

THE INVERTED-TRIANGLE TECHNIQUE OF CONVERTING THE COMPUTER NUMBER SYSTEMS (BINARY, OCTAL AND HEXADECIMAL) TO DECIMAL

S. O. O. DUKE AND J. N. OBIDINNU

(Received 27, February 2008; Revision Accepted 29, January 2009)

ABSTRACT

Number system is the method of writing numerals to represent values. It is an integral part of computer science. It is therefore, necessary to provide computer scientists with a good understanding of number system concepts. Several methods exist in converting from one number system to another. In this paper, we introduce a new method, called the Inverted-Triangular-Technique for converting other number systems to Decimal. It adopts a graphical presentation that assumes the shape of an inverted triangle. Two models have been presented: one to handle Integer number conversions and the other handles the non-integer conversions. Demonstrations of the models are presented for clarity of understanding. This technique has been tested with numerous examples, and the benefits arising from its uses have been discussed.

KEYWORDS: Inverted-Triangle, Graphical, Number System, Computer Science

1.0 INTRODUCTION

The storage of information in a computer memory corresponds to a number of two-state memory elements, each capable of storing one bit (binary digit). However, the data that a computer manipulates is not restricted simply to binary numbers. A computer may process any data which is capable of being converted to binary codes (Shawn 2002.).

Consequently, the study of Computer Science includes the study of how information is organized and stored in the memory of a computer; how data is manipulated to provide information as well as how it can be utilized. Thus, it is very important for Computer Scientists to understand the concept of number systems. The most widely used method for interpreting bit settings as non negative integers is the binary number system. In this system each bit position represents a power of 2. The right most bit position represents 2^0 , which equals 1, the next position to the left represents 2^1 , which equals 2; the next bit position represent 2^2 , which equals 4 and so on. An integer is represented as a sum of powers of 2 (John, 2007.)

Number system is the method of writing numerals to represent values. It is a framework where a set of numbers are represented by numerals in a consistent manner (Wikipedia 2007.) One interesting fact remains that we see the number system as a concept that allows the numeral "10" to be interpreted as the decimal numeral for **ten (10)**, binary numeral for **two (2)**, octal numeral for **eight (8)**, hexadecimal numeral for **sixteen (16)**, and so on for other numerals in different bases. We can list as many number bases as one thousand (1000). However, in this paper, we have restricted our discussions to the common ones that are

usually encountered in the computing field: Binary, Octal, Hexadecimal and Denary.

The binary number system is a positional numeral system that has two (2) as its base. It requires only two digits of 0 and 1. According to Encyclopedia Britannica (2007), the importance of the Binary system to information theory and computing technology derives mainly from the compact and reliable manner in which 0s and 1s can be represented electronically. The Binary numbers form the native language of computers and it is used internally by modern computers and its associated equipment due to its straight forward implementation in electronic circuitry.

Counting in Binary is similar to other number systems. It proceeds through each symbol in increasing order beginning with a single digit. Using the symbols 0 and 1, the next higher digit to the left is incremented. When the symbols for the first digit are exhausted, the counting starts over again at 0 at this point. Thus, after a digit reaches 1 in binary, an increment resets it to 0, but also causes an increment to the next digit to the left (Erik 1999.)

The Octal, Decimal, and Hexadecimal are popular in computing because they can be translated to and from the binary system. Octal is a radix of 8, which is a power of 2^3 (Mabutt, 2006.) This means that it takes exactly three binary digits to represent an octal digit. The Hexadecimal (Hex) number system is base -16 system of numeration. Hex uses a radix of 16, which is a power of 2^4 . To translate a string of binary digits into Hex, we group the binary digits into four at a time from right to left (Maxfield, and Brown 2006.) The Decimal number system is the most commonly used system of numerals by humans. It is based on ten digits. These ten digits can be combined to form larger numbers. It became

more popular than the others because of its adaptability to early computing tools such as fingers and toes (Shawn 2002.)

