WRS resource utilisation

on Xilinx US+ FPGA

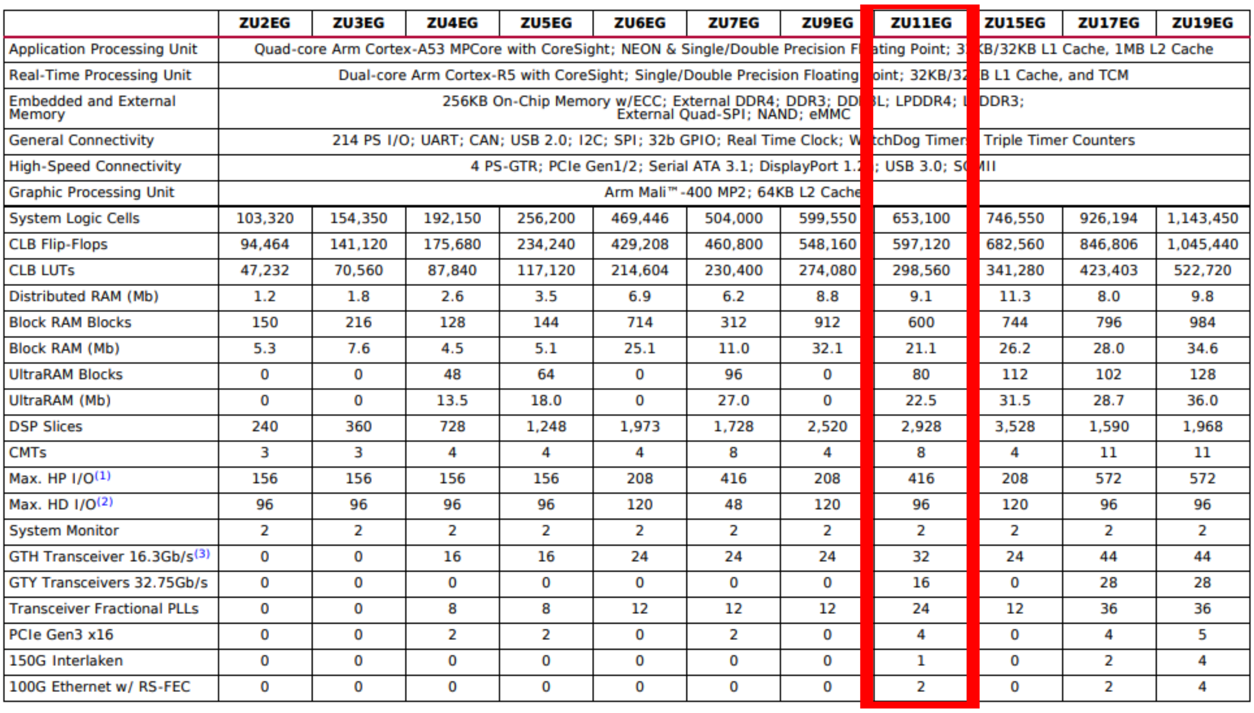
Prepared by Marek Gumiński
marek.guminski@creotech.pl

Contents

1 Introduction

Presented report summarizes evaluation of White Rabbit Switch (WRS) firmware resource evaluation. Firmware is evaluated for Xilinx Zynq UltraScale+ (US+) MPSoC XCZU11EG-1FFVC1156E (see figure 1).

Figure 1: US+ FPGA family resource overview (DS891).



