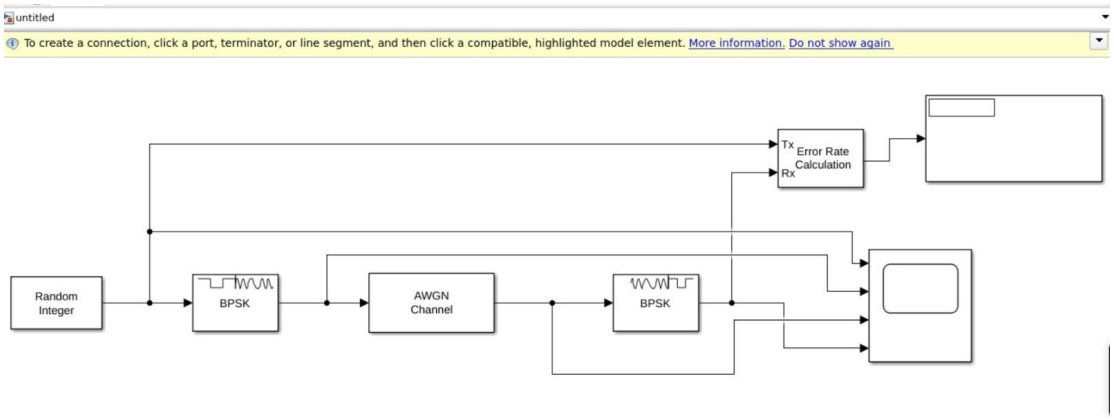
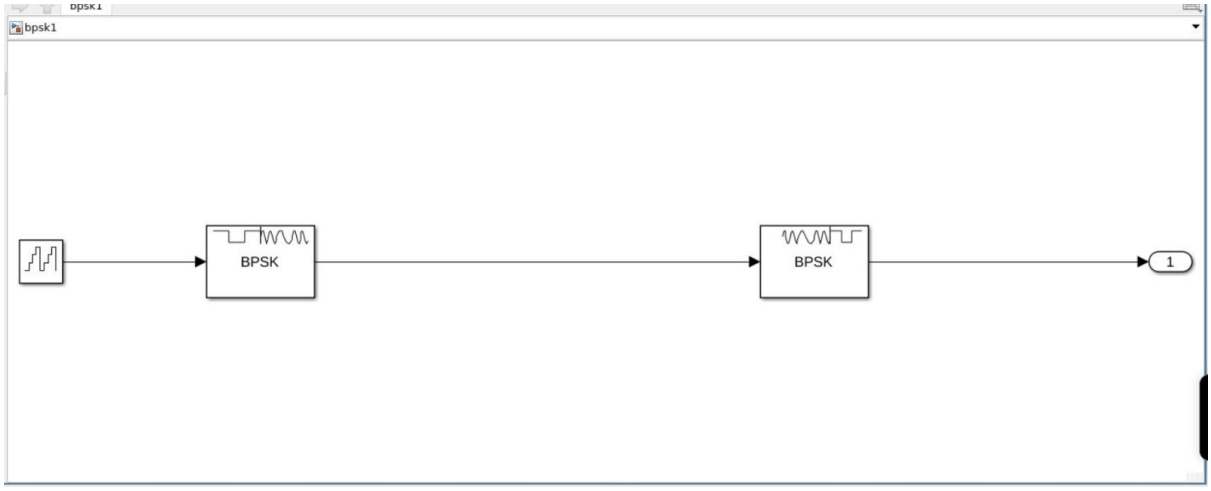
CONVERSION OF SIMULINK BLOCKS TO HDL VERILOG CODE

