Numerical relay for overcurrent protection using TMS320F2812

Article · January 2010

4 authors, including:

Yin Lee Goh
6 PUBLICATIONS **19** CITATIONS

Agileswari K. Ramasamy
Universiti Tenaga Nasional (UNITEN)
33 PUBLICATIONS **61** CITATIONS

Aidil AZWIN Zainul Abidin
Universiti Tenaga Nasional (UNITEN)
14 PUBLICATIONS **62** CITATIONS

Numerical Relay for Overcurrent Protection using TMS320F2812

YIN LEE GOH, AGILESWARI K. RAMASAMY, FARRUKH HAFIZ NAGI,
AIDIL AZWIN ZAINUL ABIDIN
Centre for Communication Service Convergence Technologies (CCSCT)
Department of Electronics and Communication Engineering
Universiti Tenaga Nasional
Jalan IKRAM-UNITEN, 43000 Kajang, Selangor
MALAYSIA
yin_lee@hotmail.com, agileswari@uniten.edu.my, farrukh@uniten.edu.my, aidilazwin@uniten.edu.my

Abstract: - Overcurrent protection is a very important element in power systems. This protection is essential in order to minimize disturbances caused by any failure in the system and to ensure continuous power delivery. Overcurrent relays are one of the devices used to achieve these purposes. The overcurrent relays initiate the corrective mechanism to determine the operation time of the relay. Thus, the overcurrent relays must have high reliability and accuracy to detect any fault currents present and determine the operation time. The entire system will be tremendously affected if the relays fail to trip or cause mal-tripping. An overcurrent relay is implemented on a high speed and high performance digital signal processor (DSP). The simulation was carried out using MATLAB/Simulink. Then, the relay was implemented on TMS320F2812 based on two different methods. In the first method, the relay simulation model from MATLAB/Simulink is downloaded into the DSP whereas in the second method, code in C is directly written on the DSP to represent the relay. The relay performance was compared with the IEC 255-3 standard and the outcomes obtained are encouraging.

Key-Word: - Power system protection, protective relaying, overcurrent protection, digital signal processors

1 Introduction

Numerical relays are the next generation relays which will soon replace the use of digital relays. The difference between the numerical relay and digital relay is the types of processor used in the relay. The numerical relay is embedded with digital signal processor (DSP) whereas a digital relay uses a microprocessor [1].

Numerous researchers have used microprocessor to implement the relay as in [2-7]. The microprocessors used for digital relays have limited processing capacity and memory causing the functionality tends to be limited and restricted largely to the protection function itself [1]. These disadvantages of using microprocessors can be overcome by using DSP.

The DSP is able to improve the performance of the relay which is very essential in order to meet the fundamental protective requirements such as reliability, sensitivity, selectivity and speed [8-9]. The processing carried out using DSP is optimized for the real-time signal processing applications [1]. Therefore numerical relays are faster.

Numerical relays are used to provide a wide range of protection functions such as overcurrent, directional overcurrent, undervoltage, overvoltage, distance and others [10]. Overcurrent protection is one of the basic protective relaying principles [11]. It is considered as the backbone of any protection strategy such for distribution circuits [12]. The basic principle of the overcurrent protection is when the current flowing into the overcurrent relay exceeds a setpoint amount, the relay operates with or without an intended time delay and trips the associated circuit breakers [13]. The time delay is also known as operation time is computed from the protection algorithm embedded in the processor.

The DSP embedded in the numerical relay is able to provide higher accuracy results for the relay which is important for power system. This processor increases the speed of the processing time since the processor has higher processing capability. This ensures that the relay does not fail to trip and avoids mal-tripping occur in the system. Thus, the implementation of overcurrent relay on DSP board, TMS320F2812 is described in this paper in order to investigate the performance of the relay using this processor.

The overcurrent relay of inverse definite minimum time (IDMT) type is adopted in the system. TMS320F2812 is a high performance processor which is operating at frequency of 150MHz [14]. The improved Harvard architecture of TMS320F2812 enables processing of high amount of data. This processor is also able to provide high speed precision calculation for the mathematical algorithm involved in the processor [15]. These features are very suitable for implementing the overcurrent relay for real-time control applications.