	ZU2EG	ZU3EG	ZU4EG	ZU5EG	ZU6EG	ZU7EG	ZU9EG	ZU11EG	ZU15EG	ZU17EG	ZU19EG
Application Processing Unit	Quad-core Arm Cortex-A53 MPCore with CoreSight; NEON & Single/Double Precision Fl ating Point; 3 KB/32KB L1 Cache, 1MB L2 Cache										
Real-Time Processing Unit	Dual-core Arm Cortex-R5 with CoreSight; Single/Double Precision Floating oint; 32KB/32 B L1 Cache, and TCM										
Embedded and External Memory	256KB On-Chip Memory w/ECC; External DDR4; DDR3; DD L; LPDDR4; L DDR3; External Quad-SPI; NAND; eMMC										
General Connectivity	214 PS I/O; UART; CAN; USB 2.0; I2C; SPI; 32b GPIO; Real Time Clock; W tchDog Timer Triple Timer Counters										
High-Speed Connectivity	4 PS-GTR; PCIe Gen1/2; Serial ATA 3.1; DisplayPort 1. ; USB 3.0; S MII										
Graphic Processing Unit	Arm Mali™-400 MP2; 64KB L2 Cache										
System Logic Cells	103,320	154,350	192,150	256,200	469,446	504,000	599,550	653,100	746,550	926,194	1,143,450
CLB Flip-Flops	94,464	141,120	175,680	234,240	429,208	460,800	548,160	597,120	682,560	846,806	1,045,440
CLB LUTs	47,232	70,560	87,840	117,120	214,604	230,400	274,080	298,560	341,280	423,403	522,720
Distributed RAM (Mb)	1.2	1.8	2.6	3.5	6.9	6.2	8.8	9.1	11.3	8.0	9.8
Block RAM Blocks	150	216	128	144	714	312	912	600	744	796	984
Block RAM (Mb)	5.3	7.6	4.5	5.1	25.1	11.0	32.1	21.1	26.2	28.0	34.6
UltraRAM Blocks	0	0	48	64	0	96	0	80	112	102	128
UltraRAM (Mb)	0	0	13.5	18.0	0	27.0	0	22.5	31.5	28.7	36.0
DSP Slices	240	360	728	1,248	1,973	1,728	2,520	2,928	3,528	1,590	1,968
CMTs	3	3	4	4	4	8	4	8	4	11	11
Max. HP I/O(1)	156	156	156	156	208	416	208	416	208	572	572
Max. HD I/O(2)	96	96	96	96	120	48	120	96	120	96	96
System Monitor	2	2	2	2	2	2	2	2	2	2	2
GTH Transceiver 16.3Gb/s(3)	0	0	16	16	24	24	24	32	24	44	44
GTY Transceivers 32.75Gb/s	0	0	0	0	0	0	0	16	0	28	28
Transceiver Fractional PLLs	0	0	8	8	12	12	12	24	12	36	36
PCIe Gen3 x16	0	0	2	2	0	2	0	4	0	4	5
150G Interlaken	0	0	0	0	0	0	0	1	0	2	4
100G Ethernet w/ RS-FEC	0	0	0	0	0	0	0	2	0	2	4

The firmware was supposed to be tested in following configurations:

1. 1G Ethernet - based on current proposed_master branch
2. 1G Ethernet with redundancy support (see figure 2)
3. 10G Ethernet
4. 10G Ethernet with redundancy support (see figure 2)

The firmware didn't have to be functional nor did it need to implement.

2 Work done

Build system

Project used to evaluate resource utilisation may be easily rebuild with HDLmake. It required small changes in syn/scb_18ports/Manifest.py file.

Block RAM

Netlist representation of Block RAM was replaced by Xilinx Parametized Macro (XPM).
Netlists were removed because they are are not supported by Vivado.

