Vivado High-Level Synthesis (HLS):

$$\sqrt{a^2 + b^2}$$

Daniele Bagni
DSP Specialist

Rationale

- Vivado HLS has a set of powerful directives to generate optimal RTL from C/C++ models
- Assume: all the right directives have been already applied
- Still space for improvements?
  - Reduce the resource occupation
  - Improve the throughput
  - Close timing

YES !

Improve C code.

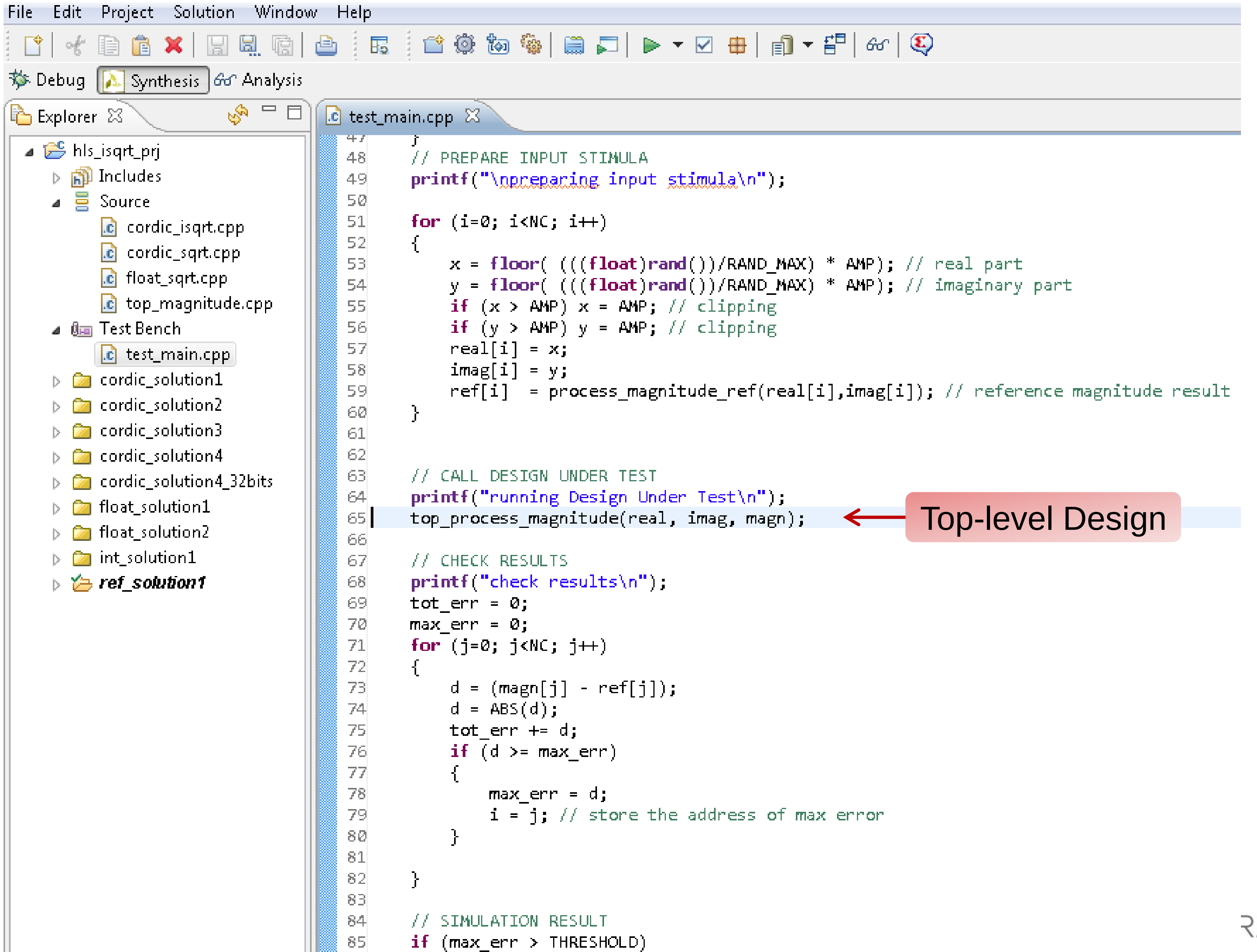
Let us start

Design Under Test

$$\sqrt{a^2 + b^2}$$

- **Design goals: small and fast (and possibly accurate)**
  - I/O signals are integer numbers
  - The real design contains 4 cores like this

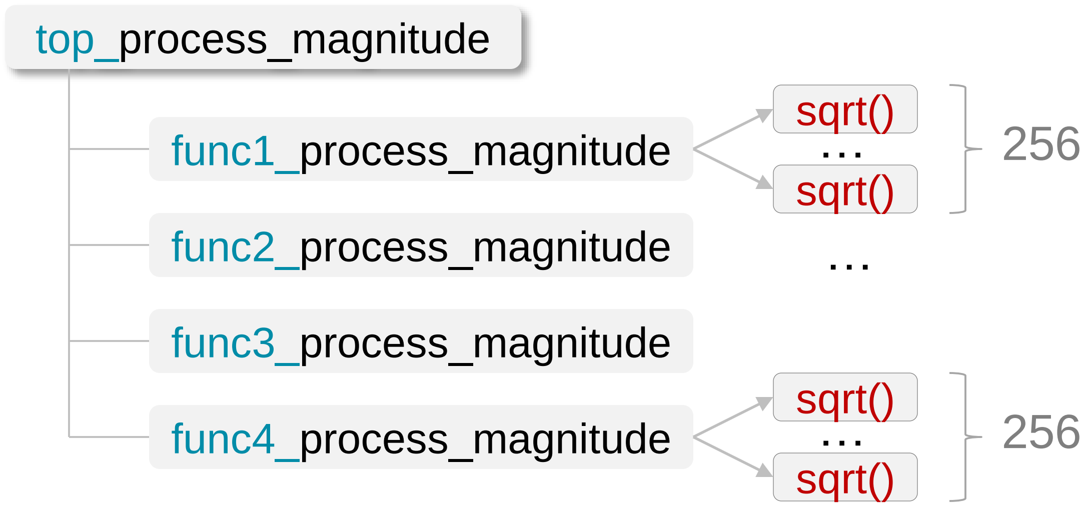
HLS Functional Test Bench



```
    }
    // PREPARE INPUT STIMULA
    printf("\npreparing input stimula\n");

    for (i=0; i<NC; i++)
    {
        x = floor( (((float)rand())/RAND_MAX) * AMP); // real part
        y = floor( (((float)rand())/RAND_MAX) * AMP); // imaginary part
        if (x > AMP) x = AMP; // clipping
        if (y > AMP) y = AMP; // clipping
        real[i] = x;
        imag[i] = y;
        ref[i]  = process_magnitude_ref(real[i],imag[i]); // reference magnitude result
    }


    // CALL DESIGN UNDER TEST
    printf("running Design Under Test\n");
    top_process_magnitude(real, imag, magn);

    // CHECK RESULTS
    printf("check results\n");
    tot_err = 0;
    max_err = 0;
    for (j=0; j<NC; j++)
    {
        d = (magn[j] - ref[j]);
        d = ABS(d);
        tot_err += d;
        if (d >= max_err)
        {
            max_err = d;
            i = j; // store the address of max error
        }

    }

    // SIMULATION RESULT
    if (max_err > THRESHOLD)
```

Top level design to be synthesized 1/2

- **Assumptions:**
  - Top-level design contains 4 functions,
  - Each one contains a `sqrt()` operation called 256 times within a loop.

