SYSTEMATIC EVALUATION OF THE EUROPEAN NG-LARGE FPGA & EDA TOOLS FOR ON-BOARD PROCESSING

Vasileios Leon[1], Ioannis Stamoulias[1], George Lentaris[1], Dimitrios Soudris[1], Rubén Domingo[2], Miguel Verdugo[2], David Gonzalez-Arjona[2], David Merodio Codinachs[3], and Isabelle Conway[3]

[1]*National Technical University of Athens, School of Electrical & Computer Engineering, Greece*
[2]*GMV Aerospace and Defense SAU, Space Segment & Robotics, Spain*
[3]*European Space Agency, European Space Research & Technology Centre, Netherlands*

ABSTRACT

The rapid growth of space applications has led the space industry to explore novel, innovative platforms for on-board data processing. The European BRAVE family of FPGAs is considered a promising solution in the generally limited pool of radiation-hardened devices. Our current work develops a methodology to drive the evaluation of the BRAVE EDA tools and devices. The paper focuses on NG-LARGE, i.e., the largest FPGA of the 65nm RHBD technology of NanoXplore. The proposed approach consists of numerous inter-dependent steps for assessing the entire FPGA design flow, including high-performance benchmarking with HDL IPs from ongoing Vision-Based Navigation activities, testing on actual HW, as well as comparisons vs state-of-the-art FPGAs.

Key words: BRAVE; NanoXplore; NG-LARGE; Assessment Methodology; Testing; Digital Signal Processing; Space Applications; European Space Agency.

1. INTRODUCTION

The proliferation of demanding workloads for on-board processing systems in space applications, such as Vision-Based Navigation (VBN) and Earth Observation (EO), marks a new era of embedded on-board computing. To achieve reconfigurable high-performance computing with restricted power budget and enhanced dependability, the space industry explores multiple new platforms/technologies. Among the existing solutions, the FPGAs have gained increased popularity due to their attractive performance-per-power ratio [3], outperforming existing rad-hard CPUs. As a result, the FPGAs are constantly being evaluated for future missions, either as main accelerators [4, 8, 9] or framing processors [5]. In this context, the new European space-grade family of FPGAs, namely BRAVE [7] by NanoXplore [1], is expected to play a key role owing to its radiation-hardness by design (RHBD), high density, and reconfiguration features, as well as its software tools providing end-to-end FPGA development and seamless chip configuration.

Currently, the number of space-grade FPGAs available in the market is relatively limited and becomes even smaller when considering European-only space-grade FPGAs. Most of these FPGAs are inferior to their Commercial Off-The-Shelf (COTS) counterparts, either in terms of performance or resource availability. The most prominent chips are the Xilinx Virtex-4QV (SRAM, 90nm) & Virtex-5QV (SRAM, 65nm), and the Microsemi RTG4 (flash, 65nm), RTAX (anti-fuse, 150nm) & RT ProASIC3 (flash, 130nm). More recently, two new FPGAs have been introduced in the market, i.e., the Microsemi RT PolarFire (SONOS, 28nm) and the Xilinx RT Kintex Ultrascale (SRAM, 20nm). The pool also includes FPGAs with limited resources, such as the Microsemi RTSX-SU (anti-fuse, 250nm) and the Atmel AT40K (350nm) & ATF280 (180nm).

The BRAVE family of FPGAs constitutes an additional option and a promising solution in the aforementioned pool of space-grade FPGAs. NanoXplore provides various BRAVE FPGAs ranging from low-end to high-end, i.e., NG-MEDIUM (65nm), NG-LARGE (65nm), and NG-ULTRA (28nm), which are radiation hardened by design, and incorporate the traditional FPGA programmable logic resources. The fabric architecture of NG-LARGE is illustrated in Figure 1, and its resources are summarized as follows: 137K Look-Up-Tables (LUTs) of 4 inputs, 129K D Flip-Flops (DFFs), 32K Carry Units (CYs), 384 Digital Signal Processors (DSPs), 192 RAM Blocks (RAMBs) of 48Kbits, 672 Register Files (RFs) of 64×16 bits, 4 Phase-Locked Loops (PLLs). NG-LARGE and NG-ULTRA include ARM processors, with the latter implementing a full System-On-Chip (SoC). Moreover, the BRAVE family provides features essential for embedding computing in space, such as the SpaceWire interface for fast I/O and chip configuration, and memory scrubbing to ensure the continuous correct functionality. Finally, in terms of EDA tools, NanoXplore supports the entire FPGA design flow via the NXmap SW tool.

