saved 12 minutes ago

EDITOR ✓NAV-TLV LOG

```
\m5_TLV_version 1d: tl-x.org
\m5

   // ========================================
   // Welcome, new visitors! Try the "Learn" menu.
   // ========================================

   //use(m5-1.0)   /// uncomment to use M5 macro library.
\SV
   // Macro providing required top-level module definition, random
   // stimulus support, and Verilator config.
   m5_makerchip_module   // (Expanded in Nav-TLV pane.)
\TLV
   // Combinational Calculators for Addition, Subtraction, Multiplication
   // $reset = *reset;
   // Stimulus ......
   |calc
      @0
         // Stimulus for Calculator .....
         $reset = *reset;
         $op[1:0] = *cyc_cnt[1:0];
         $reset_zero[31:0] = 32'b0;
         $val2[31:0] = $rand1[3:0];
         // $cnt[0:0] = *cyc_cnt[0:0];
         $valid[0:0] = & ($reset || $op);
         // $valid = &($op || $reset);

        // Stimulus for Free Runing Counter.......
         $inp1[0:0] = 1;
         $inp2[0:0] = 0;
         // Summation of next state (feedback) and trigger input "1".....
         $sum_sq[0:0] = $inp1 + >>1$cnt[0:0];
   // Arithmetic Functions (Add, Subt, Mult and Div).....
   |calc
      ?$valid
         @1
            // Arithmetic Function (Calculate ADD, SUB, MUL & DIV) in Cycle-1.....
            $Add_sq[31:0] = $val1_sq + $val2;
            $Sub_sq[31:0] = $val1_sq - $val2;
            $Mul_sq[31:0] = $val1_sq * $val2;
            $Div_sq[31:0] = $val1_sq / $val2;
            $val1_sq[31:0] = >>1$calc_out;

            // Free Runing Counter Mux (2x1) Operation ......
            $cnt[0:0] = ($reset == 1) ? $inp2:
                                        $sum_sq;

         @2
            // Mux (4x1) Operation in Cycle-2 .....
            $valid_sq[0:0] = $cnt;
            $validinv_sq[0:0] = ! ($valid_sq);
            $reset_sq[0:0] = ($validinv_sq || $reset);
            $calc_out[31:0] = ($op == 00 & $reset_sq == 0) ? $Add_sq:
                              ($op == 01 & $reset_sq == 0) ? $Sub_sq:
                              ($op == 10 & $reset_sq == 0) ? $Mul_sq:
                              ($op == 10 & $reset_sqt == 0) ? $Div_sq:
                              $reset_zero;

         //
   //...
   // Assert these to end simulation (before the cycle limit).
   *passed = *cyc_cnt > 40;
   *failed = 1'b0;
\SV
   endmodule
```