A Field-Theoretic View of Logic Functions

A Field-Theoretic View of Logic Functions

Zensho NAKAO*

Abstract Logic functions of several variables are considered as mappings from a Cartesian product of several copies of a finite (Galois) field into the field itself. Based on the well-known fact that such mappings are always polynomial, it is shown how the corresponding polynomial expressions for given logic functions are obtained through several concrete computational examples.

1. Preliminaries

Let $K=GF(q)$, $q=p^n$, where p is prime. We consider the case $q=p=3, 5$ here. Refer to [NZK] for the case $q=2, 4$. We make computation only for $p=3$ because an extension to the case $p=5$ is routine. The following are some relevant results from field theory:

(1) $K^*=K-\{0\}$ is a cyclic group of order p-1.

(2) $\forall\ x \in K^*,\ x^{p-1}=1$ (Fermat's Theorem).

(3) $\forall\ x \in K,\ x^p=x$.

(4) $K=\{x: x \text{ is a solution of } x^p-x=0\}$.

(5) $x^{p-1}-1=\prod_{i=1}^{p-1}(x-i)$, so $\prod_{i=1}^{p-1} i=(p-1)!=-1$ (Wilson's Theorem).

(6) $\forall\ x \in K$, $\exists$ a unique $\sqrt[p]{x}$

$K=GF(3)=\{0, 1, 2\}$ has the field structure given by the following addition (+) and multiplication (* or juxtaposition) tables:

+	0	1	2
0	0	1	2
1	1	2	0
2	2	0	1

*	0	1	2
0	0	0	0
1	0	1	2
2	0	2	1

2. Formulas for Polynomial Representation

We let $K=GF(3)$ once for all. There are several formulas known for polynomial representation of functions; the following results quoted from [T] are particularly useful for

Received: April 25, 1984

*Department of Electronics Engineering and Computer Science

our purpose:

(1) If $f: K \longrightarrow K$ is a function, then f can be expressed in a polynomial form over K:

$$f(x) = a_0 + a_1 x + a_2 x^2,$$

where $a_0 = f(0)$, $a_i = -\sum_{x \in K} x^{2-i} f(x)$, $(i = 1, 2)$

(2) If $f: K^2 \longrightarrow K$ is a mapping, then f can be realized as a polynomial mapping over K in two variables:

$$f(x, y) = \sum_{0 \le i, j \le 2} a_{ij} x^i y^j,$$

where $a_{00} = f(0, 0)$, $a_{i0} = -\sum_{x \in K} x^{2-i} f(x, 0)$, $a_{0i} = -\sum_{y \in K} y^{2-i} f(0, y)$, $a_{ij} = \sum_{x, y \in K} x^{2-i} y^{2-j} f(x, y)$, $(i, j = 1, 2)$

(3) If $f: K^3 \longrightarrow K$ is a mapping, then f can be realized as a polynomial mapping over K in three variables:

$$f(x, y, z) = \sum_{0 \le i, j, k \le 2} a_{ijk} x^i y^j z^k,$$

where $a_{000} = f(0, 0, 0)$, $a_{i00} = -\sum_{x \in K} x^{2-i} f(x, 0, 0)$, $a_{0j0} = -\sum_{y \in K} y^{2-j} f(0, y, 0)$, $a_{00k} = -\sum_{z \in K} z^{2-k} f(0, 0, z)$, $a_{ij0} = \sum_{x, y \in K} x^{2-i} y^{2-j} f(x, y, 0)$, $a_{i0k} = \sum_{x, z \in K} x^{2-i} z^{2-k} f(x, 0, z)$, $a_{0jk} = \sum_{y, z \in K} y^{2-j} z^{2-k} f(0, y, z)$, $a_{ijk} = -\sum_{x, y, z \in K} x^{2-i} y^{2-j} z^{2-k} f(x, y, z)$, $(i, j, k = 1, 2)$

It is to be noted that all calculations in the formulas above must be done in the Galois field $K = GF(3)$, and also that the mapping can be generalized to the one from K^n into K, i.e., of n variables.