This paper describes the implementation of overcurrent relay using TMS320F2812. Comparison results between simulation in MATLAB/Simulink and execution on TMS320F2812 based on two different implementation methods are presented in next section. For the first implementation method, the simulation model from MATLAB/Simulink is directly download into the DSP whereas for the second method, coding using C-Language is written in the DSP to represent the overcurrent relay.

2 Principle of Operation

The overcurrent relay of IDMT is the relay which starts to operate after the intended time delay. The time delay is also known as operation time. The advantage of the IDMT relay is that the greater the fault currents, the shorter are their operating time [16].

According to IEC 255-3 standard [17-20], the characteristics of IDMT overcurrent relays are represented as in equation (1). Different types of inverse characteristics which are standard inverse, very inverse, extremely inverse and long inverse can be obtained by varying α and C.

$$t = \frac{C}{(\frac{I}{I_S})^{\alpha} - 1} \times TMS \tag{1}$$

where t - relay operation time
C - constant for relay characteristics
TMS - time multiplier setting
I - current detected by relay, $I > I_S$
I_S - current setpoint
α - constant representing inverse time type, $\alpha > 0$

Table 1. Parameters for Different Types of Inverse Characteristics

Relay Characteristics Type	α	C
Standard inverse	0.02	0.14
Very inverse	1	13.5
Extremely inverse	2	80
Long inverse	1	120

3 Proposed Overcurrent Relay Model

The proposed overcurrent relay is modeled in MATLAB/Simulink before implemented on TMS320F2812 as shown in Fig.1. The overcurrent relay of IDMT type is adopted in the proposed model.

In the simulation model, the input currents of sine waveform with frequency of 50Hz are generated. The inputs are quantized since the operation of numerical relay is processed in the digital form. The input currents are sampled at the sampling frequency of 1kHz. The sampling frequency must be at least twice of the fundamental frequency. This is to ensure that the Nyquist criterion is fulfilled so that the aliasing of the input signals are avoided.

The total of 20 input current samples is obtained for each cycle of the fundamental frequency. These samples are used to compute the root mean square (RMS) value of the input current. The RMS computation is necessary in order to extract the fundamental component of the input current samples. These calculated RMS currents are supplied into the overcurrent relay.

Then, the RMS values are compared with the setpoint value to obtain the current ratio. In this proposed model, the current setpoint is set as 150A. Other setpoint values can be used according to the types of system involved.