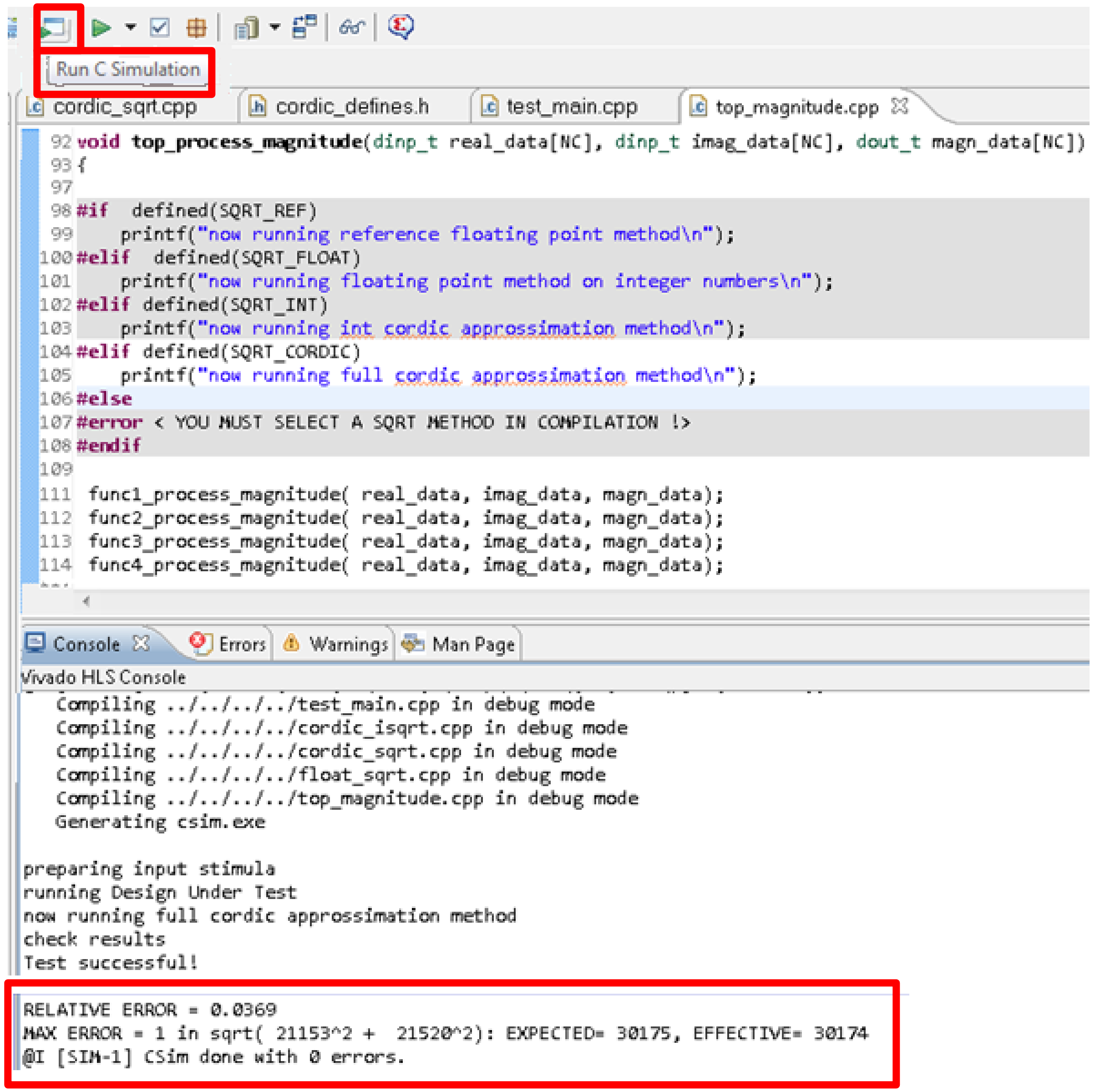


Top level design to be synthesized 2/2

- **Assumptions:**
  - Top-level design contains 4 functions,
  - Each one contains a `sqrt()` operation called 256 times within a loop.

```
void top_process_magnitude(int real_data[NC],
                           int imag_data[NC],
                           int magn_data[NC])
{

   func1_process_magnitude( real_data, imag_data, magn_data);
   func2_process_magnitude( real_data, imag_data, magn_data);
   func3_process_magnitude( real_data, imag_data, magn_data);
   func4_process_magnitude( real_data, imag_data, magn_data);

}
```

Functional verification



```
void top_process_magnitude(dinp_t real_data[NC], dinp_t imag_data[NC], dout_t magn_data[NC])
{

#if  defined(SQRT_REF)
    printf("now running reference floating point method\n");
#elif  defined(SQRT_FLOAT)
    printf("now running floating point method on integer numbers\n");
#elif defined(SQRT_INT)
    printf("now running int cordic approssimation method\n");
#elif defined(SQRT_CORDIC)
    printf("now running full cordic approssimation method\n");
#else
#error < YOU MUST SELECT A SQRT METHOD IN COMPILATION !>
#endif

 func1_process_magnitude( real_data, imag_data, magn_data);
 func2_process_magnitude( real_data, imag_data, magn_data);
 func3_process_magnitude( real_data, imag_data, magn_data);
 func4_process_magnitude( real_data, imag_data, magn_data);
```

```
   Compiling ../../../../test_main.cpp in debug mode
   Compiling ../../../../cordic_isqrt.cpp in debug mode
   Compiling ../../../../cordic_sqrt.cpp in debug mode
   Compiling ../../../../float_sqrt.cpp in debug mode
   Compiling ../../../../top_magnitude.cpp in debug mode
   Generating csim.exe

preparing input stimula
running Design Under Test
now running full cordic approssimation method
check results
Test successful!

RELATIVE ERROR = 0.0369
MAX ERROR = 1 in sqrt( 21153^2 +  21520^2): EXPECTED= 30175, EFFECTIVE= 30174
@I [SIM-1] CSim done with 0 errors.
```

- **C simulation**
  - Validate the accuracy of numerical results with bit-true C/C++ modelling
- **Fixed and Floating point results can be different:**
  - Depends on the algorithm

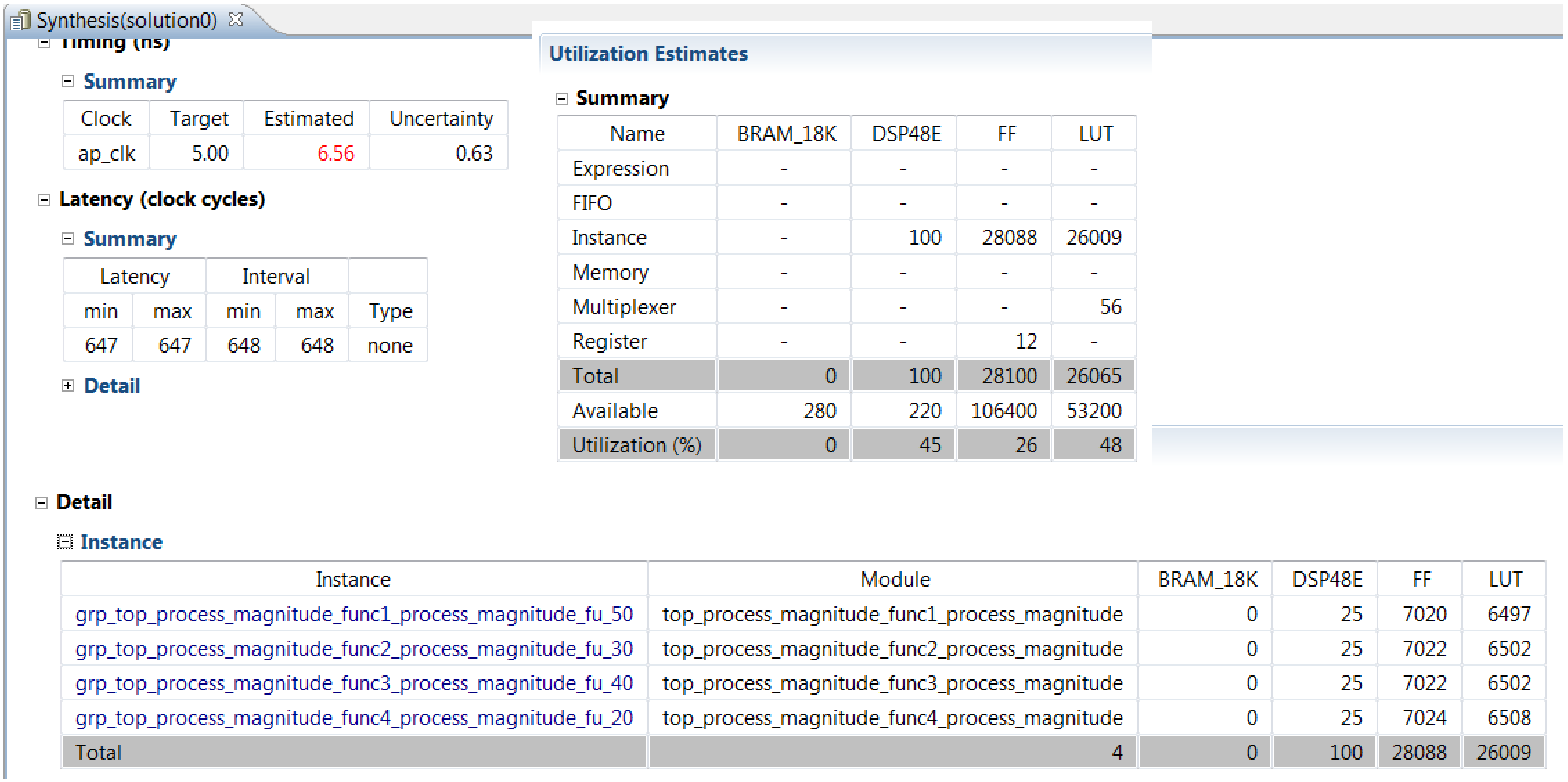
Solution 0:
everything in 64-bit Floating Point

C code of Solution 0:

Using 64-bit floating point types

```
#include <math.h>
double process_magnitude_double(double real_data,
                                double imag_data)
{
  #pragma HLS INLINE OFF
  #pragma HLS PIPELINE

  double mag_data, accu_plus, temp_datar, temp_datai;
  temp_datar = real_data * real_data;
  temp_datai = imag_data * imag_data;
  accu_plus = temp_datar + temp_datai;

  mag_data = sqrt(accu_plus);
  return mag_data;
}
```

Solution 0: Synthesis Estimation, top level



Synthesis(solution0)

Timing (ns)

Summary

Clock	Target	Estimated	Uncertainty
ap_clk	5.00	6.56	0.63

Latency (clock cycles)