BPSK



INITIAL MODEL



FINAL MODEL

BPSK CODE:

```
// File Name: hdlsrc/bpsk1/bpsk1.v
// Created: 2023-07-17 17:35:43
//
// Generated by MATLAB 9.14 and HDL Coder 4.1
//
//
// -- -------------------------------------------------------------
// -- Rate and Clocking Details
// -- -------------------------------------------------------------
```

```
// Model base rate: 0.2
// Target subsystem base rate: 0.2
//
//
// Clock Enable  Sample Time
// -- ------------------------------------------------------------
// ce_out        0.2
// -- ------------------------------------------------------------
//
//
// Output Signal                 Clock Enable  Sample Time
// -- ------------------------------------------------------------
// Out1                          ce_out        0.2
// -- ------------------------------------------------------------
//
// -------------------------------------------------------------


// -------------------------------------------------------------
//
// Module: bpsk1
// Source Path: bpsk1
// Hierarchy Level: 0
//
// -------------------------------------------------------------

`timescale 1 ns / 1 ns

module bpsk1
          (clk,
           reset,
           clk_enable,
           ce_out,
           Out1);


  input   clk;
  input   reset;
  input   clk_enable;
  output  ce_out;
  output  [7:0] Out1;  // uint8


  wire enb;
```

```
  wire [15:0] count_step;  // uint16
  reg [15:0] Counter_Free_Running_out1;  // uint16
  wire [15:0] count;  // uint16
  wire signed [15:0] BPSK_Modulator_Baseband1_out1_re;  // sfix16_En15
  wire signed [15:0] BPSK_Modulator_Baseband1_out1_im;  // sfix16_En15
  wire [7:0] BPSK_Demodulator_Baseband1_out1;  // uint8


  assign count_step = 16'b0000000000000001;



  assign enb = clk_enable;

  assign count = Counter_Free_Running_out1 + count_step;



  // Free running, Unsigned Counter
  //  initial value   = 0
  //  step value      = 1
  always @(posedge clk or posedge reset)
    begin : Counter_Free_Running_process
      if (reset == 1'b1) begin
        Counter_Free_Running_out1 <= 16'b0000000000000000;
      end
      else begin
        if (enb) begin
          Counter_Free_Running_out1 <= count;
        end
      end
    end



  BPSK_Modulator_Baseband1 u_BPSK_Modulator_Baseband1
(.in0(Counter_Free_Running_out1),  // uint16
                                     .out0_re(BPSK_Modulator_Baseband1_out1_re),  //
sfix16_En15
                                     .out0_im(BPSK_Modulator_Baseband1_out1_im)  //
sfix16_En15
                                     );
```

```
  BPSK_Demodulator_Baseband1 u_BPSK_Demodulator_Baseband1
(.in0_re(BPSK_Modulator_Baseband1_out1_re),  // sfix16_En15
                                        .in0_im(BPSK_Modulator_Baseband1_out1_im),  //
sfix16_En15
                                        .out0(BPSK_Demodulator_Baseband1_out1)  // uint8
                                        );

  assign Out1 = BPSK_Demodulator_Baseband1_out1;

  assign ce_out = clk_enable;

endmodule  // bpsk1
```

BPSK Modulator baseband code:

```
// -------------------------------------------------------------
//
// File Name: hdlsrc/bpsk1/BPSK_Modulator_Baseband1.v
// Created: 2023-07-20 17:38:01
//
// Generated by MATLAB 9.14 and HDL Coder 4.1
//
// -------------------------------------------------------------


// -------------------------------------------------------------
//
// Module: BPSK_Modulator_Baseband1
// Source Path: bpsk1/BPSK Modulator Baseband1
// Hierarchy Level: 1
//
// -------------------------------------------------------------

`timescale 1 ns / 1 ns

module BPSK_Modulator_Baseband1
          (in0,
           out0_re,
           out0_im);
```

```
  input   [15:0] in0;  // uint16
  output  signed [15:0] out0_re;  // sfix16_En15
  output  signed [15:0] out0_im;  // sfix16_En15


  wire bpsk_sel;
  wire signed [15:0] inphase_val0;  // sfix16_En15
  wire signed [15:0] inphase_val1;  // sfix16_En15
  wire signed [15:0] inphase;  // sfix16_En15
  wire signed [15:0] quadrature;  // sfix16_En15


  assign bpsk_sel = in0[0];



  assign inphase_val0 = 16'sb0111111111111111;



  assign inphase_val1 = 16'sb1000000000000000;



  assign inphase = (bpsk_sel == 1'b0 ? inphase_val0 :
            inphase_val1);



  assign out0_re = inphase;

  assign quadrature = 16'sb0000000000000000;



  assign out0_im = quadrature;

endmodule  // BPSK_Modulator_Baseband1
```

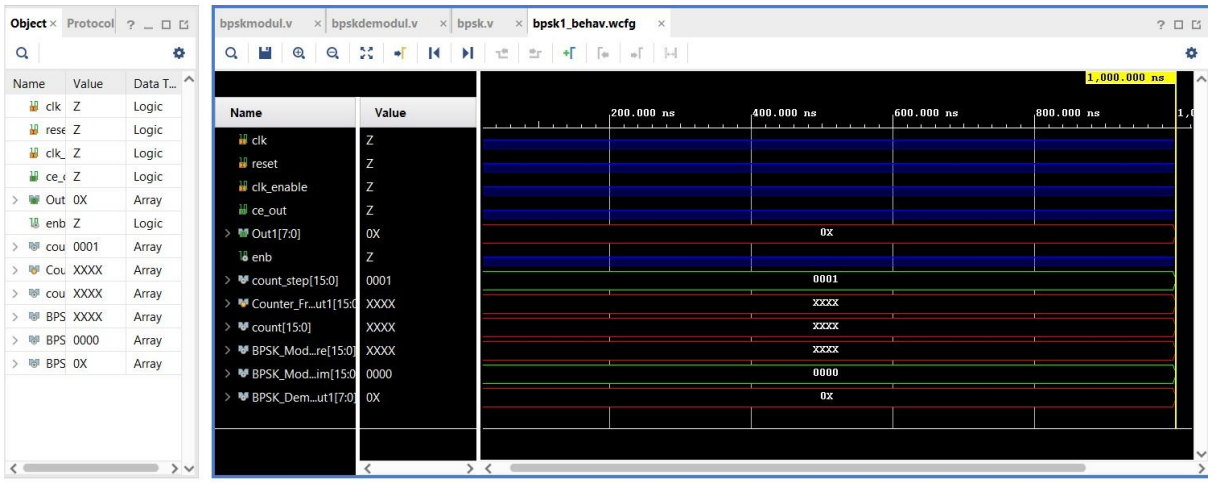
BPSK demodulator baseband code:

```
//
// File Name: hdlsrc/bpsk1/BPSK_Demodulator_Baseband1.v
```