The efficient utilization of a new FPGA family, such as that provided by NanoXplore, as well as the full exploitation of its EDA tools' capabilities, require a systematic and disciplined approach. For this reason, the European Space Agency (ESA) is supporting a set of activities involving the assessment of the NanoXplore EDA tools and the high-performance benchmarking of the NanoXplore

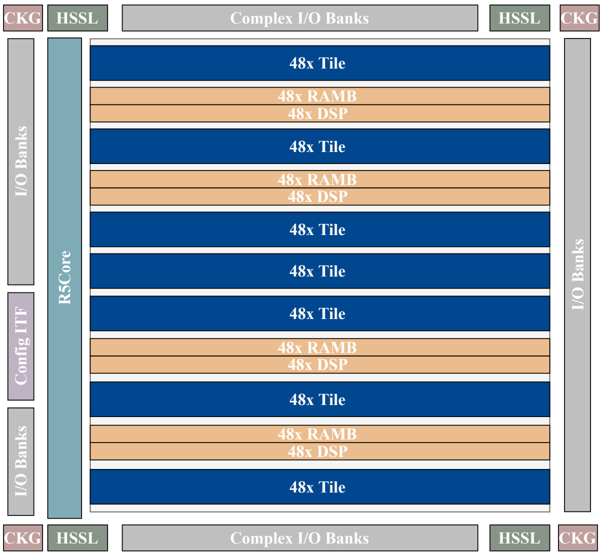


Figure 1: The fabric architecture of NG-LARGE [1, 7].

FPGAs. These activities aim to improve the NanoXplore tools and devices, and evaluate their viability as on-board data processors. In the QUEENS-FPGA activity [6], we evaluated the NG-MEDIUM FPGA based on an assessment methodology. In this paper, we enhance our methodology and present the evaluation of the next chip of the BRAVE FPGA series, i.e., NG-LARGE. The work is performed in the context of the QUEENS2 activity.

The proposed quality assessment methodology is based on the systematic verification and testing of the NanoXplore EDA tools, i.e., the FPGA development and programming tools. Our methodology involves high-performance benchmarking with HW IP cores, which are developed in house for past ESA activities, and thus, represent the performance requirements in current and future rovers/spacecraft. The contribution of this paper lies in (i) introducing an enhanced version of our assessment methodology for evaluating new devices/tools, (ii) evaluating the NanoXplore EDA tools by examining the available options throughout the entire FPGA design flow, (iii) evaluating the NG-LARGE capabilities as on-board processor with representative VBN benchmarks.

The paper is organized as follows. Section 2 describes our quality assessment methodology. Sections 3 presents experimental benchmarking results. Section 4 presents a system evaluation. Finally, Section 5 concludes the paper.

2. ASSESSMENT METHODOLOGY

In this section, we introduce our methodology for evaluating the NanoXplore EDA tools and the NG-LARGE FPGA. The methodology is divided in 5 parts: (i) benchmark selection, (ii) definitions of rating/evaluation method, (iii) synthesis assessment, (iv) place & route assessment, and (v) bitstream generation assessment.

2.1. Selection of Benchmarks

The first step of our approach is to create a pool of HDL benchmarks with diverse complexity [6], i.e., simple circuits (e.g., arithmetic/memory units), designs of medium complexity (e.g., controllers), and high-performance benchmarks (e.g., image processors). With the small circuits, we aim to examine/test specific options of the tools and/or FPGA primitives, while with the high-performance benchmarks, we stress the tools and the FPGA with algorithms from real-world space applications. Moreover, our benchmarks impose different requirements in I/O, memory and computational complexity, thus, our evaluation is diverse and covers a wide range of functionalities met in on-board processing systems.

Regarding the high-performance benchmarks, our initial pool consists of 12 HDL IPs from the signal processing and computer vision domains. To examine their suitability for our assessment methodology, we create multiple configurations for each benchmark by customizing its algorithmic parameters (e.g., image size, data bitwidth, mask size, etc.), and perform an extensive Design Space Exploration (DSE) on the 3rd party EDA tools (Intel/Altera, Xilinx, Microsemi, Synopsys). This exploration does not involve the NanoXplore tools and devices, namely, it is BRAVE-agnostic. We note that we use 3rd party FPGAs with similar features to NG-LARGE (resources, space-grade, technology node, etc.) Based on these results, as well as by considering other metrics, e.g., throughput/activity, parameterization/scalability, use of vendor's IP blocks, etc., we select the benchmarks that will drive our BRAVE evaluation. Specifically for the QUEENS2 activity, we select; FIR Filter for signal processing, Harris Corner Detector and Canny Edge Detector for feature detection, Disparity Constructor and SpaceSweep Constructor for depth extraction in 3D scene reconstruction.

2.2. Definition of Rating Method

The second step of our methodology is to identify the evaluation metrics [6]. To define the metrics, we use two groups of FPGA engineers, i.e., "black-box" engineers that have not used the NanoXplore tools and devices and "grey-box" engineers that have started using them. Indicatively, such metrics are the resource utilization, maximum clock frequency, power consumption, tool runtime & memory, tool reports, tool options & attributes, floorplan capabilities, GUI flexibility, etc.

Next, we introduce a process to rate/evaluate NG-LARGE compared to the 3rd party devices [6]. For each one of the measurable evaluation metrics, we calculate a reference value, which is the average value of all the results obtained from the 3rd party tools/devices. Then, we apply the rating of BRAVE by comparing the NanoXplore value with the reference value. Our rating process has 5 ranges, which are defined by thresholds: deficient (D), if NanoXplore is more than 20% worse, acceptable (A), if NanoXplore is 20%–5% worse, good (G), if NanoXplore is less than 5% worse, very good (V), if NanoXplore is 0.1%–5% better, excellent (E), if NanoXplore is more than 5% better. We note that this rating system is applied at each one of the following methodology steps, which assess the typical processes of the FPGA design flow, i.e., Synthesis, Place & Route (P&R), and Bitstream Generation.

