Hardware Prototyping of Digital Residential Energy Meter

M. B. I. Reaz, F. Mohd-Yasin, M. S. Sulaiman, F. Choong, A. Alaudeen
Faculty of Engineering
Multimedia University
63100 Cyberjaya, Selangor
Malaysia
mamun.reaz@mmu.edu.my

Abstract – **This paper describes the hardware prototyping of a digital energy meter. This digital energy meter development is focused on home usage since it employs a single-phase power system. The model architecture comprises of four main modules: power, energy, billing and display. Reprogrammable devices such as Field Programmable Gate Arrays (FPGAs) are highly attractive options for hardware implementations of proposed algorithm as they provide potentially much higher performance than software solutions and generate low-cost prototypes of designs. In the design, each module is modeled individually using behavioral VHDL and combined using structural VHDL. This is followed by the timing analysis for the validation, functionality and performance of the models using Aldec Active HDL. After successful simulation, the design was synthesized using Synplify for the effectiveness of the proposed hardware realization of the applications with achieved a maximum clock frequency of 31.4 MHz. It is proven that the model has been tested successfully and follows the behavior of a conventional energy meter.**

I. INTRODUCTION

The most popular energy measurement method is the analogue method of the electromechanical meter. However, there are certain drawbacks with the analogue meter. An analogue solution is not synchronized to a clock or crystal but calculates the product of current and voltage continuously. Analogue meters lack stability; accuracy drifts over large variations in operating environment and long periods of time. They use a series of dials to display kWh consumption [1]. Most consumers either don't know how to interpret these dials or never read the meter to double-check the bill unless it is inconsistent with past bills. Electronic solutions have been used in the past to duplicate the behavior of a classic electromechanical meter [2]-[4]. Examples of electronic analogue signal processing include Hall effect multiplication and discrete analogue multipliers.

In this paper, a digital hardware approach is introduced. The technology used to build digital hardware has evolved dramatically over the past four decades [5]. Until the 1960s, logic circuits were constructed with bulky components, such as transistors and resistors that came as individual parts. In contrast to standard chips that have fixed functionality, it is possible to construct chips that contain circuitry that can be configured by the user to implement a wide range of different logic circuits. These chips have a very general structure and include a collection of programmable switches that allow the internal circuitry in the chip to be configured in many different ways. Such chips are known as *programmable logic devices* (PLDs). One of the most sophisticated types of PLD is known as a field-programmable gate array (FPGA). FPGA has emerged as a quick and very effective means to generate low-cost prototypes of designs. In recent times, FPGAs have grown from a tiny market niche to a significant portion of the IC market. The complexity and speed of the FPGAs available in the market has been increasing at a rapid pace. Simultaneously, the cost per gate of FPGAs has been fast decreasing. The Synopsys FPGA Compiler has been developed primarily to target FPGA technology libraries. Here FPGA synthesis with special focus on Altera FLEX10K family of FPGAs will be discussed. The field of VLSI was chosen because it greatly reduces the size of a meter, offers a higher reliability, improved security, higher performance, higher accuracy and also the ability to add on features easily [6]-[8].

The behavior and description of the digital energy meter was modeled using VHDL, a hardware independent language, which supports a design to be modified easily to suit requirements. This can save valuable time. Digital circuitries used are quite simple, for example, adders, multipliers, dividers and comparators and are easily implemented in low cost programmable devices or ASICs. Electricity usage is determined by reading the electric meter at residences monthly and keyed into a computer by the meter reader to generate the monthly billing. However, this leads to two possible errors: reading error from the meter causes an inaccurate bill to be generated and the customers are unaware of the wrong entry due to typing error [9]. With the digital energy meter, there is no more tedious calculation or method of reading the value and calculating the bill as the readings can be observed from the display. This digital energy meter also displays the total energy consumed in number of units and the corresponding charge at any time and thus it offers the user a choice of controlling the energy usage [10].

II. DESIGN OVERVIEW

The whole system is fully synchronous as it uses a single system clock. All the different modules are made to work in parallel. This increases the speed of the entire system [11]. First the bare bone block diagram for the meter was designed and the inputs and outputs are determined. Two input values, which are single-phase voltage and single-phase current, will be fed into the digital energy meter and the output is expected to be the energy consumed and the corresponding billing. The suggested special feature to be introduced to this meter is the ability to calculate the total amount, at any point in time in accordance to the total energy consumed that the user has to pay, which will be convenient to the user.