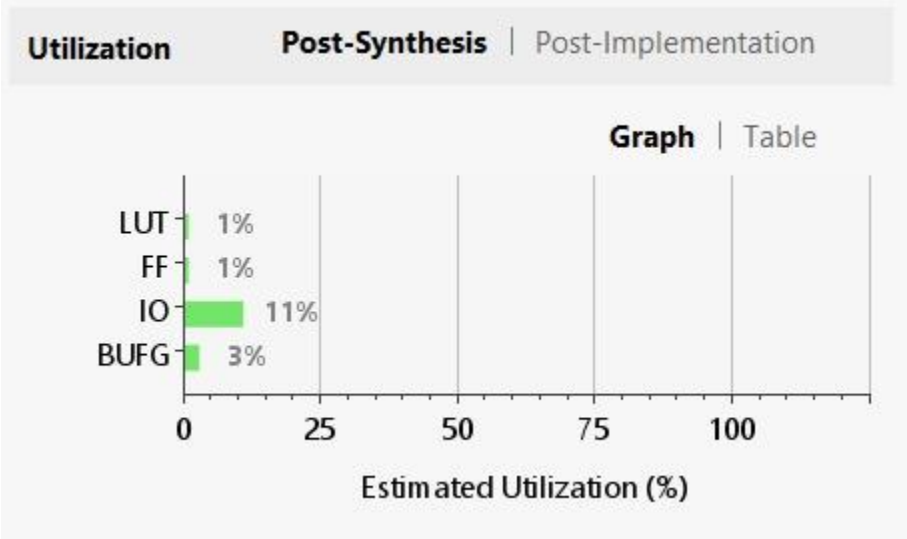
```
// Created: 2023-07-20 17:38:01
//
// Generated by MATLAB 9.14 and HDL Coder 4.1
//
// -----------------------------------------------------------


// -----------------------------------------------------------
//
// Module: BPSK_Demodulator_Baseband1
// Source Path: bpsk1/BPSK Demodulator Baseband1
// Hierarchy Level: 1
//
// -----------------------------------------------------------

`timescale 1 ns / 1 ns

module BPSK_Demodulator_Baseband1
          (in0_re,
           in0_im,
           out0);


  input   signed [15:0] in0_re;  // sfix16_En15
  input   signed [15:0] in0_im;  // sfix16_En15
  output  [7:0] out0;  // uint8


  wire inphase_lt_zero;
  wire inphase_eq_zero;
  wire quadrature_eq_zero;
  wire [2:0] decisionLUTaddr;  // ufix3
  wire [0:7] DirectLookupTable_1;  // ufix1 [8]
  wire hardDecision;  // ufix1


  assign inphase_lt_zero = in0_re < 16'sb0000000000000000;



  assign inphase_eq_zero = in0_re == 16'sb0000000000000000;
```

```
  assign quadrature_eq_zero = in0_im == 16'sb0000000000000000;


  assign decisionLUTaddr = {inphase_lt_zero, inphase_eq_zero, quadrature_eq_zero};


  assign DirectLookupTable_1[0] = 1'b0;
  assign DirectLookupTable_1[1] = 1'b0;
  assign DirectLookupTable_1[2] = 1'b1;
  assign DirectLookupTable_1[3] = 1'b0;
  assign DirectLookupTable_1[4] = 1'b1;
  assign DirectLookupTable_1[5] = 1'b1;
  assign DirectLookupTable_1[6] = 1'b1;
  assign DirectLookupTable_1[7] = 1'b0;
  assign hardDecision = DirectLookupTable_1[decisionLUTaddr];


  assign out0 = {7'b0, hardDecision};


endmodule  // BPSK_Demodulator_Baseband1
```

Verilog output :