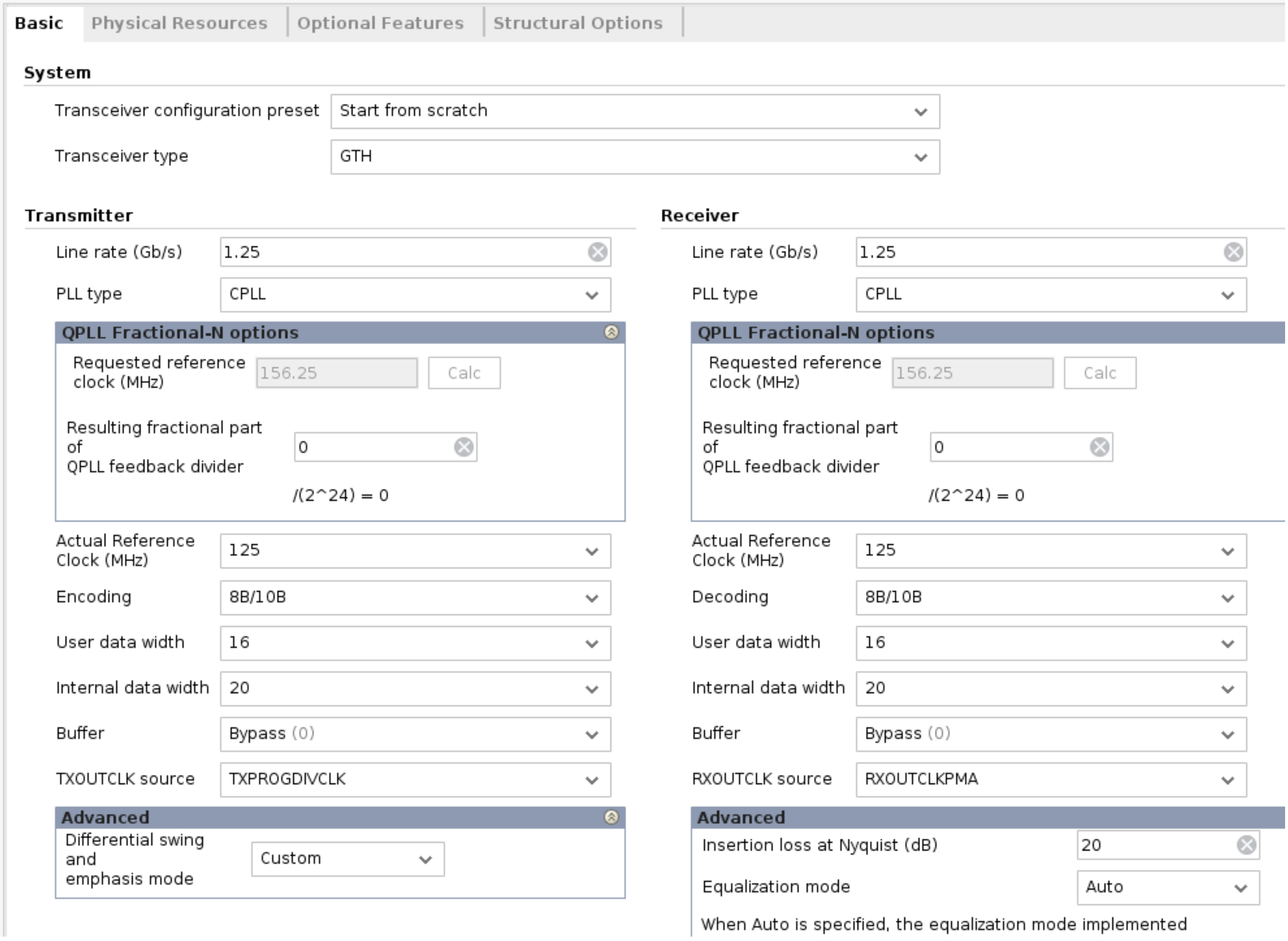
Xilinx primitives

Input and output buffer primitives have been replaced by US+ family counterparts.
Some changes were also required in block platform/xilinx/oserdes_8_to_1.vhd. Oserdes resolution has increased in US+.

Gigabit transceivers

Original Ethernet phys have been replaced with Xilinx IPcore "UltraScale FPGAs Transceivers Wizard". Separate versions were created for 1G Ethernet (line rate 1.25 Gb/s, 125 MHz clock, 20b internal data width as shown in figure 2) and 10G Ethernet (line rate 12.5, 156.25 MHz clock, 40b internal data width as shown in figure 3).
A generate loop (concatenate_gen) was added to scb_top_synthesis to adjust existing phy interface to IPcore ports. Interconnect between the design and phys is unlikely to be functional, but should prevent Vivado from removing any major components.

