Digital Synchronverter Design Flow for Renewable Energy

James K. Reed, Qing-Chang Zhong and Jafar Saniie

Department of Electrical and Computer Engneering
Illinois Institute of Technology, Chicago, Illinois, USA

***Abstract*— In today's energy grid there are very few options to handle the severe energy swings of renewable energy sources. A Synchronverter is designed to allow an asynchronous energy source like wind mills and solar panels to appear as a synchronous energy sources. To this day, the designs have been analog with large size and large assortment of components. This paper proposes a digital design that matches the proposed analog design's output, while using common digital components in order to lower cost and size. The primary focus for the digital design is to use the most optimal mathematical block solutions for improved performance. An FPGA is used as the base component while implementing common FIR integrators. For the mathematical sine and cosine functions, the CORDIC method is used due to its efficient calculation of the sine and cosine functions. Using scaling and 8-bit multiplication blocks, the data integrity can be maintained, while the keeping scale to a minimum. Overall this paper provides a digital synchronverter solution that is optimal for small FPGA use, translating to small dedicated chips, with peak performance.**

I. Introduction

In today's world, we use terawatt-hours upon terawatt-hours of electricity to power the digital and electrical age that we live in. Humanity has developed many forms of energy conversion to satisfy this ever-growing power consumption, including oil, natural gas, coal, nuclear energy, as well as environmentally friendly sources (solar, wind, water, and geothermal). With the grid having grown large enough to supply thousands of terawatts-hours per year, it has developed multiple major issues. One significant one is the stability of the grid. The power grid is notorious for having momentary power outages as well as power surges, especially around power hungry areas like large cities. The solution to this issue is to take all modern day's computational devices and combine them with our generators and motors to create what is called a smart grid. This smart grid will be able to allow devices to be plug-and-play, very like how a USB device operates in today's world, but reduce the amount of power surges and droops seen throughout the grid.

The biggest hindrance to the idea of a smart grid is the fact that we will need to utilize renewable energy sources. The nature of these power sources is asynchronous. Their energy is generally from a source that is not very consistent. Thus, integrating it into a well-controlled smart grid is nigh impossible in its current state. Thus, much research has been put into a device that is able to simulate a synchronous output with an asynchronous source. This device is self-titled Synchronverter and makes an asynchronous system appear synchronous [1]. Thus, with the proper controller and refinement of system feedback loops, this new component could become a self-synchronous device and easily controlled by an external source [2-4]. With many years of investment in this device, many more improvements were made and the current synchronverter design has seen many newer applications such as improved solar panel control and the ability to emulate a HVDC transmission system [5-10]. Below in Figure 1 is the current block diagram of the standard synchronverter design [2]:

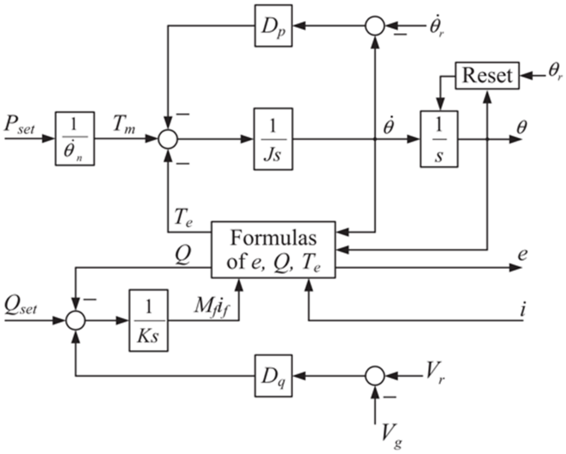

Figure 1. Current Standard Synchronverter Block Diagram [2]

A synchronverter is a major design point to smart grid that allows for this plug-and-play system. The idea of the synchronverter came from the concept that renewable energy was becoming a more readily available power source. Synchronverters have been mathematically proven and tested using basic analog components along with microcontroller systems. The next major step in the future of the synchronverter is to develop a cheaper, smaller, smarter, and more robust version; to take the mathematical models and construct a digital synchronverter that can either be implemented with a single digital chip or implemented easily within a digital circuit.

The purpose of this paper is to determine if it is practical to develop a single chip synchronverter solution using a FPGA.