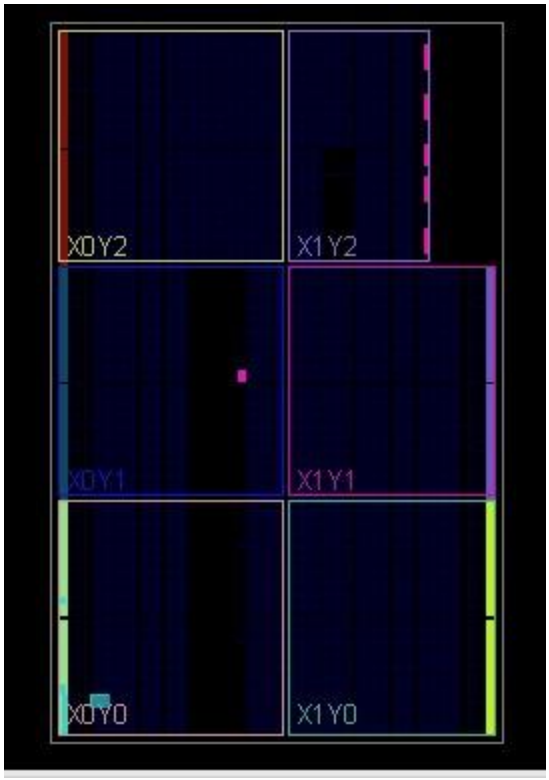
POWER UTILIZATION GRAPH AND TABLE :



Utilization **Post-Synthesis** | Post-Implementation

Graph | **Table**

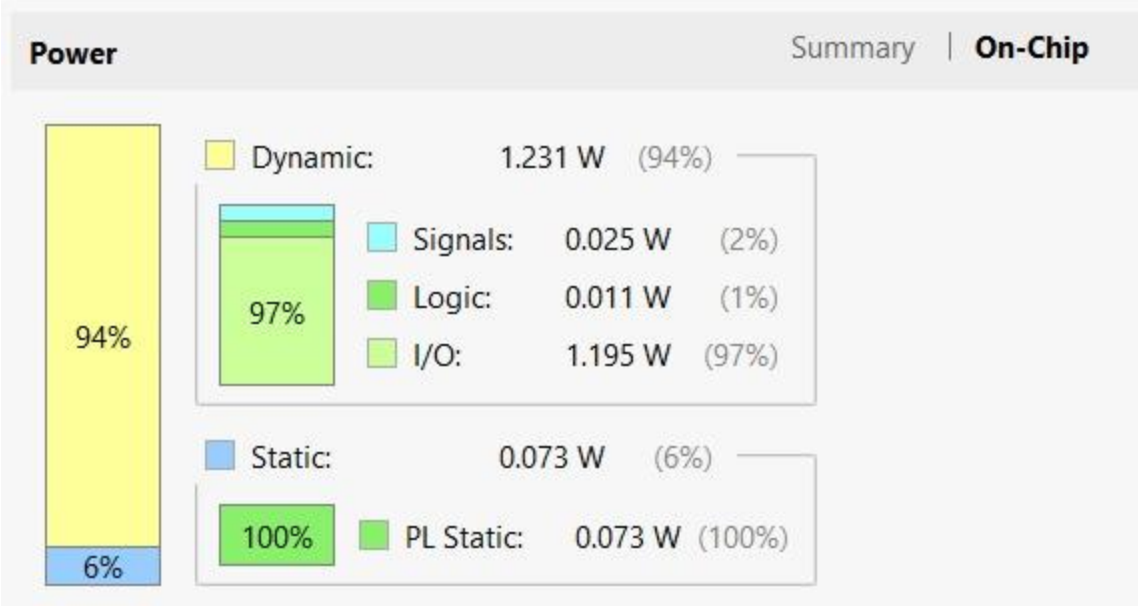
Resource	Estimation	Available	Utilization...
LUT	1	20800	0.01
FF	1	41600	0.01
IO	12	106	11.32
BUFG	1	32	3.13