Figure 2: Gigabit transceivers configuration



System

Transceiver configuration preset: Start from scratch
Transceiver type: GTH

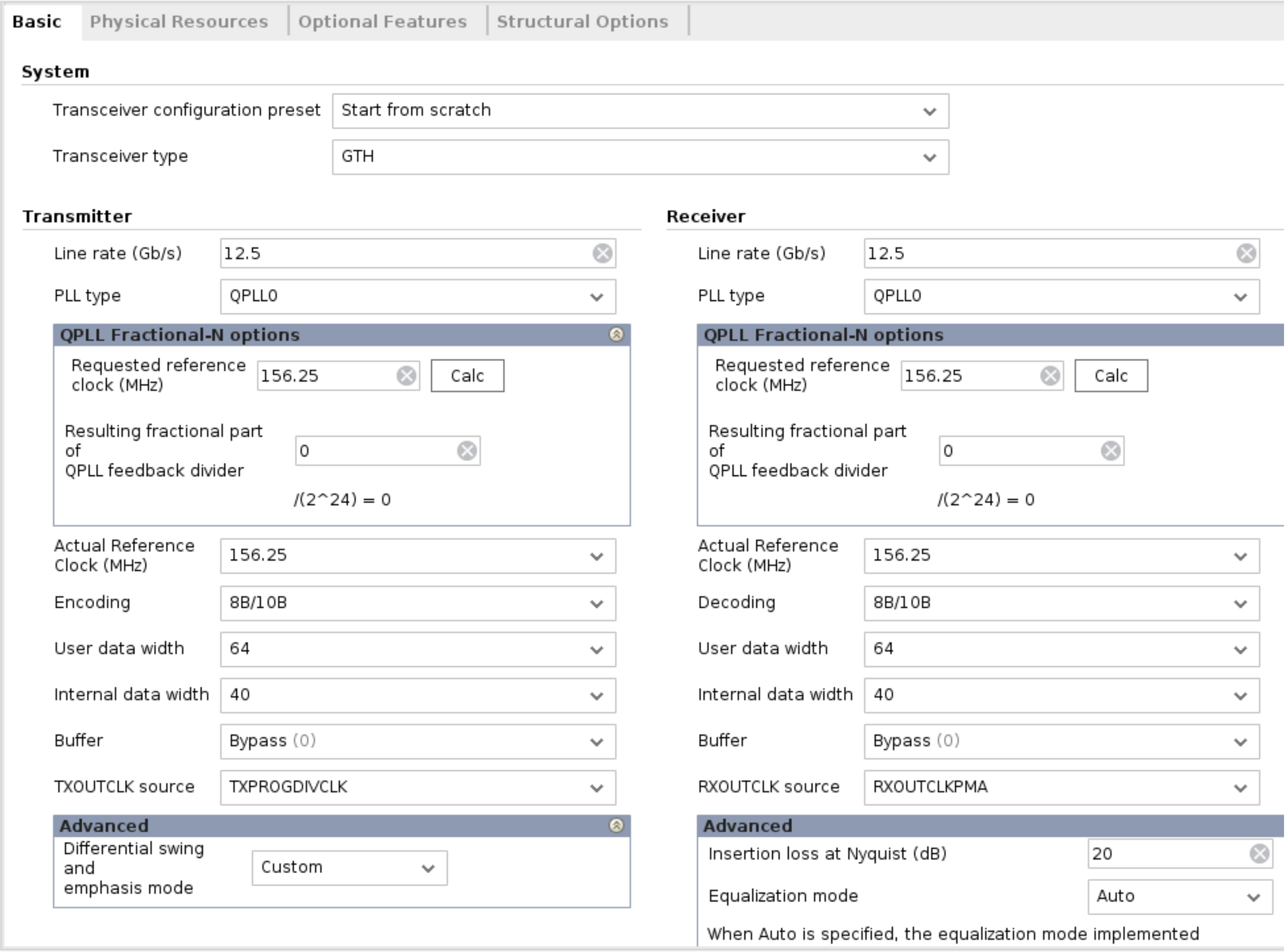
Transmitter

Line rate (Gb/s): 1.25
PLL type: CPLL
QPLL Fractional-N options
Requested reference clock (MHz): 156.25
Resulting fractional part of QPLL feedback divider: 0
/(2^24) = 0
Actual Reference Clock (MHz): 125
Encoding: 8B/10B
User data width: 16
Internal data width: 20
Buffer: Bypass (0)
TXOUTCLK source: TXPROGDIVCLK
Advanced
Differential swing and emphasis mode: Custom