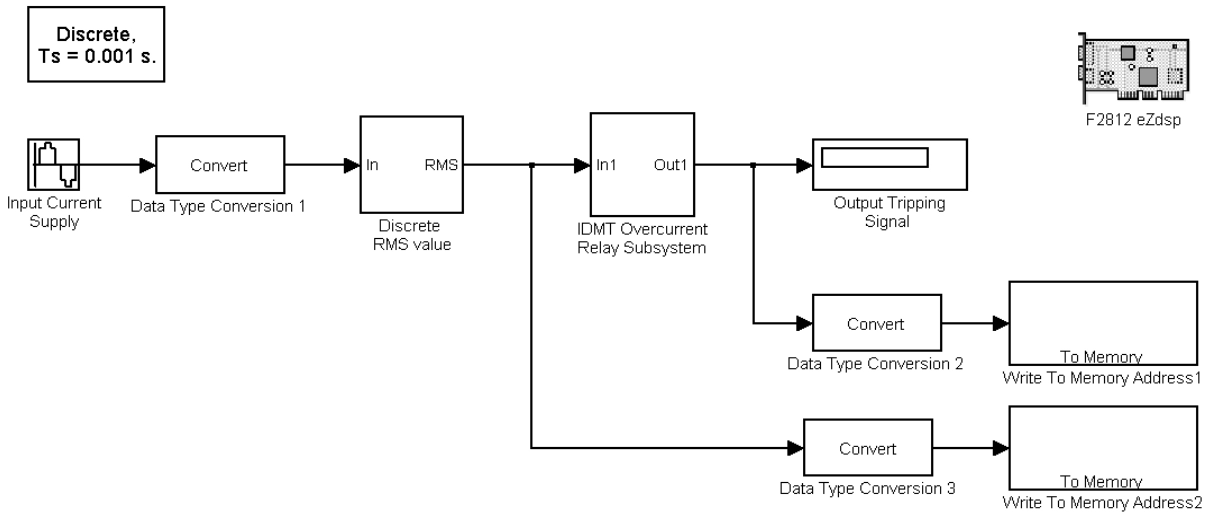

Fig. 1. The proposed overcurrent relay model

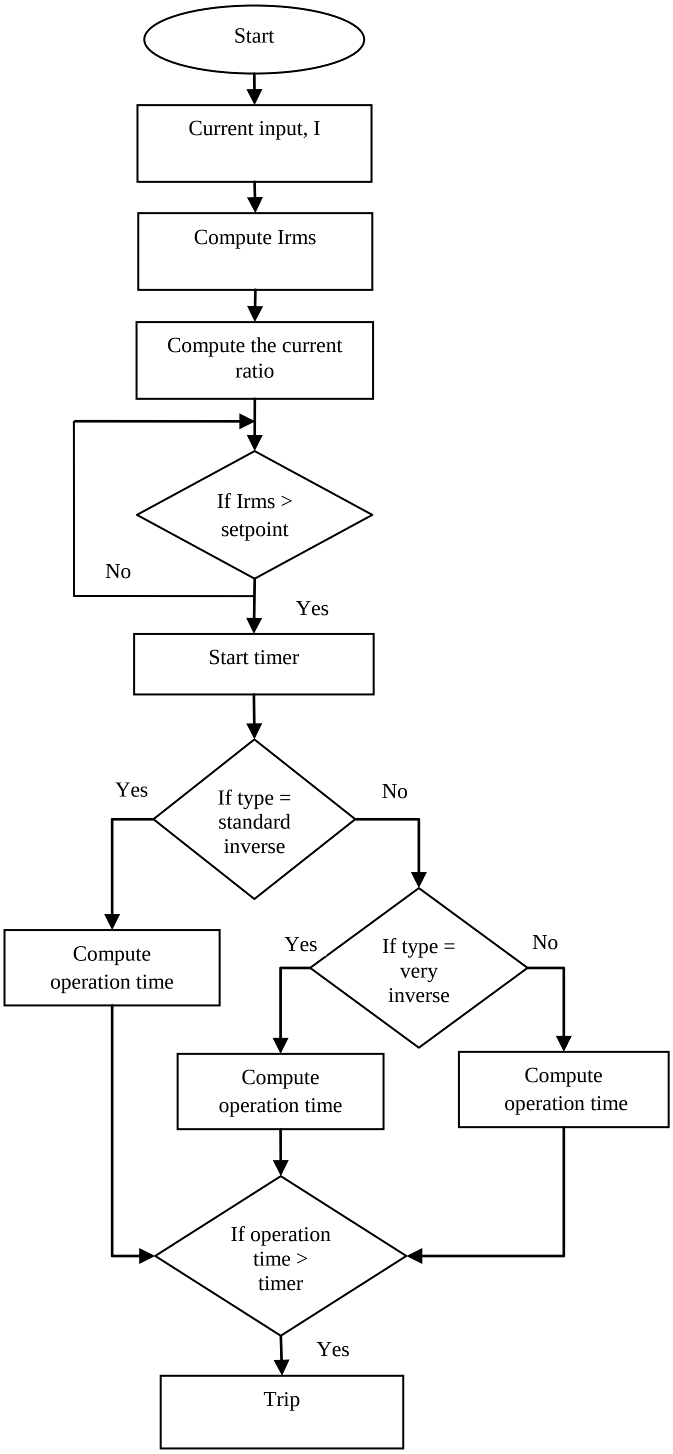

Fig. 2. Flow chart of the protection algorithm of the overcurrent relay

If the current ratio is more than 1, this indicates that the RMS input current exceeded the setpoint value. Hence, the protection algorithm embedded in the relay starts the process to determine the operation time of the relay. The flow chart for the protection algorithm of the overcurrent relay is shown in Fig.2.

The protection algorithm is built based on IEC standard which includes different types of inverse characteristics of the IDMT relay. The algorithm requires current setpoint value, TMS and type of inverse characteristics to determine the operation time. In this model, very inverse characteristic of IDMT type with TMS of 0.1 is used to investigate performance of the overcurrent relay.

An output tripping signal is generated if the RMS input current exceeds the setpoint value or in other word when the current ratio is more than 1. The overcurrent relay will trip after the intended operation time which is calculated from the protection algorithm.

