

B.0 – Fabric Analysis Benchmark

General Description

The purpose of this fabric characterisation benchmark is to provide a simple, fast and application-independent estimation of the power consumption in fine-grain FPGAs and characterise their power consumption under different scenarios. The power scenarios considered are active processing, inactivity and dedicated sleep or low-power states. These power scenarios are inspired by the different real-world scenarios that a device could be operated in. Rather than measuring static and dynamic power, we define processing scenarios by specifying the behaviour of so-called activity modes and measure the power in these modes. The benchmark is extensible by allowing to specify additional activity modes that are based on the characteristics of the device's low-power modes. The benchmark uses a pseudo random number generator (RNG) as a synthetic test circuit to stress the device and emulate intense active processing. The actual benchmark circuit size is scaled to the target device size and results are normalised to device size and clock frequency in order to allow the comparison of different devices. Further aspects of this benchmark are the thermal properties of the device. We characterise how the devices heats up under different activation levels. In a second test scenario, the temperature dependency of static power is measured directly by heating up the device externally.

Benchmark Core: Random Number Generator

The benchmark is based on a pseudo random number generator (RNG) as described in [1]. This RNG uses binary linear recurrences where each bit of the next state is generated based on a linear combination of the current state. Compared to linear feedback shift registers (LFSR), the most common type of pseudo random number generators, this improves quality of the random numbers. For power benchmarking purposes, this also has the benefit of yielding a circuit where LUTs are heavily interconnected. LUT placement cannot be optimised and hence, the circuit exercises all different kinds of short and long wires of the routing fabric. The circuit also does not provide any opportunity for logic optimisation. The lack of optimisation potential is an important aspect to reduce the influence of the implementation tools on the result of our characterisation. The RNG circuit is also characterised by a high and uniformly distributed toggle rate of 50%. Because of its high toggle rate and its dense interconnect, the circuits is capable of causing a very high activity in the device. The RNG has 512 taps and should map to 512 LUT and flip-flop pairs. The RNG circuit has a parameter *width* which specifies the number of inputs per tap. This parameter should match the LUT size of the target device so that each tap maps to one LUT exactly. The RNG circuit is provided as HDL source with this benchmark.

Benchmark Hardware Implementation

One RNG benchmark core maps to 512 LUT / flip-flop pairs. In order to allow benchmarking and comparison of devices of different sizes, multiple instances of the RNG core are implemented so that the 60% of all available LUT / flip flops resources are utilised. A logic utilisation of 60% with this highly active circuit is high enough to cause substantial active power consumption that can be considered worst-case for most real applications. On the other hand it is lax enough to maintain routability of this very dense circuit.

