LA-UR-04-7084

*Approved for public release;
distribution is unlimited.*

Title:

Evaluation of Power Costs in Applying TMR to FPGA Designs (Paper)

Author(s):

Nathaniel Rollins, Brigham Young University
Michael J. Wirthlin, Brigham Young University
Paul S. Graham, ISR-3, Los Alamos National Laboratory

Submitted to:

7th Annual Military and Aerospace Programmable Logic Devices International Conference, Washington D.C., 8-10 September 2004

Los Alamos
NATIONAL LABORATORY

Evaluation of Power Costs in Applying TMR to FPGA Designs

Nathan Rollins[1],Michael J. Wirthlin[1], and Paul Graham[2]

nhr2@ee.byu.edu, *wirthlin@ee.byu.edu*, and *grahamp@lanl.gov*

[1]Department of Electrical and Computer Engineering, Brigham Young University, Provo, UT. 84602

[2]Los Alamos National Laboratory, Los Alamos, NM

Abstract

Triple modular redundancy (TMR) is a technique commonly used to mitigate against design failures caused by single event upsets (SEUs). The SEU immunity that TMR provides comes at the cost of increased design area and decreased speed. Additionally, the cost of increased power due to TMR must be considered. This paper evaluates the power costs of TMR and validates the evaluations with actual measurements. Sensitivity to design placement is another important part of this study. Power consumption costs due to TMR are also evaluated in different FPGA architectures. This study shows that power consumption rises in the range of 3x to 7x when TMR is applied to a design.

I. Introduction

Triple modular redundancy (TMR) is a technique commonly used to make designs reliable in the presence of single event upsets (SEUs)[1]. This design hardening technique triplicates all of the resources used in a design and then uses a majority voter to vote on the outputs of the triplicated design. TMR can implemented on a design in different ways. The TMR style used in this study is shown in Figure 1. The top level design circuit is triplicated and the top level output ports connect to triplicated voters. This style of TMR will protect a design from SEUs, but this reliability comes at great cost.

Previous studies have shown that TMR can be used to make a design immune to SEUs[2] but at great cost in terms of design area and speed. A completely SEU immune design comes at the cost of at least 3x in area. In addition to these costs, the power increase due to TMR must be considered.

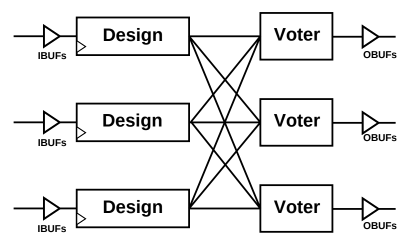


Figure 1: Triple modular redundancy (TMR) style which triplicates the top level design and provides triplicated voters

Power consumption is becoming a defining design criterion for semi-conductor devices[3]. FPGAs in particular, consume relatively more power than other semi-conductor devices such as ASICs. FPGAs are less power efficient than ASICs due to their flexibility and large routing matrix. The re-programmability of SRAM-based FPGAs causes them to require a larger number of transistors than ASICs. A larger number of transistors leads to larger leakage current. Leakage, or static power, previously considered insignificant compared to dynamic power, can no longer be neglected. Our study shows that static power makes up a large portion of consumed power. Power characteristics of an FPGA affect the density, performance, reliability, and cost of a device[4]. For some applications such as space-based applications where device cooling is an integral design consideration, but SEU immunity is essential, power consumption is certainly non-trivial.

The goal of this study is to evaluate power consumption of TMR. Triplicating an entire design suggests that the amount of power consumed will increase by at least 3x. Tripling power consumption is significant. In addition to evaluating the power costs of TMR, this paper investigates the effect of design placement on power consumption, and compares the power consumption of different Xilinx architectures.

II. Power Evaluation Tools

Reliable power measuring tools are necessary to determining how costly TMR is in terms of power. In order to verify the results of our study, we use a power measurement tool to verify the results of a power estimation tool. The two tools we use in our study are JPower, a tool which measures the amount of actual current flowing in a circuit, and Xilinx's XPower tool, which estimates the amount of power which a design would consume.