2.3. Assessment of Synthesis

The Synthesis assessment aims to: (i) explore and test the correct functionality of all the NXmap's settings and attributes, (ii) examine the quality of the results for different NXmap settings, (iii) evaluate the ability of the synthesizer to map efficiently the RTL designs on the NG-LARGE primitives, (iv) evaluate the quality of the synthesis reports, (v) rate the resource utilization via systematic comparisons to state-of-the-art 3rd party tools.

The proposed methodology for realizing the aforementioned goals is illustrated in Figure 2. Initially, we adapt the algorithmic parameters of the selected benchmarks according to the features of NG-LARGE (e.g., available resources, architecture of FPGA primitives, etc.). Next, we perform a naive Synthesis with the default NXmap settings of to retrieve the "default" reports and detect possible issues. This step is also considered as test for the synthesizer's flexibility to automatically balance the resource utilization and provide a viable solution.

The naive Synthesis is followed by the phase of "Program-Agnostic Tuning", which explores the available Synthesis-related settings and assesses their capability to drive the Synthesis process according to the user's choices and preferences. In this phase, it is not required for the user to be familiar with the HDL code of the benchmarks. Indicatively, we mention that the tool settings involve choices regarding the mapping effort of the synthesizer, the mapping target of the arithmetic/memory components, the DSP utilization ratio, the register duplication, the style of the FSM encodings, etc. The evaluation is performed both in standalone and combinatorial fashion. Subsequently, we compare the NXmap results with the results of the 3rd party tools based on our rating methodology. In case spikes are observed, a lower level exploration takes place by recursively decomposing the benchmark architecture to smaller building blocks and testing them individually at HDL level. This is an essential modification of the QUEENS-FPGA methodology [6], which allows for in depth investigation of various optimization issues and/or errors which otherwise, would be very difficult to be detected at higher level.

The exploration with the building blocks is performed at the phase of "Programming-Level Tuning", which investigates the capability of the synthesizer to efficiently map the HDL code on the underlying NG LARGE architecture. In this phase, we use standard template-based coding and attributes/directives to express memories, FSMs,

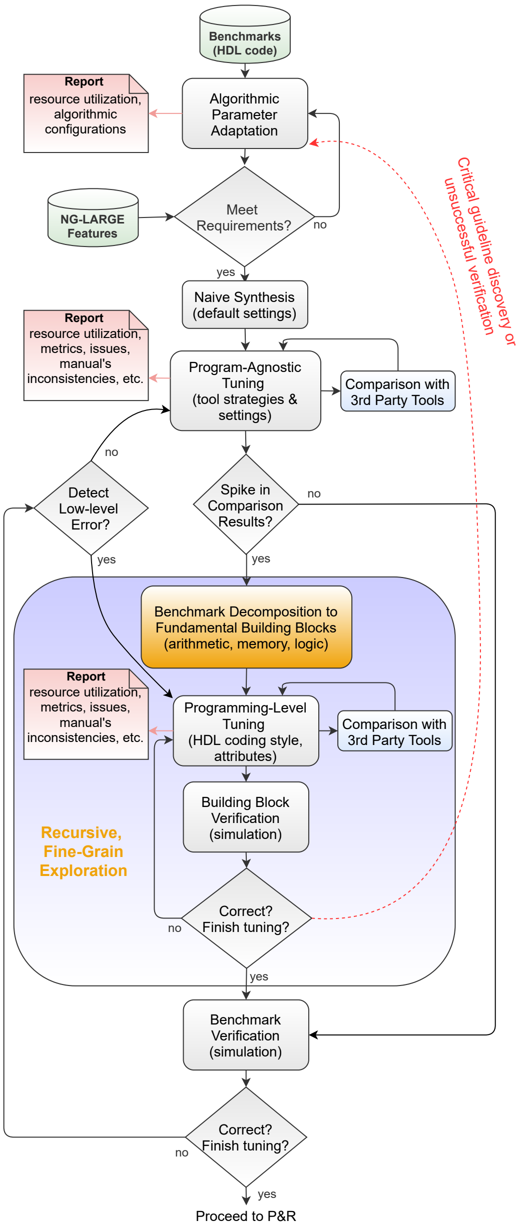


Figure 2: The assessment methodology for Synthesis.

multipliers, etc. If we identify a type of HDL coding that leads to improved results, or notice inability of the design to fit in NG-LARGE in spite of our efforts, we use our feedback loop (red dashed line) to re-customize the algorithmic parameters and proceed with a new benchmark configuration. Similarly to the previous phase, every exploration is followed by systematic comparison with the 3rd party tools.

In both main exploration phases, we keep records of the examined metrics, i.e., resource utilization, provided features, efficiency of the existing Synthesis settings, proposed design guidelines, as well as the detected issues. Moreover, both phases include functional verification via simulations with 3rd party simulators (ModelSim/QuestaSim). Specifically, the post-Synthesis netlists of the benchmarks and/or the basic building blocks are simulated, and the outputs are compared to the ground-truth data obtained by the RTL/behavioral simulation.