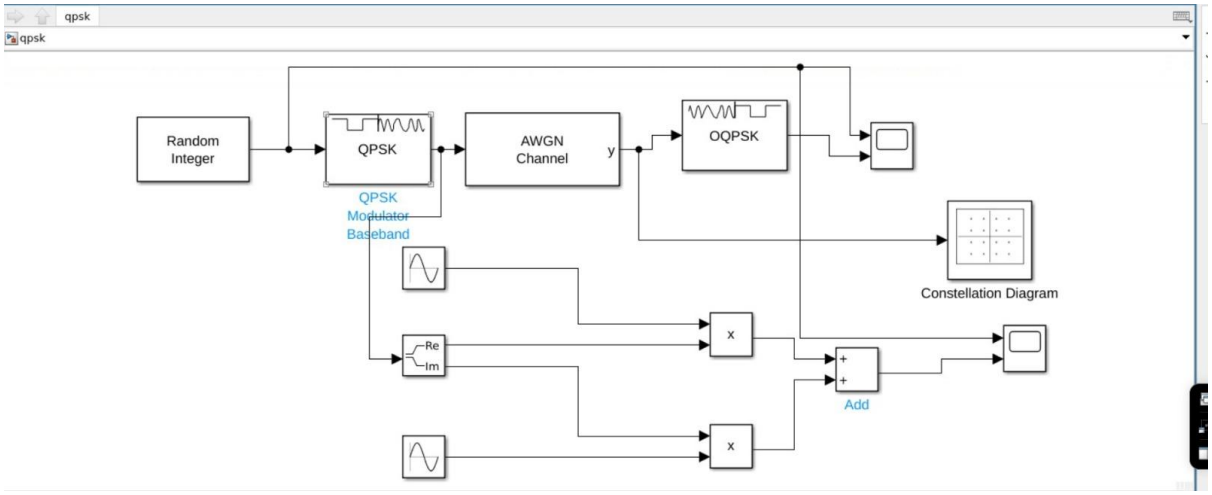
Power **Summary** | On-Chip

Total On-Chip Power:	**1.304 W**
Junction Temperature:	**31.5 °C**
Thermal Margin:	53.5 °C (10.6 W)
Effective ϑJA:	5.0 °C/W
Power supplied to off-chip devices:	0 W
Confidence level:	Low

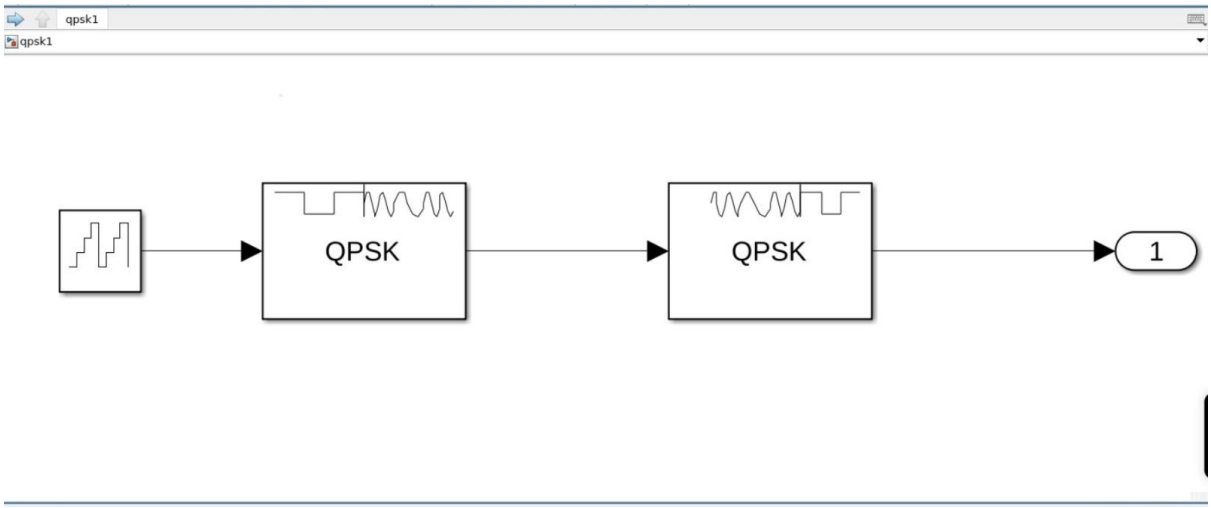
Implemented Power Report


Power
Summary | On-Chip
94%
6%
Dynamic: 1.231 W (94%)
97%
Signals: 0.025 W (2%)
Logic: 0.011 W (1%)
I/O: 1.195 W (97%)
Static: 0.073 W (6%)
100%
PL Static: 0.073 W (100%)