Receiver

Line rate (Gb/s): 1.25
PLL type: CPLL
QPLL Fractional-N options
Requested reference clock (MHz): 156.25
Resulting fractional part of QPLL feedback divider: 0
/(2^24) = 0
Actual Reference Clock (MHz): 125
Decoding: 8B/10B
User data width: 16
Internal data width: 20
Buffer: Bypass (0)
RXOUTCLK source: RXOUTCLKPMA
Advanced
Insertion loss at Nyquist (dB): 20
Equalization mode: Auto
When Auto is specified, the equalization mode implemented

Figure 3: 10 Gigabit transceivers configuration



System

Transceiver configuration preset: Start from scratch

Transceiver type: GTH

Transmitter

Line rate (Gb/s): 12.5

PLL type: QPLL0

QPLL Fractional-N options

Requested reference clock (MHz): 156.25

Resulting fractional part of QPLL feedback divider: 0 /(2^24) = 0

Actual Reference Clock (MHz): 156.25

Encoding: 8B/10B

User data width: 64

Internal data width: 40

Buffer: Bypass (0)

TXOUTCLK source: TXPROGDIVCLK

Advanced

Differential swing and emphasis mode: Custom

Receiver

Line rate (Gb/s): 12.5

PLL type: QPLL0

QPLL Fractional-N options

Requested reference clock (MHz): 156.25

Resulting fractional part of QPLL feedback divider: 0 /(2^24) = 0

Actual Reference Clock (MHz): 156.25

Decoding: 8B/10B

User data width: 64

Internal data width: 40

Buffer: Bypass (0)

RXOUTCLK source: RXOUTCLKPMA

Advanced

Insertion loss at Nyquist (dB): 20

Equalization mode: Auto

When Auto is specified, the equalization mode implemented

Redundancy support

Code required to implement link redundancy was developed by Maciej Lipiński. It is available in ohwr repository in branches ML-PTP-support-150317 and TRUandRTUandEndpointAndSWcoreAndTATSU.
Branch TRUandRTUandEndpointAndSWcoreAndTATSU was already merged with proposed master. Following generics had to be activated in scb_top_bare instantiation in order to evaluate resource utilisation of TRU and TATSU components: g_with_TRU, g_with_TATSU.
Branch ML-PTP-support-150317 required merging into proposed_master. Regular merging proved to be difficult, due to multiple merge conflicts. Instead ML-PTP-support-150317 was rebased on proposed master. This way conflicts in consecutive commits could be resolved one by one. A PSU (component that is added in this branch) may be enabled with generic g_with_PSU.

10 Gb link

Migration to 10 Gb link requires changing reference clock frequency and link data width. Frequency change doesn't effect resource utilisation. Timing is not even verified during synthesis.
Change of the data width requires changes in some parts of the design.
Most of components passes the data as is, so it is sufficient to change record declaration in vhdl packages.
Vectors t_wrf_source_out.dat, t_phyif_output.tx_data and t_phyif_input.rx_data were resized from 16 to

64 bytes.
The endpoint on the other hand contains multiple comparisons and assignments that assume certain (16 bit) data width. Changing data width in this component would require rewriting multiple FSM's.
It was decided to instantiate 4 endpoints in parallel instead. Each of the multiplicated endpoints processes 16 bit of the 64 data word received from the phys. Other control signals (rx_k, tx_disparity etc) are connected to all 4 endpoints. Data output of the endpoints is concatenated back into 64 bit array. Other outputs of the endpoints are xored in order to make sure that nothing will be over optimised during synthesis. Vector t_ep_internal_fabric.data has original 16 bit width.
The interface between original existing signals and multiplicated endpoint is located inside U_Real_Top/gen_endpoints_and_phys generate loop.

3 Found issues

Ext PLL

A minimal frequency of MMCM block was increased to 14Mhz in US+ family. A U_Ext_PLL1 that generated 100 MHz clock from 10 MHz input will will not work correctly in US+ FPGA.

Latch

A latch reported by Vivado was fixed in commit 16c02da485ed4aaea17b8c242b52b40d2cc2481c.