A. JPower

JPower is a tool that measures the amount of current flowing in the SLAAC-1V FPGA computing board[5]. JPower measures the current from the SLAAC-1V ADC by means of the SLAAC-1V C API and then stores the value as a 10-bit unsigned number. This registered value is then multiplied by a constant (4.8828125 mA) to produce the current value in mA (rounded to the nearest mA). JPower can measure current on the SLAAC-1V board in the range of 0 to 4995 mA.

The SLAAC-1V board ADC has three different channels from which to sample current. Channel 0 reports the board's 5V current, channel 1 reports the 2.5V current, and channel 2 reports the 3.3V current. The ADC can be sampled at a rate of up to 120 kHz divided by the number of channels being sampled. In our study we are only concerned with the power consumed by the actual circuit on the FPGA. In our study we disregard any I/O related current (channel 2), which means we only need to sample the current on the 2.5 supply.

In order to get accurate current measurements, a collection of ADC samples are taken and averaged. The amount of time between samples must be no less than 8.33 μs (120 kHz sample rate). When a sufficient number of samples are randomly taken and averaged, we find that JPower produces consistent results to within 2 mA. It is important to note that this averaged value includes the current from our design as well as from other sources.

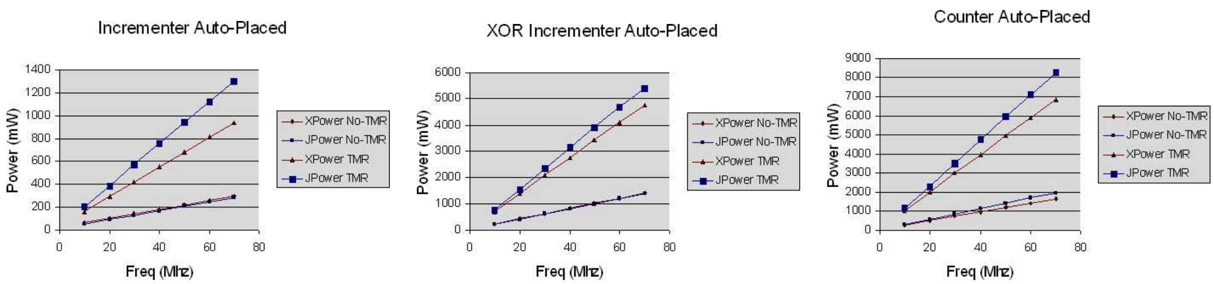
JPower reports the amount of current flowing through the entire SLAAC-1V board. Among other things, the SLAAC-1V board includes three Virtex V1000 FPGAs and multiple on-board memories. It is important therefore, to be able to distinguish between the current in the FPGA device we wish to examine and the current used by all other devices. The amount of current consumed by these other devices must be subtracted from the value measured from the ADC in order to isolate the current flowing through our design.

A simple equation was derived which tells us how much current to subtract from the measured ADC value. In order to derive this equation, current from channel 1 is sampled with no designs in any of the three FPGAs (a default design is automatically placed in the FPGA which communicates with the host). The SLAAC-1V board is run at a range of different frequencies and at each frequency, an averaged current value is recorded. At each frequency an averaged value was recorded when the clock was both running and stopped. The resulting formula is therefore a function of frequency as well as whether or not the clock is running. It is interesting to note that even when the clock is stopped, the amount of power consumed is a function of frequency.

JPower's ability to take true power consumption measurements for a design is invaluable. Unfortunately however, since the JPower tool is linked to the SLAAC1V board, it's use is limited to designs based on Xilinx's Virtex FPGA architecture.

B. XPower

Xilinx has a power estimation tool called XPower[6] which can estimate power consumption of designs for a variety of Xilinx FPGA architectures (not just Virtex). This tool is different from JPower in that it does not measure the actual current flowing in an FPGA. Instead, based on the input design, it calculates a power consumption estimate. This estimation is based on the design resources as well as the activity rates of the nets in the design. In order for XPower to be able to perform this estimation, every net in the design must have an activity rate assigned to it.