The system will take in incoming instantaneous values of current (from 0 to 15000mA) and voltage (scaled down

to 5V); multiply the values together to obtain the electrical power. The individual power values are then compared with the previous value. If it is found to be the same, it will be added to the previous total value and the duration (in terms of clock cycles) recorded. However, if it is not equal to the previous power value, the previous total power and the duration are multiplied together to obtain the total energy for that particular duration. The energy values will keep on accumulating and each time it reaches 1 kWh, 1 unit of energy has been used up. The billing for the total number of units consumed will be calculated. The total number of units and the corresponding charge will be converted to BCD digits and displayed using 7-segment decoders. A digital clock is used to display the time as well as to keep track of the number of days elapsed. If the end of the month is reached, a signal will be sent.

III. SYSTEM LEVEL DESIGN

The core of the energy meter has been sub-divided into several distinct modules. Each of these modules performs a specific function or task. The top-level diagram or architecture designed is shown in Figure 1. The function of each individual module is given in sub section 3.1 to 3.9.

A. Clock Divider Module

The system clock of frequency 20MHz is too fast to verify the design and thus a clock divider is added. This module takes a 20MHz clock input and output a 100Hz square wave. Dividing 20Mhz by 200000 does this and flipping a bit each time it reaches that value. A counter of 18 is needed so that it can hold the value of 200000.

B. Scale down Module

This module takes in input current values and outputs a 11-bit value in binary. The output is fed in as the input to the Power Calculation module which, together with the fixed voltage of 5V, the power values are calculated. The current values are taken between 0 to +15000 mA that is the maximum current is fixed at 15 A. Incoming analog values are scaled down by a factor of 10 to produce values between 0 to 1500. The output current values will then be divided again by a factor of 100 in the Energy Store module. The reason for this is simply to obtain values in terms of Amperes.

C. Power Calculation Module

The function of this module is simply to multiply the values of current by voltage to produce the individual values of power where P = v x i. The module has three inputs and two outputs. The input values are the 5-bit supply voltage value, the 11-bit current values and the clock signal. The product of current and voltage is a 16-bit value. This output is fed to the Power Store module.

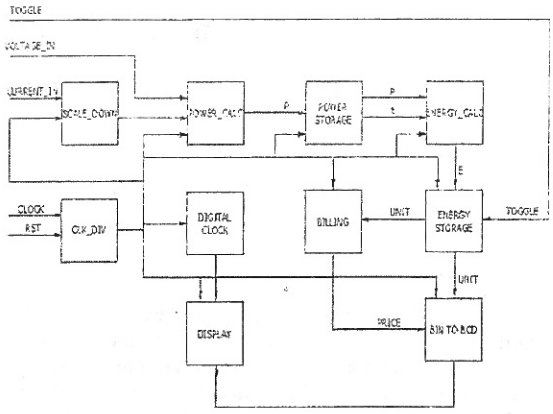

Figure 1. System top-level diagram

D. Power Store Module

This module is one of the important modules in the whole system. It employs an adder to add all the power samples, a comparator to detect variations in power levels and also a register to store these values to be fed into the Energy Calculation module when appropriate. It first compares the incoming power value with the previously recorded value. If it is found to be equal, then the incoming value will be added to the total power and the counter will be incremented by 1. This counter value denoted the amount of time in number of clock cycles elapsed since the last change was recorded. In other words, it is the duration at which a particular power value is constant. If the value is not equal, then the value of the total power together with its corresponding time is sent to the Energy Calculation module to calculate the energy.

