Super Fast Battery Charger Design with NeuFuz™

National Semiconductor
Application Note 959
Dilip S, Zafar Ullah
June 1994

FEATURES

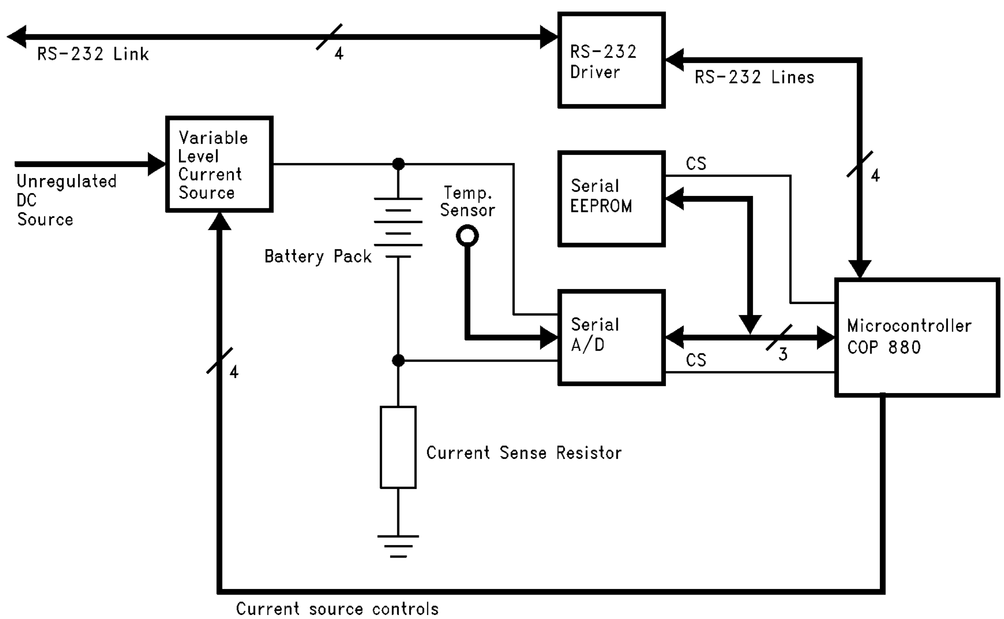
- Provides accurate Fast Charge Control based on NeuFuz algorithm
- Communicates battery status, historical battery data, and present state of charge information with host
- Battery operational limits accessed from a EEPROM
- Low power consumption during normal operation
- Halt-mode operation possible to further reduce power consumption
- Provides maximum and minimum temperature, voltage, recording in EEPROM memory
- Records the number of charge/discharge cycles
- Firmware extremely modular, and is designed to be either stand alone unit or embedded in battery packs
- Firmware for new charging algorithm design is possible without writing any microcontroller code

INTRODUCTION

Fast Charging of secondary cells is a problem which is difficult, if not expensive, to solve using conventional techniques. Most conventional fast charging techniques rely on some trip-point based approach, which subjects the secondary cells to undue thermal and chemical stress and could reduce the life of the cell. Most conventional techniques result in overcharging the battery, which wastes energy and reduces battery life.

This application note focuses on a design based on National Semiconductor's patented NeuFuz based Neural-Fuzzy technology to arrive at a charge control scheme implemented on National Semiconductor's low-cost COP8™ microcontroller. NeuFuz technology allows for the automated generation of a fuzzy logic control engine based on training data. In this case, the training data was taken from the battery characteristics provided by the battery manufacturer. The NeuFuz based design methodology followed here allows for tailoring the charger to suit different battery characteristics without having to rewrite the code for battery charging.

Battery chemistries are changing rapidly, and each battery chemistry results in different battery characteristics. The NeuFuz-based charge controller design provides a quick and low cost path to tailor the charger to specific battery chemistries. No microcontroller code need be written in to tailor the battery charger design to different battery characteristics.