2.4. Assessment of Place & Route

A similar assessment methodology is designed for the P&R process. This methodology takes as input the post-Synthesis netlist of the benchmark or the netlist of a small building block with problematic behavior. The evaluation of the building blocks as standalone components is another addition to our initial methodology for the P&R assessment [6]. It is very important, as it enables the isolation and testing of individual components, which may result in malfunction of the entire benchmark when tested on HW. For both cases, we evaluate the P&R metrics (resources, estimated performance, power etc.) in a sequential fashion, by exploring the various settings and physical constraints of the NXmap tool. Our main exploration/evaluation procedure consists of two phases.

Firstly, we evaluate all the options for Placement. In this phase, we assess the capability of the tool to perform location specific placements via constraints regarding the targeted region, placement of groups, etc., either in at fine-grain (e.g., LUTs, DFFs) or coarse-grain (e.g., Tiles, DSPs, RAMBs) level. Furthermore, we examine the quality of the reports and the efficiency of the available Placement settings, such as the placing effort.

Subsequently, we evaluate all the Routing-related settings. In this phase, we examine if the tool is capable of delivering efficient solutions under stressing the implementation towards performance and/or increased routing congestion. Our exploration is driven by various timing constraints (e.g., timing driven, set false path, set max delay, create clock) and router's settings (e.g., router effort, router mode). We note that we apply different Placement constraints from the previous phase. Also, in this phase we assess the Static Timing Analysis (STA) reports.

Similarly to our Synthesis assessment methodology, every experimentation is accompanied by; (i) systematic comparison with the 3rd party tools, (ii) functional and timing verification via post-Place and post-Route netlist simulations, and (iii) floorplan inspection. Moreover, considering that the Synthesis and P&R processes are tightly coupled (different post-Synthesis netlists may lead to different P&R results), we explore various scenarios by combining tool settings from both stages.

2.5. Assessment of Bitstream Generation

The final step of our methodology is to evaluate the Bitstream Generation, as well as the FPGA Programming. Specifically, in this evaluation phase, we examine the correct bitstream generation for all the relevant tool options, the bitstream size, the programming speeds via the available configuration interfaces (JTAG, SpaceWire, EPROM), and the correctness of the actual HW execution on NG-LARGE. The latter is performed by establishing I/O communication between FPGA and host-PC (e.g., via UART, SpaceWire) and comparing the outputs of the FPGA with ground-truth data obtained from behavioral or post-P&R NanoXplore netlist simulations.

Table 1: Algorithmic Configuration of Benchmarks

	Data			Mask	
	In. Size	**In. Partition**	**I/O Bits**	**Size**	**Bits**
FIR	$N\times1$	contin.	16/16	64×1	16×16
Harris	1024×1024	1024×32	8/32	7×7	8×14
Canny	1024×1024	contin.	8/4	3×3	8×3
Disparity	1024×1024	1024×32	8/10	7×7	8×7
SpaceSweep	1024×1024	1024×16	8/32	13×13	8×8

3. BENCHMARKING ON NXMAP & NG-LARGE

The benchmarking evaluation results are produced by NXmap3 v2020.3. The selected benchmarks are configured as shown in Table 1 (e.g., Harris inputs a 1024×1024 8-bit image partitioned in 1024×32 pixel stripes and performs convolutions with 7×7 14-bit kernels to output 32-bit corners). Our evaluation is performed at two levels: (i) at SW level, by evaluating the NXmap's options and examining the resource utilization and the tool requirements, (ii) at HW level, by evaluating the chip's maximum frequency, the power consumption, the benchmarks' throughput and the FPGA configuration times.

The functional verification of the benchmarks was performed via post-Synthesis and post-P&R simulations on realistic datasets. A natural signal sampled at 110K samples/sec with 16 bits for FIR, 1024×1024 synthetic stereo images depicting a rover's view on Martian terrain were employed for Disparity and SpaceSweep, 1024×1024 images depicting rocky Martian terrains were used for Harris and Canny. The derived results were compared with the RTL behavioral simulation and revealed a fully functional error-free operation. All benchmarks were also tested on the NG-LARGE HW utilizing a serial UART communication for transmitting the input data and receiving the results.