2.0 STATEMENT OF PROBLEM

The digit-times-place-value conventional method of number systems conversion has been popular over the years. However, we observed that it usually takes a long time to verify the accuracy or otherwise of the eventual result obtained through its application. This is because a lot of care has to be taken to ensure that exponentiation values are correctly computed. More so, you may require doing a reversal of the process before you can be sure of your result.

Furthermore, the conventional method does not specifically present a formula, which can be used to make the teaching and learning of number systems easier. Rather, it offers rules that can be followed, in the process of translation. These rules may not be easily assimilated. This paper therefore, is aimed at introducing a method which will provide a scientific formula that can be followed in number systems conversion. It is our assertion that pictures speak better than words. Consequently, we seek to devise a graphical presentation that will make the assimilation and verification of computations faster and easier.

3.0 METHODOLOGY

We introduce two models for the conversion of number systems from one base to another. The models are designed to handle integer number conversion, and non-integer number conversion.

3.1 THE INVERTED-TRIANGULAR TECHNIQUE FOR CONVERTING INTEGER NUMERAL IN ANY BASE TO BASE-10

The model formulated for doing this is presented below:

$$X_p = (((k_1 * p + k_2) * p + k_3) * p + \dots + k_n)_{10} \dots \dots 1$$

where

- o X_p is the number to be converted;
- o p represents the base.
- o $k_1, k_2, k_3, \dots, k_n$ represents the digits positions or place values in the number to be converted.
- o k_n is the right most digit of the number.

From the model (equation 1), the underlying principle is built on the idea that:

- i) The left most digit in the given number to be converted is multiplied by the given base and the product is added to the next digit to the right in the innermost parenthesis; and again the sum is further multiplied by the given base. At this point the innermost parenthesis disappears.
- ii) The result obtained in (i) above is added to the next digit and the sum is multiplied by the. This process continues until the final digit has been reached, at which time the resultant will provide the base-10 equivalent of the original number, see **DEMO 1**, **DEMO 2** and **DEMO 3** for integer conversion.

It should be noted that the number of parenthesis present is dependent on the number of digits in the given number base to be converted. Each inner parenthesis disappears as each step is solved. As the conversion process progresses, solving from the leftmost digit to the right most digit the triangle tapers downward until a single equivalent value in base-10 is obtained. This is illustrated in fig. 1.

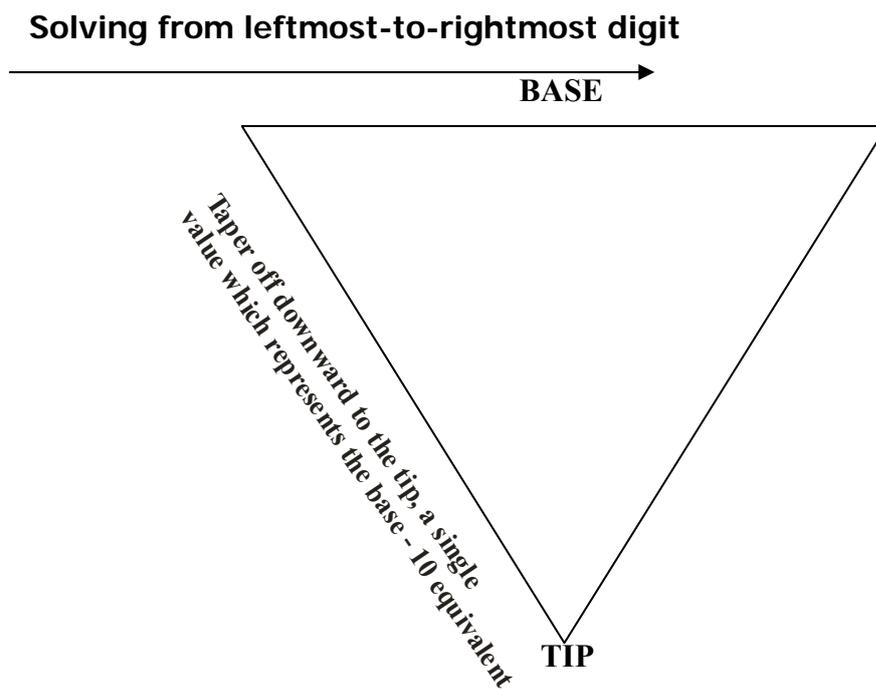


Fig. 1: Graphical model solution process of the Inverted-Triangular Technique for the Integer Number System Conversion.