Summary

Latency		Interval		
min	max	min	max	Type
647	647	648	648	none

Detail

Utilization Estimates

Summary

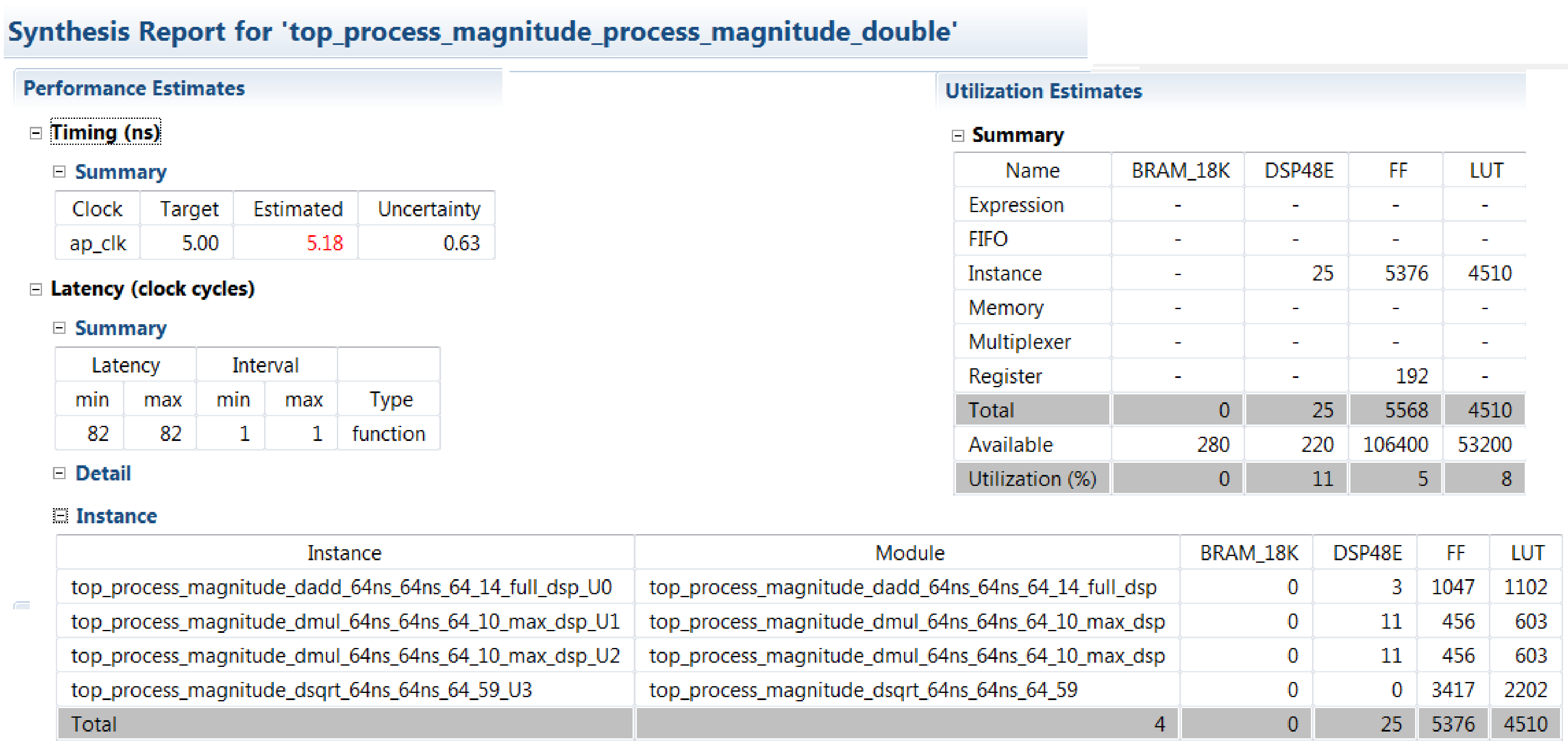
Name	BRAM_18K	DSP48E	FF	LUT
Expression	-	-	-	-
FIFO	-	-	-	-
Instance	-	100	28088	26009
Memory	-	-	-	-
Multiplexer	-	-	-	56
Register	-	-	12	-
Total	0	100	28100	26065
Available	280	220	106400	53200
Utilization (%)	0	45	26	48

Detail

Instance

Instance	Module	BRAM_18K	DSP48E	FF	LUT
grp_top_process_magnitude_func1_process_magnitude_fu_50	top_process_magnitude_func1_process_magnitude	0	25	7020	6497
grp_top_process_magnitude_func2_process_magnitude_fu_30	top_process_magnitude_func2_process_magnitude	0	25	7022	6502
grp_top_process_magnitude_func3_process_magnitude_fu_40	top_process_magnitude_func3_process_magnitude	0	25	7022	6502
grp_top_process_magnitude_func4_process_magnitude_fu_20	top_process_magnitude_func4_process_magnitude	0	25	7024	6508
Total	4	0	100	28088	26009

Solution 0: Synthesis Estimation, inner function



Synthesis Report for 'top_process_magnitude_process_magnitude_double'

Performance Estimates

⊟ **Timing (ns)**

⊟ **Summary**

Clock	Target	Estimated	Uncertainty
ap_clk	5.00	5.18	0.63

⊟ **Latency (clock cycles)**

⊟ **Summary**

Latency		Interval		
min	max	min	max	Type
82	82	1	1	function

⊟ **Detail**

⊟ **Instance**

Instance	Module	BRAM_18K	DSP48E	FF	LUT
top_process_magnitude_dadd_64ns_64ns_64_14_full_dsp_U0	top_process_magnitude_dadd_64ns_64ns_64_14_full_dsp	0	3	1047	1102
top_process_magnitude_dmul_64ns_64ns_64_10_max_dsp_U1	top_process_magnitude_dmul_64ns_64ns_64_10_max_dsp	0	11	456	603
top_process_magnitude_dmul_64ns_64ns_64_10_max_dsp_U2	top_process_magnitude_dmul_64ns_64ns_64_10_max_dsp	0	11	456	603
top_process_magnitude_dsqrt_64ns_64ns_64_59_U3	top_process_magnitude_dsqrt_64ns_64ns_64_59	0	0	3417	2202
Total	4	0	25	5376	4510

Utilization Estimates

⊟ **Summary**

Name	BRAM_18K	DSP48E	FF	LUT
Expression	-	-	-	-
FIFO	-	-	-	-
Instance	-	25	5376	4510
Memory	-	-	-	-
Multiplexer	-	-	-	-
Register	-	-	192	-
Total	0	25	5568	4510
Available	280	220	106400	53200
Utilization (%)	0	11	5	8

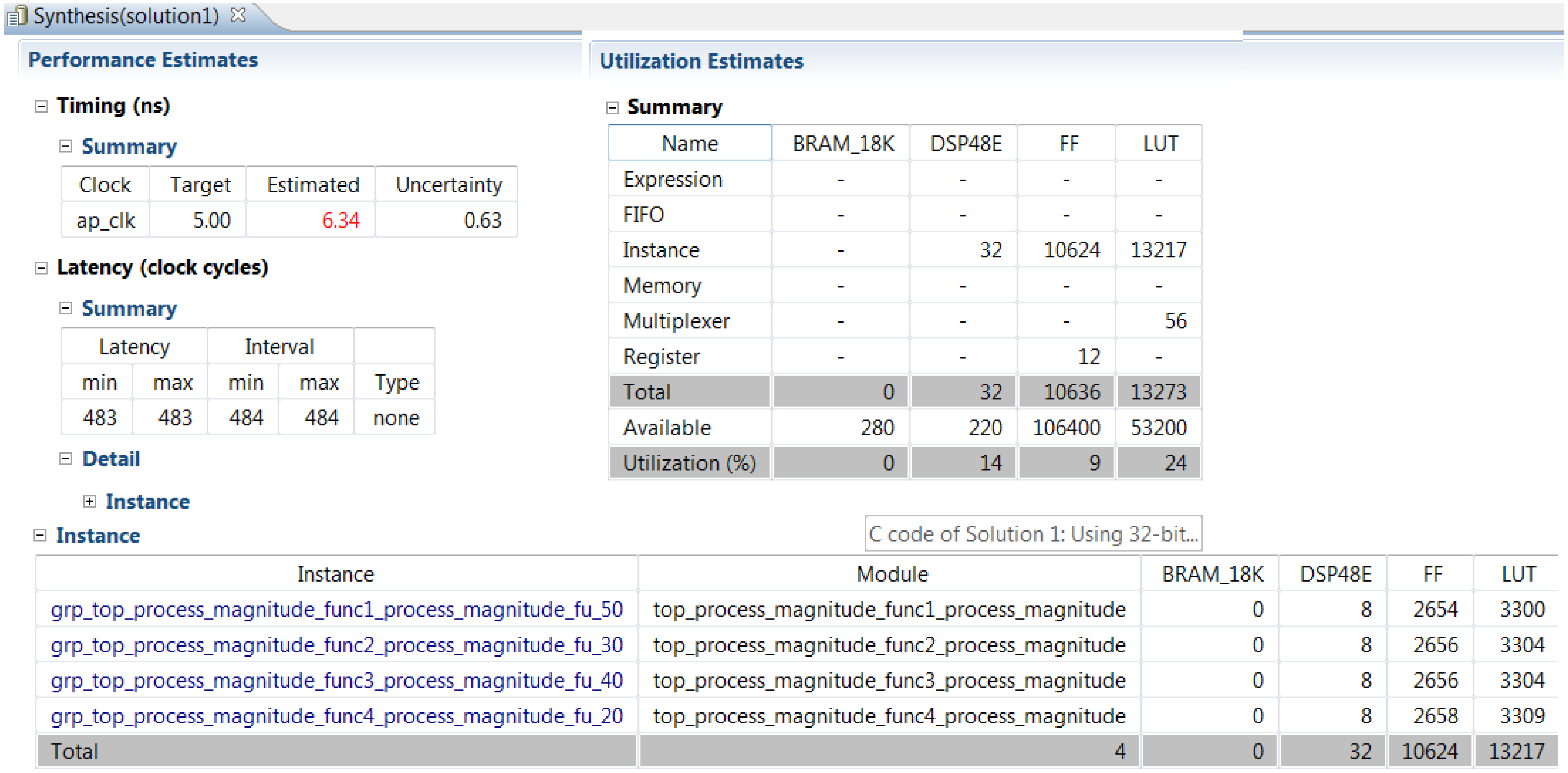
First method:
everything in 32-bit Floating Point