4 Verification

1 Gb link

The design was verified by comparing resource utilisation of original WRS firmware for Virtex 6 FPGA and US+ version.
The Virtex firmware was implemented with keep_hierarchy property disabled, as it is in the repository, so the resource distribution between components may be distorted. US+ version is synthesised with keep_hierarchy enabled.
Comparison of LUT and Register usage is simple, as both ISE and Vivado report them in a similar manner.
Comparison of BRAM usage is more complicated because ISE reports number of BRAM instantiations without diverting between 18 kB and 36 kB blocks. Vivado counts 18 kB blocks as 0.5 of a BRAM.
Comparison is given in table 1. LUT utilisation of subcomponents (i.e U_RTU, U_Endpoint_X) differ quite significantly, but total utilisation of U_Real_Top and top are very similar. The differences in subcomponents are probably caused by flatten hierarchy.
Register utilisation matches almost perfectly.
BRAM utilisation seem do differ a lot, but it is caused by different way of reporting 18 kB BRAMs. Design summary shows that the whole design uses 165 RAMB36 and 121 RAMB18 (286 in total). Vivado would report it as 165+0.5*121=225, so the difference is 9 BRAMs.
The resource utilisation of firmware with redundancy was compared with the original one. As expected utilisation didn't change of any of subcomponents listed in table 1 except for increase in top and U_Real_Top caused by addition of subcomponents U_TRU, U_PSU, U_TATSU. Total utilisation of new subcomponents match the increase in top module usage.

Table 1: Extract of resource utilisation report of original firmware synthesized for Virtex 6 and US+ FPGA.

		LUT	REG	BRAM
top	Virtex 6	96k	82k	286
	US+	108k	91k	216
U_Real_Top	Virtex 6	92k	80k	286
	US+	102k	82k	216
U_RT_Subsystem	Virtex 6	2.4k	5.7k	34
	US+	5.1k	5.7k	40.5
U_Swcore	Virtex 6	37.4k	29.5k	83
	US+	44.8k	30.6k	55
U_RTU	Virtex 6	0.8k	5.4k	24
	US+	6.4k	5.4k	23
U_Nic	Virtex 6	0.2k	0.8k	11
	US+	0.8k	0.8k	10.5
U_Endpoint_X	Virtex 6	0.2k	1.7k	6
	US+	1.7k	1.8k	3.5

Table 2: Comparison between original version and one with redundancy support.

		LUT	REG	BRAM
top	Virtex 6	96k	82k	286
	US+	108k	91k	216
	US+ Redundancy	114k	94k	240
U_Real_Top	Virtex 6	92k	80k	286
	US+	102k	82k	216
	US+ Redundancy	108k	85k	240
U_TRU	Virtex 6	-	-	-
	US+	-	-	-
	US+ Redundancy	4.5k	1.5k	24
U_PSU	Virtex 6	-	-	-
	US+	-	-	-
	US+ Redundancy	0.7k	0.3k	0.5
U_TATSU	Virtex 6	-	-	-
	US+	-	-	-
	US+ Redundancy	0.5k	0.3k	0

Table 3: Comparison of resource utilisation between 1 Gb and 10 Gb firmware without redundancy support.

		LUT	REG	BRAM
top	1 Gb	108k	91k	216
	10 Gb	212k	200k	506
U_Real_Top	1 Gb	102	82k	216
	10 Gb	212k	200k	506
U_RT_Subsystem	1 Gb	5.1k	5.7k	40.5
	10 Gb	5.1k	5.7k	40.5
U_Swcore	1 Gb	44.8k	30.6k	55
	10 Gb	57.8k	51.7k	115.5
U_RTU	1 Gb	6.4k	5.4k	23
	10 Gb	6.4k	5.4k	23
U_Nic	1 Gb	0.8k	0.8k	10.5
	10 Gb	0.9k	0.9k	11.5
U_Endpoint_X	1 Gb	1.7k	1.8k	3.5
	10 Gb	1.7k	1.8k	3.5

10 Gb link

Table 3 shows resource utilisatoin comparison of 1 Gb and 10 Gb versions of the original (no redundancy) firmware.

As described in section 2 1 Gb endpoints were used in 10 Gb version of the firmware. The endpoints were multiplicated to simulate increased complexity. As shown in table 3 resource utilisation of single endpoint didn't change, but total utilisation increased 4 times (there were 19 endpoints, now there is 4*19=76).