TL/DD/12096–1

FIGURE 1. NeuFuz Based Fast Charger and Battery Management Unit Block Diagram

COP8™, NeuFuz™ and Simple-Switcher™ are trademarks of National Semiconductor Corporation.

RRD-B30M75/Printed in U. S. A.

The Battery Management Unit performs data collection, battery parameter recording and communications with the host system. The battery management functions monitors the batteries usage patterns, and saves this information in the EEPROM. It also provides real-time information on the state of charge of the battery.

HARDWARE CONFIGURATION

The hardware for the fast charger and battery management unit demo is shown in *Figure 1.* The hardware comprises of a COP880 microcontroller with 4k ROM and 128 bytes of RAM. Around the microcontroller, there is a low cost A/D converter, 16 x 128-bit EEPROM memory, a RS-232 driver and a 3.5A constant current source. This hardware provides for all the functionality required in a battery management unit, and intelligent charger. The hardware is built with parts which consume very little power. All the parts used here are also available in compact surface mount packaging, which allows this design to be built into battery packs. The constant current source uses a National Semiconductor Simple-Switcher™ switching regulator circuit with current feedback. The level of current output by the switching regulator circuit is controlled by toggling four I/O lines of the COP8 microcontroller.

Firmware modules accomplish the following tasks:

1. Battery data comprising of voltage, current and temperature are acquired using the 4-channel serial A/D converter.
2. Charge control is performed by an intelligent NeuFuz based algorithm, which uses the parameters acquired by the A/D converter and outputs the charging current required. The output of the charge control algorithm modulates the current level output by the constant current source.
3. Charge Status monitoring is performed using the parameters acquired from the battery.
4. Serial communications with the host provide it with information on the charge status, historical battery parameters and the present levels of battery usage.
5. Historical data of the maximum and minimum voltage, current and temperature are recorded in the EEPROM.
6. Historical information on the number of charge/discharge cycles are recorded in the EEPROM.

The firmware for the battery management and charger unit is extremely modular. It can therefore be tailored to design a system with a mix and match of needed features.

CHARGING SCHEMES

A multitude of fast charging schemes exist, suitable for use with quick-charge batteries. The most commonly used fast charge control schemes are described as follows:

1. *Constant-rate fast charge with trip points.* A high current (Charge rate $\leq$ 1C) constant charging rate is maintained until one or more of the measured parameters exceeds the safe limits (trip point). When such a condition is detected the charging level is reduced to a safe level. This is continued until the battery is fully charged. This approach is easy to implement, but does not achieve very fast charge times.
2. *Multiple rate fast charge with trip points.* The charger is capable of charging the battery at multiple rates, and the transition between the various charging rates is based on several trip points. These trip points are resident in firmware and are activated on the measured battery parameters, which alter charging levels. This continues until the battery is fully charged. This type of charge control scheme is preferred but only about 50 trip points can be accommodated in 2.1 kbytes of ROM. An elaborate scheme is required to check for the activation of the trip points and determining the level of charge desired.
3. *NeuFuz based fast charge control.* Instead of using multiple trip points, the charge control is performed by an intelligent algorithm, which accounts for nonlinearities in the battery charging characteristics. This algorithm can effectively compress a look up table of up 600 trip point with the appropriate inference engine in 2.1 kbytes of ROM memory in a COP8 processor. The charging algorithm is based on fuzzy logic and is inherently nonlinear. The charging algorithm is derived from training the battery characteristics using a neural network to generate the fuzzy logic implementation.

CHARGER DESIGN USING NeuFuz

This application note shows the complete Neuro-Fuzzy development cycle to create the charging algorithm. The following steps determine the implementation:

— Define the data set by studying battery parameters.
— Preprocess the data set to tailor it to the hardware.
— Configure the NeuFuz neural network.
— Train the neural network.
— Find an optimized fuzzy representation.
— Generate code.
— Integrate code with other code in the target system.