4 Hardware Implementation

The implementation of the overcurrent relay consists of DSP of TMS320F2812 and a computer as the DSP host. The block diagram of the hardware implementation is shown in Fig.3. TMS320F2812 is a high speed processor which operates at frequency of 150MHz [14]. The TMS320F2812 is also equipped large memory capacity of 18K words on-chip RAM and 64K words off-chip SRAM memory [14].

The two methods used to implement the relay on DSP required codes. These codes needed to be complied, linked, downloaded and executed on TMS320F2812. For the downloaded simulation model,

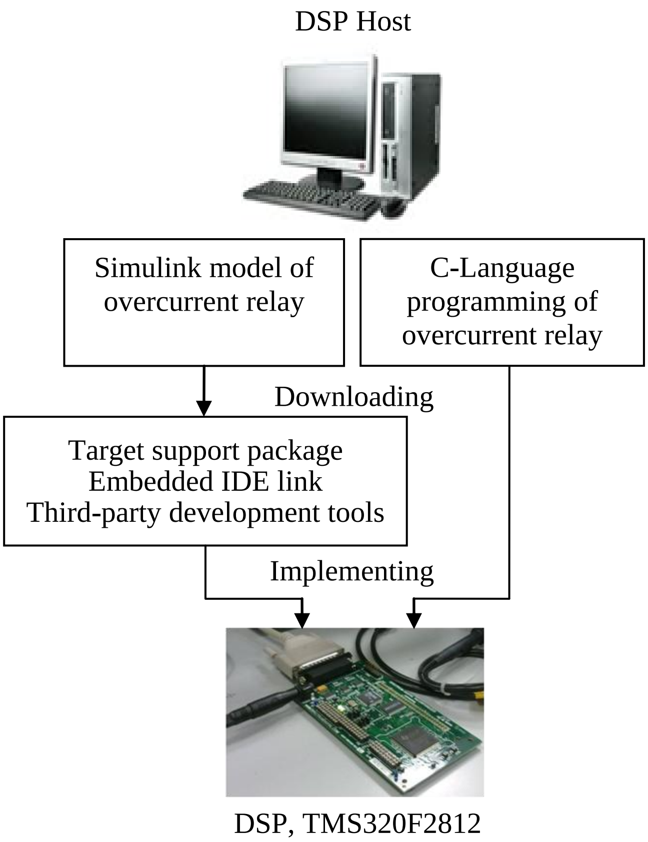

Fig. 3. Hardware implementation of the overcurrent relay on TMS320F2812

the code is automatically generated by the Target Support Package MATLAB Package along with Real-Time Workshop, Embedded Integrated Development Environment (IDE) Link and third-party development tools [21]. Meanwhile for second method, the code is written in the processor itself.

The code is downloaded into the processor through the DSP host using parallel port connector. The results from the execution are viewed from Code Composer Studio (CCS). CCS provides a single user interface where the contents of the memory saved in the TMS320F2812 can be viewed.

5 Implementation on the DSP

The two different implementation methods are used to implement the overcurrent relay on the DSP, TMS320F2812. For the first method, the overcurrent relay simulation model from MATLAB/Simulink is directly downloaded into the DSP. Meanwhile, for the second method, coding in C-Language for the relay is written in the DSP.

5.1 Simulation Model

In the first method, the simulation model from Fig.1 which was simulated in MATLAB/Simulink is directly download into the DSP. The relay simulation model which uses Simulink blocks is implemented on the DSP using this method. This is due to the integration between MATLAB and Texas Instrument which automate the process of generating the algorithm code. The results of the RMS value, current ratio and operation time are stored in three different memory locations of the processor. After the execution, the results can be viewed from the particular memory locations from CCS.

5.2 Using C-Language Programming

For the second method, the proposed overcurrent relay model is written in the C-Language coding rather than downloading the simulation model into the DSP. The program is written in the CCS.

The sine waveform input current of 50Hz is generated in the programming itself. The input currents are sampled at sampling rate of 1kHz. Each of the current samples is buffered in the processor. This is to ensure that the DSP has sufficient time and accurate number of data for at least a cycle to compute the RMS values in this real-time operation. The RMS computation used in the programming is based on equation (2).