3.1. Evaluation Results: NXmap SW Tool

Regarding the Synthesis process, Table 2 presents the resource utilization when using the default configuration/options of the NXmap tool. With this setup, none of the DSPs are employed for FIR, as all the arithmetic operations are mapped onto CY units (61% utilization). Similarly, most of the multiplications of Harris are not mapped to DSPs, but onto CY units. To balance the utilization, we used our Synthesis exploration procedure

Table 2: Synthesis Resource Utilization on NG-LARGE with Default NXmap Settings

	LUT	DFF	CY	DSP	RF	RAMB
FIR	0 (0%)	7136 (6%)	19440 (61%)	0 (0%)	0 (0%)	0 (0%)
Harris	6210 (5%)	16398 (13%)	13794 (43%)	27 (8%)	0 (0%)	69 (36%)
Canny	1845 (2%)	2348 (2%)	1167 (4%)	2 (1%)	0 (0%)	177 (93%)
Disparity	1000 (1%)	3628 (3%)	4548 (15%)	4 (2%)	0 (2%)	85 (45%)
SpaceSweep	5500 (5%)	10222 (8%)	6277 (20%)	50 (14%)	8 (2%)	74 (39%)

Table 3: Synthesis Resource Utilization on NG-LARGE with Tailored NXmap Settings

	LUT	DFF	CY	DSP	RF	RAMB
FIR	2 (1%)	7170 (6%)	1008 (4%)	64 (17%)	0 (0%)	0 (0%)
Harris	6110 (5%)	15304 (12%)	7112 (23%)	81 (22%)	0 (0%)	69 (36%)
Canny	1845 (2%)	2299 (2%)	1086 (4%)	4 (2%)	0 (0%)	177 (93%)
Disparity	1000 (1%)	3628 (3%)	4548 (15%)	4 (2%)	0 (2%)	85 (45%)
SpaceSweep	5499 (5%)	10222 (8%)	6277 (20%)	50 (14%)	0 (0%)	79 (42%)

Table 4: P&R Resource Utilization on NG-LARGE with Default NXmap Settings

	LUT	DFF	CY	DSP	RAMB	MHz
FIR	0 (0%)	7136 (6%)	19440 (61%)	0 (0%)	0 (0%)	214
Harris	6205 (5%)	16516 (13%)	13794 (43%)	27 (8%)	69 (36%)	31
Canny	1844 (2%)	2412 (2%)	1167 (4%)	2 (1%)	177 (93%)	35
Disparity	994 (1%)	3664 (3%)	4548 (15%)	4 (2%)	85 (45%)	47
SpaceSweep	5493 (5%)	10280 (8%)	6277 (20%)	50 (14%)	74 (39%)	51

Table 5: P&R Resource Utilization on NG-LARGE with Tailored NXmap Settings

	LUT	DFF	CY	DSP	RAMB	MHz
FIR	0 (0%)	7136 (6%)	19440 (61%)	0 (0%)	0 (0%)	214
Harris	6105 (5%)	15413 (13%)	7112 (23%)	81 (22%)	69 (36%)	40
Canny	1843 (2%)	2349 (2%)	1086 (4%)	2 (1%)	177 (93%)	38
Disparity	998 (1%)	3672 (3%)	4548 (15%)	4 (2%)	85 (45%)	50
SpaceSweep	5458 (5%)	10276 (8%)	6277 (20%)	50 (14%)	79 (42%)	52

(see Section II) to customize the tool options according to the requirements of each benchmark. More specifically, we shared the arithmetic operations between the DSPs and CYs to balance their utilization. The results of our customization decisions are reported in Table 3. The employment of DSPs in FIR decreased the CY utilization from 61% to 4%, providing a 17% increase in DSPs. For Harris, we get a balanced usage between CY and DSP blocks, i.e., from 43% and 8% to 23% and 22%, respectively. The Harris and Canny benchmarks achieve better timing performance when the multiplication operations are forced to be mapped onto DSPs, but FIR, Disparity, and SpaceSweep achieve better timing when the default mapping is used. We used this custom mapping on the FIR and observed a 44% decrease in the achieved frequency. For Disparity and SpaceSweep, the decrease was only 1MHz and 0.65MHz, respectively, so during the P&R evaluation, we decide to remove the custom mapping and affect only the P&R-related options.

Similarly, we used our P&R exploration procedure (see Section II) to experiment with the tool options and achieve the best possible timing performance. Table 4 reports the P&R resource utilization for the implementations with the default P&R option values, while Table 5 reports the results for the customized P&R option values. For the memory and arithmetic operations (RFs, RAMBs, CYs, DSPs), the resources remain equal to that of Synthesis results. Ultimately, we observe that, with exception of the FIR benchmark, all the other benchmarks can achieve better timing performance by changing some of the default P&R option values. For all the benchmarks, in terms of resources, the variations between the available tool configurations (*DensityEffort, CongestionEffort, PolishingEffort, RoutingEffort, and BypassingEffort*) are negligible (±10 LUTs). In terms of maximum frequency, which is reported by the tool's static timing analysis, it is possible to achieve better results by affecting one or more P&R option values, depending on the features of the benchmark. For Harris, the *PolishingEffort* option was set to ”low” rather than ”medium”, giving an increase of 9.5MHz. For Disparity, the *PolishingEffort* option was set to ”low” rather than ”medium” and the *DensityEffort* to ”medium” rather than ”low”, delivering an increase of 2.6MHz. For Canny, the *PolishingEffort* option was set to ”high” rather than ”medium”, providing an increase of 2.7MHz. For SpaceSweep, the *CongestionEffort* option was set to ”medium” rather than ”high”, delivering an increase of 0.7MHz. We also notice that for FIR, the tool achieves almost the double maximum frequency, i.e., 214MHZ from 121MHz, when we do not employ a custom mapping for the arithmetic operations.