E. 24 Binary to BCD Converter Module

This module converts a binary number representation to a BCD representation. It takes in a 24-bit binary number and outputs a 32-bit binary number that represents a set of eight BCD numbers. The 24-bit binary number is captured synchronous to a rising Clock edge from the port Dual when Indicator equals one. This triggers the conversion process. The Indicator signal should be '1' only during the capturing Clock edge. The binary input should be stable when the Indicator signal goes high. The result BCD is valid at the next rising Clock edge when Ready equals one. This module has been designed in a way that it will accept the next incoming input only when the process of converting the previous input to BCD has been completed. To achieve this, the Ready output port is connected to the Status Check module. The module is scaled by the three generics DUALBITS (the number of bits of the dual number), BCDBITS (the number of bits required for the BCD representation) and BCDBLKS (the number of BCD-blocks in the BCD representation which is equal to BCDBITS/4). The output that comes out will display the contents of the seven registers that hold all the intermediate results of conversion before the final results is obtained.

In designing this module, a very efficient technique, the "add-3" conversion algorithm was employed [12]. The conversion from binary to BCD form can be done by

repeated multiply-by-2 operations. The multiplication by 2 is accomplished by a left-shift operation followed by an adjustment of the of the BCD digit, if necessary. For example, a left-shift of BCD $1001(9_{10})$ yields 1 0010, which is 18 in binary but only 12 in BCD. Adding $0011(3_{10})$ before the shift gives 1 1000, which the required value of 18. The same result could also be obtained by adding $0110(6_{10})$, since a left shift multiplies any number by 2.

BCD digits in the range 0-4 will not require an adjustment after a left shift, because the new number will be in the range 0-8, which can still be contained in a four-bit BCD digit. However, if the number to be shifted is in the range 5-9, then an adjustment is necessary after a left shift, because the new range will be 10-18, which requires two BCD digits. Thus, if the BCD digit to be shifted has a value of 5-9, then 3 is added to the digit and the digit is shifted left one bit position. Register B contains n binary bits and is initially loaded from a data bus. The BCD number is formed in register A, which is initialized to 0. The adjustment is performed before each shift by inspecting each BCD digit. If a digit requires adjustment, then 3 is added to the digit and the sum is reloaded into the appropriate position in register A. Then registers A and B are shifted left one bit position in concatenation. The process is performed n times under the control of a sequence counter until all n bits have been shifted out of register B.

F. Status Check Module

This module takes in as input, the output signal of the previous module, the Binary to BCD module, checks the status of the BCD conversion process and outputs a '1' if the conversion is complete and a '0' otherwise. This output is connected back into the Binary to BCD module to initiate the next conversion. In the Reset mode, the Status signal is given the value of '1' for 1 clock cycle to enable the first conversion process.

G. Display Module

The Display module takes in a 32-bit binary input and generates what appears to be the continuous display of 8 digits by lighting each digit up in sequence. The hex_display component is used as the means to drive the segment pins. A 14-bit counter is declared whereby each time the count "rolls over", switch to the next digit. There is also a selector to select the corresponding digit to be displayed. The value of the selector is used to decide which bit of data to put on the cur_digit lines (which feed the hex_display decoder). The hex display decoder is instantiated in the display module. It uses concurrent signal assignment rather than the process structure that the primary implementation uses. It receives input data in BCD form and converts it to a seven- segment output. In this project, a logic-1 signal is fixed to illuminate the segment and logic-0 signal turns the segment off. The display multiplexes 8 seven segment displays on one set of display driver lines. When the display is rapidly scanned using the FPGA it appears to the human eye as if all of the digits are lit simultaneously. Further, it saves FPGA resources since it only consume 16 I/O lines but have eight digits to be displayed.

H. Digital Clock Module

This module mainly displays the time in blocks of hours, minutes and seconds. It also monitors the number of days that have elapsed and sends a signal to the billing module to signal the end of a month. When the module is reset, 00:00:00 is displayed. This module employs counters for seconds, minutes, hours, ten seconds, ten minutes and then hours. The counter for seconds and minutes are fixed to count from 0 to 9. The counter for ten seconds and ten minutes are fixed from 0 to 5. The counter for hours is fixed from 0 to 4. Finally, the counter for ten hours is fixed from 0 to 2. The output from these counters is fed to decoders that will convert the integer output of the counters to signals that drive the seven-segment display. The LEDs are fixed at '0' to represent lighted and a '1' represents not lighted.