(a) 72 8-bit incrementers

(b) 416 XOR'ed 8-bit incrementers

(c) 416 8-bit up/down loadable counters

Figure 2: JPower and XPower results for the calibration designs with and without TMR applied

III. Testbench Designs

In order to callibrate the tools we compare the results of the two power evaluation tools. In order to perform this comparison, we employ the use of a set of simple test designs. The tools are used to estimate and measure the power consumed by each design run at a range of different frequencies. TMR is then applied to each design and the power tools again measure the amount of power dissipated at a range of frequencies. By comparing the amount of power consumed in the TMR designs with the amount of power used in the non-TMR designs, we can see the cost of TMR in terms of power.

In previous TMR studies[2] two simple designs were used to evaluate the area and speed costs of an SEU-immune design. The two designs used in these previous tests are an 8-bit incrementer and an 8-bit loadable counter. In our power study, we use these simple designs as part of our testbench designs to examine the power costs due to TMR. Since we will be using the JPower tool, all of the calibration designs are based on the Virtex FPGA architecture.

A single-bit incrementer and a single-bit counter each fit inside one slice of a Xilinx CLB. It is difficult for the tools to precisely measure the power consumption of an 8-bit incrementer or an 8-bit loadable counter alone. Therefore, in order to obtain significant power measurements from JPower and XPower, these designs are replicated a large number of times. In order to ensure that the nets of each design remain relatively active, we again restrict the bitwidth of each of the replicated incrementers and counters to be 8 bits wide.

	Non-TMR			TMR		
	INC	XOR	CNT	INC	XOR	CNT
Frequency vs. Power Slopes						
JPower	1.54	7.85	11.08	7.37	31.13	47.53
XPower	1.54	7.95	9.26	5.23	27.06	39.03
Area Costs						
LUTs	576	3250	3328	1728	9750	19968

Table 1: Frequency vs. power slopes for the calibration designs.

The replicated 8-bit incrementers are used in two different testbench designs for our power studies. In the first design, the incrementer is replicated 72 times and the output of each incrementer is fed to an output IOB. In the second design, the incrementer is replicated 416 times. In this second design, the outputs of the incrementers are divided into groups. The incrementer outputs in a group are XOR'ed together, and the XOR outputs are then fed to output IOBs.

A third testbench design is created from the 8-bit loadable counters. In this design, the 8-bit counter is replicated 416 times. The output of one counter is fed into the data input of the next. This creates a large chain of counters with the final counter's outputs leading to IOBs.

IV. Power Calibration Results

For each of the different testbench designs, the power evaluation tools are used to measure or estimate the power of each design at a range of different frequencies. Taking power measurements in a range of frequencies enables us to create a plot of frequency vs. power from which we can interpolate a slope which has units of mW per MHz. TMR is applied to each design and the power tools are again used to evaluate power at a range of different frequencies. Comparing the slope of a design with TMR implemented vs. the slope of a design without TMR provides the cost of TMR in terms of power.

Figure 2 displays four graphs. Both JPower and XPower are used in each graph to create frequency vs. power slopes for each of the calibration designs with and without TMR applied. In the first three graphs (Figure 2(a)-2(c)) the bottom two slopes show the power consumption for the design without TMR applied (one slope reports the JPower measurements, the other reports the XPower estimates). The top two slopes show the power consumption after TMR has been applied.

Table 1 shows the slopes of the graphs in Figure 2. The slopes are in units of mW per MHz. This table shows that the two tools are fairly close in their measurments. For example both tools report a slope of 1.54 mW per MHz for the array of 72 incrementers without TMR. The slopes, given for both JPower and XPower, enable us to determine the cost of TMR in terms of power. This cost is calculated from the ratio of the slope of a TMR applied design vs. the slope of a design without TMR. Before we investigate this ratio further, we first consider how design placement can affect frequency vs. power slopes.