Overall, with respect to the reported resource utilization, for FIR, the NXmap tool correctly omits the RAMBs for storage purposes, because FIR describes a deeply pipelined filter with a big sequence of registers. Moreover, the tool correctly occupies 64 DSPs (when we employ that mapping directive), which coincide with the number of filter's 64 taps/coefficients. Regarding Harris, Canny, Disparity, and SpaceSweep, several of the available RAMB configurations are employed, i.e., 24K×2,

Table 6: NXmap Tool Requirements for the Benchmark Implementation on NG-LARGE

	Synthesis Runtime (s)	P&R Runtime (s)	Total Runtime (s)	Peak Memory (KB)
FIR	8	55	118	1099
Harris	141	270	577	1576
Canny	1334	104	1521	1225
Disparity	47	117	287	1291
SpaceSweep	95	158	382	1445

System: Intel Xeon E5-2650 @2.60GHz ×16, 64GB RAM

12K×4, 2K×24, etc., and thus, reasonable RAMB utilization is derived for 1024-pixel-wide images (36%, 93%, 45% and 42%, respectively).

In terms of resource requirements, NXmap is a lightweight tool as shown by the overall runtime and memory usage (Table 6). Apart from FIR, all the other benchmarks are very demanding, but even for those, both runtime and peak memory usage remain in relatively low levels, manageable even by low-end CPUs. We also notice that even the elapsed time (real-world time) for the entire process up to the bitstream generation is just a couple of minutes (<6). Canny is the only benchmark that required several minutes (25), and the reason is the high RAMB utilization that is at 93%.

3.2. Evaluation Results: Comparison to 3rd Party

The comparison between NanoXplore and the 3rd part vendors shows that the BRAVE results are promising. Depending on the benchmark, NXmap provides comparable P&R resource utilization for some primitives, and even better in some cases.

For the Harris benchmark, NXmap provides a good LUT utilization, i.e., 3.2× less LUT vs the reference value. When considering the pass-thru LUTs from the CY utilization, the total number of LUTs increases. The LUT utilization should be examined along with the number of employed DSPs, where NXmap utilizes 1.5× less. Regarding the RAM resources, NXmap delivers less RAMBs (it has larger RAMB size) and the total RAMB kbits are less to the reference value. We note that the maximum clock frequency is less than the reference value, however, there is a small increase vs the frequency the previous tool versions. For Canny, NXmap provides average LUT utilization with an increase of 6%, but if we also consider the route-thru LUTs from the CYs the utilization, it is increased by 48%. The DFF utilization is increased by almost 50%, but it is still within acceptable limits. The DSP and the RAMB utilization is excellent, as NXmap provides the same number of resources.

For the Disparity benchmark, NXmap provides promising LUT utilization, as it employs a small number even when considering the CY resources. Regarding RAM resources, NXmap is below the reference value in both blocks and kbits. In terms of frequency, NXmap provides less MHz, however, this value was again increased by almost 15% vs the previous tool version (tool is improving). For SpaceSweep, NXmap also provides a very good LUT utilization. It achieves better results by 52% and drops to comparable results when considering the CYs, with just a 3% LUT increase. The DFF utilization, it is also better by 5%. Finally, the DSP and RAMB utilization is excellent, as NXmap outperforms the average values of the other tools by 20% and 30%, respectively.

Table 7: Maximum Frequency & Throughput on NG-LARGE with Tailored NXmap Settings

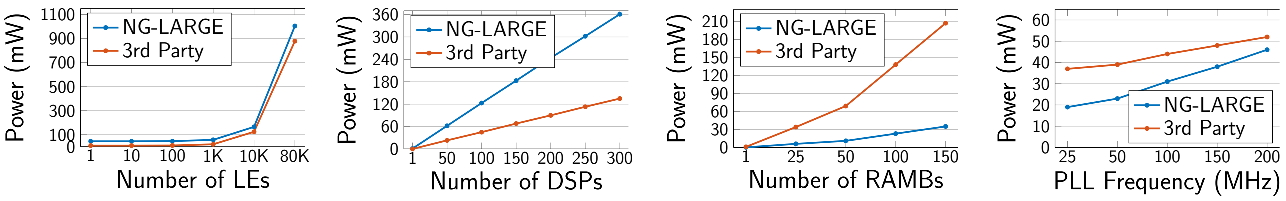
	Frequency (MHz)	Runtime (s)	Throughput (*)
FIR	214	continuous	214 MSPS
Harris	40	0.19 / frame	5.3 FPS
Canny	38	0.10 / frame	10 FPS
Disparity	50	6.7 / frame	18 MPDS
SpaceSweep	52	10.8 / frame	29 MPDS

* Throughput excludes I/O and differs per benchmark: MSPS = mega samples per second, FPS = frames per second, MPDS = mega pixel disparities per second

3.3. Evaluation Results: NG-LARGE HW Board

Table 7 presents the maximum frequency of the benchmarks. Accordingly, a throughput metric is presented for each benchmark to highlight the potential of NG-LARGE for real-time operation. Overall, we note that NG-LARGE provide sufficient resources and frequency to all benchmarks. The achieved throughput of FIR can support a multitude of applications, e.g., for telecomm, while its resource utilization allows even for complementary VHDL components to be placed in the chip. The time required for a complete reconstruction using Disparity and SpaceSweep could improve the conventional depth extraction of Mars rovers by an order of magnitude (in terms of resolution and speed). We note that, in the tested configuration, SpaceSweep examines 3× depth levels and it provides much higher accuracy than Disparity. Furthermore, given that most VBN applications require 1–10 FPS, we conclude that Harris' and Canny's throughput leaves enough room for the complementary components of an algorithmic chain to finish on time.