3. Examples and Computations

We pick several examples of logic functions and express them as polynomial in one, two, or three variables. Refer to the Appendix for the BASIC programs used.

(1) Monadic operators

Define logic functions of one variable $x^{(0,2)}$ and $x^{(1)}$ as in [HK] by

$$x^{(02)} = \begin{cases} 2, & x \in \{0, 2\} \\ 0, & \text{otherwise} \end{cases}$$

$$x^{(1)} = \begin{cases} 2, & x = 1 \\ 0, & \text{otherwise} \end{cases}$$

By formula (2.1), we obtain the coefficients for $x^{(0,2)}$ and $x^{[1]}$, respectively, as follows:

$$a_0 = 2,\ a_1 = a_2 = -1 = 2;$$
$$a_0 = 0,\ a_1 = a_2 = -2 = 1$$

Hence, their polynomial representation now becomes:

$$x^{(0,2)} = 2\,(x^2 + x + 1) = 2\,(x + 2)^2,$$
$$x^{[1]} = x + x^2 = x\,(x + 1)$$

(2) Dyadic Operators

Define logic functions of two variables Max(x,y) and Min(x,y) by

$$\mathrm{Max}(x, y) = \begin{cases} x, & x \geqq y \\ y, & \text{otherwise} \end{cases}$$

$$\mathrm{Min}(x, y) = \begin{cases} x, & x \leqq y \\ y, & \text{otherwise} \end{cases}$$

By formula (2.2), we obtain the coefficients for Max(x,y) and Min(x,y), respectivly, as follows:

$$a_{00} = a_{20} = a_{02} = 0,\ a_{10} = a_{01} = -2 = 1,\ a_{11} = 2,\ a_{21} = a_{12} = a_{22} = 1;$$

$$a_{00} = a_{10} = a_{01} = a_{20} = a_{02} = 0,\ a_{11} = 1,\ a_{21} = a_{12} = a_{22} = 2$$

Thus, their polynomial representation is given by:

$$\mathrm{Max}(x, y) = x + y + 2\,xy + x^2y + xy^2 + x^2y^2 = (x + y) + xy\,[2 + (x + y) + xy],$$
$$\mathrm{Min}(x, y) = xy + 2x^2y + 2xy^2 + 2x^2y^2 = xy\,[\,1 + 2\,(x + y + xy)\,]$$

(3) Triadic Operator

Define a logic function of three variables T(x,y,z) by the following table [HK] :

z \ x, y	x=0, y=0	x=0, y=1	x=0, y=2	x=1, y=0	x=1, y=1	x=1, y=2	x=2, y=0	x=2, y=1	x=2, y=2
0	1	0	1	1	2	2	1	0	1
1	1	2	2	2	2	0	0	2	2
2	1	0	1	2	0	2	0	2	0

By formula (2 . 3) its coefficients are given by the following:

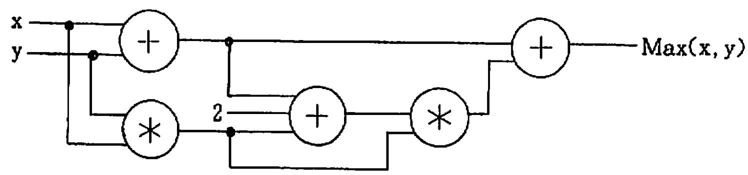
$$a_{000}=a_{110}=a_{011}=a_{210}=a_{102}=a_{012}=1,\ a_{100}=a_{001}=a_{200}=a_{002}=a_{101}=a_{111}=a_{201}=a_{120}=a_{021}$$
$$=a_{220}=a_{202}=a_{022}=a_{211}=a_{222}=0,\ a_{010}=a_{020}=a_{112}=a_{221}=a_{212}=a_{122}=-2=1,\ a_{121}=-1=2$$