THE INVERTED- TRIANGULAR MODEL FOR CONVERTING NON-INTEGER NUMBERS IN ANY BASE TO BASE- 10

The Inverted-Triangular model for converting a non-integer number in any base to base – 10 is as shown in equation 2.

$$X_p = 1/p*(k_1+1/p*(k_3...+1/p*(k_{n-1}+k_n/p))))_{10} \dots\dots\dots 2$$

where

- o X_p is the number to be converted;
- o $k_1, k_2,$ and k_3 represent the digit's position in the number to be converted;
- o k_{n-1} represents the digit before the rightmost digit;
- o p represents the given base.
- o k_n is the rightmost digit of the given base.

The underlying principle built for equation 2 is built on the idea that:

- i) Working from the rightmost digit to the leftmost digit; the rightmost digit is divided by the given base and the dividend is added to the next digit (k_{n-1}) to left.
- ii) Next the sum obtained in (i) is multiplied by the reciprocal of the given base being converted. At this stage, the innermost parenthesis disappears from right to left. This process continues until the final leftmost digit has been reached, at which time the resultant digit will provide the base-10 equivalent of the original number. **(See DEMO 4, DEMO 5 and DEMO 6).**

It should be observed that, as the conversion progresses step-wisely, each inner parenthesis disappears at the completion of each step. This leads to the tapering of the inverted – triangle to a tip. The tip of inverted triangle gives the base-10 non integer equivalent of the given non-integer base.

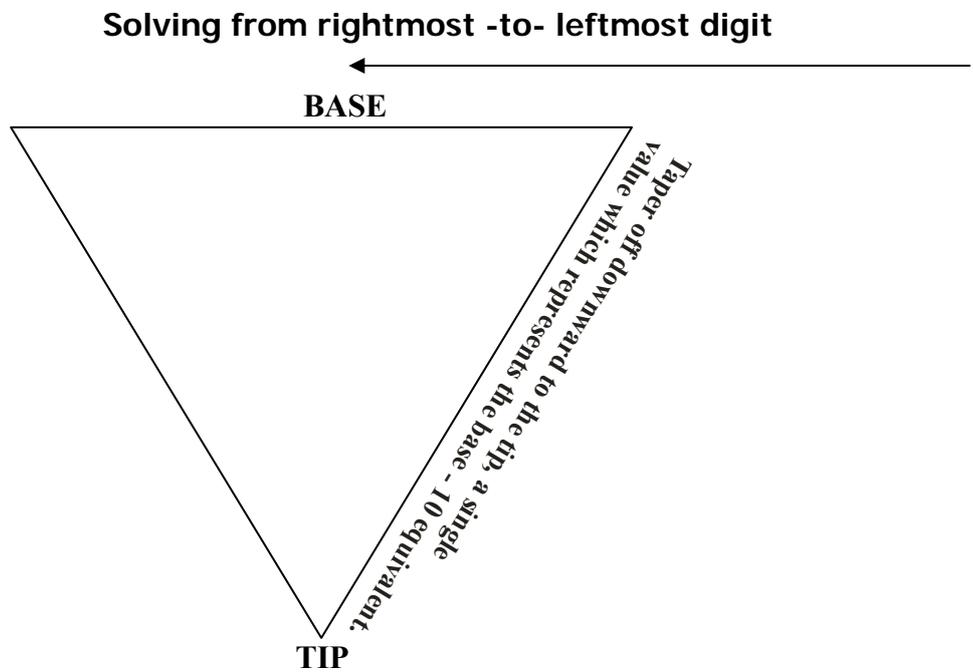


Fig 2: Graphical representation of the Inverted – Triangular Solution process for non-integer number systems conversion.