Little difference of resource utilisation is seen in U_RTU, U_RT_Subsystem and U_Nic.

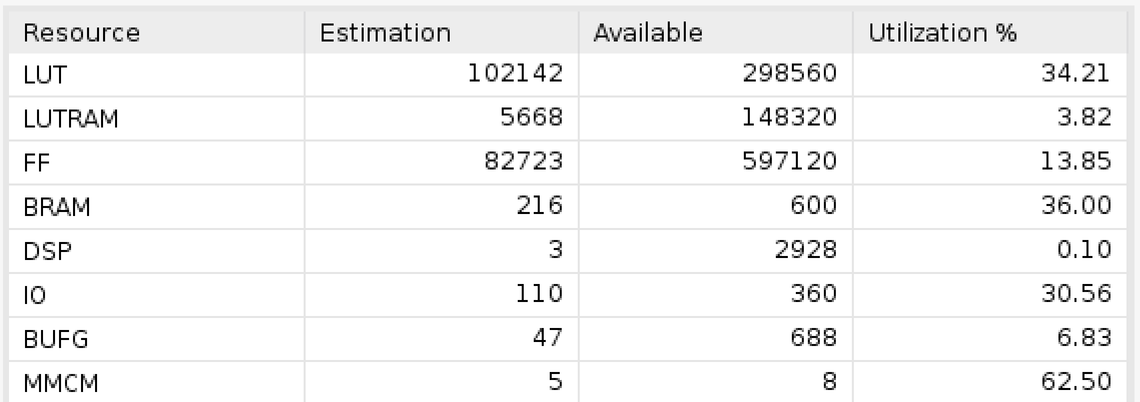
The biggest difference is visible in U_Swcore. BRAM utilisation changed mainly due to increase of "MULTIPORT_MEMORY", that is used for buffering of data packets. LUT and register utilisation increased approximately 1.5 times.

Total increase of top and U_Real_Top roughly matches increased consumption of multiplicatated endpoints and changed U_Swcore.

5 Summary

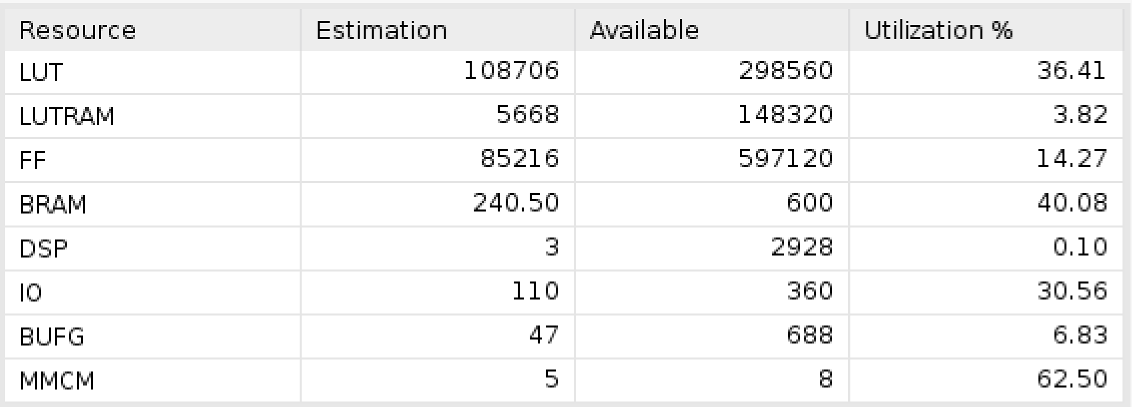
Figures 4, 5, 6 and 7 show resource utilisation estimated by Vivado IDE.

Figure 4: Resource utilisation with 1Gb serial links and without redundancy components.



