

## DEVELOPMENT OF AN EFFICIENT AND FLEXIBLE ALGORITHM FOR IMPLEMENTING A DIGITAL HOUSEHOLD ENERGY METER: AN FPGA APPROACH

M. I. Reaz<sup>\*</sup>, M. S. Rahman<sup>\*\*</sup>, F. Mohd-Yasin<sup>\*</sup>, M. S. Sulaiman<sup>\*</sup>, F. Choong<sup>\*</sup> and A. Alaudeen<sup>\*</sup>

<sup>\*</sup>Faculty of Engineering, Multimedia University, 63100 Cyberjaya, Selangor, Malaysia

<sup>\*\*</sup>Dept. of Electrical and Electronic Engineering, BUET, Dhaka, Bangladesh

E-mail: saif672@yahoo.com

### ABSTRACT

This paper describes the development of an efficient algorithm that implements a flexible and affordable digital energy meter intended for home usage. As a first step, the algorithm is downloaded into an FPGA prototype board. The algorithm architecture comprises of four main modules: power, energy, billing and display. Two digitized inputs, which are assumed to come from 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 timing analysis has been performed on Aldec Active HDL, and 'Synplify' has been used for circuit synthesis. Using digital synthetic test data it has been proven that the model has been tested successfully. This work forms the first phase of developing a commercial but affordable digital energy meter for home usage employing ASIC.

### 1. INTRODUCTION

Although very popular, the analogue electromechanical energy meters lack stability; and the accuracy drifts with operating environment and long periods of time. These meters use a series of dials to display kWh consumption [1]. Electronic solutions have been used in the past to duplicate the behaviour of a classic electromechanical meter [2-4]. In this paper, a digital hardware approach is introduced, which has evolved dramatically over the past four decades [5]. Until the 1960s, logic circuits were constructed with bulky components, and the standard chips had fixed functionality. It is now possible to construct chips, known as *programmable logic devices* (PLDs), that can be configured by the user to implement a wide range of different logic circuits. Field-programmable gate arrays (FPGAs)

are special types of PLD, which are highly attractive options for hardware implementation of the proposed digital energy meter. VLSI technology is used here, as 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 is modeled using VHDL. Electricity usage at households is generally 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, the readings can be easily observed from the display, which gives the total energy units consumed and the corresponding charge at any time, and thus it offers the user a choice of controlling the energy usage [10].

### 2. DESIGN OVERVIEW

In the proposed implementation of the digital energy meter, all the different modules are made to work in parallel and the system is run by a single system clock to operate in full synchronism. This increases the speed of the entire system [11]. First, the bare bone block diagram for the meter is designed and the inputs and outputs are determined. Two input values (single-phase voltage and single-phase current) are fed into the digital energy meter and the outputs are the energy consumed and its billing. This meter can calculate the total amount of energy consumed at any point in time.

The system takes in incoming instantaneous values of current (0-15000mA) and voltage (scaled down to 5V); multiply the values together to obtain the total electrical power. The power and the duration are

then multiplied together to obtain the total energy for the particular duration. The energy values keep on accumulating, and one unit of energy is used up each time it reaches 1 kWh. The total number of billing units and the corresponding charge are then converted to BCD digits and displayed using 7-segment decoders.

### 3. SYSTEM LEVEL DESIGN

The core of the energy meter has been sub-divided into several distinct modules, each of which performs a specific task. The top-level diagram is shown in Fig. 1.

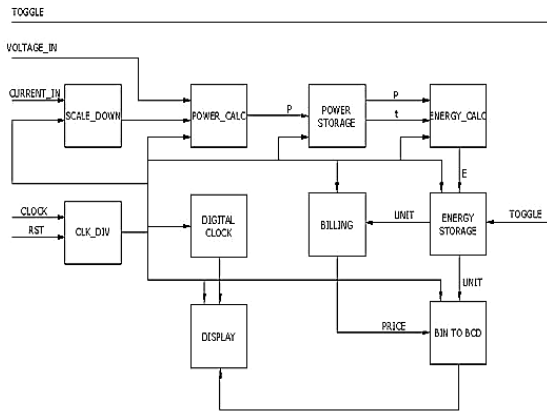


Fig. 1: System top-level diagram

The function of each individual module is then described in sub-sections 3.1 to 3.9.

#### 3.1 Clock Divider Module

The system clock frequency is 20MHz, which is too fast to verify the design, and thus a clock divider is added. It takes a 20MHz clock input and outputs a 100Hz square wave. Dividing 20MHz by 200000 does this job, which is stored in an 18-bit counter, and it reaches the final value flipping a bit each time.

#### 3.2 Scale down Module

This module takes in input current values and outputs an 11-bit value in binary. The output is fed to the Power Calculation module.

#### 3.3 Power Calculation Module

The function of this module is to multiply the values of current by voltage to produce the individual values of power  $P$ , where  $P = v \times i$ . 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. The output is then fed to the Power Store module.

#### 3.4 Power Store Module

This module 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. It first compares the incoming power value with the previously recorded value. If it is the same, the incoming value is added to the total power and the counter is incremented by '1'. This value denotes the amount of time in terms of number of clock cycles elapsed since the last change was recorded. It is the duration at which a particular power value remains constant. If the power is not the same, the value of the total power and the corresponding time is sent to the Energy Calculation module.