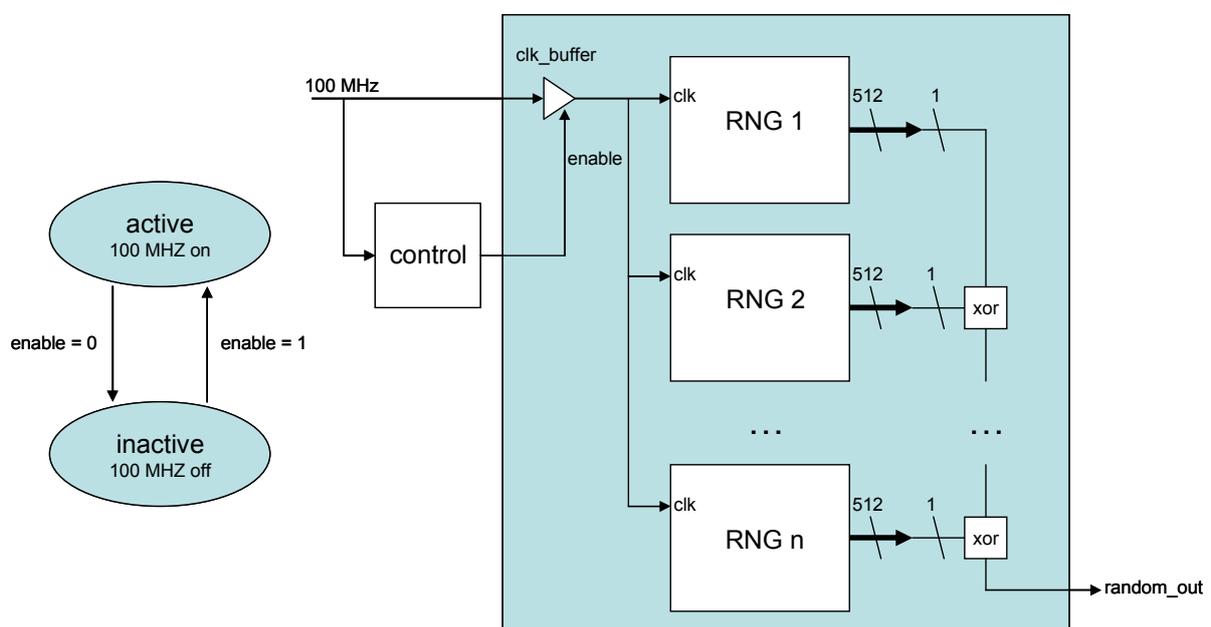


Figure 1 - Block diagram of a possible implementation of the benchmark circuit.

Figure 1 illustrates a possible implementation of the benchmark circuit. The device is filled with multiple instances of the RNG core to achieve a logic utilisation of 60%. One output pin of each RNG is fed into an XOR chain to avoid the cores being optimised away by the implementation tools. The output of the XOR chain is connected a device output. However, this is only one example of how to avoid logic optimisation. The outputs of the RNGs are the not important for the purpose of this benchmark and do not have to be used or verified. However, it should be verified that the benchmark circuit is implemented and functioning properly. It is also recommended to initialise all RNGs instances with different seeds to avoid possible optimisations by the synthesis tools. The benchmark circuit should be driven with a clock frequency that is realistic or representative for the target device or target application. As a reference point, 100 MHz can be used in most cases as this is supported by most commercial devices. However, if this is too unrealistic a more suitable clock rate that is representative of

the targeted domain may be chosen. For example, 32 MHz might be more appropriate for low-power devices targeting the mobile market. For high performance parts, the frequency will most likely be higher. Section “Benchmarking and Reporting” explains how different clock frequencies are dealt with in the benchmark report. Figure 1 also illustrates how the clock can be enabled by an external controller through and FPGA clock buffer. In many current devices, clock gating is the most appropriate way of switching the circuit between “active” and “inactive” modes. Details of this implementation however can depend on the particular target device. This is further explained in the next section.

Activity Modes

A key aspect of this benchmark is to characterise the power consumption of a device in different activity modes. These activity modes specify how the device behaves in a certain mode rather than by which means this mode is implemented. For this benchmark, measuring the power within the context of a certain behaviour is more meaningful than measuring dynamic and static power.

The two basic modes that are applicable to all devices are *active* and *inactive*. In *active* mode, the test circuit continuously generates random numbers at its given clock frequency. The power consumption in this mode is a combination of static and high dynamic power. In *inactive* mode, the circuit does not generate random numbers. However, its state must be preserved and it also must be able to switch back into *active* mode instantly. These are the most basic activity modes and the *inactive* mode can usually be implemented with a simple clock-gating approach as illustrated in figure 1. However, we are only concerned about the power consumption of the inactive device with preservation of state and instant wake-up capability and not the details of its technical implementation. Depending on the implementation details of the *inactive* mode there might be supporting circuitry such as clock managers that are still operating and drawing dynamic power. It is important to point out that the focus of this benchmark is not measuring pure static power but rather the minimal power necessary to implement the *inactive* mode. Only FPGA-internal resources should be used, i.e. do not use external clock buffers.

In order to compare devices with advanced low-power modes, we allow the specification of further activity modes. The behaviour of these modes can be defined according to the device’s capabilities and the power consumption in these modes can then be compared against the two basic modes. Table 1 illustrates an example with the two basic activity modes *active* and *inactive*, and two hypothetical user-defined modes *sleep* and *hibernate*. The behaviour of the basic modes is fixed, while the behaviour of the advanced modes is based on the technical details of the target device.

Warning: In *active* mode, some devices can consume unusually high amounts of power. This can lead to significant heat development and potentially even damage the device if run for longer periods. It is therefore recommended to monitor the device temperature when the benchmark circuit is active.

	activity mode			
	standard		device-specific	
	active	inactive	sleep	hibernate
generate output	yes	no	no	no
retain state	-	yes	yes	no
wakeup time	-	instant	500 μ s	50 ms

Table 1 – Example of activity modes. The first modes are fixed, further modes can be defined based on the device capabilities.

Benchmarking Method

This section explains how the benchmark is performed. This includes how to measure power, how to perform the benchmark and report results and how to use a well defined temperature to measure the device temperature.