QPSK



INITIAL MODEL



FINAL MODEL

QPSK CODE :

```
// -------------------------------------------------------------
//
// File Name: hdlsrc/qpsk1/qpsk1.v
// Created: 2023-07-17 17:49:33
//
// Generated by MATLAB 9.14 and HDL Coder 4.1
//
//
// -- -------------------------------------------------------------
```

```
// -- Rate and Clocking Details
// -- -------------------------------------------------------
// Model base rate: 0.2
// Target subsystem base rate: 0.2
//
//
// Clock Enable  Sample Time
// -- -------------------------------------------------------
// ce_out        0.2
// -- -------------------------------------------------------
//
//
// Output Signal                 Clock Enable  Sample Time
// -- -------------------------------------------------------
// Out1                          ce_out        0.2
// -- -------------------------------------------------------
//
// -------------------------------------------------------------


// -------------------------------------------------------------
//
// Module: qpsk1
// Source Path: qpsk1
// Hierarchy Level: 0
//
// -------------------------------------------------------------

`timescale 1 ns / 1 ns

module qpsk1
          (clk,
           reset,
           clk_enable,
           ce_out,
           Out1);


  input   clk;
  input   reset;
  input   clk_enable;
  output  ce_out;
  output  [7:0] Out1;  // uint8
```

```
  wire enb;
  wire [15:0] count_step;  // uint16
  reg [15:0] Counter_Free_Running_out1;  // uint16
  wire [15:0] count;  // uint16
  wire signed [15:0] QPSK_Modulator_Baseband_out1_re;  // sfix16_En15
  wire signed [15:0] QPSK_Modulator_Baseband_out1_im;  // sfix16_En15
  wire [7:0] QPSK_Demodulator_Baseband_out1;  // uint8


  assign count_step = 16'b0000000000000001;



  assign enb = clk_enable;

  assign count = Counter_Free_Running_out1 + count_step;



  // Free running, Unsigned Counter
  //  initial value   = 0
  //  step value      = 1
  always @(posedge clk or posedge reset)
    begin : Counter_Free_Running_process
      if (reset == 1'b1) begin
        Counter_Free_Running_out1 <= 16'b0000000000000000;
      end
      else begin
        if (enb) begin
          Counter_Free_Running_out1 <= count;
        end
      end
    end



  QPSK_Modulator_Baseband u_QPSK_Modulator_Baseband
(.in0(Counter_Free_Running_out1),  // uint16
                                    .out0_re(QPSK_Modulator_Baseband_out1_re),  //
sfix16_En15
                                    .out0_im(QPSK_Modulator_Baseband_out1_im)  //
sfix16_En15
                                    );
```

```
  QPSK_Demodulator_Baseband u_QPSK_Demodulator_Baseband
(.in0_re(QPSK_Modulator_Baseband_out1_re),  // sfix16_En15
                                        .in0_im(QPSK_Modulator_Baseband_out1_im),  //
sfix16_En15
                                        .out0(QPSK_Demodulator_Baseband_out1)  // uint8
                                        );

  assign Out1 = QPSK_Demodulator_Baseband_out1;

  assign ce_out = clk_enable;

endmodule  // qpsk1
```

QPSK Modulator baseband code:

```
// -------------------------------------------------------------
//
// File Name: hdlsrc/qpsk1/QPSK_Modulator_Baseband.v
// Created: 2023-07-20 18:13:57
//
// Generated by MATLAB 9.14 and HDL Coder 4.1
//
// -------------------------------------------------------------


// -------------------------------------------------------------
//
// Module: QPSK_Modulator_Baseband
// Source Path: qpsk1/QPSK Modulator Baseband
// Hierarchy Level: 1
//
// -------------------------------------------------------------

`timescale 1 ns / 1 ns

module QPSK_Modulator_Baseband
          (in0,
           out0_re,
           out0_im);


  input   [15:0] in0;  // uint16
```

```
  output  signed [15:0] out0_re;  // sfix16_En15
  output  signed [15:0] out0_im;  // sfix16_En15


  wire [1:0] constellationLUTaddress;  // ufix2
  wire signed [15:0] constellationLUT_t1_re [0:3];  // sfix16_En15 [4]
  wire signed [15:0] constellationLUT_t1_im [0:3];  // sfix16_En15 [4]


  assign constellationLUTaddress = in0[1:0];



  assign constellationLUT_t1_re[0] = 16'sb0101101010000010;
  assign constellationLUT_t1_re[1] = 16'sb1010010101111110;
  assign constellationLUT_t1_re[2] = 16'sb0101101010000010;
  assign constellationLUT_t1_re[3] = 16'sb1010010101111110;
  assign constellationLUT_t1_im[0] = 16'sb0101101010000010;
  assign constellationLUT_t1_im[1] = 16'sb0101101010000010;
  assign constellationLUT_t1_im[2] = 16'sb1010010101111110;
  assign constellationLUT_t1_im[3] = 16'sb1010010101111110;
  assign out0_re = constellationLUT_t1_re[constellationLUTaddress];
  assign out0_im = constellationLUT_t1_im[constellationLUTaddress];



endmodule  // QPSK_Modulator_Baseband
```