4.0 APPLICATIONS / DEMONSTRATIONS

This section presents the applications / demonstrations of the Inverted – Triangular Technique for converting both integer and non-integer computer number systems (binary, octal and hexadecimal) to their base-10 equivalents. The demonstration features the Inverted-Triangular Technique vis-à-vis the conventional digit-times-place-values method.

4.1 INTEGER CONVERSION DEMONSTRATION

DEMO 1: BINARY CONVERSION

Convert 1110111_2 to base -10

Solution

(a) By Inverted – Triangular Technique

$$k_1=1; k_2=1; k_3 = 1; k_4 = 0; k_5 =1; k_6 = 1; k_7 = 1; p = 2$$

$$X_p = 1110111_2$$

Model $\longrightarrow X_p = (((k_1*p+k_2)*p+k_3)*p+...+k_n)_{10}$

Xp = ((((((k₁*p+k₂)*p+k₃)*p+k₄)*p+k₅)*p+k₆)*p+k₇)₁₀

(i) In its Simplest Form solution at a Glance

$$((((((1*2+1)*2+1)*2+0)*2+1)*2+1)*2+1)_{10} = 119_{10}$$

Fig. 3: The Inverted-Triangular Technique simplest form solution at a glance (single line solution) for integer binary number

(ii) **In its step-wise solution**

$$\begin{aligned}
 X_p &= ((((((k_1*2+k_2) *p+k_3) *2+k_4) *p+k_5) *p+k_6) *p+k_7)_{10} \\
 1110111_2 &= ((((((1*2+1) *2+1) *2+0) *2+1) *2+1) *2+1)_{10} \\
 &((((((2+1) *2+1) *2+0) *2+1) *2+1) *2+1)_{10} \\
 &((((((3) *2+1) *2+0) *2+1) *2+1) *2+1)_{10} \\
 &((((((6+1) *2+0) *2+1) *2+1) *2+1)_{10} \\
 &((((((7) *2+0) *2+1) *2+1) *2+1)_{10} \\
 &((((((14+0) *2+1) *2+1) *2+1)_{10} \\
 &(((14) *2+1) *2+1) *2+1)_{10} \\
 &(((28+1) *2+1) *2+1)_{10} \\
 &((29) *2+1) *2+1)_{10} \\
 &((58+1)*2+1)_{10} \\
 &((59)*2+1)_{10} \\
 &(118+1)_{10} \\
 &\mathbf{19}_{10}
 \end{aligned}$$

Fig. 4: The Inverted-Triangular step-wise solution (step-by-step solution) for integer binary number.

(b) **By the conventional method**

1110111 ₂ =	1	1	1	0	1	1	1
	2	2	2	2	2	2	2
	6	5	4	3	2	1	0
	$1 * 2^6 + 1*2^5 + 1*2^4 + 0*2^3 + 1*2^2 + 1*2^1 + 1*2^0$						
	$1*64+1*32+1*16+0*8+1*4+1*2+1*1$						
	$64 + 32 + 16 + 0 + 4 + 2 + 1 = 119_{10}$						

Fig. 5: Conventional digit-times-place-value method solution for integer binary number

DEMO 2: OCTAL CONVERSION

Convert 67652_8 to base -10

Solution

a) By the Inverted – Triangular Technique

$$k_1= 6; k_2 =7; k_3=6; k_4=5; k_5=2; p = 8$$

$$X_p=67652_8$$

Model → $X_p = (((k_1*p+k_2+k_3)*p+...+k_n)_{10}$

$$X_p = (((((k_1*p+k_2)*p+k_3)*p+k_4)*p+k_5)_{10}$$

(i) In its Simplest Form solution at a Glance

$$67652_8 = (((((6*8+7)*8+6*8+5)*8+2)_{10} = \mathbf{28586}_{10}$$

Fig. 6: The Inverted-Triangular Technique simplest form solution at a glance (single line solution) for octal integer number