Then, the RMS values are used to calculate the current ratio of the input before proceeding to determine the operation time. The algorithm used to determine the operating time of the relay for different types of inverse characteristic is written.

$$RMS = \sqrt{\frac{1}{n} \sum_{n=0}^{n} I^2} \qquad (2)$$

where n - number of samples
I - current value

Once the relay setting is selected, the relay trips if the RMS input current exceeds the current setpoint value. In order to ensure that the relay is operating correctly, the output tripping signal is also buffered in the processor. The results of the RMS value, current ratio, operation time and output tripping signal are also viewed from CCS either in the form of graph or value. Comparison results of the operation time of the overcurrent relay obtained between simulation in MATLAB/Simulink and execution on TMS320F2812 for two different methods are performed as described in next section.

6 Results

Different input current values are supplied into the overcurrent relay in order to investigate the accuracy of the operation time obtained using different types of implementation methods. The ratio of currents obtained from the simulation and execution on the TMS320F2812 is shown in Table 2. Meanwhile the results from simulation and execution on TMS320F2812 for two different methods are shown in Table 3.

From Table 3, the operation time is obtained for very inverse type characteristics of the IDMT overcurrent relay with TMS of 0.1. The current setpoint of the relay is 150A. The overcurrent relay will start to compute the operation time if the RMS input current exceeds the setpoint value. Thus, the processor must be able to operate efficiently in the real-time to detect the input current value.

The operation time obtained from the simulation and hardware implementations have small variation as shown in Table 3. The C-Language programming which is embedded in the processor produces closely same results with the simulation and according to IEC standard. The operation time obtained from the programming has higher accuracy compared to downloaded simulation model.

Since the overcurrent relay will only compute the operation time when the RMS input exceeds the setpoint value thus there are no results obtained if the RMS input current less than the setpoint value. These are shown for the amplitude input current of 170 A and 210A.

Table 2. Ratio of Currents for the Overcurrent Relay

Amplitude input current (A)	Ratio current		
	Simulation on MATLAB/Simulink	Hardware implementation	
		Downloaded from simulation model	C-Language programming
170	0.8014	0.8013009	0.8013878
210	0.9899	0.9888265	0.9899496
213	1.004	1.004158	1.004092
227	1.07	1.069787	1.070089
500	2.357	2.357023	2.357022

Table 3. Operation Time of the Overcurrent Relay

Amplitude input current (A)	Operation time (s)			
	Calculated from IEC standard	Simulation on MATLAB/Simulink	Hardware implementation	
			Downloaded from simulation model	C-Language programming
170	-	-	-	-
210	-	-	-	-
213	329.9419	329.9	324.6737	329.9426
227	19.2614	19.26	19.34454	19.26133
500	0.9948	0.9948	0.994825	0.9948252

For the amplitude input current of 213A is the minimum value that could cause the relay to trip since the ratio of this input value is 1.004. With this minimum value, the theoretical value for the operation time is 329.9419s. Compared to MATLAB/Simulink simulation is 329.9s and hardware implementation using simulation model and programming is 324.6737s and 329.9426s respectively.

The memory capacity used in the hardware implementation shows a large difference between the downloaded simulation model and programming. The downloaded simulation model into TMS320F2812 requires large amount of memory to store the code generated by MATLAB. The memory capacity of the 32-bit processor used by the downloaded model is 3639 addresses and programming is 2681 addresses. By writing the algorithm code in the DSP, small memory allocation is used. Thus, this definitely increases the speed and reduces the time taken to execute the protection algorithm embedded in the DSP [22].

Overall, the results obtained from two different methods show that the TMS3202812 are suitable to be use as the processor for the overcurrent relay. Since the operation time obtained has small difference at most of 1.6% compared to the IEC standard. But by using coding method, less memory is used and results obtained are faster.

7 Conclusions

This paper describes the implementation of overcurrent relay on DSP, TMS320F2812. The implementation of the overcurrent relay on DSP improves the performance in the terms of accuracy and speed. From the results obtained for two different implementation methods, DSP yield similar results compared to IEC 255-3 standard for computing the operation time of the relay. Both implementation methods are suitable to implement the overcurrent relay. However, implementation using coding method used less memory usage. An improved power system protection is achieved by using DSP that is able to maintain high degree of service reliability to avoid any failure to trip or mal-trip of the relay in the power system. Future works which could be performed such as implementing different types of protection control algorithm on the DSP for power system protection applications.

References:

[1] *Network Protection & Automation Guide*, Areva T&D, 1995.