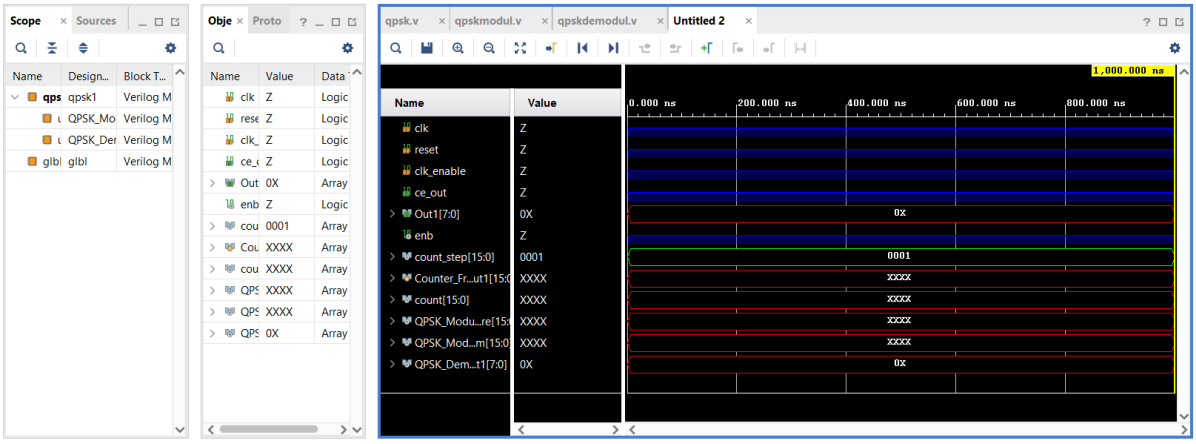
QPSK Demodulator baseband code:

```
// -------------------------------------------------------------
//
// File Name: hdlsrc/qpsk1/QPSK_Demodulator_Baseband.v
// Created: 2023-07-20 18:13:57
//
// Generated by MATLAB 9.14 and HDL Coder 4.1
//
// -------------------------------------------------------------
```

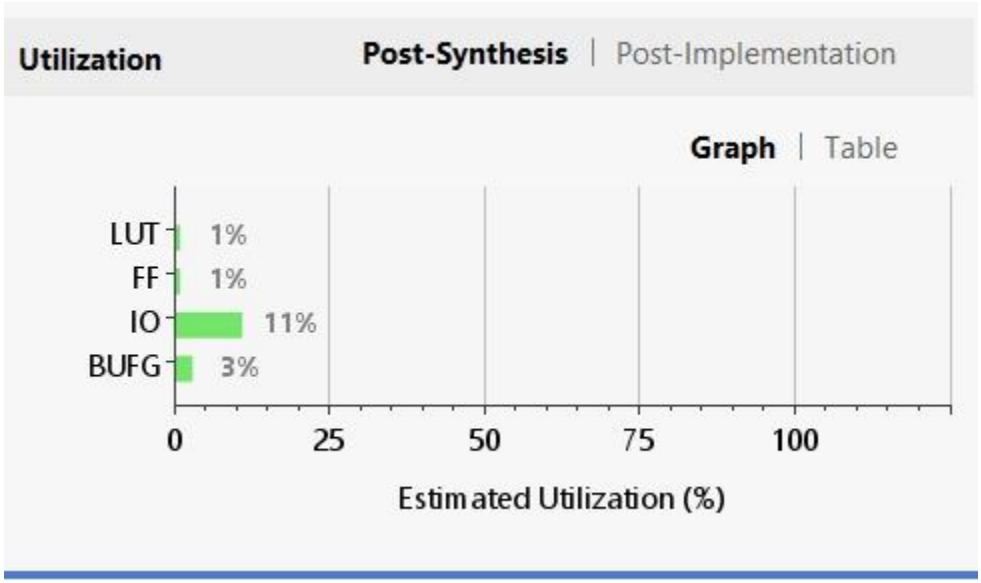
```
// -------------------------------------------------------------
//
// Module: QPSK_Demodulator_Baseband
// Source Path: qpsk1/QPSK Demodulator Baseband
// Hierarchy Level: 1
//
// -------------------------------------------------------------

`timescale 1 ns / 1 ns

module QPSK_Demodulator_Baseband
          (in0_re,
           in0_im,
           out0);


  input   signed [15:0] in0_re;  // sfix16_En15
  input   signed [15:0] in0_im;  // sfix16_En15
  output  [7:0] out0;  // uint8


  wire inphase_lt_zero;
  wire inphase_eq_zero;
  wire quadrature_lt_zero;
  wire quadrature_eq_zero;
  wire [3:0] decisionLUTaddr;  // ufix4
  wire [1:0] DirectLookupTable_1 [0:15];  // ufix2 [16]
  wire [1:0] hardDecision;  // ufix2


  assign inphase_lt_zero = in0_re < 16'sb0000000000000000;



  assign inphase_eq_zero = in0_re == 16'sb0000000000000000;



  assign quadrature_lt_zero = in0_im < 16'sb0000000000000000;



  assign quadrature_eq_zero = in0_im == 16'sb0000000000000000;
```

```
  assign decisionLUTaddr = {inphase_lt_zero, inphase_eq_zero, quadrature_lt_zero,
quadrature_eq_zero};



  assign DirectLookupTable_1[0] = 2'b00;
  assign DirectLookupTable_1[1] = 2'b00;
  assign DirectLookupTable_1[2] = 2'b10;
  assign DirectLookupTable_1[3] = 2'b00;
  assign DirectLookupTable_1[4] = 2'b01;
  assign DirectLookupTable_1[5] = 2'b00;
  assign DirectLookupTable_1[6] = 2'b10;
  assign DirectLookupTable_1[7] = 2'b00;
  assign DirectLookupTable_1[8] = 2'b01;
  assign DirectLookupTable_1[9] = 2'b11;
  assign DirectLookupTable_1[10] = 2'b11;
  assign DirectLookupTable_1[11] = 2'b00;
  assign DirectLookupTable_1[12] = 2'b00;
  assign DirectLookupTable_1[13] = 2'b00;
  assign DirectLookupTable_1[14] = 2'b00;
  assign DirectLookupTable_1[15] = 2'b00;
  assign hardDecision = DirectLookupTable_1[decisionLUTaddr];



  assign out0 = {6'b0, hardDecision};



endmodule  // QPSK_Demodulator_Baseband
```