Power Measurement

All power results in this benchmark should report FPGA core power only. FPGAs usually provide separate voltage rails for the core logic and IOs. IO power consumption is heavily dependent on the IO configuration and loads attached to the IOs. IO power consumption is therefore of no concern for this benchmark. Some FPGAs provide separate voltage rails for core logic and supporting internal circuitry such as clock managers. All of these are considered to contribute to core power and the reported number should therefore be the combination of the power consumption on all these voltage rails. However, a benchmark user may additionally report a breakdown of the total core power into its components. For example, it might be interesting to point out that the logic fabric itself is very power efficient in inactive mode while some auxiliary circuits on a separate voltage rail continue to draw significant amounts of power.

Power measurements with good accuracy can be taken by inserting a precision current measurement resistor into a voltage rail and determining the current on this rail by measuring the voltage drop over the resistor. The resistor has to be selected carefully so that the voltage drop does not cause the input voltage on the device to drop below the minimum allowed value. Alternatively, a dedicated power supply with integrated current or power measurement facilities can be used.

Benchmarking and Reporting

There are two different benchmarking methods for cold and hot devices. All measurements are performed in an ambient temperature of 25°C. If the maximum temperature of the device in continuous active mode does not exceed 35°C then the device is considered as “cold” and the benchmark does not further consider the influence of device temperature. If the device exceeds this temperature, a more complex benchmarking method is used that characterises the influence of circuit activity on the device temperature.

Measuring cold devices

If the device temperature in continuous active mode does not exceed 35°C then the device is benchmarked by simply measuring the power in all activity modes separately. The results are reported as absolute numbers and also as normalised to the number of LUT/flip-flop pairs. This is further illustrated in table 2.

Active power	Relative active power	Inactive power	Power in dedicated low-power mode
Absolute value of power consumption in active mode. Report in combination with clock frequency used and with device temperature	Active power relative to clock frequency. This value is calculated as (active power minus inactive power) divided by clock frequency.	Absolute value of power consumption in inactive mode.	Absolute value of power consumption when in a dedicated low-power mode, e.g. “sleep”. Depends on the availability of low-power modes as explained in table 1.
The above value normalised to the number of LUT/flip-flop pairs in the device.	The above value normalised to the number of LUT/flip-flop pairs in the device.	The above value normalised to the number of LUT/flip-flop pairs in the device.	The above value normalised to the number of LUT/flip-flop pairs in the device.

Table 2 - Reporting power consumption for “cold” devices in absolute and normalised values.

The first column displays the power consumption in active mode. This value should be reported in combination with the temperature that the device reaches during continuous active processing and the clock frequency that was used to drive the benchmark circuit. The second column presents the active power normalised to clock frequency. The value is calculated as active power minus inactive power divided by clock frequency. This allows estimating how the power in the device scales with different clock frequency and also enables the comparison with devices that were benchmarked at different clock frequencies. In the third column, the power in inactive mode is reported. If the device supports additional low-power as illustrated in table 1, then the report can be amended with additional columns to report the power in these modes. Power measurements in inactive and all additional low-power modes should be taken at a device temperature of 25°C, i.e. before the device is warmed up by active processing or after it has cooled down sufficiently. Table 3 shows an example report for a hypothetical device with 10k LUTs /flip-flops.

Active power	Relative active power	Inactive power	Sleep power	Hibernate power
1.2 W (100 MHz, 31°C)	11 mW / MHz	100 mW	17 mW	240 µW
120 µW / LUT	1.1 µW / LUT · MHz	10 µW / LUT	1.7 µW / LUT	24 nW / LUT

Table 3 - Example of a power report for a device with 10k LUTs/FFs

Measuring hot devices

If the device temperature exceeds 35°C in active mode then a more complex characterisation should be performed. This characterisation shows how the device temperature depends on the level of activity in the device. Since static power consumption is highly dependent on temperature and inactive power is most likely to be dominated by static power, we also want to characterise how inactive power degrades with higher activity. In order to adjust the level of activity in the device we periodically switch the device between active and inactive mode with various duty cycles. Measurements of temperature, active and inactive power are taken for all duty cycles from 0% to 100% in 5% increments. To take a measurement for one given duty cycle, the benchmark user should first wait until the device temperature reaches its final value. This is illustrated in figure 2. Once the temperature has settled to its final value, power measurements are taken. Active power is not measured as average active power but as

instantaneous active power during the “on” phase of the duty cycle. The inactive power is measured during the “off” phase.