C code of Solution 1:

Using 32-bit floating point types

```
#include <math.h>
float process_magnitude_ref(float real_data, float imag_data) {
  #pragma HLS INLINE OFF
  #pragma HLS PIPELINE

  float mag_data, accu_plus, temp_datar, temp_datai;

  temp_datar = real_data * real_data;
  temp_datai = imag_data * imag_data;
  accu_plus = temp_datar + temp_datai;

  mag_data = sqrtf(accu_plus);

  return mag_data;
}
```

Solution 1: Synthesis Estimation, top level



Synthesis(solution1)

Performance Estimates

Timing (ns)

Summary

Clock	Target	Estimated	Uncertainty
ap_clk	5.00	6.34	0.63

Latency (clock cycles)

Summary

Latency		Interval		
min	max	min	max	Type
483	483	484	484	none

Detail

Instance

Utilization Estimates

Summary

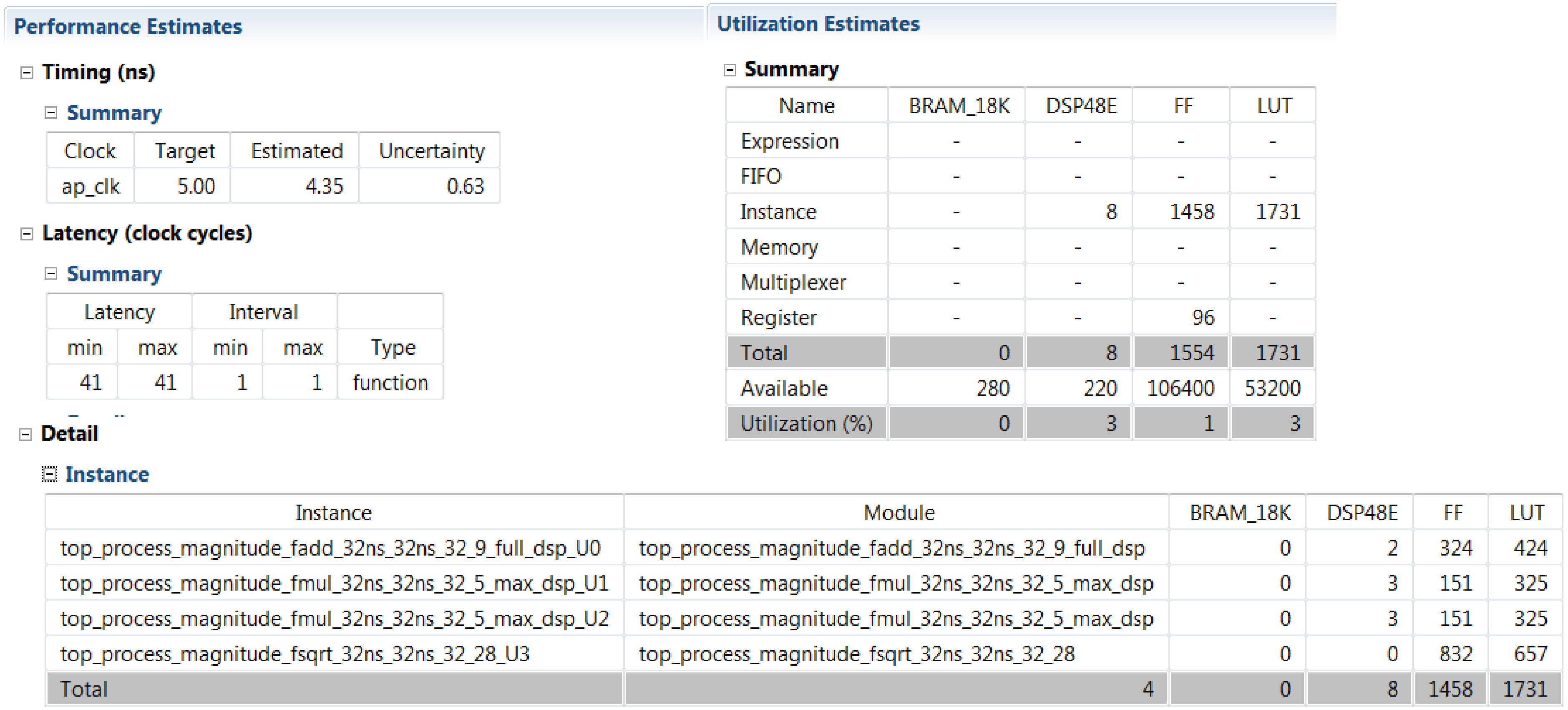
Name	BRAM_18K	DSP48E	FF	LUT
Expression	-	-	-	-
FIFO	-	-	-	-
Instance	-	32	10624	13217
Memory	-	-	-	-
Multiplexer	-	-	-	56
Register	-	-	12	-
Total	0	32	10636	13273
Available	280	220	106400	53200
Utilization (%)	0	14	9	24

C code of Solution 1: Using 32-bit...

Instance

Instance	Module	BRAM_18K	DSP48E	FF	LUT
grp_top_process_magnitude_func1_process_magnitude_fu_50	top_process_magnitude_func1_process_magnitude	0	8	2654	3300
grp_top_process_magnitude_func2_process_magnitude_fu_30	top_process_magnitude_func2_process_magnitude	0	8	2656	3304
grp_top_process_magnitude_func3_process_magnitude_fu_40	top_process_magnitude_func3_process_magnitude	0	8	2656	3304
grp_top_process_magnitude_func4_process_magnitude_fu_20	top_process_magnitude_func4_process_magnitude	0	8	2658	3309
Total	4	0	32	10624	13217

Solution 1: Synthesis Estimation, inner function



Performance Estimates

Timing (ns)

Summary

Clock	Target	Estimated	Uncertainty
ap_clk	5.00	4.35	0.63

Latency (clock cycles)

Summary

Latency		Interval		
min	max	min	max	Type
41	41	1	1	function

Detail

Instance

Instance	Module	BRAM_18K	DSP48E	FF	LUT
top_process_magnitude_fadd_32ns_32ns_32_9_full_dsp_U0	top_process_magnitude_fadd_32ns_32ns_32_9_full_dsp	0	2	324	424
top_process_magnitude_fmul_32ns_32ns_32_5_max_dsp_U1	top_process_magnitude_fmul_32ns_32ns_32_5_max_dsp	0	3	151	325
top_process_magnitude_fmul_32ns_32ns_32_5_max_dsp_U2	top_process_magnitude_fmul_32ns_32ns_32_5_max_dsp	0	3	151	325
top_process_magnitude_fsqrt_32ns_32ns_32_28_U3	top_process_magnitude_fsqrt_32ns_32ns_32_28	0	0	832	657
Total	4	0	8	1458	1731