(ii) In its Step-wise Solution

$$\begin{aligned} 67652_8 &= (((((6 * 8 + 7) * 8 + 6) * 8 + 5) * 8 + 2)_{10} \\ &(((48+7) * 8 + 6) * 8 + 5) * 8 + 2)_{10} \\ &(((55) * 8 + 6) * 8 + 5) * 8 + 2)_{10} \\ &(((440 + 6) * 8 + 5) * 8 + 2)_{10} \\ &(((446) * 8 + 5) * 8 + 2)_{10} \\ &((3568 + 5) * 8 + 2)_{10} \\ &((3573) * 8 + 2)_{10} \\ &(28584 + 2)_{10} \\ &\mathbf{28586}_{10} \end{aligned}$$

Fig. 7: Inverted-Triangular step-wise solution (step-by-step solution) for integer octal number

b) By the conversional method

6	7	6	5	2
8	8	8	8	8
4	3	2	1	0

$$\begin{aligned} &6*8^4 + 7*8^3 + 6*8^2 + 5*8^1 + 2*8^0 \\ &6*8*8*8*8 + 6*8*8*8+ 5*8*8+ 2*8 \\ &6*4096 + 7*512 + 6*64 + 5*8 + 2*1 \\ &24576 + 3584 + 384 + 40 + 2 = \mathbf{28586} \end{aligned}$$

Fig. 8: Conventional digit-times-place-value method solution for integer octal number

DEMO 3: HEXADECIMAL CONVERSION

Convert 7FFBDE₁₆ to base -10

Solution:

k₁=7; k₂=F; k₃=F; k₄=B; k₅=D; k₆=E; p=16

X_p=7FFBDE₁₆

a) By the Inverted – Triangular Technique

X_n = (((((k₁ * p + k₂) * p + k₃) * p + k₄) * p + k₅) * p + k_n)₁₀

7FFBDE₁₆ = ((((((7 * 16 + F) * 16 + F) * P + B) * P + D) * P + E)₁₀
 (((((112+15)*16+15)*16+11)*16+13)*16+14)₁₀
 (((((127)*16+15)*16+11)*16+13)*16+14)₁₀
 (((((2032+15)*16+11)*16+13)*16+14)₁₀
 (((((2047)*16+11)*16+13)*16+14)₁₀
 (((((32752+11)*16+13)*16+14)₁₀
 (((((32763)*16+13)*16+14)₁₀
 (((((524208+13)*16+14)₁₀
 (((((524221)*16+14)₁₀
 ((8387536+14)₁₀
8387550₁₀

Fig. 9: Inverted-Triangular step-wise solution (step-by-step solution) for integer hexadecimal number

(c) By the conventional method

7FFBDE ₁₆ =	7	F	F	B	D	E
	16	16	16	16	16	16
	5	4	3	2	1	0
	$7*16^5 + F*16^4 + F*16^3 + B*16^2 + D*16^1 + E*16^0$					
	$7*16*16*16*16*16 + 15*16*16*16*16 + 15*16*16*16 + 11*16*16 + 13*16 + 14*1$					
	$7340032 + 983940 + 61440 + 2816 + 208 + 14 =$					8387550₁₀

Fig. 10: Conventional digit-times-place-value method solution for integer hexadecimal number

4.2 NON-INTEGERS CONVERSION DEMONSTRATION

DEMO 4: BINARY CONVERSION

Convert 0.111011₂ to base- 10

Solution:

a) By the Inverted – Triangular Technique

$$k_1=1; k_2=1; k_3=1; k_4=0; k_5=1; k_6=1; p=2$$

$$X_p = 0.111011_2$$

$$\text{Model} \longrightarrow X_p = 1/p * (k_1 + 1/p * (k_2 + 1/p * (k_3 + \dots + 1/p * (k_{n-1} + k_n/p))))_{10}$$

$$X_p = 1/p * (k_1 + 1/p * (k_2 + 1/p * (k_3 + 1/p * (k_4 + 1/p * (k_5 + k_6/p))))_{10}$$