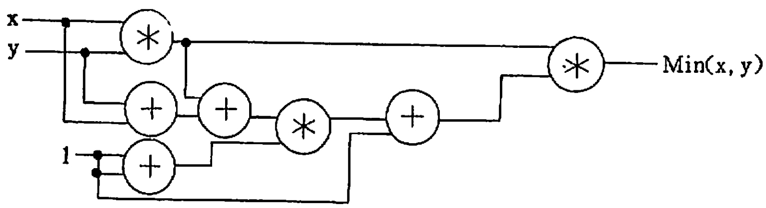
Therefore, the polynomial expression for T(x,y,z) is given by:

$$T(x,y,z)=1+y+y^2+xy+yz+x^2y+xz^2+yz^2+2xy^2z+xyz^2+x^2y^2z+x^2yz^2$$
$$+xy^2z^2=1+y+y(x+y+z+x^2+z^2)+xyz(xy+xz+yz+2y+z)+xz^2$$

4. Conclusions

If we introduce two logic gates $=\!\oplus\!-$ and $=\!\circledast\!-$, realizing the two field operators (+) and (*), respectively, then we can easily depict the functions in logic diagrams. Take, for example, Max(x,y) and Min(x,y) ; we get the following logic diagrams:





An obvious advantage for having polynomial expressions is that all calculations can be made efficiently on { 0 , 1 , 2 } with the field structure of GF(3), i.e., with the four arithmetic operations of GF(3) itself as we normally do with the real (or complex) number field, thereby turning the manipulation of logic functions into ordinary computation of polynomials. Also, the more familiar polynomial form reveals the configuration of strings of symbols more concretely than otherwise.

Acknowledgment The author is grateful to Unisoft, Inc. who provided him with free use of its microcomputer facility.

Appendix (BASIC programs)

THE PROGRAM EXPRESSES TERNARY LOGIC FUNCTIONS IN POLYNOMIAL FORM.

```
10  '--------- GF(3) - ONE VARIABLE
20  '--------- THIS PROGRAM EXPRESSES LOGIC FUNCTIONS IN ONE VARIABLE
30  '--------- IN POLYNOMIAL FORM OVER GALOIS FIELD GF( 3 )
40  DIM F(3), A(3)
50  '--------- DEFINE (+) AND (*) MODULO3
60  DEF FNM (X*Y) =X*Y-3 *INT ( (X*Y) / 3 )
70  DEF FNA (X*Y) =X+Y-3 *INT ( (X+Y) / 3 )
80  '--------- DEFINE F(X)
90  CLS 1
100 PRINT" THE INPUTS MUST BE 0 , 1 , 2 . TYPE IN THREE NUMBERS FOR
    F ( 0 ), F ( 1 ), F ( 2 ), "
110 PRINT" SEPARATED BY RETURN. "
120 FOR I=0 TO 2
130       INPUT F( I )
140 NEXT I
150 CLS 1
160 PRINT : LPRINT
170 PRINT" POLYNOMIAL EXPRESSIONS FOR LOGIC FUNCTIONS "
180 LPRINT" POLYNOMIAL EXPRESSIONS FOR LOGIC FUNCTIONS "
190 PRINT STRING $ (42, " - " )
200 LPRINT STRING $ (42, " - " )
210 PRINT : LPRINT
220 PRINT" X" ; " -----> " : " F (X)"
230 LPRINT" X" ; " -----> " : " F (X)"
240 '--------- TABLE FOR X----> F (X)
250 FOR I=0 TO 2
260       RINT I; " -----> " ; F ( I )
270       LPRINT I; " -----> " ; F ( I )
280 NEXT I
290 PRINT STRING $ (20, " - " )
300 LPRINT STRING $ (20, " - " )
310 '--------- INITIALIZE
320 N=0 : N1 =0
330 '--------- DEFINE YF(X)
340 DEF FND (X)=FNM (X, F (X) )
350 FOR I=0 TO 2
360       N=FNA (N, FND ( I ) ) : 'A ( I )
370 NEXT I
```

```
380  FOR I=0  TO  2
390        N1 =FNA (N1 ,F (I)):'A (2)
400  NEXT I
410  '--------- OUTPUT THE FORMULA FOR F(X)
420  PRINT"  F (X) = " ;F (0) ; " - " ;N; "X- " ;N1 ; "XX. "
430  LPRINT"  F (X) = " ;F (0) ; " - " ;N; "X- " ;N1 ; "XX. "
440  PRINT  STRING $ (20, " - ")
450  LPRINT  STRING $ (20, " - ")
460  PRINT : LPRINT
470  PRINT"  CONT=RETURN : STOP= 0 " ; : A $ =INPUT $ ( 1 )
480  IF  A $ <> CHR $ (13)  AND  A $ <> " 0 "  THEN  470
490  IF  A $ =CHR $ (13)  THEN  90  ELSE  500
500  END
```