Utilization Estimates

Summary

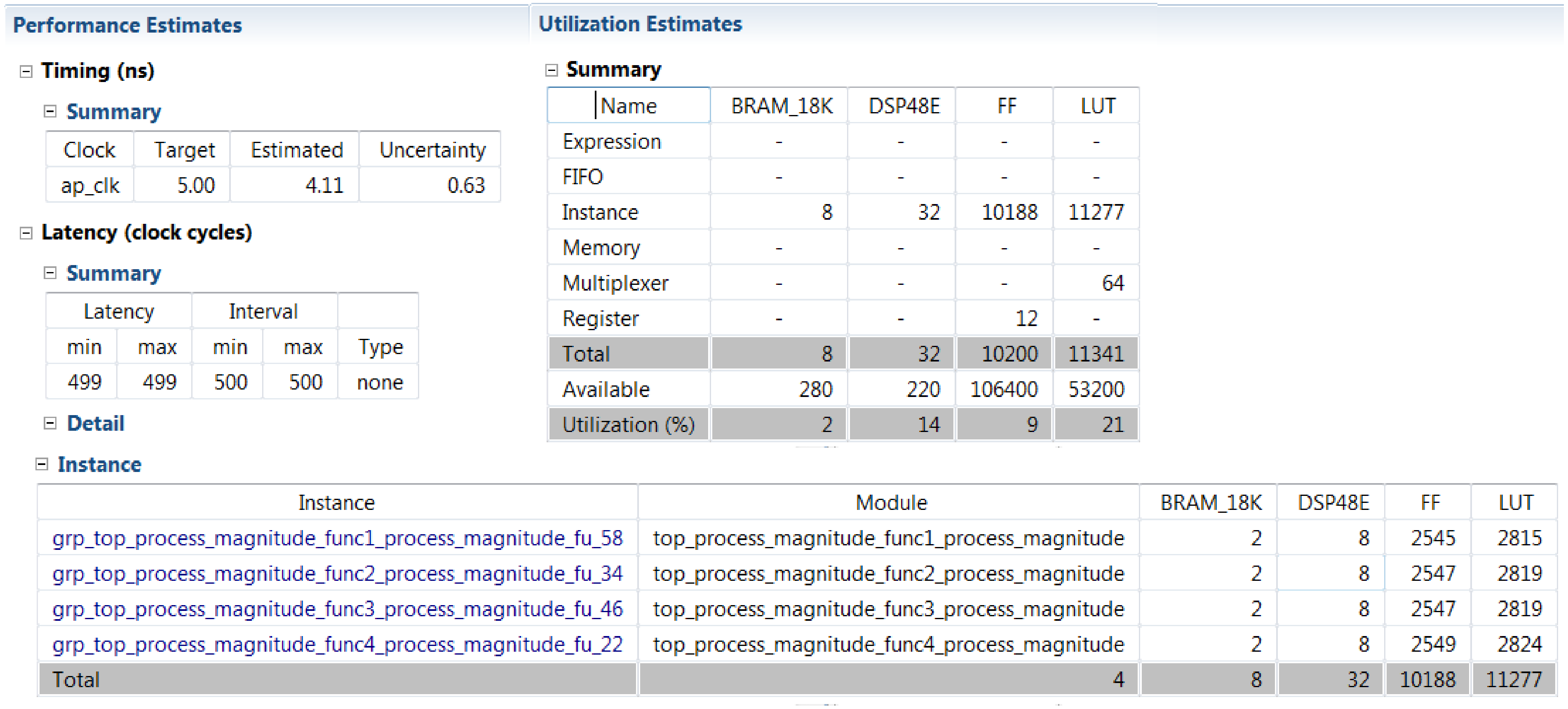
Name	BRAM_18K	DSP48E	FF	LUT
Expression	-	-	-	-
FIFO	-	-	-	-
Instance	-	8	1458	1731
Memory	-	-	-	-
Multiplexer	-	-	-	-
Register	-	-	96	-
Total	0	8	1554	1731
Available	280	220	106400	53200
Utilization (%)	0	3	1	3

Second method:
mixing 32-bit Fixed and Floating Point

C code of Solution 2:
Using 32-bit integer & floating point types

```
#include <math.h>
typedef long long int acc_t; // 64-bit data type
int process_magnitude_float(int real_data, int imag_data) {
#pragma HLS INLINE OFF
#pragma HLS PIPELINE

  int mag_data; // 32-bit data type
  acc_t accu_plus, temp_datar, temp_datai;

  // 32x32 to 64-bit integer multiplications
  temp_datar = (acc_t)real_data * (acc_t)real_data;
  temp_datai = (acc_t)imag_data * (acc_t)imag_data;
  accu_plus = temp_datar + temp_datai;

  mag_data = (int) floor(sqrtf( (float)accu_plus ));
  return mag_data;
}
```

Solution 2: Synthesis Estimation, top level



Performance Estimates

⊟ **Timing (ns)**

⊟ **Summary**

Clock	Target	Estimated	Uncertainty
ap_clk	5.00	4.11	0.63

⊟ **Latency (clock cycles)**

⊟ **Summary**

Latency		Interval		
min	max	min	max	Type
499	499	500	500	none

⊟ **Detail**

Utilization Estimates

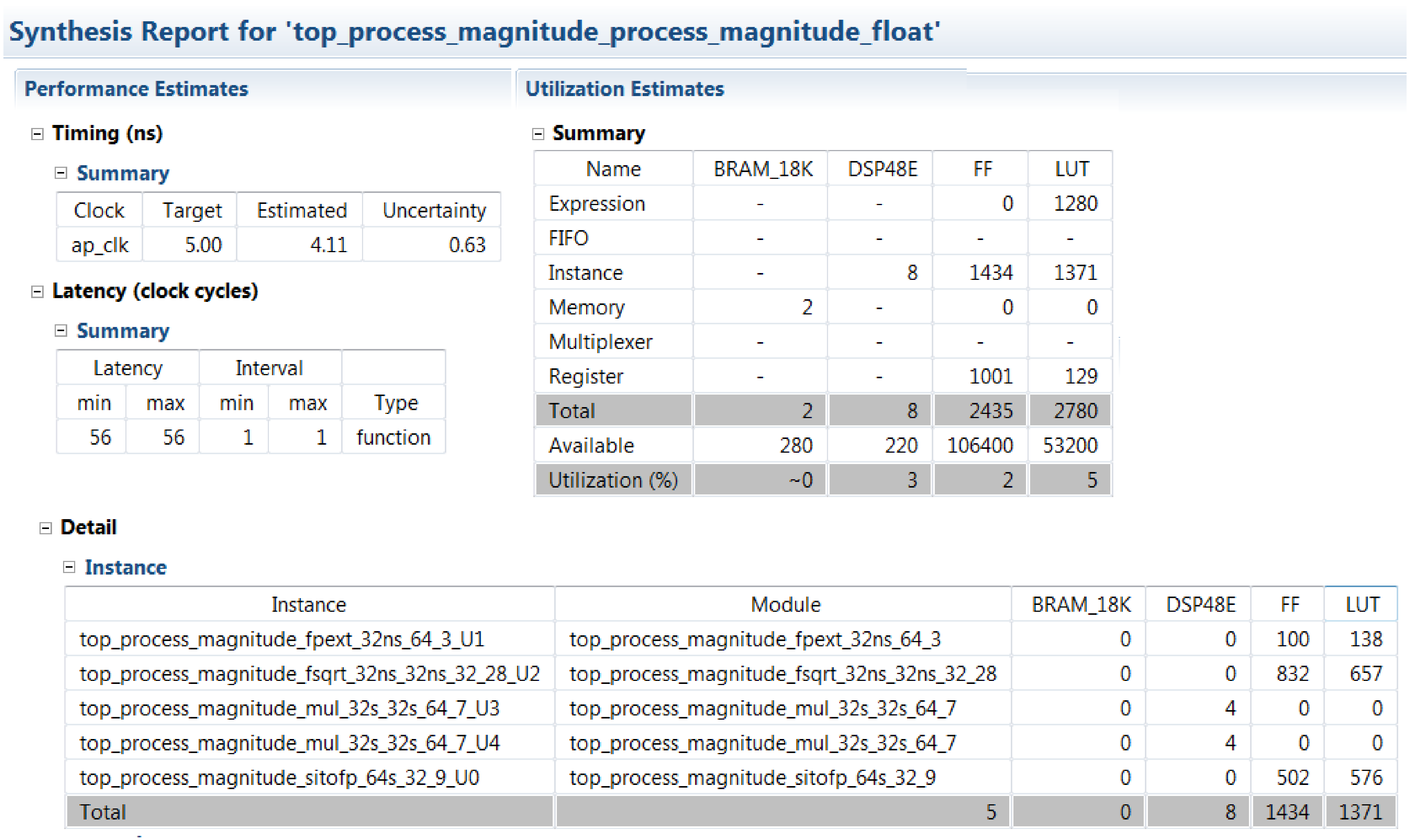
⊟ **Summary**

Name	BRAM_18K	DSP48E	FF	LUT
Expression	-	-	-	-
FIFO	-	-	-	-
Instance	8	32	10188	11277
Memory	-	-	-	-
Multiplexer	-	-	-	64
Register	-	-	12	-
Total	8	32	10200	11341
Available	280	220	106400	53200
Utilization (%)	2	14	9	21

⊟ **Instance**

Instance	Module	BRAM_18K	DSP48E	FF	LUT
grp_top_process_magnitude_func1_process_magnitude_fu_58	top_process_magnitude_func1_process_magnitude	2	8	2545	2815
grp_top_process_magnitude_func2_process_magnitude_fu_34	top_process_magnitude_func2_process_magnitude	2	8	2547	2819
grp_top_process_magnitude_func3_process_magnitude_fu_46	top_process_magnitude_func3_process_magnitude	2	8	2547	2819
grp_top_process_magnitude_func4_process_magnitude_fu_22	top_process_magnitude_func4_process_magnitude	2	8	2549	2824
Total	4	8	32	10188	11277

Solution 2: Synthesis Estimation, inner function



Synthesis Report for 'top_process_magnitude_process_magnitude_float'

Performance Estimates

⊟ Timing (ns)

⊟ Summary

Clock	Target	Estimated	Uncertainty
ap_clk	5.00	4.11	0.63

⊟ Latency (clock cycles)

⊟ Summary

Latency		Interval		
min	max	min	max	Type
56	56	1	1	function

Utilization Estimates

⊟ Summary

Name	BRAM_18K	DSP48E	FF	LUT
Expression	-	-	0	1280
FIFO	-	-	-	-
Instance	-	8	1434	1371
Memory	2	-	0	0
Multiplexer	-	-	-	-
Register	-	-	1001	129
Total	2	8	2435	2780
Available	280	220	106400	53200
Utilization (%)	~0	3	2	5