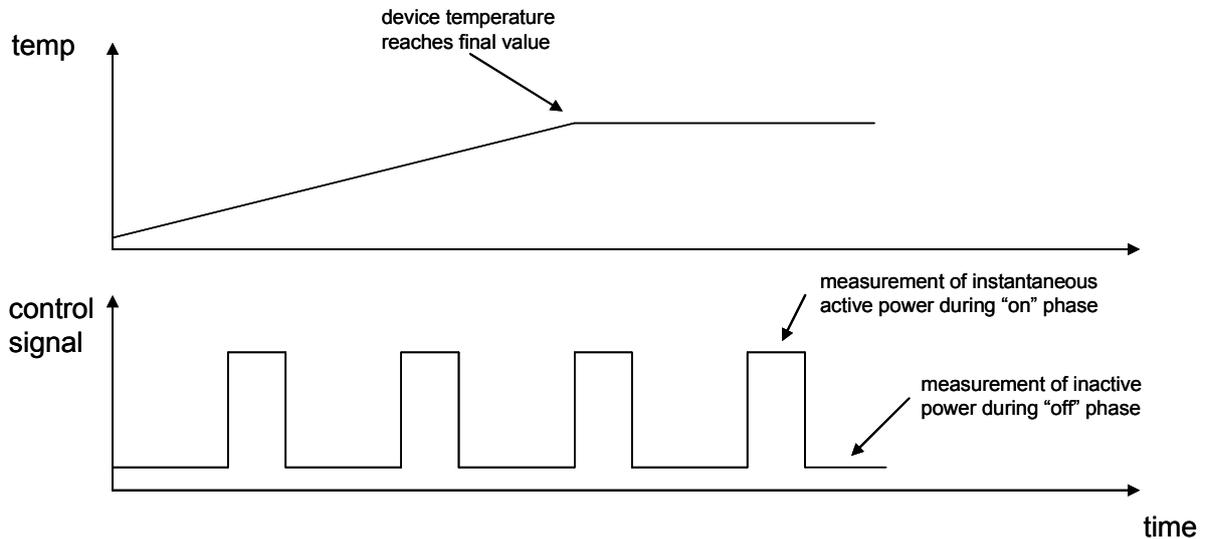


Figure 2 - Temperature and power measurements

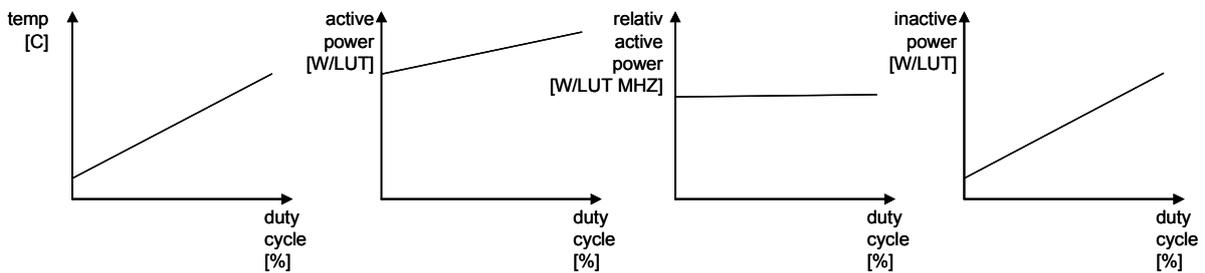


Figure 3 – Reporting temperature and power in a diagram over duty cycle.

Results for temperature, active and inactive power should be reported in diagram from over duty cycle as illustrated in figure 3. The report should also mention the clock frequency used to run the benchmark circuit. Similar to the method for cold devices, the relative active power should be calculated by subtracting inactive from active power and dividing it by clock frequency. All results should be reported as absolute and LUT-normalised values. As illustrated in figure 3, there is probably a strong dependency of inactive power on temperature. Active power is usually dominated by dynamic power and therefore less dependent on temperature. Temperature will most likely show little to no influence on relative active power. If the device supports further low-power modes, they should be measured as well. Measurements for these advanced modes are taken without any duty cycle at 25°C.

Environment

In order to minimise the environmental influences on the measurement the ambient temperature should be fixed to 25 °C. The device should not use any heatsinks or active cooling. The airflow should be reduced to natural convection around the device. This can be achieved placing the board in an open cardboard enclosure. The walls reduce the flow of surrounding air while the open top stops heat from building up inside the enclosure. If the ambient temperature cannot be set to 25 °C it should be at least constant throughout the experiment and reported with the results.