Data Set Definition

The first step in charger design is to decide on the control input parameters, which represent the internal state of the battery. The parameters chosen should be easy to measure, using a low cost sensor and A/D converter. The relationship of these parameters to battery life and battery environment must be known. It should be known if the battery exhibits unique characteristics such as the memory effect. The battery should be a fast charge type (i.e., typical fast charge batteries are rated for charge time of 1.5 hours or less). The data set should contain information about nonlinearities in the battery characteristics and more data points are needed around the critical nonlinear portions of the characteristics.

Preprocess The Data Set

Once these parameters are known, a table containing the values and corresponding output are made. It is recommended that sufficient data points are available to account for the nonlinearities of the system. The data points must cover all the possibilities of input values within the input space. For a nickel-cadmium battery pack, 400 to 700 data point were adequate to represent the input space. This table must be in the form of an ASCII flat file. The NeuFuz user manual provides useful information on preparing the ASCII flat file.

Configure The Neural Net

The configuration parameters for training the neural net, the number of fuzzy membership functions desired and the absolute accuracy desired from the system need to be defined.

Train The Neural Network

Training the neural network is an iterative process. This requires the user to study the error generated during training and to modify the learning neural network's parameters.

Find An Optimized Fuzzy Representation

The fuzzy logic solution obtained from the trained neural network needs to be verified for accuracy and size. Verify the accuracy of the solution over the entire range of input space. This fuzzy logic solution can be further optimized directly from NeuFuz using a deletion factor to eliminate some of the less significant rules, with minimal effect on the accuracy of the solution.

Generate Code

Once the neural network has been trained and the accuracy of the fuzzy logic solution found acceptable, NeuFuz package automatically generates COP8 code. The code generated by NeuFuz comprises of relocatable COP8 assembly code. The code generated also includes the definitions for the RAM requirements. A log file indicating the amount of RAM and ROM used for this algorithm is also generated. The COP8 code includes some general purpose math routines for multiplication and division and can be shared by other firmware modules.

Integrate Code With Other Firmware Modules

To integrate the NeuFuz generated fuzzy logic assembly code with the rest of the code is required to pre-process the inputs and output externally. In this application, it was required to process the input data acquired from the A/D converter and scale it. The scaled input is an 8-bit value and is stored in RAM locations IN1, IN2 and IN3. The fuzzy logic algorithm reads data from these RAM locations and writes the output in RAM locations labelled OUT1 to OUT6.

One of the most significant benefits of using the fuzzy logic assembly code produced by NeuFuz is that the RAM used by it can be reused by other assembly modules.

Should the NeuFuz generated code be interrupted during execution, it is necessary to protect all the contents of RAM used by NeuFuz. Special care must be taken not to overwrite the RAM locations that NeuFuz uses.

RESULTS

The results of NeuFuz based fast battery charger are compared with conventional fast battery charger using multiple trip points in Table I.

TABLE I. Comparison of NeuFuz and Conventional Fast Chargers

Item	Conventional Trip-Point Based Charge Controller	NeuFuz Based Charge Controller
Processor Used for Charger	COP8	COP8
Charging control	Trip point based PWM current controller, resulting in 3 charging phases	Adaptable multilevel current controller
Charging cycle for NiCd batteries	30 ~ 70 minutes	10 ~ 20 minutes
Code size for Charging Algorithm (ROM)	800 bytes	2100 bytes
Data Memory Used (RAM)	30 bytes	52 bytes
Loop Update time	50 ms ~ 100 ms	100 ms ~ 150 ms
Data Measured and Recorded	Temperature, voltage, current	Temperature, voltage, current

ANALYSIS OF RESULTS

Fast Charge Time

Fast charge time is accomplished by using a nonlinear control algorithm, which can control the charging current by closely following optimal battery characteristics. Higher charge currents are used on the battery during initial stages of the charge cycle. Subsequently, the charge levels are lowered to by monitoring the battery parameters, as determined by the NeuFuz based control engine.

