

気づいたら

8月 ...

Bluespec

Bluespec のライセンスをもらった(?) のでライセンス設定の確認がてら遊んでみた .

Bluespec デザインコンテストのときは ,

シミュレーションばかりで FPGA 上での動作を確認しなかったので ,

まずはカウンタを作って FPGA 上で動作させてみるどころから .

Bluespec 的には次のようなソースを書く .

```
interface Test_ifc;
  method Bit#(26) result();
endinterface

module mkTest(Test_ifc);
  Reg#(Bit#(26)) counter <- mkReg(0);

  rule r0;
    counter <= counter + 1;
  endrule

  method Bit#(26) result();
    return readReg(counter);
  endmethod

endmodule
```

で ,

```
bsc -verilog -g mkTest test.bsv
```

として Verilog コードを生成すると ,

```
module mkTest(CLK,
              RST_N,
              result,
              RDY_result);
  input  CLK;
  input  RST_N;

  // value method result
  output [25 : 0] result;
  output RDY_result;
  /* snip */
  always@(posedge CLK)
  begin
    if (!RST_N)
    begin
      counter <= `BSV_ASSIGNMENT_DELAY 26'd0;
    end
    else
    begin
      if (counter$EN) counter <= `BSV_ASSIGNMENT_DELAY counter$D_IN;
    end
  end
  /* snip */
endmodule // mkTest
```

という何とも素直なコードが .

手書きのコードと最終的な回路規模を比較するまでもありません .

トップモジュールを ,

```
`default_nettype none
module top(USER_CLOCK, CPU_RESET, GPIO_LED);
  input wire USER_CLOCK;
  input wire CPU_RESET;
  output wire [7 : 0] GPIO_LED;

  wire rst_x;
  wire [25:0] result;

  assign rst_x = CPU_RESET;
  assign GPIO_LED = result[25:18];

  mkTest U0(.CLK(USER_CLOCK), .RST_N(rst_x), .result(result), .RDY_result());
endmodule
```

と書いて合成 , 配置配線で bit ファイルができるので書き込めばよい .