initial begin
  $display("-- Stopping");
  $stop;
  $display("-- Finishing");
  $finish;
end
endmodule
```

```
module systask2_tb ();
integer keep,loop;
reg[9:0] temp;
initial begin
  keep=$time;
  #10 $display("-- SimTime:[%%d:%d][%%g:%g]",keep,keep);
  #10 $display("-- SimTime:[%%5d:%5d] => numeric prefix sets spacing",$time);
  #10 $display("-- SimTime:[%%05d:%05d] => zero prefix for 0-padding",$time);
  #10 $display("-- SimTime:[%%5g:%5g] => same for g",$time);
  #10 $display("-- SimTime:[%%05g:%05g]",$time);
  temp=$time;
  #10 $display("-- SimTime:[%%8b:%8b] =>  b is always 0-padded",temp);
  #10 $display("-- SimTime:[%%08b:%08b]",temp);
  for (loop=0;loop<3;loop=loop+1) begin
    #10 $display("-- SimTime:[%05g]",$time);
  end
  $stop;
end
endmodule
```

```
module systask3_tb ();
initial begin
  $write("-- 1st Line... ");
  $write("still here!\n");
  $display("-- Next Line");
  $display("-- Next Line");
  $stop;
end
endmodule
```

That's all!

MY1