V. Effects of Design Placement on Power

An important part of this study involves investigating the effects of design placement on power costs associated with TMR. Our studies show that the amount of power a design consumes is highly dependent on how it is placed. To demonstrate this dependence we use the our first calibration design (the array of 72 8-bit incrementers).

Figure 3 shows three different hand placements of the first calibration design. The first placement is a poor placement; the incrementers are spread far apart from each other and therefore long nets are required to connect to the voters. The second placement is an improvement on the first, but the third

	Incrementer			
	Auto-Place	Place 1	Place 2	Place 3
Frequency vs. Power Slopes (TMR)				
JPower	7.37	10.65	6.15	4.76
XPower	5.23	6.20	5.21	4.78
Power Increase Due to TMR				
JPower	4.79x	7.04x	4.06x	3.10x
XPower	3.40x	4.04x	3.39x	3.10x

Table 2: TMR power costs for different placements of an array of 72 8-bit incrementers

placement is the best placement. Along with these three hand placements, we have the 'auto-placed' design which the Xilinx place and map tools provide. The results shown in Figure 2 and Table 1 are auto-placed results.

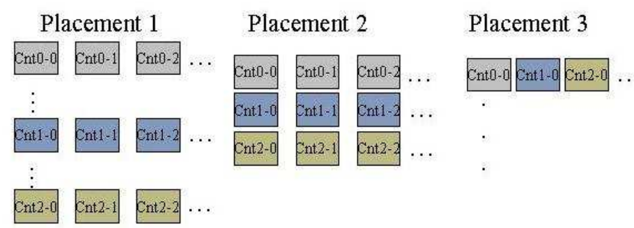
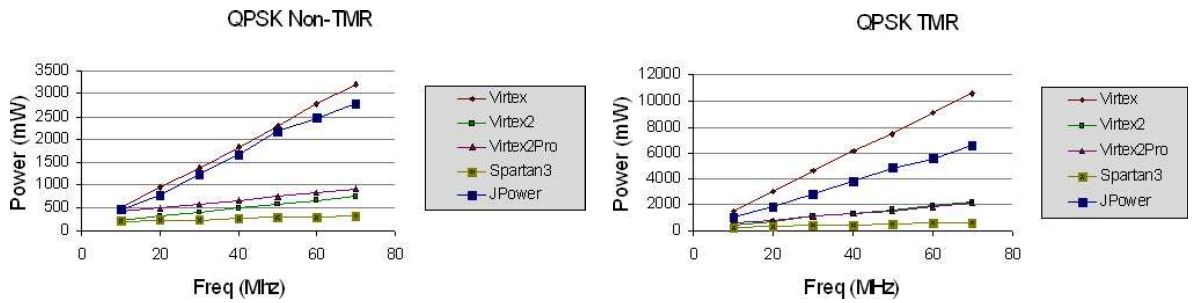


Figure 3: Three different hand placements of the array of 72 8-bit incrementers

Table 2 shows the power costs due to TMR for the four different placements of the array of 72 8-bit incrementers. The cost is determined by the ratio of the frequency vs. power slope of the placed design with TMR applied to the frequency vs. power slope of the design without TMR.

We can see from the table that JPower is more sensitive than XPower to design placement. For the poor hand placement JPower reports a power cost of 7.04x while XPower reports a power cost of 4.04x. Notice however that for the optimal placement that both JPower and XPower report a power cost of 3.10x. This result agrees with our intuition that when we triplicate a design, the power will also triple. These results also indicate that power consumption is indeed linked to design placement.