Temperature dependency of static power

In this test scenario we want to measure the temperature dependency of static power directly. In contrast to the previous test cases, static power is measured directly rather than inactive power. This should be done by configuring the board and stopping the clock after the configuration is completed. The power is then measured on the configured but unlocked device. The reason for measuring a configured rather than an unconfigured device is that the power consumption can vary between these two states. Measurements for static power should be taken at the following temperatures: 25 °C, 40°C, 55°C, 70°C and 85°C. The device should be externally heated up in a temperature chamber and measurements should only be taken once the temperature has reached an equilibrium throughout the board and the FPGA. The results should be reported as absolute and LUT-normalised values.

Implementation Guide

The following guide outlines the key steps for implementing the benchmark.

VHDL Package

This benchmark includes two VHDL packages `RNG_test_circuit` and `PWM_driver`. `RNG_test_circuit` contains a top level file and a set of RNG circuits for testing the device. `PWM_driver` contains a design to implement the pulse generator that can be used as a controller on a second FPGA board to drive the device under test with various duty cycles.

VHDL Preparation

Prepare the top_level VHDL code for the RNG test circuit:

- Select the correct RNG version. The number of taps has to match the LUT size of the target device. *Example: The target device has 4_LUTs. Choose a 4-tap version.*
- Determine how many instances are needed to fill 60% of the device. *Example: The device provides 10k LUT/FF pairs. 12 RNG instances will require 6144 LUTs/FFs which is 61.44% of 10,000.*
- Modify the component declaration and instantiation accordingly. *In the current example, use `rng_width512_taps4`.*
- Modify the generic “size” to specify the number of RNGs needed. *In the current example specify `size = 12`.*
- Determine an appropriate way of implementing the inactive mode with FPGA-internal resources. If clock-gating is used, determine the appropriate clock buffer.
- Modify the clock buffer declaration and instantiate according to the correct proprietary primitive. Check the vendor’s device primitive libraries.

Synthesis, Constraints and Implementation

The VHDL code can be synthesised with vendor specific tools or third party tools. In some cases, vendor specific libraries have to be specified or included in the code. The mapping of logic functions to hardware resources is straightforward i.e. 12 RNG instances should use *exactly* $12 * 512 = 6144$ LUTs and flip flops. However, some tools can make unexpected choices and map some 4-input functions into a combination of one 3-LUT and one 2-LUT. The synthesis report file should be checked if the output and resource utilisation is as expected. Changing the synthesis options or optimisation efforts can help solving this problem.

The implementation constraints have to be chosen according to the target device and board. Make sure IO pins are located correctly. Check for low-active inputs or outputs and adjust the VHDL if necessary. *Make sure the clock buffer is disabled by default! Otherwise, the device could be unintentionally in active mode if uncontrolled.* In many devices, clock buffers are associated with preferred input pins. Place the clock buffer accordingly to minimise the timing delay and power. The clock tree between input pin and clock buffer is the only clocked resource during the inactive phase. Optimal placement of the clock buffer can therefore reduce the inactive power. Set timing constraints for the design according to the clock frequency chosen. Timing violations will go unnoticed because the RNG output is not verified but the measured active power could be inaccurate.

Map, place and route the design with the vendor tools. After completion, check the reports for timing violations and resource utilisation. The resource utilisation should be exactly as expected, e.g. in our example 6144 LUTs and flip-flops.

Experiment and Measurement

Connect the power measurement setup to the FPGA board. Configure the FPGA and set the device into active mode. Carefully monitor the device temperature. Determine if the device has to be measured according to the “cold” or “hot” scenario. If the device is “cold” simply measure the active power in combination with the device temperature. Inactive and low-power measurements should be taken after the device has cooled back down the 25°C.

A “hot” device has to be measured under various duty cycles. The `PWM_driver` design can be used to implement a duty cycle controller on a second FPGA board. Alternatively, a function generator or microcontroller can be used. Start the first measurement with 0% duty cycle, i.e. completely inactive. Then step up the duty cycle to 5% and check the temperature until it does not increase any further. Measure the inactive and instantaneous active power. If the “on” phase of the duty cycle is too short to measure active power, extend this phase to get a reading. Afterwards, wait for the temperature to normalise again. Increase the duty cycle in 5% increments and repeat the measurement procedure.

If a device supports further low power modes, these should be measured as well. In this case, the device does not have to respond to the control signal of the pulse generator. The device is simply transferred into the low-power mode and its power consumption is recorded. Advanced low-power modes should be measured at 25°C.

References

[1] David B. Thomas and Wayne Luk, "High Quality Uniform Random Number Generation Using LUT Optimised State-transition Matrices", *The Journal of VLSI Signal Processing*, Springer, 2007