Next, we evaluate the power consumption of NG-LARGE by comparing it to one of the most prominent space-grade FPGAs (labeled as "3rd party device"), which has similar technology node and resources. The static power consumption has been measured when the FPGAs are powered up and no bitstream is loaded, using the physical components and chipscope tools. The results are similar for both devices, i.e., 1.99W for NG-LARGE and 1.91W for the 3rd party device. For the dynamic power consumption, which mainly depends on the number of utilized resources, the clock frequency and



(a) (b) (c) (d)

Figure 3: Dynamic power consumption of NG-LARGE and 3rd party device with respect to the (a) LE, (b) DSP & (c) RAMB utilization, and (d) the generated clock frequency of PLL.

the toggle rates, we have performed various experiments in the same environment conditions using the provided power analyzer tools. The derived results, discussed in the next paragraph and illustrated in Figure 3, show that NG-LARGE provides comparable dynamic power, and in some cases, even lower compared to the 3rd party device.

Figure 3a shows the dynamic power consumption of the Logic Elements (LEs) (each LE of NG-LARGE consists of 1 LUT and 1 DFF). NG-LARGE delivers 5× higher power consumption than the 3rd party device. Nevertheless, this difference decreases for bigger LE utilization, and specifically, it is reduced up to 1.1× when almost all the LEs are utilized. Figure 3b reports the power consumption for different DSP utilization. In this case, NG-LARGE consumes 2.6× higher power than the 3rd party device. We note that the DSPs of both FPGAs have similar architecture and operation word-length. Figure 3c illustrates the scaling of the power consumption with respect to the RAMB utilization. It is important to notice that NG-LARGE provides memory blocks of 48Kbits, while the 3rd party device provides smaller blocks. Despite that, NG-LARGE delivers 6× lower power compared to the 3rd party device. For all the aforementioned experiments, we have used a clock frequency of 25MHz. The last step was to examine the power consumption of the PLL, when assigned to generate different clock frequencies (the input frequency is 25MHz). In Figure 3d, we observe the lower power consumption values of NG-LARGE compared to 3rd party device. Nevertheless, the 3rd party device shows smaller increases (0.08mW/MHz) compared to NG-LARGE (0.16mW/MHz). This implies that NG-LARGE provides high power efficiency for low frequencies, which deteriorates for higher frequencies, but is still better than the 3rd party device.

Finally, in Table 8 we report the bitstream size of each benchmark, and the time required for the NG-LARGE to be configured via JTAG. As shown, the configuration time of NG-LARGE is almost proportional to bitstream size: 384 KBytes per second are handled for FIR, 389 KBytes for Harris, 452 KBytes for Canny, 381 for Disparity and 399 for SpaceSweep. Also, we observe that Canny, which has 93% RAMB utilization, results with a large bitstream, i.e., around 1KB larger than the rest of the benchmarks, and thus, it requires more time (2-3 extra seconds) to be configured.

Table 8: Bitstream Configuration Results on NG-LARGE

	Bitstream Size	**JTAG Config. Time**
	(KB)	(s)
FIR	960	2.5
Harris	1751	4.5
Canny	2669	5.9
Disparity	1563	4.1
SpaceSweep	1719	4.3

System: Intel Core i7-4500U @1.80GHz ×4, 8GB RAM

4. SYSTEM-LEVEL TESTING: VBN PIPELINE

In this section, we evaluate NG-LARGE when implementing a VBN system used for Rover localization tasks. More specifically, we present some preliminary results from the implementation of ”SPARTAN VBN2” on NG-LARGE, which is custom optimization of a vision-based autonomous navigation system from past ESA activities and codes [2]. The entire system includes SpaceWire communication interfaces to connect the image processing board (i.e., the FPGA), with an on-board computer, and a telemetry and telecommand CODEC IP based on CCSDS Space Packet standard and PUS services. In terms of image processing algorithms, the Harris Corner Detector and the SIFT Descriptor are implemented in VHDL, along with their arbiters to control their execution. For comparative purposes, we implement the same system on a prominent 3rd party space-grade FPGA with similar resources and technology node. We also note that the input data are pairs of 512×512 stereo images.