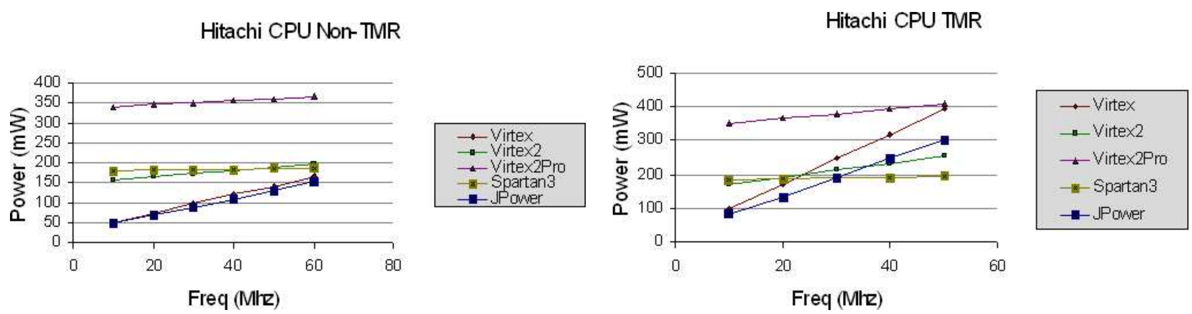
A less thorough demonstration of how design placement relates to power consumption is shown in



(a) QPSK demodulator without TMR

(b) QPSK demodulator with TMR applied

Figure 4: Frequency vs. power slopes for the QPSK demodulator with and without TMR applied, for different Xilinx FPGA architectures



(a) 8-bit Hitachi CPU without TMR

(b) 8-bit Hitachi CPU with TMR applied

Figure 5: Frequency vs. power slopes for the 8-bit Hitachi CPU with and without TMR applied, for different Xilinx FPGA architectures

Table 3. In this table the frequency vs. power slopes are shown for two different placements of all of the calibration designs. The auto-placement is shown as well as an optimized hand placement. Also shown in the table is a ratio of JPower to XPower - indicating how well the two tools agree in their results. A value of 1 indicates the two tools agree in their results. We can draw similar conclusions from this table as we could from Table 2: power consumption is directly affected by design placement and JPower is more sensitive to design placement than XPower.

VI. Power Costs of Different Architectures

Having compared the results of the two power evaluation tools we can now use these tools to evaluate the cost of TMR in terms of power on some real designs. The two designs that we use to measure the cost of TMR in terms of power consumption are an 8-bit Hitachi CPU and a QPSK demodulator. Both designs are implemented on the Virtex architecture as well as the Virtex2, Virtex2Pro and Spartan3 architectures. Implementing these designs on different architectures allows us to examine power consumption characteristics of each architecture.

Before looking at the power costs of TMR on theses designs, we first look at the costs of TMR for these designs in terms of area and speed. Table 4 shows these costs. The area costs listed are strictly in terms of the number of LUTs required for the design. The cost in terms of other resources such as IOBs,

	Incrementer		XOR Incrementer		Up/Down Counter	
	Auto-Place	**Hand-Place**	**Auto-Place**	**Hand-Place**	**Auto-Place**	**Hand-Place**
Frequency vs. Power Slopes						
JPower	7.37	4.78	31.13	22.18	47.53	41.22
XPower	5.23	4.76	27.06	25.10	39.03	36.40
JP / XP	1.41	1.00	1.15	0.88	1.22	1.13

Table 3: Frequency vs. power slopes for different placements of the calibration designs

		QPSK	Hitachi
Virtex	**Area Cost**	3.03x	3.01x
	Speed Cost	4.8%	29.9%
Virtex2	**Area Cost**	3.03x	3.00x
	Speed Cost	15.4%	0.0%
Virtex2Pro	**Area Cost**	3.03x	3.00x
	Speed Cost	18.1%	19.2%
Spartan3	**Area Cost**	3.02x	3.00x
	Speed Cost	2.8%	13.0%

Table 4: TMR costs in terms of area and speed for an 8-bit Hitachi CPU and a QPSK demodulator

BRAMs, TBUFs, and multipliers also reported an area cost of 3x in all cases. The speed costs report how much slower the maximum clock speed of the design with TMR can run compared to the maximum clock speed of the design without TMR.