This module also has an internal counter which increments by 1 each time 24 hours is reached. This is to keep track of the number of days that has elapsed. Once the count value reaches 30, the month indicator will be '1' to indicate the end of the month has been reached. This '1' output will be sent as input to the billing module to determine the total amount for a particular month to be displayed on the seven-segment. In the source code, compounded if statements was used so that there would not be a need to define states, because otherwise the number of lines in the code will increase.

I. User Interface Terminal

The user interface terminal serves as the main interface between the consumer (as well as the meter reader) and the digital energy meter system. It enables the consumer to monitor electricity consumption and the meter reader for ease of meter reading at the end of every month. The terminal consists of 1 LED, 22 seven-segment displays and 3 push buttons. Reset button is used to reset the whole system and the Power button is to on or off the system. These two buttons are embedded in the digital meter and can only be controlled by the party who will be carrying out the meter installation. The LED (also called the supervisory indicator) is used to indicate the end of every month. This reminds the customer on his/her electricity bill payment. A toggle button is used to switch between the current total cost for the present month and the previous month. 22 seven-segment displays are present to display the current time (6 displays), total energy consumed in units (8 displays) and total cost (8 displays). The current time (6 seven-segment displays) are controlled by 42 signal lines from the FPGA. The total energy and total cost, both using 8 seven-segment displays are controlled by 15 signal lines from the FPGA (7 data lines and 8 enable lines- one for each display). The BCD to seven-segment conversion is done internally in the FPGA and the data is written to each display at a frequency fast enough to avoid flickering of the display.

IV. SIMULATION

The system was coded in IEEE-compliant VHDL and compiled and simulated using the Aldec Active-HDL version 3.5 suite. This provides an opportunity to detect and correct errors early in the design process. Each subcomponent was designed and tested in isolation before being incorporated into the higher levels of the design. A test bench is used to in simulation.

A. Individual Modules

The individual modules were first simulated to verify their functionalities. Each module was fed a fix inputs and the correct outputs were observed.

After the successful individual simulations were performed, the modules were integrated together. This enables detailed simulation at the top level. There are two integrations performed, as discussed in sub sections *B* and *C*.

B. Structural Combination (Part 1)

This first integration consists of Clock Divider, Scale Down, Power Calculation and Power Store modules. This structural combination of four modules takes in the current and multiplies by voltage to obtain power. The power value is compared with the previous value stored, if it is equal, it will be added up and the counter will be incremented by '1'. The counter value represents the total number of clock cycles that has elapsed for a particular power value. If it is not equal, the total power and its corresponding duration will be sent out as input to the Energy_calculation module to calculate the corresponding energy value. The energy values will be added up in the Energy_store module until 1kWh is reached, which represents 1 unit. The Billing module will calculate the corresponding cost for for the total number of units consumed. Figure 2 shows the timing diagram for the simulation performed.

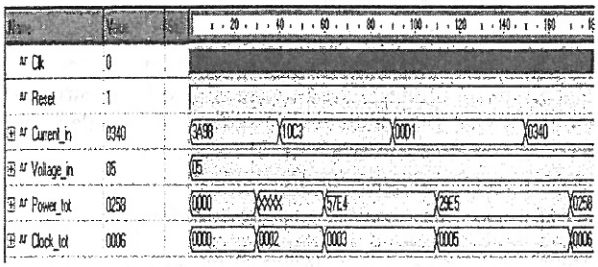

Figure 2: Part 1 Structural Combination

C. Structural Combination (Part 2)

The second combination consists of Binary2BCD Energy, Binary2BCD Billing, Display Energy, Display Billing, Status Check Unit, Status Check Bill, Digital Clock and Seven Segment Decoder. This structural combination of 8 modules takes in the incoming values of the total unit of electricity consumed and its corresponding cost and converts the values to BCD to be displayed at the seven-segment display. The digital clock will send a signal at the end of each month and this will reset the system to calculate the total number of units and billing for the next month. The previous value will be stored separately. Figure 3 shows the timing diagram of the initial results. Figure 4 shows the timing diagram of the final results.

It is concluded from the Figure 3-4 that all the modules performed as expected from the individual simulation as well as the top-level simulations.