Table 9 reports the resource utilization and the maximum clock frequency of the two FPGAs. Due to a different LE and LUT architecture (the 3rd party device integrates 4× more LUTs in a LE, and also has 6-input LUTs), NG-LARGE results in 6.5× and 2× higher LE and LUT utilization, respectively. Nevertheless, when considering the total number of available LEs and LUTs in both devices, we observe that the utilization percentage of LEs and LUTs is only 3% and 9% higher, respectively. Similarly, NG-LARGE utilizes 1.4× more DFFs, however, its utilization percentage is better by 4%. Regarding the DSPs, NG-LARGE delivers 2× higher utilization, but this is due to mapping arithmetic operations onto DSPs to save logic

Table 9: Resources of ”SPARTAN VBN2” System

	LE	LUT	DFF	DSP	RAMB	MHz
NG-LARGE	98279 (76%)	98279 (71%)	56296 (44%)	250 (65%)	113 (58%)	22
3rd Party	14894 (73%)	50427 (62%)	39008 (48%)	129 (40%)	228 (77%)	30

Table 10: Performance of ”SPARTAN VBN2” System

	Harris	**SIFT**	**SpW I/O**	**Total System**	
	Time[1]	**Time**[1]	**Time**[1]	**Time**[2]	**Throughput**[2]
NG-LARGE	208ms	395ms	28ms	1251ms	0.8 FPS
3rd Party	104ms	196ms	28ms	624ms	1.6 FPS

[1] refers to one 512×512 image.

[2] refers to a localization step with one 512×512 stereo pair.

resources. In contrast, NG-LARGE utilizes half of the RAMBs employed by the 3rd party device. This is explained by the bigger storage capacity of NanoXplore’s RAMBs. Finally, in terms of maximum clock frequency, both devices achieve comparable MHz (there is a small difference of 8MHz).

Table 10 summarizes the performance results for the two implementations. The system clock of each FPGA is configured according to the maximum frequency reported by the tool, hence, at 12.5MHz in NG-LARGE and 25MHz in the 3rd party device. The first two table columns report the execution times of the Harris and SIFT algorithms for one of the pair’s stereo images. As expected (due to having 2× faster clock), the 3rd party device demonstrates around 2× less processing time. We note that the execution time varies among different input images, as it is affected by the number of features detected. Regarding the I/O time via SpaceWire configured at 100mbps, it is around 28ms for both implementations (1.75ms for the transmission of each one of the 16 image bands). The last two columns report the performance of the entire system (I/O + FPGA processing) for a pair of stereo images. Again, as expected, the 3rd party device provides 2× throughput. In any case, the NanoXplore provides comparable performance, which is improved as the EDA tools become more mature.

5. CONCLUSION & FUTURE WORK

In this work, we presented a custom methodology for evaluating the new European NG-LARGE FPGA and associated EDA tools, which is formed according our experience with the NanoXplore tools and devices. The experimental evaluation, which is driven by HDL benchmarks from past ESA activities, involves various results regarding the SW tool and the FPGA chip, as well as discussions about their comparison to 3rd party vendors. NG-LARGE can effectively implement high-performance designs with sufficient resource utilization, throughput, and power consumption, which are all comparable to the 3rd-party solutions. Our future work includes: (i) benchmarking on BRAVE with HDL IPs from other fields, e.g., telecommunications, (ii) testing and exploration on the new versions of the SW tools, and (iii) evaluation of NG-ULTRA, i.e., the next FPGA of the BRAVE series, which will also include the embedded processor.

ACKNOWLEDGMENTS

This work was supported by the European Space Agency via the QUEENS2 activity #4000128041/19/NL/AR/va. The authors would like to thank the NanoXplore team for the support throughout the activity.

REFERENCES

[1] https://www.nanoxplore.com.

[2] G. Lentaris et al. SPARTAN/SEXTANT/COMPASS: Advancing Space Rover Vision via Reconfigurable Platforms. In *Int’l. Symp. on Applied Reconfigurable Computing*, pages 475–486. Springer, 2015.

[3] G. Lentaris et al. High-Performance Embedded Computing in Space: Evaluation of Platforms for Vision-Based Navigation. *Journal of Aerospace Information Systems*, 15(4):178–192, Feb. 2018.

[4] G. Lentaris et al. High-Performance Vision-Based Navigation on SoC FPGA for Spacecraft Proximity Operations. *IEEE Trans. on Circuits and Systems for Video Technology*, 30(4):1188–1202, 2020.

[5] V. Leon et al. Improving Performance-Power-Programmability in Space Avionics with Edge Devices: VBN on Myriad2 SoC. *ACM Trans. on Embedded Computing Systems*, 20(3), Mar. 2021. doi: 10.1145/3440885.

[6] K. Maragos et al. Evaluation Methodology and Reconfiguration Tests on the New European NG-MEDIUM FPGA. In *NASA/ESA Conf. on Adaptive Hardware and Systems (AHS)*, pages 127–134, 2018.

[7] J. L. Mauff. NX RHBD FPGA Solutions for OBDP Applications. In *European Workshop on On-board Data Processing*, pages 1–34, 2019.

[8] A. Perez et al. Run-Time Reconfigurable MPSoC-Based On-Board Processor for Vision-Based Space Nvigation. *IEEE Access*, 8:59891–59905, 2020.

[9] F. Rittner et al. Broadband FPGA Payload Processing in a Harsh Radiation Environment. In *NASA/ESA Conf. on Adaptive Hardware and Systems (AHS)*, pages 151–158, 2014.