⊟ Detail

⊟ Instance

Instance	Module	BRAM_18K	DSP48E	FF	LUT
top_process_magnitude_fpext_32ns_64_3_U1	top_process_magnitude_fpext_32ns_64_3	0	0	100	138
top_process_magnitude_fsqrt_32ns_32ns_32_28_U2	top_process_magnitude_fsqrt_32ns_32ns_32_28	0	0	832	657
top_process_magnitude_mul_32s_32s_64_7_U3	top_process_magnitude_mul_32s_32s_64_7	0	4	0	0
top_process_magnitude_mul_32s_32s_64_7_U4	top_process_magnitude_mul_32s_32s_64_7	0	4	0	0
top_process_magnitude_sitofp_64s_32_9_U0	top_process_magnitude_sitofp_64s_32_9	0	0	502	576
Total	5	0	8	1434	1371

Third method:
CORDIC SQRT 32-bit approximation

C Code of Solution 3:
Cordic-based fully 32-bit integer sqrt function

```
int process_magnitude_cordic(int real_data, int imag_data)
{
  int mag_data; // 32-bit data type

  mag_data = (int) cordic_sqrt<40,32>(real_data, imag_data);

  return mag_data;
}
```

Solution 3: CORDIC in C++ code

```
#define ROT 11 // number of iterations (rotations)
template <int TOT, int INT>
ap_fixed<TOT, INT> cordic_sqrt(int x0, int y0) {

  static const signed short int atan_lut[ROT] = {
   804, 475, 251, 127, 64, 32, 16, 8, 4, 2, 1 };
  signed   short int z, zp;
  unsigned char i;

  ap_fixed<TOT, INT> x, y, xp, yp, x2; // HLS fractional data type

  const ap_fixed<16,1, AP_RND_MIN_INF, AP_SAT> inv_G =
       0.607253031529134; // to compensate cordic gain;

  xp=x0; yp=y0; zp=0; // initialization
```

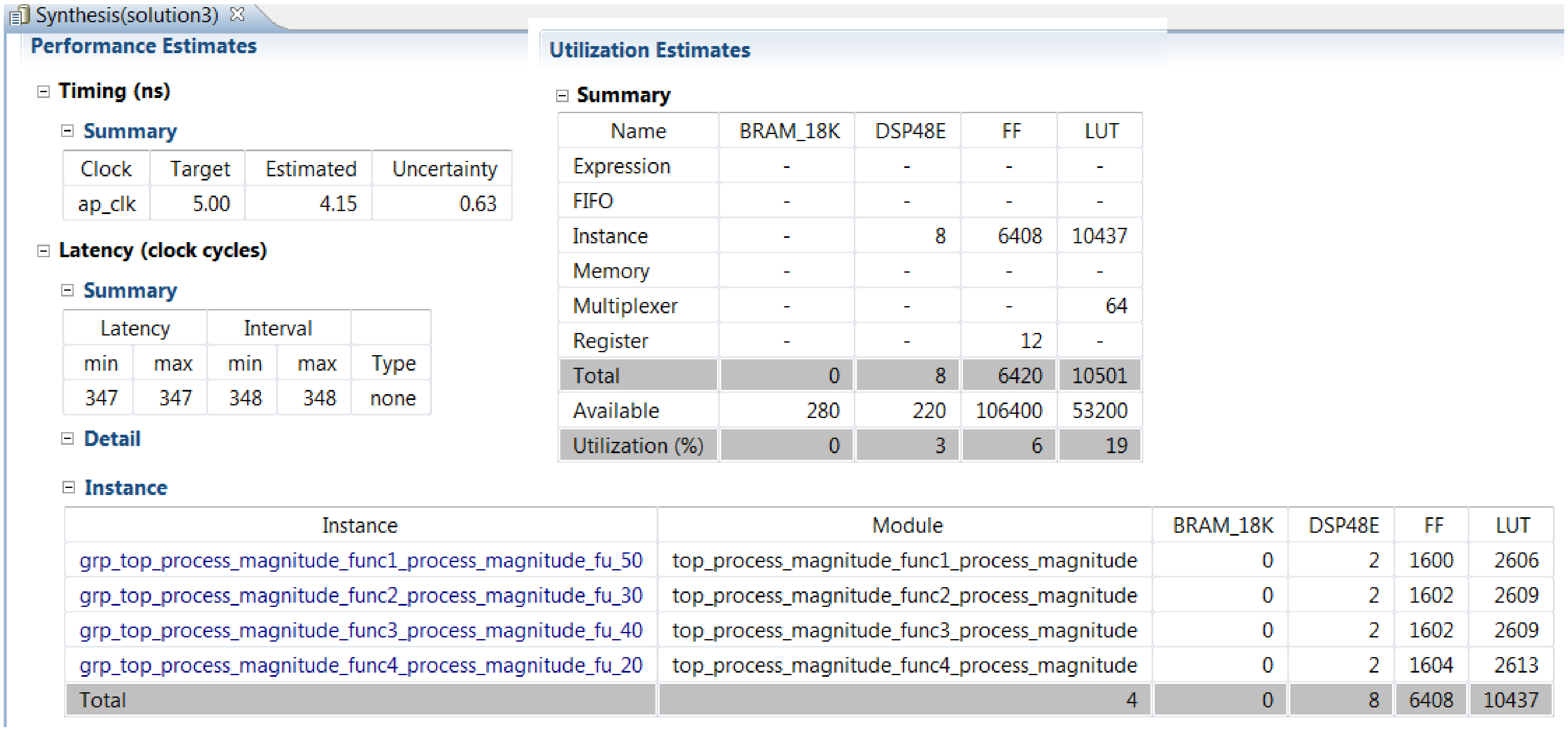
Solution 3: CORDIC in C++ code

```
  for (i=0;i<ROT;i++) {
#pragma HLS PIPELINE II=1
    if (yp<0) { // rotations by tan(2^-i)
      x = xp - (yp>>i); y = yp + (xp>>i); z = zp - atan_lut[i];
    } else {
      x = xp + (yp>>i); y = yp - (xp>>i); z = zp + atan_lut[i];
    }
    xp=x; yp=y; zp=z; // update
  }

  // compensating the cordic gain
#pragma HLS RESOURCE variable=x2 core=MUL6S
  x2     =  xp * inv_G; // x2 = (xp*453)/746;

  return x2;
}
```

Solution 3: CORDIC sqrt

- CORDIC generates $\frac{746}{453}\sqrt{a^2+b^2}$ output.
  - The gain has to be compensated by a multiplication per 453/746
- CORDIC is a very elegant and well documented algorithm (see Xilinx Product Guide 105 for the HW IP core in the references) .
- For a very good accuracy we selected 32 bits integer and 8 further bits fractional (32+8=40)
  - this will cost 2 DSP48 slices per one 40x15 bits multiplication
- No limits on the range of I/O

Solution 3: Synthesis Estimation, top level



Synthesis(solution3)

Performance Estimates

Timing (ns)

Summary

Clock	Target	Estimated	Uncertainty
ap_clk	5.00	4.15	0.63

Latency (clock cycles)

Summary

Latency		Interval		
min	max	min	max	Type
347	347	348	348	none

Detail

Instance

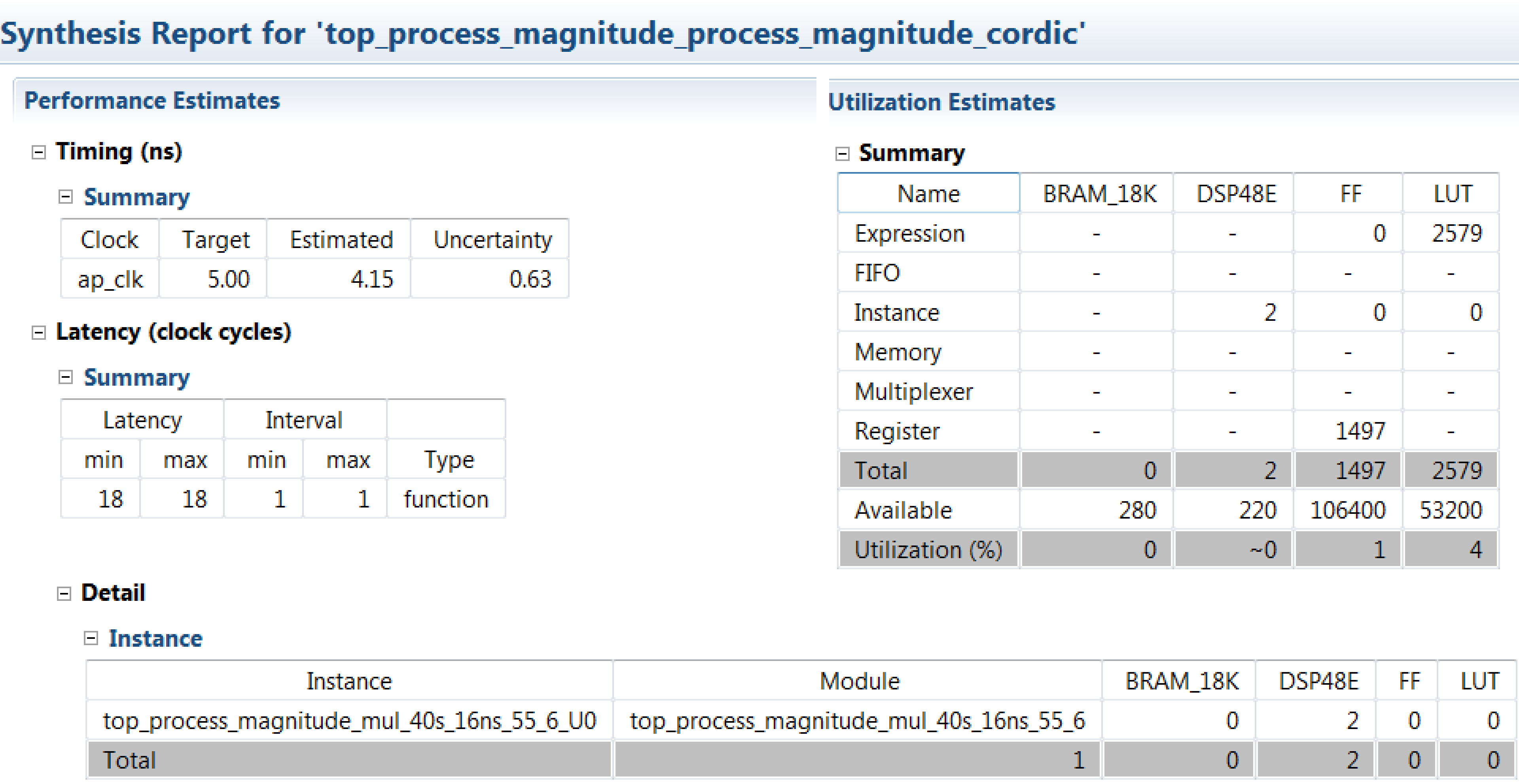
Instance	Module	BRAM_18K	DSP48E	FF	LUT
grp_top_process_magnitude_func1_process_magnitude_fu_50	top_process_magnitude_func1_process_magnitude	0	2	1600	2606
grp_top_process_magnitude_func2_process_magnitude_fu_30	top_process_magnitude_func2_process_magnitude	0	2	1602	2609
grp_top_process_magnitude_func3_process_magnitude_fu_40	top_process_magnitude_func3_process_magnitude	0	2	1602	2609
grp_top_process_magnitude_func4_process_magnitude_fu_20	top_process_magnitude_func4_process_magnitude	0	2	1604	2613
Total	4	0	8	6408	10437