$$\frac{1}{2} * (1 + \frac{1}{2} * (1 + \frac{1}{2} * (1 + \frac{1}{2} * (0 + \frac{1}{2} * (1 + \frac{1}{2}))))_{10}$$

$$\frac{1}{2} * (1 + \frac{1}{2} * (1 + \frac{1}{2} * (1 + \frac{1}{2} * (0 + \frac{1}{2} * (3/2))))_{10}$$

$$\frac{1}{2} * (1 + \frac{1}{2} * (1 + \frac{1}{2} * (1 + \frac{1}{2} * (0 + 3/4))))_{10}$$

$$\frac{1}{2} * (1 + \frac{1}{2} * (1 + \frac{1}{2} * (1 + \frac{1}{2} * (3/4))))_{10}$$

$$\frac{1}{2} * (1 + \frac{1}{2} * (1 + \frac{1}{2} * (1 + 3/8)))_{10}$$

$$\frac{1}{2} * (1 + \frac{1}{2} * (1 + \frac{1}{2} * (11/8)))_{10}$$

$$\frac{1}{2} * (1 + \frac{1}{2} * (1 + 11/16))_{10}$$

$$\frac{1}{2} * (1 + \frac{1}{2} * (27/16))_{10}$$

$$\frac{1}{2} * (1 + 27/32)_{10}$$

$$\frac{1}{2} * (59/32)_{10}$$

$$(59/64)_{10}$$

$$\mathbf{0.921875}_{10}$$

Fig. 11: Inverted-Triangular step-wise solution (step-by-step solution) for non-integer binary number

b) By the Conventional Method

$$\begin{array}{rcccccccc}
 0.1110_2 & = & 0. & 1 & 1 & 1 & 0 & 1 & 1 \\
 & & 2. & 2 & 2 & 2 & 2 & 2 & 2 \\
 & & 0. & 1 & 2 & 3 & 4 & 5 & 6 \\
 & & 0*2^0 & + & 1*2^{-1} & + & 1*2^{-2} & + & 1*2^{-3} & + & 0*2^{-4} & + & 2^{-5} & + & 1*2^{-6} \\
 & & 0*1 & + & 1/2 & + & 1/4 & + & 1/8 & + & 0 & + & 1/32 & + & 1/64 \\
 & & 0 & + & 0.5 & + & 0.25 & + & 0.125 & + & 0 & + & 0.03125 & + & 0.015625 & = & \mathbf{0.921875}_{10}
 \end{array}$$

Fig. 12: Conventional digit-times-place-value method solution for non-integer binary number

DEMO 5: OCTAL CONVERSION

Convert 0.76576_8 to base-10

Solution:

$k_1=7; k_2=6; k_3=5; k_4=7; k_5=6; k=8; X_p = 0.76576_8$

a) By the Inverted – Triangular Technique

$$\begin{aligned} \text{Model} \longrightarrow X_p &= 1/p*(k_1+1/p*(k_2+1/p*(k_3+\dots 1/p*(k_{n-1}+k_n/p))))_{10} \\ X_p &= 1/p*(k_1+1/p*(k_2+1/p*(k_3+1/p*(k_4+k_5/p))))_{10} \end{aligned}$$

$$\begin{aligned} 0.76576_8 &= 1/8 *(7 + 1/8 *(6 + 1/8 *(5+1/8 *(7 + 6/8))))_{10} \\ &= 1/8 *(7 + 1/8 *(6 + 1/8 *(5 + 1/8 *(62/8))))_{10} \\ &= 1/8 *(7 + 1/8 *(6 + 1/8 *(5 + 62/64)))_{10} \\ &= 1/8 *(7 + 1/8 *(6 + 1/8 *(382/64)))_{10} \\ &= 1/8 *(7 + 1/8 *(6 + 382/512))_{10} \\ &= 1/8 *(7 + 1/8 *(3454/512))_{10} \\ &= 1/8 *(7 + (3454/4096))_{10} \\ &= 1/8 *(32126/4096)_{10} \\ &= (32126/32768)_{10} \\ &= \mathbf{0.980408}_{10} \end{aligned}$$