978-1-5090-4767-3/17/$31.00 ©2017 IEEE

The focus is to develop the initial synchronous machine core. This core is a complex piece of the current synchronverter design and its size and performance will dictate the feasibility of a single chip solution. Below in Figure 2 is the block diagram of the core synchronous machine initially developed along with the equations within this system:

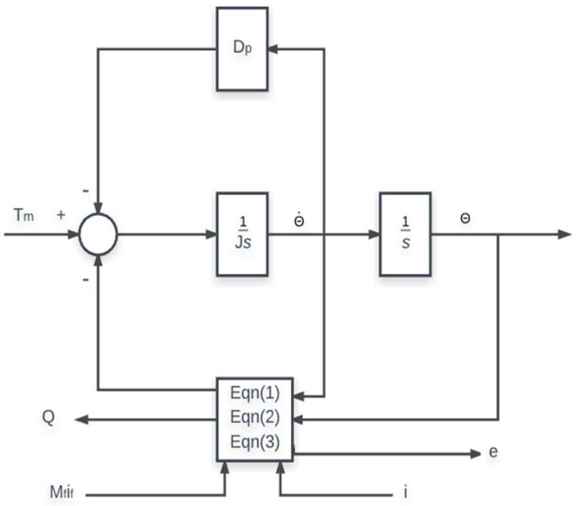

Figure 2. Synchronverter Design in Control System Format, with Equations 1-3 Referenced Below [1]

$$T_e = \rho M_i i_f \langle i, \sin\theta \rangle \tag{1}$$

$$e = \dot{\theta} M_i i_f \sin\theta \tag{2}$$

$$Q = -\dot{\theta} M_i i_f \langle i, \cos\theta \rangle \tag{3}$$

II. Synchronverter Design Flow

There are many potential problems to consider when converting the concept of a synchronverter into a digital design. This includes:

1. Understanding the system design to know the data you expect out of it
2. Implementing the individual functions described in the block diagram (shown in Figure 2)
3. Optimizing the system to be efficient, low cost, and practical.

The basis of a synchronverter [1] is to allow for synchronous DC-AC conversion. This is because many of the power sources are units that either supply DC power like solar panels, or systems that convert AC to DC power like wind mills. Thus, a device needs to interpret the provided DC power and supply it as AC power to the grid. The problem, however, is that the DC power needs to be supplied at 60 Hz, and much of the DC power is known to fluctuate due to inconsistency in the power sources. Synchronverters are components that can perform this interpretation and allow for many of the asynchronous devices to appear as 60 Hz synchronous generators. Looking at the design (seen in Figure 2) we see an intricate system that allows for conversion between DC and AC power. This system, paired with the power diagram seen in Figure 3 below and a controller, allows this synchronous DC-AC conversion.

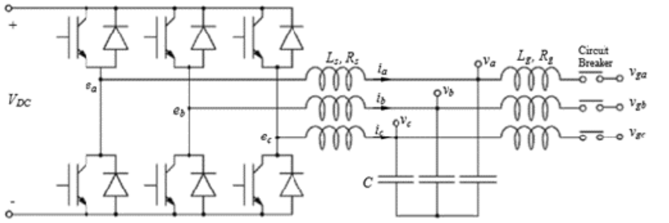

Figure 3. Power Section of Synchronverter [2]

The first major section of the design is the desired torque of motor, T_m, and is provided by the controller. The motor torque runs through an addition block that will subtract the current torque of the generator, T_e, and the torque calculated from the damping of the inverter circuit, D_p. The torque of the generator is determined by Equation 1 of the equation block. For the damping block, D_p, the system takes the projected angular velocity of the generator and multiplies it by the change in torque required. Thus, it generates an internal loop that feeds itself, basically providing a self-check system to determine if the end goal has been achieved.

The torque difference is then passed through an integrator with a magnitude gain of 1/J. The term J is determined by the end user rather than the system. It is designed to be the amount of energy gain the system needs to experience. The output of the integrator is the angular speed, $\dot{\theta}$, the system needs to achieve in order to produce the torque difference. This is then supplied to the damping block, another integrator that produces the angular position, θ, desired for the given speed, and fed to the equation block for further analysis. The angular position is the key comparison factor as it tells the system what angular frequency position is needed to achieve the proper power into the grid.

The first equation, referenced as Equation 1, determines the torque generated by the current generator T_e. This equation uses: the angular position of the generator; the current of the generator; the coil winding coil/current multiplier; and the number of desired system poles. The input current from the power section of the synchronverter is sent to the system via a 1x3 vector, meaning there are a total of three current values sent to this block. This is because the power section needs three lines, all offset by approximately 120 degrees. Thus, to get it down to one value, the current needs to be multiplied by a 1x3 angular position vector that uses θ and adds 120 and 240 degrees to get the second and third value of the 1x3 vector, with the angular position being the first. These positions then run through a sine function to get the angular position vector in radians. This resulting vector is then multiplied by the generator current vector and provides one, single number to be multiplied by the number of poles and winding coil/current multiplier. Equation 2 and 3 produce the electromagnetic force, e, and the complex power, Q, being generated by the real power and generator currently. The major difference between each of the individual equations is the different parameters overall, even though they use many of the same inputs. Equation 2 focuses on just the angular position and angular speed with no current reference or pole reference, when compared to Equation 1. While Equation 3 removes the pole dependency and adds in the angular velocity and changes the current and angular position calculation to be based on the cosine of angular position, rather than sine.