Verilog output :



POWER UTILIZATION GRAPH AND TABLE :



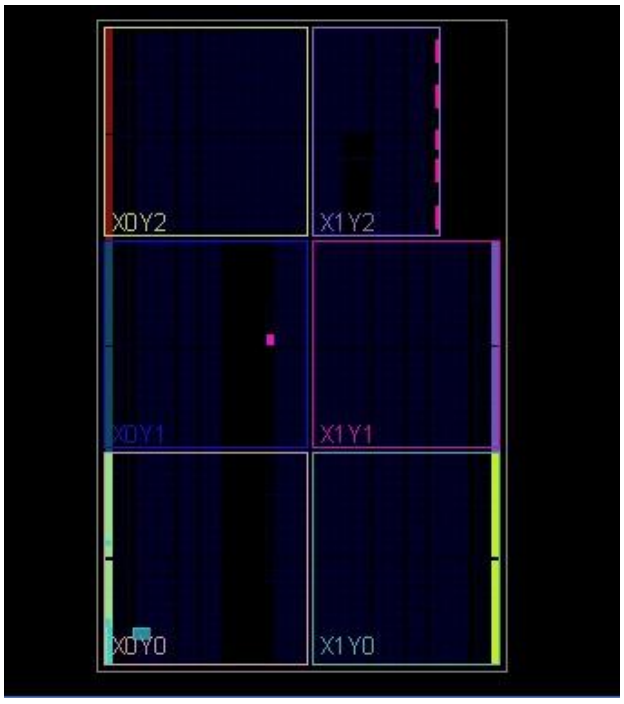
Utilization **Post-Synthesis** | Post-Implementation

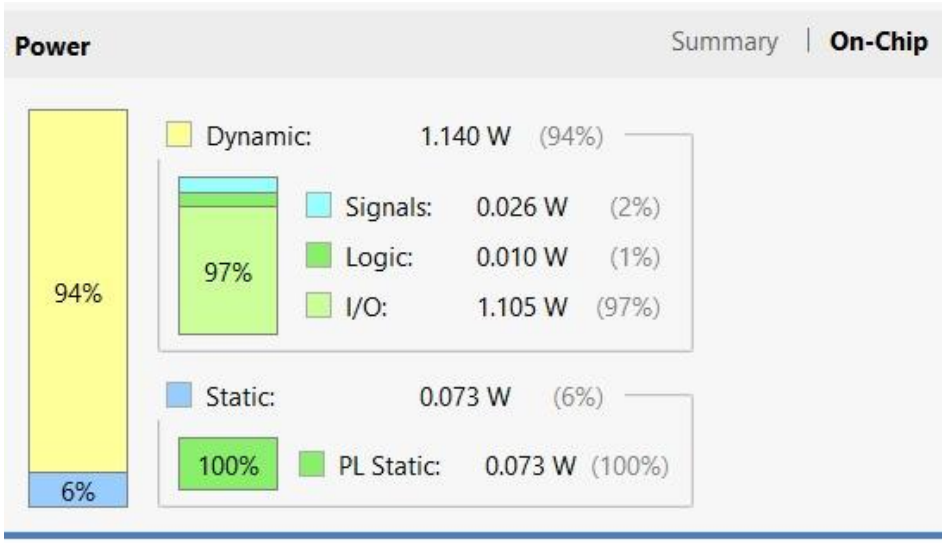
Graph | **Table**

Resource	Estimation	Available	Utilization %
LUT	1	20800	0.01
FF	2	41600	0.01
IO	12	106	11.32
BUFG	1	32	3.13

Power **Summary** | On-Chip

Total On-Chip Power:	**1.213 W**
Junction Temperature:	**31.1 °C**
Thermal Margin:	53.9 °C (10.7 W)
Effective ϑJA:	5.0 °C/W




Power
Summary | On-Chip
94%
6%
Dynamic: 1.140 W (94%)
97%
Signals: 0.026 W (2%)
Logic: 0.010 W (1%)
I/O: 1.105 W (97%)
Static: 0.073 W (6%)
100%
PL Static: 0.073 W (100%)