THE PROGRAM EXPRESSES TERNARY LOGIC FUNCTIONS IN POLYNOMIAL FORM.

```
10   '-------- GF (3) - TWO VARIABLES
20   '-------- THIS PROGRAM EXPRESSES LOGIC FUNCTIONS OF TWO
     VARIABLES
30   '-------- IN  POLYNOMIAL  FORM  OVER  GALOIS  FIELD  GF( 3 ) .
40   DIM  F(3, 3) ,  A(3, 3)
50   '--------- DEFINE (+)  AND  (*)  MODULO  3
60   DEF  FNM (X , Y) =X*Y- 3 *INT ( (X*Y) / 3 )
70   DEF  FNA (X , Y) =X+Y- 3 *INT ( (X+Y) / 3 )
80   CLS  1
90   '-------- DEFINEF (X,Y)
100  PRINT"  THE INPUTS MUST BE 0 , 1 , 2 . TYPE IN 9 NUMBERS F(0 , 0),F
     ( 0 , 1 ), ··· ,  F ( 2 , 2 ), "
110  PRINT"  SEPARATED  BY  RETURN. "
120  FOR I=0  TO  2
130        FOR  J=0  TO  2
140              INPUT  F (I , J)
150        NEXT  J
160  NEXT  I
170  CLS  1
180  LPRINT : LPRINT
190  PRINT " POLYNOMIAL  EXPRESSIONS  FOR  LOGIC  FUNCTIONS "
200  LPRINT " POLYNOMIAL  EXPRESSIONS  FOR  LOGIC  FUNCTIONS "
210  LPRINT  STRING $ (42, " - ")
220  PRINT  STRING $ (42, " - ")
```

```
230 LPRINT : PRINT
240 '---------- TABLE FOR X , Y ----->F ( X , Y )
250  PRINT"  X              " ; "Y" ; "  ----->   " ; "F (X , Y)"
260  LPRINT"  X              " ; "Y" ;" ----->   " ; "F (X , Y)"
270 FOR I=0  TO  2
280       FOR J=0  TO  2
290             PRINT I; "              " ; J ; "  -----> " ; F (I , J)
300             LPRINT I; "              " ; J ; "  -----> " ; F (I , J)
310       NEXT J
320 NEXT I
330 PRINT  STRING $ (42 , " - " )
340 LPRINT  STRING $ (42 , " - " )
350 '--------- DEFINE  XF ( X, 0 )
360 DEF  FND (X) =FNM (X , F (X , 0) )
370 '--------- DEFINE  XF ( 0 ,X )
380 DEF  FNE (X) =FNM (X , F ( 0 , X) )
390 '--------- DEFINE  XYF (X,Y)
400 DEF  FNT (X , Y) =FNM (X , FNM (Y , F (X , Y) ) )
410 '--------- DEFINE  XF (X,Y)
420 DEF  FNR (X , Y) =FNM (X , F (X , Y) )
430 '--------- DEFINE  YF (X,Y)
440 DEF  FNP (X , Y) =FNM (Y , F (X , Y) )
450 '--------- INITIALIZE
460 N=0 : N1 =0 : N2 =0 : N3 =0 : S=0 : R=0 : P=0 : Q=0
470 FOR I=0  TO 2
480       N=FNA (N , FND ( I ) )             : 'A (1 , 0)
490       N1 =FNA (N1 , F (I, 0) )           : 'A (2 , 0)
500       N2 =FNA (N2 , F (0 , I) )          : 'A (0 , 2)
510       N3 =FNA (N3 , FNE ( I ) )          : 'A (0 , 1)
520 NEXT I
530 FOR I=0  TO  2
540       FOR J=0  TO  2
550             S=FNA (S , FNT (I , J) )                : 'A (1 , 1)
560             R=FNA (R , FNR (I , J) )                : 'A (1 , 2)
570             P=FNA (P , FNP (I , J) )                : 'A (2 , 1)
580             Q=FNA (Q , F (I , J) )                  : 'A (2 , 2)
590       NEXT J
600 NEXT I
610 '--------- OUTPUT THE FORMULA FOR F (X,Y)
620 LPRINT"  F(X,Y) = " ;F (0 , 0) ; " - " ;N; "X- " ;N3 ; "Y+ " ;S; "XY- " ;N1 ; "XX- " ;
    N2 ; "YY+ "
```

```
630 LPRINT P; "XXY+" ;R; "XYY+" ;Q; "XXYY. "
640 PRINT" F(X,Y) =" ;F(0 , 0) ; "-" ;N; "X-" ;N3 ; "Y+" ;S; "XY-" ;N1 ; "XX-" ;
    N2 ; "YY+"
650 PRINT P; "XXY+" ;R; "XYY+" ;Q; "XXYY. "
660 PRINT STRING$ (42, "-")
670 LPRINT STRING$ (42, "-")
680 PRINT:PRINT
690 '--------- CONTINUE?
700 PRINT"CONT=RETURN, STOP=0" ; :A$ =INPUT$ (1)
710 IF A$ <> CHR$ (13) AND A$ <> "0" THEN 700
720 IF A$ =CHR$ (13) THEN 60
730 END
```