Utilization Estimates

Summary

Name	BRAM_18K	DSP48E	FF	LUT
Expression	-	-	-	-
FIFO	-	-	-	-
Instance	-	8	6408	10437
Memory	-	-	-	-
Multiplexer	-	-	-	64
Register	-	-	12	-
Total	0	8	6420	10501
Available	280	220	106400	53200
Utilization (%)	0	3	6	19

Solution 3: Synthesis Estimation, inner function



Synthesis Report for 'top_process_magnitude_process_magnitude_cordic'

Performance Estimates

Timing (ns)

Summary

Clock	Target	Estimated	Uncertainty
ap_clk	5.00	4.15	0.63

Latency (clock cycles)

Summary

Latency		Interval		
min	max	min	max	Type
18	18	1	1	function

Detail

Instance

Instance	Module	BRAM_18K	DSP48E	FF	LUT
top_process_magnitude_mul_40s_16ns_55_6_U0	top_process_magnitude_mul_40s_16ns_55_6	0	2	0	0
Total	1	0	2	0	0

Utilization Estimates

Summary

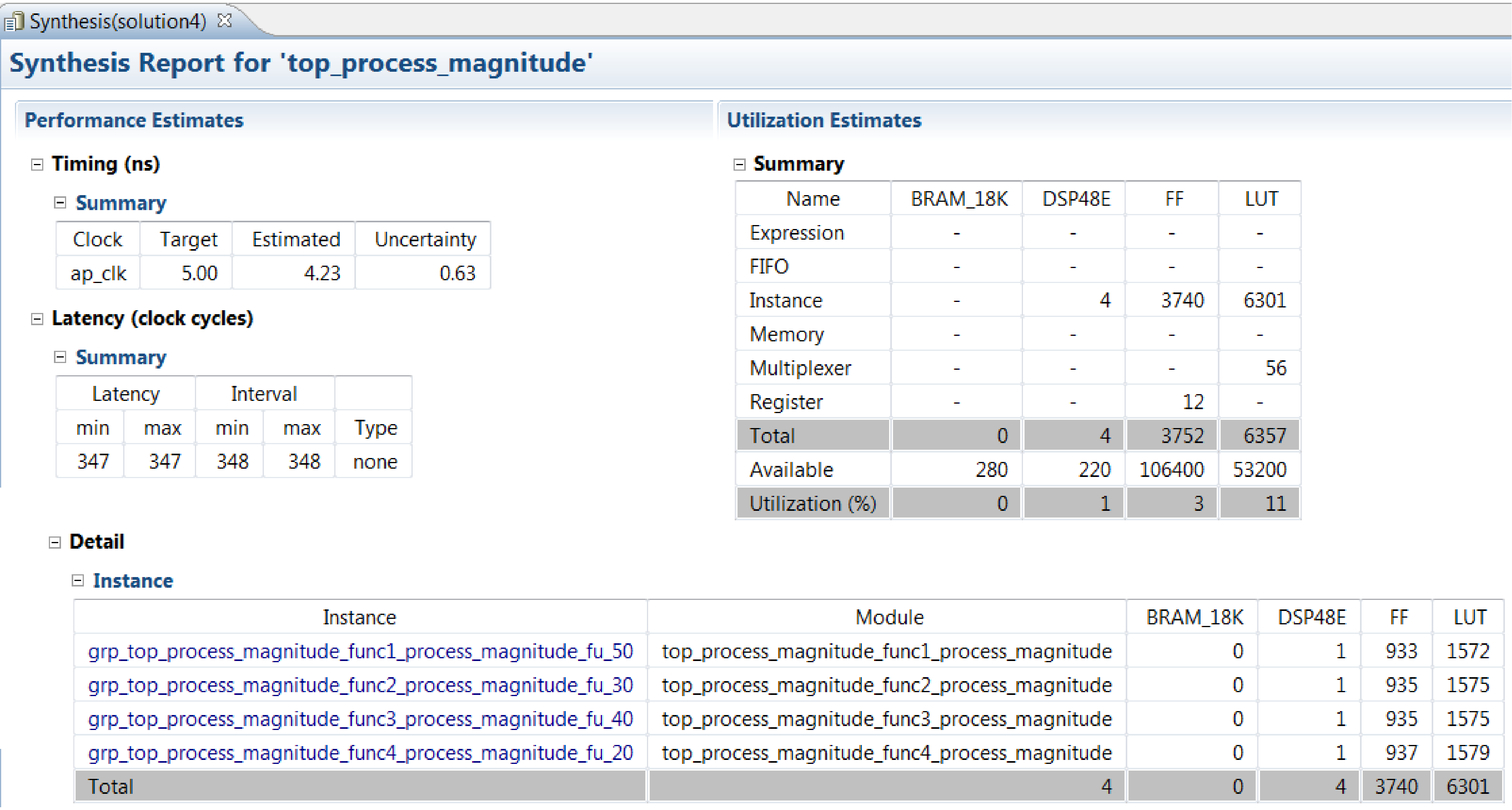
Name	BRAM_18K	DSP48E	FF	LUT
Expression	-	-	0	2579
FIFO	-	-	-	-
Instance	-	2	0	0
Memory	-	-	-	-
Multiplexer	-	-	-	-
Register	-	-	1497	-
Total	0	2	1497	2579
Available	280	220	106400	53200
Utilization (%)	0	~0	1	4

Solution 4:
18-bit CORDIC

C Code of Solution 4:
Cordic-based 18-bit integer sqrt function

```
typedef ap_int<10>  teta_t;
typedef ap_int<18>   dinp_t;
typedef ap_int<24>   dout_t;

dout_t process_magnitude_cordic(dinp_t real_data, dinp_timag_data)
{
  dout_t mag_data; // 24-bit data type

  mag_data = (int) cordic_sqrt<24,18>(real_data, imag_data);

  return mag_data;
}
```

Solution 4: Synthesis Estimation, top level



Synthesis(solution4)

Synthesis Report for 'top_process_magnitude'

Performance Estimates

Timing (ns)

Summary

Clock	Target	Estimated	Uncertainty
ap_clk	5.00	4.23	0.63

Latency (clock cycles)

Summary

Latency		Interval		
min	max	min	max	Type
347	347	348	348	none

Utilization Estimates

Summary

Name	BRAM_18K	DSP48E	FF	LUT
Expression	-	-	-	-
FIFO	-	-	-	-
Instance	-	4	3740	6301
Memory	-	-	-	-
Multiplexer	-	-	-	56
Register	-	-	12	-
Total	0	4	3752	6357
Available	280	220	106400	53200
Utilization (%)	0	1	3	11