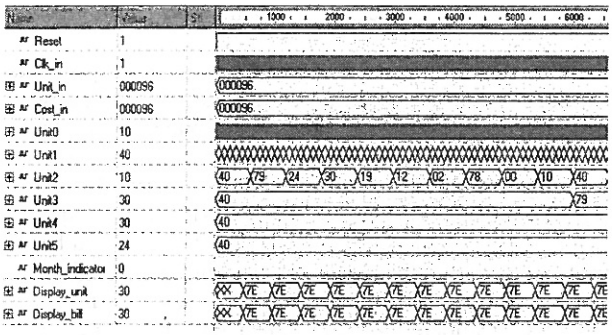

Figure 3. Part 2 Structural Combination (Initial result)

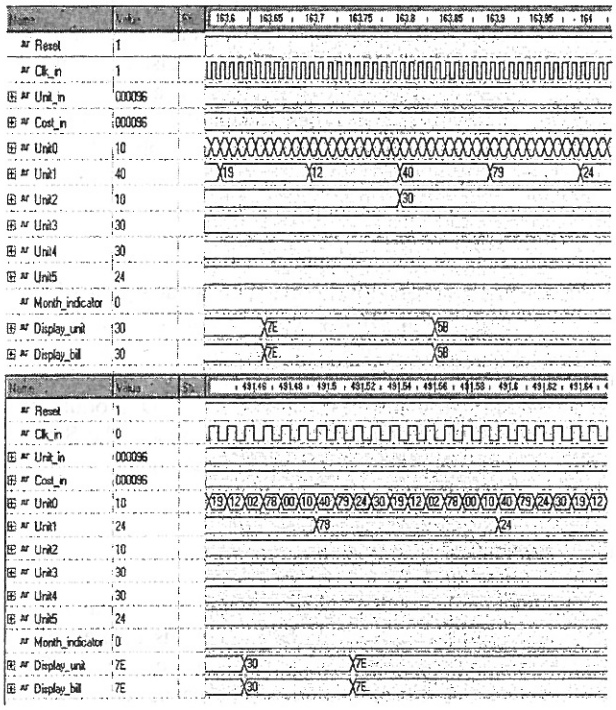

Figure 4. Part 2 Structural Combination (Final result)

V. SYNTHESIS

The individual components and finally the entire architecture were synthesized using Synplify version 7.0 by considering ALTERA FLEX10K family, and the particular chip is EPF10K10LC84-3 with a PC84 package. The FLEX 10K family provides the density, speed, and features to integrate entire systems, including multiple 32-bit buses into a single chip. The maximum frequency was achieved 31.4 MHz. The design took a minimum resource i.e. 273 logic element, which is 47.4% of the device EPF10K10LC84. Table 1 shows a details report of the usage of resources. Partial view of sheet 4 (out of 16) of the top level RTL view and the sheet 3 (out of 10) of top-

level technology view are shown in Figure 5 and 6 respectively.

TABLE 1
THE USAGE OF RESOURCES

Logic Resources (EPF10K10LC84-3)	Logic resources: 273 LEs of 576 (47.4%)
	Number of Nets: 241
	Number of Inputs: 976
	I/O cells: 42
	Cells in logic mode: 237
	Cells in cascade mode: 29