Charge Control Algorithm Code Size

The control code embedded in microcontroller's ROM for charging comprises of two major sections. The first section is the fuzzy logic inference engine, and the second is the fuzzy logic data tables, which correspond to battery specific charging characteristics. The size of the fuzzy logic inference engine is fixed, and is approximately 600 bytes. The fuzzy logic rule base has 294 rules for a Nickel Cadmium battery pack. The number of fuzzy membership functions correspond to 7 for voltage, 7 for temperature and 6 for time. The algorithm is compute intensive and does the loop update 6 to 10 times per second.

Accuracy Of State Of Charge Indication

The state of charge is based on calculating the level of charge from the measured battery parameters of voltage and temperature. Rechargeable batteries exhibit varying self discharge characteristics with time and temperature. The accuracy of the state of charge indication is $\pm 3\%$ when only self discharge and load are considered. The ROM code size for the state of charge indication is around 400 bytes.

Maximum Charge Current

The present current source hardware can deliver up to 3.5A of charge current and provides up to 5C rate for batteries of 700 mAH capacity. National Semiconductor offers many types of switching regulators, and changing the rating of the current source is just a matter of altering the switching regulator.

PRACTICAL IMPLICATIONS

The other features offered by the existing hardware unit are recording of battery parameters in a EEPROM and communications with the host system.

The historical data record the maximum and minimum battery temperature, battery voltage, previous charge level and number of complete discharge cycles. There is space allocated in the EEPROM to record the battery manufacturer's warranty information. The maximum and minimum safe levels of battery operation are also stored in the EEPROM. This design to acts as a final safety cut-off to shut the charger should the charge control algorithm or peripheral hardware fail.

Serial host communications are performed using a software UART written in firmware with RS-232 protocols with XON/XOFF handshaking. The present battery conditions, the historical battery values stored in the EEPROM can be sent to the host. This serial link can be used to program the warranty information, safe maximum and minimum battery charging parameters in the EEPROM.

The hardware and firmware designs are modular and lend themselves to the design of both stand-alone chargers and embedded chargers in battery packs. Most stand-alone chargers feature quick battery charging with accurate state of charge display in the charging unit. Most embedded chargers in battery packs feature charge control, state of charge indication on the battery pack, historical battery information recording, and serial host communications. Since the battery parameters are very closely coupled to the battery type, battery packs with embedded battery management units are becoming very popular.

National Semiconductor Corporation
2900 Semiconductor Drive
P.O. Box 58090
Santa Clara, CA 95052-8090
Tel: 1(800) 272-9959
TWX: (910) 339-9240

National Semiconductor GmbH
Livry-Gargan-Str. 10
D-82256 Fürstenfeldbruck
Germany
Tel: (81-41) 35-0
Telex: 527649
Fax: (81-41) 35-1

National Semiconductor Japan Ltd.
Sumitomo Chemical Engineering Center
Bldg. 7F
1-7-1, Nakase, Mihama-Ku
Chiba-City,
Ciba Prefecture 261
Tel: (043) 299-2300
Fax: (043) 299-2500

National Semiconductor Hong Kong Ltd.
13th Floor, Straight Block,
Ocean Centre, 5 Canton Rd.
Tsimshatsui, Kowloon
Hong Kong
Tel: (852) 2737-1600
Fax: (852) 2736-9960

National Semiconductores Do Brazil Ltda.
Rue Deputado Lacorda Franco
120-3A
Sao Paulo-SP
Brazil 05418-000
Tel: (55-11) 212-5066
Telex: 391-1131931 NSBR BR
Fax: (55-11) 212-1181

National Semiconductor (Australia) Pty, Ltd.
Building 16
Business Park Drive
Monash Business Park
Nottinghill, Melbourne
Victoria 3168 Australia
Tel: (3) 558-9999
Fax: (3) 558-9998