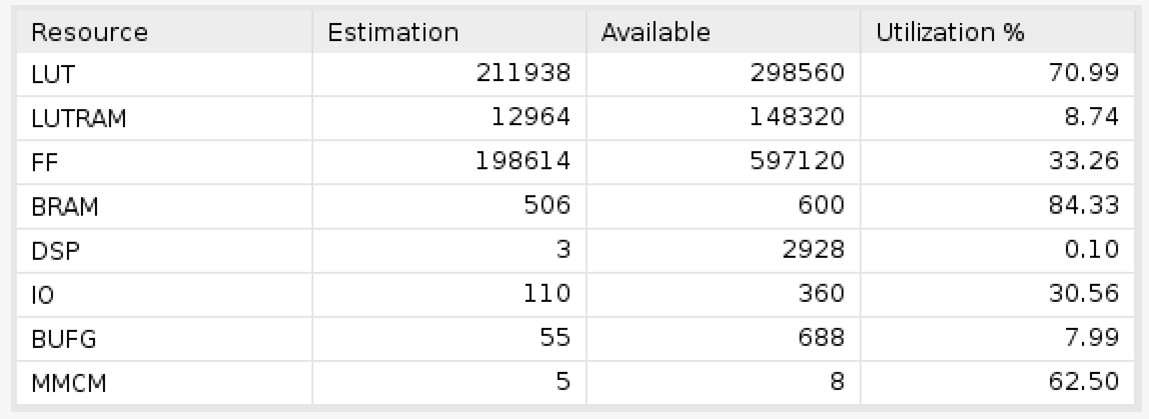
Resource	Estimation	Available	Utilization %
LUT	102142	298560	34.21
LUTRAM	5668	148320	3.82
FF	82723	597120	13.85
BRAM	216	600	36.00
DSP	3	2928	0.10
IO	110	360	30.56
BUFG	47	688	6.83
MMCM	5	8	62.50

Figure 5: Resource utilisation with 1Gb serial links and with redundancy components enabled.



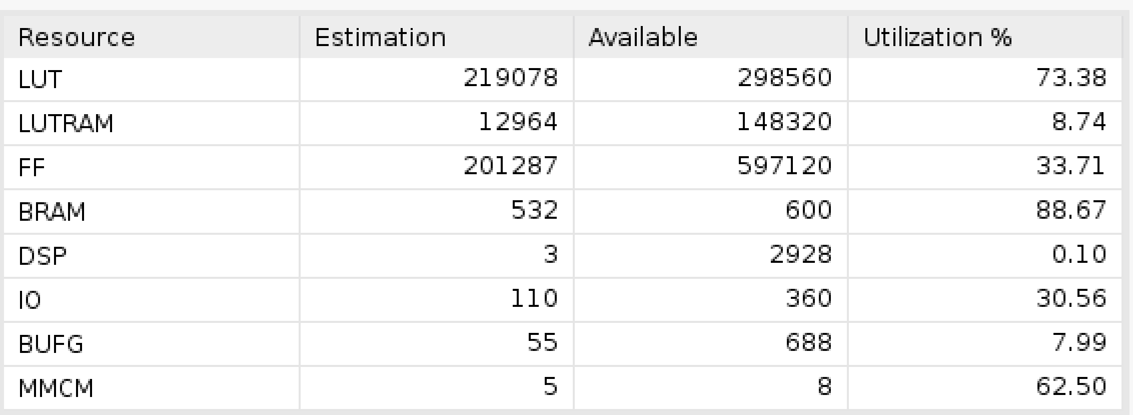
Resource	Estimation	Available	Utilization %
LUT	108706	298560	36.41
LUTRAM	5668	148320	3.82
FF	85216	597120	14.27
BRAM	240.50	600	40.08
DSP	3	2928	0.10
IO	110	360	30.56
BUFG	47	688	6.83
MMCM	5	8	62.50

Figure 6: Resource utilisation with 10Gb serial links and without redundancy components.



Resource	Estimation	Available	Utilization %
LUT	211938	298560	70.99
LUTRAM	12964	148320	8.74
FF	198614	597120	33.26
BRAM	506	600	84.33
DSP	3	2928	0.10
IO	110	360	30.56
BUFG	55	688	7.99
MMCM	5	8	62.50

Figure 7: Resource utilisation with 10Gb serial links and with redundancy components enabled.



Resource	Estimation	Available	Utilization %
LUT	219078	298560	73.38
LUTRAM	12964	148320	8.74
FF	201287	597120	33.71
BRAM	532	600	88.67
DSP	3	2928	0.10
IO	110	360	30.56
BUFG	55	688	7.99
MMCM	5	8	62.50