Fig.13: Inverted-Triangular step-wise solution (step-by-step solution) for non-integer octal number

b) By Conventional Method

$$\begin{aligned} 0.76576_8 &= 0. \quad 7 \quad 6 \quad 5 \quad 7 \quad 6 \\ & \quad 8 \quad 8 \quad 8 \quad 8 \quad 8 \quad 8 \\ &= 0. \quad 1 \quad 2 \quad 3 \quad 4 \quad 5 \\ &= 0x8^0 + 7x8^{-1} + 6x8^{-2} + 5x8^{-3} + 7x8^{-4} + 6x8^{-5} \\ &= 0+7/8 + 6/64 + 5/512 + 7/4096 + 6/32768 \\ &= 0+0.875 + 0.09375 + 0.009766 + .001709 + 0.000183 = \mathbf{0.980408}_{10} \end{aligned}$$

Fig. 14: Conventional digit-times-place-value method solution for integer non-octal number

DEMO 6: HEXADECIMAL CONVERSION

Convert 0.CA6CB216 to base -10

Solution:

$K_1=C; K_2=A; K_3=6; K_4=C; K_5=B; K_6=2; P=16; X_p=0.CA6CB2_{16}$

a) By Inverted – Triangular Technique.

$$\begin{aligned}
 \text{Model} \longrightarrow X_p &= 1/p^*(k_1+1/p^*(k_2+1/p^*(k_3+1/p^*(k_{n-1}+k_n/p))))_{16} \\
 X_p &= 1/p^*(k_1+1/p^*(k_2+1/p^*(k_3+1/p^*(k_4+1/p^*(k_5+k_n/p))))_{16} \\
 0.CA6CB2_{16} &= 1/16*(C+1/16*(A+1/16*(6+1/16*(C+1/16*(B+2/16))))_{10} \\
 &= 1/16*(C+1/16*(A+1/16*(6+1/16*(C+1/16*(11+2/16))))_{10} \\
 &= 1/16*(C+1/16*(A+1/16*(6+1/16*(C+1/16*(178/16))))_{10} \\
 &= 1/16*(C+1/16*(A+1/16*(6+1/16*(12+178/256))))_{10} \\
 &= 1/16*(C+1/16*(A+1/16*(6+1/16*(3250/256))))_{10} \\
 &= 1/16*(C+1/16*(A+1/16*(6+3250/4096)))_{10} \\
 &= 1/16*(C+1/16*(A+1/16*(27826/4096)))_{10} \\
 &= 1/16*(C+1/16*(A+27826/65536))_{10} \\
 &= 1/16*(C+1/16*(10+27826/65536))_{10} \\
 &= 1/16*(C+1/16*(683186/65536))_{10} \\
 &= 1/16*(12+683186/1048576)_{10} \\
 &= 1/16*(13266098/16777216)_{10} \\
 &= (13266098/16777216)_{10} \\
 &= \mathbf{0.79072058}_{10}
 \end{aligned}$$

Fig. 15: Inverted-Triangular step-wise solution (step-by-step solution) for non-integer hexadecimal number

b) By the Conventional Method.

$0.CA6CB2_{16} =$	0.	C	A	B	C	B	2
	16	16	16	16	16	16	16
	0	1	2	3	4	5	6
	$0*16^0 + C*16^{-1} + A*16^{-2} + B*16^{-3} + C*16^{-4} + B*16^{-5} + 2*16^{-6}$						
	$0*1 + C/16^1 + A/16^2 + B/16^3 + C/16^4 + B/16^5 + 2/16^6$						
	$0+12/16 + 10/256 + 11/4096 + 12/65536 + 11/1048576 + 2/16777216$						
	$0+0.75 + 0.390\ 625 + 0.00268555 + 0.00018311 + 0.0000/0490 + 0.000000012$						
	$\mathbf{0.79072058}_{10}$						

Fig. 16: Conventional digit-times-place-value method solution for non-integer hexadecimal number.