[2] T. S. Sidhu, M. S. Sachdev, and H. C. Wood, Design of a microprocessor-based overcurrent relay, *IEEE Western Canada Conference on Computer, Power*

and Communications Systems in a Rural Environment, 29–30 May 1991, pp. 41–46.

[3] Mudhafar A. Al-Nema, Sinan M. Bashi, and Abdulhadi A. Ubaid, Microprocessor-based overcurrent relays, *IEEE Transactions on Industrial Electronics*, Vol.IE-33, No.1, 1986, pp. 49–51.

[4] C. A. Kramer, and W. A. Elmore, Flexible inverse overcurrent relaying using a microprocessor, *IEEE Transactions on Power Delivery*, Vol.5, No.2, 1990, pp. 915–923.

[5] T. S. Sidhu, M. S. Sachdev, and H. C. Wood, A computer-aided design tool for developing digital controllers and relays, *IEEE Transactions on Industry Applications*, Vol.28, No.6, 1992, pp. 1376– 1383.

[6] G. Benmouyal, Design of a digital multi-curve time-overcurrent relay, *IEEE Transactions on Power Delivery*, Vol.6, No.22, 1991, pp. 656–665.

[7] M. A. Manzoul, Interrupt-driven microprocessor-based overcurrent relay, *IEEE Transactions on Industrial Electronics*, Vol. 38, No.1, 1991, pp. 8–9.

[8] E. Price, The next step in the evolution of protection and control implementation, *Annual Conference for Protective Relay Engineers*, 2010, pp. 1–16.

[9] M. Khederzadeh, Back-up protection of distance relay second zone by directional overcurrent relays with combined curves, *IEEE Power Engineering Society General Meeting*, 2006.

[10] Walter A. Elmore, *Protective Relaying Theory and Applications*, ABB Power T&D Company Inc, 1994.

[11] T. Keil, and J. Jager, Advanced coordination method for overcurrent protection relays using nonstandard tripping characteristics, *IEEE Transactions on Power Delivery*, Vol.23, No.1, 2008, pp. 52–57.

[12] D. N. Vishwakarma, and Z. Moravej, ANN based directional overcurrent relay, *IEEE/PES Transmission and Distribution Conference and Exposition*, Vol.1, 28 October–2 November 2001, pp. 29–64.

[13] J. C. Tan, P. G. McLaren, R. P. Jayasinghe, and P. L. Wilson, Software model for inverse time overcurrent relays incorporating IEC and IEEE standard curves, *Canadian Conference on Electrical and Computer Engineering*, Vol.1, 12–15 May 2002, pp. 37–41.

[14] *Data Manual: TMS320F2810, TMS320F2811, TMS320F2812, TMS320C2810, TMS320C2811, TMS320C2812 Digital Signal Processors*, Texas Instruments, 2010.

[15] K. Shuang, C. H. Chen, N. Xiao and D. Z. Yu, A new digital electronic governor based on 32 bits DSP for a gas engine, *9th International Conference on Control, Automation, Robotics and Vision, 5–8 December 2006*, pp. 1–6.

[16] Computer representation of overcurrent relay characteristics, *IEEE Power Engineering Review*, Vol.9, July 1989, pp. 50–51.

[17] IEC Publication 255-3 (1989–05), Single input energizing quality measuring relays with dependent or independent, 1989.

[18] Jarm-Long Chung, Ying Lu, Wen-Shiow Kao and Chih-Ju Chou, Study of solving the coordination curve intersection of inverse-time overcurrent relays in subtransmission systems, *IEEE Transactions on Power Delivery*, Vol.23, No.4, 2008, pp. 1780–1788.

[19] F. Wang, and M. H. J. Bollen, Quantification of transient current signals in the viewpoint of overcurrent relay, *IEEE Power Engineering Society General Meeting*, Vol.4, 2003.

[20] A. A. Zainul Abidin, A. Ramasamy, I. Z. Abidin, and F. H. Nagi, Overcurrent time delay determination using gain scheduled PID controllers, *3rd International Conference on Energy and Environment*, 2009, pp. 89–93.

[21] *Target Support Package*, The MathWorks.

[22] J. M. Chang, and E. F. Gehringer, A high-performance memory allocator for object-oriented systems, *IEEE Transactions on Computers*, Vol.45, No.3, 1996, pp. 357–366.

ISBN: 978-960-474-262-2