Since the area costs of TMR for these two designs are about 3x we expect that if the designs are placed relatively well, the power costs of TMR will also be about 3x. The graphs in Figures 4 and 5 show the frequency vs. power slopes of the two designs for a variety of Xilinx FPGA architectures. These slopes are recorded in Table 6 as dynamic power. The intercept of these slopes gives us a value for static power. The cost of TMR in terms of power is determined from the ratio of dynamic power without TMR to the dynamic power with TMR. Table 5 shows this ratio for the Hitachi and QPSK designs for each architecture. For a design placement performed by the Xilinx place and map tools, we see that the cost of TMR in terms of power is relatively close to 3x.

Table 6 also provides important information about static power. As we move from the Virtex architecture to the Virtex2 architecture and then to the Virtex2Pro and Spartan3 architectures, static power increases while dynamic power decreases. In Figure 5(b) we see that at 50MHz the overall power for Virtex, Virtex2, and Spartan3 architectures are almost the same. Below 50MHz, the Virtex architecture consumes less overall power due to its lower static power consumption. Above 50MHz, the Spartan3 architecture consumes less power overall due to its lower dynamic power consumption. The graphs in Figures 4 and 5 show that the overall power consumption is dependent on the design, the FPGA architecture, and on the clock frequency at which we run the design.

	JPower	Virtex	Virtex2	Virtex2Pro	Spartan3
Dynamic Power Increase For TMR					
QPSK	2.53x	3.30x	3.51x	3.06x	3.39x
Hitachi	2.66x	3.12x	2.66x	2.88x	2.50x

Table 5: TMR costs in terms of power for an 8-bit Hitachi CPU and a QPSK demodulator

VII. Conclusion

This paper investigates the cost of TMR in terms of power. Since previous studies[2] have shown that the cost of TMR in terms of area can be 3x, it is reasonable to expect that the power consumption will also triple. When TMR is performed at the top design level, and the design is relatively well placed we have shown that indeed the power consumption is also triplicated. We have also shown how power consumption is affected by design placement. Evaluating the power costs of TMR on different FPGA architectures has shown how static power in many cases contributes more to the overall power consumption than dynamic power. Overall power consumption is affected by the design implemented, by the FPGA architecture the design is implemented on, by the design placement in the FPGA and on the clock frequency the design runs at.

	Non-TMR					TMR				
	JPower	Virtex	Virtex2	Virtex2Pro	Spartan3	JPower	Virtex	Virtex2	Virtex2Pro	Spartan3
Dynamic Power (mW / MHz)										
QPSK	40.50	45.71	8.60	8.16	1.97	93.75	150.64	30.17	24.98	6.68
Hitachi	2.06	2.34	0.79	0.48	0.12	5.48	7.30	2.10	1.39	0.30
Static Power (mW)										
QPSK	28.57	22.14	150.00	336.86	179.83	26.43	37.86	139.50	334.71	180.23
Hitachi	27.17	26.43	150.00	337.07	180.00	28.25	27.50	150.00	337.50	180.34

Table 6: Static and dynamic power consumption of an 8-bit Hitachi CPU and

References

[1] J. von Neumann. Probabilistic logics and the synthesis of reliable organisms from unreliable components. *Automata Studies*, (Annals of Math Studies No. 34), 1956. Princeton University Press.

[2] Nathan Rollins, Michael Wirthlin, Michael Caffrey, and Paul Graham. Evaluating TMR techniques in the presence of single event upsets. In *Proceedings of the 6th Annual International Conference on Military and Aerospace Programmable Logic Devices (MAPLD)*, September 2003. To Be Published.

[3] A. Allan, D. Edenfeld, W. Joyner Jr, A. Khang, M. Rogers, and Y. Zorian. 2001 technology roadmap for semiconductors. *Computer*, 35:42–53, January 2003.

[4] Xilinx. FPGAs, power and packages. *XCell*, 1997.

[5] USC-ISI East. *SLAAC-1V User VHDL Guide*, October 1, 2000. Release 0.3.1.

[6] Xilinx, Inc. *XPower Manual*.