5.0 DISCUSSION

From the demonstrations presented in section 4.1 and 4.2 in contrast, the new Inverted- Triangular Technique:

- (1) Is the faster and easier method of converting any number in any base to its base-10 equivalent compared to the conventional digit-times-place-

value method. This is because it has models (formulae) for converting both integer and non-integer numbers from any base to base-10 while the conventional digit-times-place-value method does not. (See equations 1 and 2). These models show the mathematical/scientific rule, steps and processes which are involved when doing the

conversion, thereby making the process of conversion with the new method easier and faster. Also it deals with only two (2) mathematical operators (Multiplication (*), addition (+)). It does not also involved exponentiation or raise to power. The conventional digit-times-place-value method on the other hand deals with three (3) mathematical operators (multiplication (*), exponentiation or raise to power (^) and addition (+)). (See Fig. 3, Fig.4, and Fig.5 in DEMO 1, for an integer binary number conversion; Fig. 6, Fig.7, Fig.8 in DEMO 2, for an integer octal number conversion; Fig.9 and Fig.10 in DEMO 3, for an integer hexadecimal number; conversion; Fig.11 and Fig.12 in DEMO 4, for a non-integer binary number conversion; Fig.13 and Fig.14 in DEMO 5, for a non-integer octal number conversion; Fig.15 and Fig.16 in DEMO 6 for a non-integer hexadecimal number conversion).

The exponentiation in the conventional digit-times-place-value method expands the mathematical processes involved in it, thereby making it uneasy and time consuming when doing conversion with it. It produces large values, which may be difficult to handle.

- (2) Saves spaces when working both manually and on computer because it does not involve exponentiation or raise to power. On the computer, this memory space trade-off optimizes the memory speed, thereby making the models to execute rapidly and efficiently. On the other hand, the conventional digit-times-place-value method does not optimize space because it deals with exponentiation.
- (3) Provides a highly beautiful structural/graphical presentation with its step-by-step solution that looks like an inverted triangle, which the name is coined from, than the conventional digit-times-place-value method.
- (4) Looks more scientific in presentation with its step-by-step solution.

6.0 CONCLUSION

Conversions of number systems from other number bases to Decimal have always been carried out by scholars using different methods. The method

presented in this paper introduces a new approach into the existing methods. When compared with other methods of number systems conversion, the Inverted-Triangular Technique presents a beautiful structure that is highly scientific, and can be easily adopted by numerical scientists.

REFERENCES

- Maxfield, C. and Brown, A., 2006. The History of Number Systems. Internet Download: Source on-line at <http://www.diycalculator.com/popup-h-numsys.shtml> Retrieved on the 20th May 2007.
- Encyclopedia Britannica- the Online Encyclopedia , 2008. Binary Number System. Internet Download: Online at <http://www.britannica.com/eb/article-9079220/binary-number-system> Retrieved on 25th February 2007
- Erik, O., 1999. Binary Number System. Internet Download. Source on-line at <http://www.erikoest.dk/bimnary.htm> Retrieved on 20th May 2007.
- John, V. W. H., 2007. Understanding the Binary Number System. Source: <http://ezinearticles.com/Understanding-the-Binary-Numbering-System&id=393145> Retrieved on 26th February 2007
- Mabbut, D., 2006. Computer Number Systems. Internet Download: Online at <http://visualbasic.about.com/od/4sevb6/aa0329036.htm>. Retrieved 26th February 2007
- Shawn, S., 2002. Understanding Binary and Hex numbers. Internet downloads. Source on-line at <http://www.codemastershaw.co/library/tutorial/hexbin-numbers.pbp>. Retrieved on 30th March 2007
- Wikipedia, the free encyclopedia, 2006. Numeral System. Source on-line at http://en.wikipedia.org/wiki/Numeral_system Retrieved on 26th February 2007