#### 3.5 24 Binary to BCD Converter Module

This module takes in a 24-bit binary number and outputs a 32-bit binary number, which represents a set of eight BCD numbers.

In designing this module, a very efficient technique, called the "add-3" conversion algorithm is 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 BCD digit, if necessary.

#### 3.6 Status Check Module

This module takes in the output signal of the Binary to BCD module as input, 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 initiate the first conversion.

#### 3.7 Display Module

The Display module takes in a 32-bit binary input and generates a display of 8 digits by lighting each digit up in sequence. 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.

#### 3.8 Digital Clock Module

This module displays the time in blocks of hours, minutes and seconds. It employs counters for seconds, minutes, hours, ten seconds, ten minutes and then hours. The outputs from the counters are fed to decoders that convert the integer output of the counters to signals that drive the seven-segment display. This module also has an internal counter 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 that the end of the month has been reached. This '1' is sent as input to the billing module to determine the total amount for that month and display the output.

### 3.9 User Interface Terminal

The user interface terminal enables the consumer to monitor electricity consumption and the meter reader to take meter reading at the end of every month. The terminal consists of 1 LED to indicate the end of the month, 22 seven-segment displays and 3 push buttons.

## 4. SIMULATION

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

### 4.1 Individual Modules

The individual modules are first simulated to verify their functionalities. Each module is fed a fixed input and the appropriate outputs are observed. After the successful individual simulations, the modules are integrated together, which enables detailed simulation at the top level. The two integrations performed are discussed in sub-sections 4.2 and 4.3.

### 4.2 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. Fig. 2 shows the timing diagram for the simulation performed. Then the Energy calculation module calculates the energy.

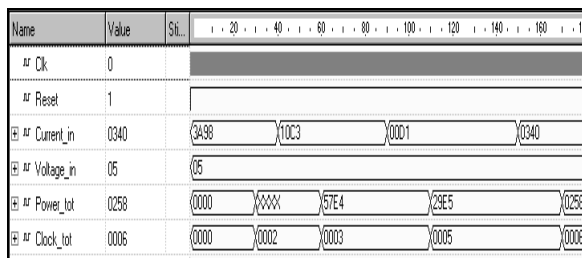


Fig. 2: Part 1 Structural Combination

The energy values are added up in the Energy\_store module until 1kWh is reached. Finally, the Billing module calculates the total cost of energy consumed.

### 4.3 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 the corresponding cost, and converts the values to BCD to be displayed at the seven-segment display. Fig. 3 shows the timing diagram of the initial results and Fig. 4 of the final results.

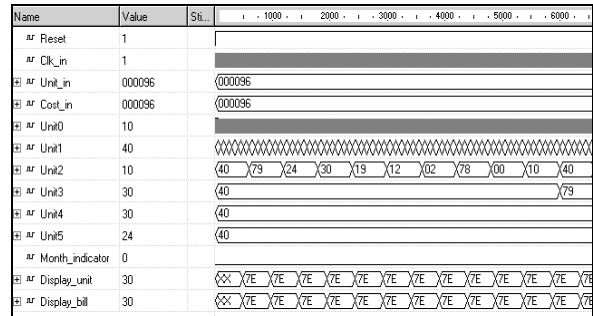


Fig. 3: Part 2 Structural Combination (Initial result)

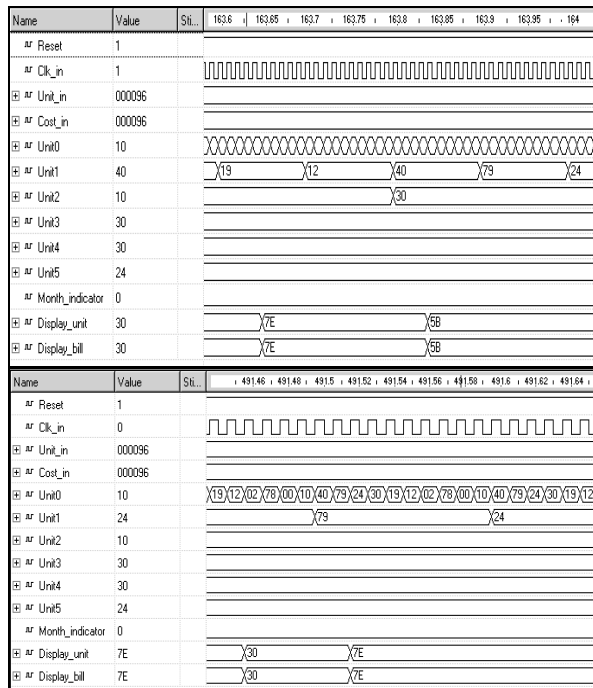


Fig. 4: Part 2 -Structural Combination (Final result)

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

## 5. SYNTHESIS

The individual components and finally the entire architecture were synthesized using "Synplify version 7.0" by considering ALTERA FLEX10K family, and the chip is EPF10K10LC84-3 with a PC84 package. It took a minimum resource (47.4% of the device EPF10K10LC84). Table 1 shows the