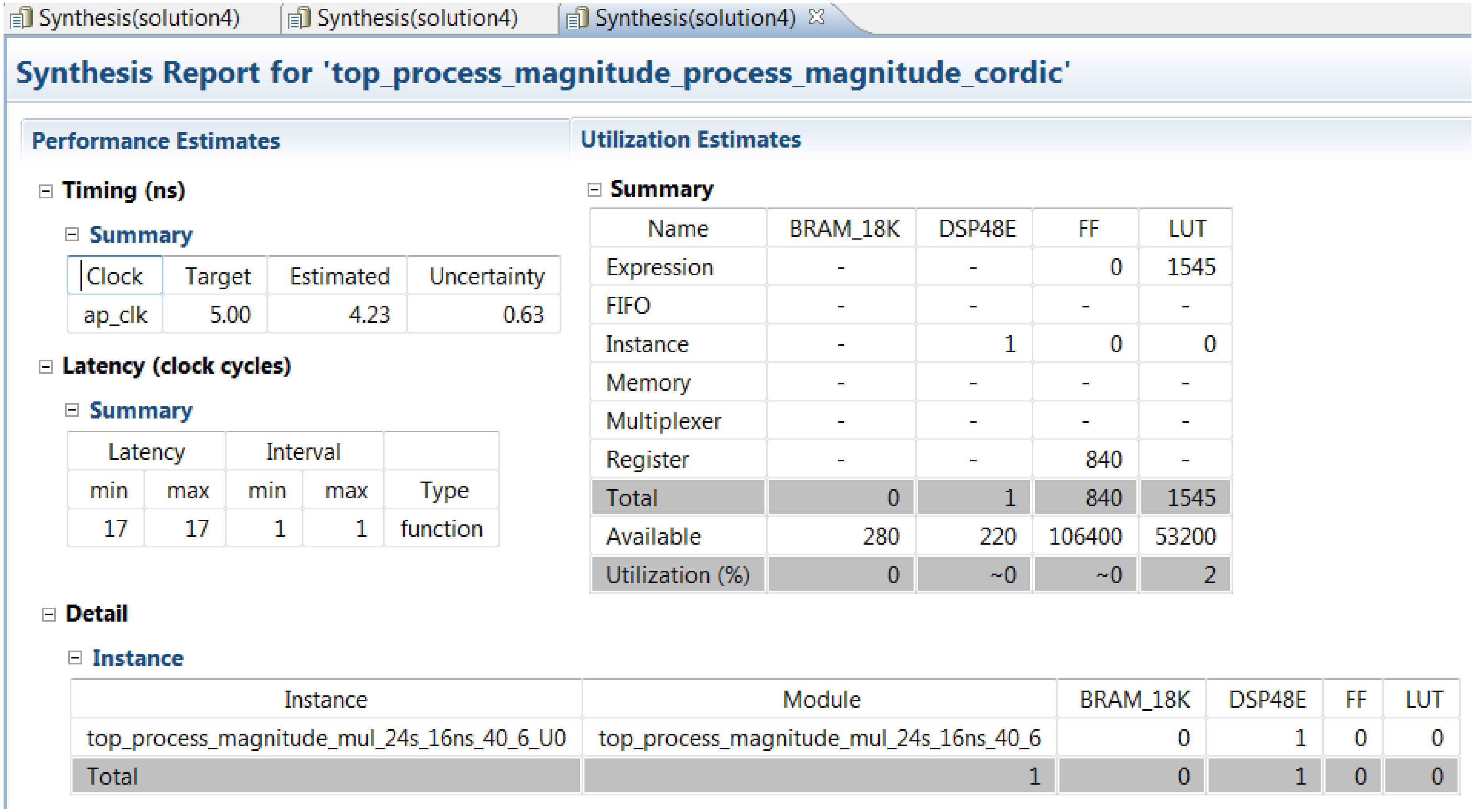
Detail

Instance

Instance	Module	BRAM_18K	DSP48E	FF	LUT
grp_top_process_magnitude_func1_process_magnitude_fu_50	top_process_magnitude_func1_process_magnitude	0	1	933	1572
grp_top_process_magnitude_func2_process_magnitude_fu_30	top_process_magnitude_func2_process_magnitude	0	1	935	1575
grp_top_process_magnitude_func3_process_magnitude_fu_40	top_process_magnitude_func3_process_magnitude	0	1	935	1575
grp_top_process_magnitude_func4_process_magnitude_fu_20	top_process_magnitude_func4_process_magnitude	0	1	937	1579
Total	4	0	4	3740	6301

XILINX ALL PROGRAMMABLE.

Solution 4: Synthesis Estimation, inner function



Synthesis(solution4) | Synthesis(solution4) | Synthesis(solution4)

Synthesis Report for 'top_process_magnitude_process_magnitude_cordic'

Performance Estimates

Timing (ns)

Summary

Clock	Target	Estimated	Uncertainty
ap_clk	5.00	4.23	0.63

Latency (clock cycles)

Summary

Latency		Interval		
min	max	min	max	Type
17	17	1	1	function

Utilization Estimates

Summary

Name	BRAM_18K	DSP48E	FF	LUT
Expression	-	-	0	1545
FIFO	-	-	-	-
Instance	-	1	0	0
Memory	-	-	-	-
Multiplexer	-	-	-	-
Register	-	-	840	-
Total	0	1	840	1545
Available	280	220	106400	53200
Utilization (%)	0	~0	~0	2

Detail

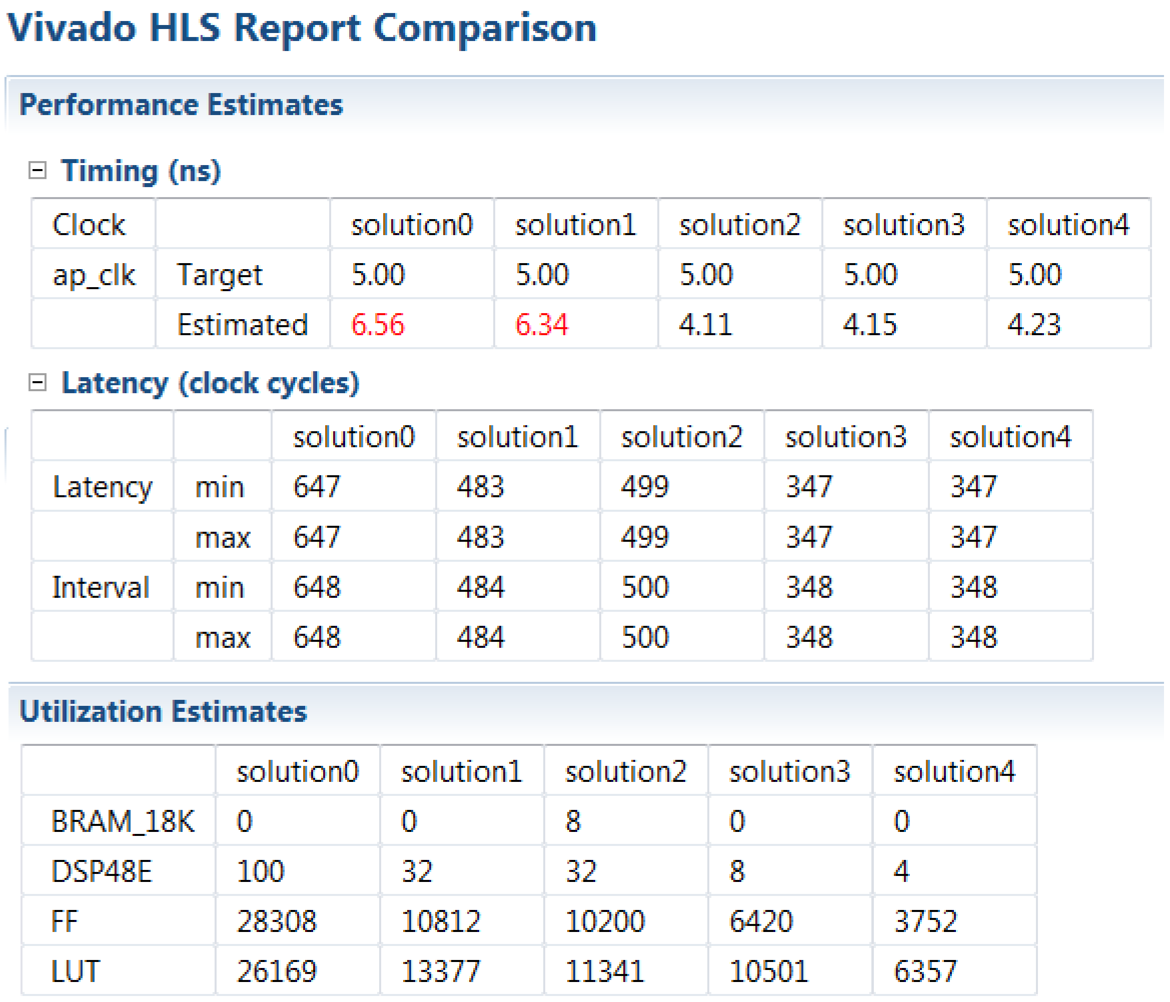
Instance

Instance	Module	BRAM_18K	DSP48E	FF	LUT
top_process_magnitude_mul_24s_16ns_40_6_U0	top_process_magnitude_mul_24s_16ns_40_6	0	1	0	0
Total	1	0	1	0	0

XILINX ALL PROGRAMMABLE.

Summary

Synthesis Performance Estimates Comparison



Vivado HLS Report Comparison

Performance Estimates

⊟ **Timing (ns)**

Clock		solution0	solution1	solution2	solution3	solution4
ap_clk	Target	5.00	5.00	5.00	5.00	5.00
	Estimated	6.56	6.34	4.11	4.15	4.23

⊟ **Latency (clock cycles)**

		solution0	solution1	solution2	solution3	solution4
Latency	min	647	483	499	347	347
	max	647	483	499	347	347
Interval	min	648	484	500	348	348
	max	648	484	500	348	348

Utilization Estimates

	solution0	solution1	solution2	solution3	solution4
BRAM_18K	0	0	8	0	0
DSP48E	100	32	32	8	4
FF	28308	10812	10200	6420	3752
LUT	26169	13377	11341	10501	6357