THE PROGRAM EXPRESSES TERNARY LOGIC FUNCTIONS IN POLYNOMIAL FORM.

```
10  '--------- GF (3)-THREE VARIABLES
20  '--------- THIS PROGRAM EXPRESSES LOGIC FUNCTIONS OF THREE
    VARIABLES
30  '--------- IN POLYNOMIAL FORM OVER GALOIS FIELD GF(3).
40  DIM F(3, 3, 3), A(3, 3, 3), B(27)
50  '--------- DEFINE (+) AND (*) MODULO 3
60  DEF FNM (X, Y) =X*Y-3*INT ((X*Y) /3)
70  DEF FNA (X.Y) =X+Y-3*INT ((X+Y) /3)
80  CLS 1
90  '--------- DEFINE F (X,Y,Z)
100 PRINT " THE INPUTS MUST BE 0, 1, 2. TYPE IN 27 NUMBERS F(0, 0, 0)
    F(0, 0, 1),"
110 PRINT" F(0, 0, 2),···, F(2, 2, 2), SEPARATED BY RETURN."
120 FOR I=0 TO 2
130       FOR J=0 TO 2
140             FOR K=0 TO 2
150                   INPUT F(I, J, K)
160             NEXT K
170       NEXT J
180 NEXT I
190 CLS 1
200 LPRINT:LPRINT
210 PRINT " POLYNOMIAL EXPRESSIONS FOR LOGIC FUNCTIONS "
220 LPRINT " POLYNOMIAL EXPRESSIONS FOR LOGIC FUNCTIONS "
230 LPRINT STRING$ (42, "-")
```

```
240 PRINT STRING$ (42, "-")
250 LPRINT : PRINT
260 '--------- TABLE FOR X,Y,Z----> F(X,Y,Z)
270 PRINT" X        ";" Y          ";" Z";" ---->";" F (X,Y,Z)"
280 LPRINT" X        ";" Y          ";" Z";" ---->";" F (X,Y,Z)"
290 FOR I=0 TO 2
300     FOR J=0 TO 2
310         FOR K=0 TO 2
320             PRINT I;"      ";J;"      ";K;"---->";F (I,J,K)
330             LPRINT I;"      ";J;"      ";K;"---->";F (I,J,K)
340         NEXT K
350     NEXT J
360 NEXT I
370 PRINT STRING$ (79, "-")
380 LPRINT STRING$ (79, "-")
390 '--------- DEFINE XF (X,0,0)
400 DEF FNX1 ( X ) =FNM (X, F (X, 0, 0))
410 '--------- DEFINE YF (0,Y,0)
420 DEF FNY1 ( Y ) =FNM (Y, F (0, Y, 0))
430 '-------- INITIALIZE
440 FOR L=1 TO 26
450     B ( L ) =0
460 NEXT L
470 '--------- DEFINE ZF (0,0,Z)
480 DEF FNZ1 ( Z ) =FNM ( Z, F (0, 0, Z ))
490 FOR I=0 TO 2
500     B (1) =FNA ( B (1), FNX1 ( I ))               'A (1, 0, 0)
510     B (2) =FNA ( B (2), F ( I, 0, 0))             'A (2, 0, 0)
520     B (3) =FNA ( B (3), FNY1 ( I ))               'A (0, 1, 0)