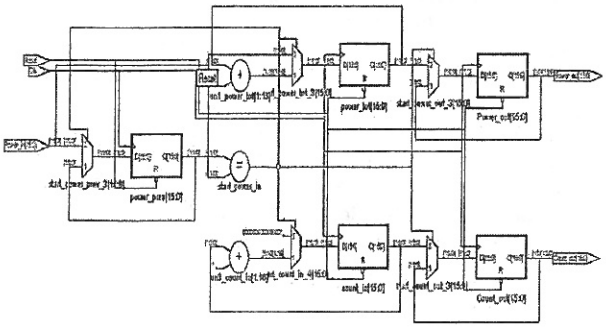

Figure 5: Top level RTL view – partial sheet 4 of 16

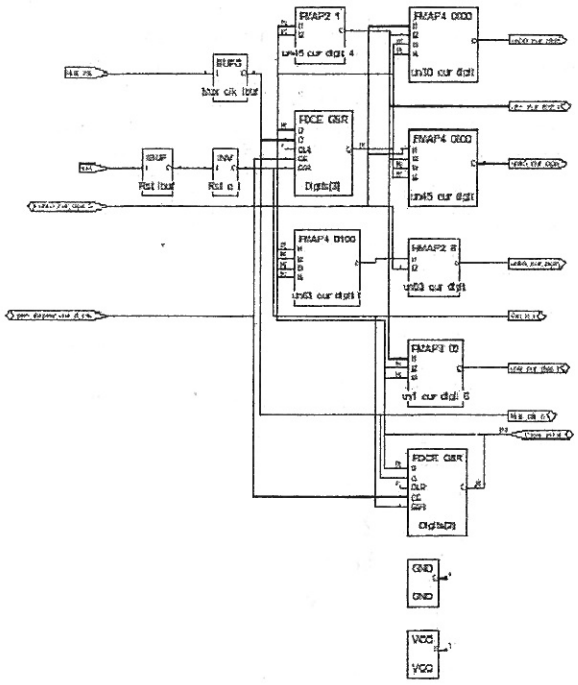

Figure 6: Top level technology view – sheet 3 of 10

VI. CONCLUSION

The objective of this project was to design and implement a digital energy meter system on FPGA device EPF10K10LC84-3 using VHDL. The initial design parameters were determined, and general system architecture was designed. General system functions were then broken down into modules, which were designed and coded separately using VHDL. After successfully compiling and simulating each module individually, the modules were combined using structural VHDL to simulate the entire system and successfully synthesized with the tools Synpify 7.0 with achieved maximum frequency of 31.4 MHz and a minimum resource usage of 47.4% of the total logic elements. The hardware implementation demonstrated complete, correct functionality and met all the initial system requirements. Using IEEE floating point-numbering format would enhance the further accuracy of the design. Currently, we are conducting the further research that considers the floating-point numbering format, the further reduction in the hardware complexity and increase the critical frequency in terms of synthesis by improving the VHDL coding.

VII. REFERENCES

[1] Djokic, B., Bosnjakovic and P. Bogovic, M., "New method for reactive power and energy measurement", Instrumentation and Measurement, IEEE Transactions on, vol. 41, Issue-2, April 1992, pp. 280-285.

[2] S. Toral, S.M. Quero and L.G. Franquelo, "Power Energy Metering based on Random Signal Processing", (EC-TPS), IEEE, 1998, vol. III, page 435

[3] A.Hashimoto, K. Yasui and S. Kusui, "Self-Calibrating Standard Watthour Meter", Japan Electric Meters Inspection Corporation, Nippon Institute of Technology Metering and Tarriffs for Energy Supply, 3-5 July 1996, Conference Publication No. 426, IEEE

[4] P.A.V. Loss, M.M. Lamego Vieira and G.C.D. Sousa, "A Single Phase Microcontroller Based Energy Meter", IEEE Instrumentation and Measurement, Technology Conference, 1998

[5] M.F. Schwendtner, "Digital Measurement System for Electricity Meter", 3-5 July, 1996, Conference Publication No. 426, IEEE

[6] Alexandre Schmid, Yusuf Leblebici, and Daniel Mlynek, "Hardware Realization of A Hamming Neural Network with On-Chip Learning," IEEE International Symposium on Circuits and Systems, Monterrey CA, 1998, vol. III, pp. 191-194.

[7] B. K. Fawcett, "Tools to Speed FPGA Development," IEEE Spectrum, November 1994, vol. 31, pp. 88-94.

[8] C. E. Cummings, "Verilog Simulation Xilinx Designs," Proc. Int. Verilog HDL Conf., Santa Clara, CA, 1994, pp. 93-100.

[9] Dragun Pejic, Vladimir Vujicic, "Accuracy limit of High-Precision Stochastic Watt-hour meter", IEEE Transaction June 2000

[10] J. Zhang and W. Oghanna , "A DSP Based Electricity Meter with Remote Reading", Proceedings of ICSP'98

[11] Stefan Sjoholm, Lennart Lindh, "VHDL for Designers", 1st edition, Prentice Hall, Europe, 1997, 473 pages

[12] Stephen Brown, Zvonko Vranesic, "Fundamentals of Digital Logic with VHDL Design", International Edition, McGraw-Hill International Edition, 2000, 840 pages.