In the past, there has been only direct analog implementation of a synchronverter. In recent years, however, there has been much discussion on how to implement the synchronverter in a single chip package. There are multiple ways to implement this system in a pure digital approach. The three-main implementations are: an embedded microcontroller, a computer using a USB board to read and interpret all the data, and lastly is an FPGA to simulating the direct analog implementation. Ultimately, an FPGA digital design was chosen due to its powerful computational system-on-chip (SoC) capability, and ease of interfacing to a complex environment such as power grid.

There are five main blocks within the system: an equation block, two integrators, a dampener, and a summation block. Designing each component for minimum size with best performance will be critical. The first major component looked at was the equation block due to its overall complexity and consuming a large proportion of the overall size. Within this block, the major focus was the sine/cosine function as it is a major computational moment and is quite large with it handling a 1x3 vector. There are two methods that are most commonly done to implement sine/cosine: look-up table or CORDIC algorithm. When looking at the overall size of each depending on the input sizes you get the following results in Table I:

TABLE I. SINE FUNCTION IMPLEMENTATION, SIZE COMPARISON

	4-bit	5-bit	8-bit	16-bit
Look-Up Table (LUTs Used)	8	15	127	32767
CORDIC (LUTs Used)	N/A	N/A	296	1021

Because the input of the equation block for theta is 24-bits, this size comparison shows that three CORDIC blocks will save space not only due to their ability to do both cosine and sine together, but also because the number of LUTs is much lower the larger the input bus width.

The next focal point was the integrators. Because they are generally large memory blocks used to store past results, it was a matter of balancing size versus the accuracy of the integration itself. Below in Figure 4 we can see the error percentage versus the size of the integrator itself and Equation 4 demonstrates the integrator equation being used:

$$y(n) = \frac{x(n) + y(n-1) + y(n-2) + \cdots + y(n-N-1)}{N} \quad (4)$$

The results from Figure 4 show that as the size N increases past 8-bits, we begin to see a major decrease of error to the overall size of the integrators. Thus, it seems N greater than 28 ends up being the desirable integrator size for both accuracy and efficiency.

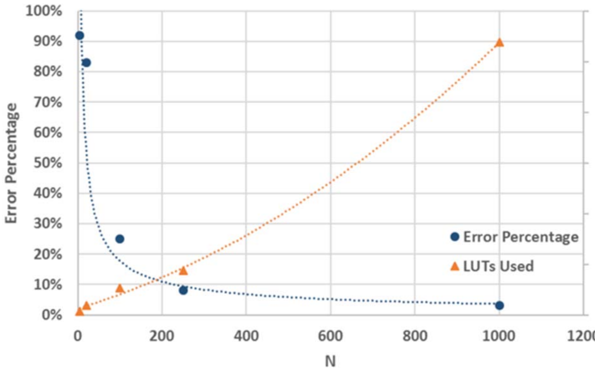

Figure 4. Size of Integrator vs Error Percentage of Results

The last major design piece is the dampener, summation, and bus size of the system. All kind of correlate to one another as they require uniform movement throughout the system. To strike a balance for accuracy and efficiency, we use buses of 16-, 24-, and 32-bit. This allows for easy scaling and movement from one block to the next. Figure 6 below shows the overall system broken down into the buses per line:

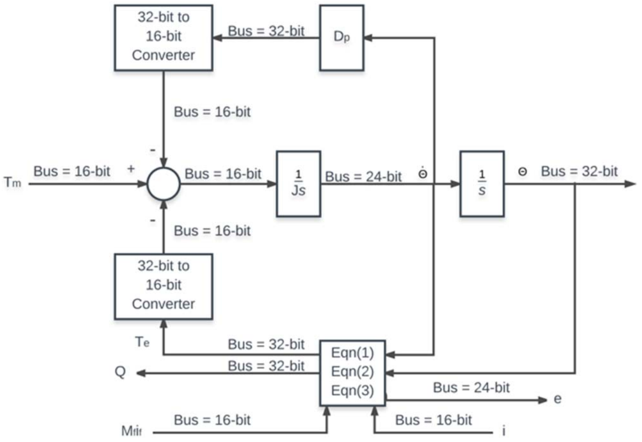

Figure 5. Proposed Bus-Size for Synchronverter Design

III. SIMULATION RESULTS

When building the entire system, we see that the system is moderately large and highly practical to be implemented in an FPGA such as Spartan-3e by Xilinx. The system gave an LUT usage of 5874 and 1479 slices. The primary focus of the system resources was mostly from logic gates with 4142 used, which makes sense due to the large amount of computations occurring in this system. Overall, this system is quite small compared to the average size of today's FPGAs, the average FPGA uses around 5000 logic gates and 2000 slicers, so this could be put on quite small and affordable FPGAs of today. The major hindrance, however, is the number of needed pins; the outputs total to 128 pins and the inputs require 48pins. This can be remedied by scaling the outputs to a more reasonable number of outputs, aka scale a 32-bit bus down to 8-bit, but this would require sacrificing overall resolution.

The other major piece looked at was the timing analysis of the overall system. Sure, it could be small enough to fit on smaller FPGAs, but if it cannot produce the desired reaction times, it is of no use overall. When running a static timing analysis, we see that the system at its slowest will react when any of the current, i, input pins are changed. The average time seemed to be around 25 nanoseconds. This makes sense due to the large vector computation that is needed. Comparing this to the data provided on the analog synchronverter, we see this system runs in the milliseconds range and thus this response time is well within the range of the desired system.

IV. Conclusion

When taking the given simulation results into consideration, the system is clearly fast enough to replicate the analog system. Many of the analog systems of today run in the millisecond range; the system designed can react as quick as 25 nanoseconds and would be able to stabilize in roughly 6.4 microseconds. This amount of reaction time allows for a very stable yet reactive system.

The size of the design is small enough to fit onto smaller Spartan-6 FPGA chips. The current design was using the Spartan-3e chip and fit with ease with only using 36% of the chips capable functions.

The next major step of this project is to run many iterations of this design and work on overall usability, stability, and size. Using more complex integrators and multipliers could help decrease system size while improving overall system stability and response time. Another potential area of focus would be to absorb the synchronverter controller. This would allow for an all-in-one solution and with the current size being quite manageable, it is very possible. With enough iterations of these two concepts it might even allow for stand-alone chip variations, making wide spread adoption that much easier.

References

[1] Q. Zhong and G. Weiss, "Synchronverters: inverters that mimic synchronous generators." IEEE Trans. Ind. Electron., vol. 58, no 4, pp. 1259-1267, Apr. 2011.

[2] Q. Zhong , P. Nguyen, Z. Ma, and W. Sheng. "Self-synchronized synchronverters: inverters without a dedicated synchronization unit." IEEE Trans. Power Electron., vol. 29, iss 2, pp. 617-630, May 2013.

[3] Z. S. Bahron. "Development of synchronverter control for induction motor connection." UTHM Instit. Rep., id 6715, April 2015.

[4] S. Dong and Y. C. Chen. "Adjusting synchronverter dynamic response speed via damping correction loop." IEEE Trans. Energy Conv., vol. PP, iss. 99, pp. 1, Dec. 2016

[5] P. Piya and M. Karimi-Ghartemani. "A stability analysis and efficiency improvement of synchronverter." 2016 IEEE App. Power Electron. Conf. & Expo. (APEC), doi: 10.1109/APEC.2016.7468317, May 2016.

[6] C. Zhang, J. Meng, S. Chen, X. Chen, Q. Zhong, Z. Lv, and Q. Huang. "An improved synchronverter model and its dynamic behavior comparison with synchronous generator." 2nd IET Renew. Power Gen. Conf. (RPG 2013), doi: 10.1049/cp.2013.1879, Jan. 2014.

[7] R. Aouini, B. Marinescu, K. B. Kilani, and M. Elleuch, "Synchronverter-based emulation and control of HVDC transmission." IEEE Transact. Power Sys., vol. 31, Jan. 2015.

[8] E. Brown and G. Weiss. "Using synchronverters for power grid stabilization." 2014 IEEE 28th Conv. Elec. & Electron. Eng. Israel (IEEEI). doi:10.1109/eeei.2014.7005736, Jan. 2015.

[9] A. Thomas and S. D. Smitha, "A self-synchronized synchronverter technology for integrating PV inverters to grid without using a phase locked loop." International Journal Innov. Research Tech. (IJIRT), vol. 2, pp. 24-29, Oct. 2015.

[10] S. Mishra, D. Pullaguram, S. A. Buragappu, and D. Ramasubramanian. "Single-phase synchronverter for a grid-connected roof top photovoltaic system." IET Renew. Power Gen., vol. 10, iss. 8, pp. 1187-1194, Sept. 2016.