530     B (4) =FNA ( B (4), F (0, I, 0))              'A (0, 2, 0)
540     B (5) =FNA ( B (5), FNZ1 ( I ))               'A (0, 0, 1)
550     B (6) =FNA ( B (6), F (0, 0, I ))             'A (0, 0, 2)
560 NEXT I
570 '--------- DEFINE XYF (X,Y,0 ),YF (X,Y,0 ),XF (X,Y,0 )
580 DEF FNX2 (X, Y) =FNM (X, FNM (Y, F (X, Y, 0)))
590 DEF FNX3 (X, Y) =FNM (Y, F (X, Y, 0))
600 DEF FNX4 (X, Y) =FNM (X, F (X, Y, 0))
610 FOR I=0 TO 2
620     FOR J=0 TO 2
630         B (7) =FNA (B (7), FNX2 (I, J))           'A (1, 1, 0)
640         B (8) =FNA (B (8), FNX3 (I, J))           'A (2, 1, 0)
```

```
650              B (9) =FNA (B (9), FNX4 (I, J) )                 'A (1 , 2 , 0)
660              B (10) =FNA (B (10), F (I, J, 0) )               'A (2 , 2 , 0)
670          NEXT J
680   NEXT I
690   '--------- DEFINE XZF (X,0,Z), ZF (X,0,Z), XF (X,0,Z)
700   DEF FNZ1 (X, Z) =FNM (X, FNM (Z, F (X, 0, Z) ) )
710   DEF FNZ2 (X, Z) =FNM (Z, F (X, 0, Z) )
720   DEF FNZ3 (X, Z) =FNM (X, F (X, 0, Z) )
730   FOR I=0 TO 2
740          FOR J=0 TO 2
760              B (11) =FNA (B (11), FNZ1 (I, J) )               'A (1 , 0 , 1)
770              B (12) =FNA (B (12), FNZ2 (I, J) )               'A (2 , 0 , 1)
780              B (13) =FNA (B (13), FNZ3 (I, J) )               'A (1 , 0 , 2)
790              B (14) =FNA (B (14), F (I, 0, J) )               'A (2 , 0 , 2)
810          NEXT J
820   NEXT I
830   '--------- DEFINE YZF (0,Y,Z) ,ZF (0,Y,Z) ,YF (0,Y,Z)
840   DEF FNY1 (Y, Z) =FNM (Y, FNM (Z, F (0, Y, Z) ) )
850   DEF FNY2 (Y, Z) =FNM (Z, F (0, Y, Z) )
860   DEF FNY3 (Y, Z) =FNM (Y, F (0, Y, Z) )
870   FOR I=0 TO 2
880          FOR J=0 TO 2
890              B (15) =FNA (B (15), FNY1 (I, J) )               'A (0 , 1 , 1)
900              B (16) =FNA (B (16), FNY2 (I, J) )               'A (0 , 2 , 1)
910              B (17) =FNA (B (17), FNY3 (I, J) )               'A (0 , 1 , 2)
920              B (18) =FNA (B (18), F (0, I, J) )               'A (0 , 2 , 2)
930          NEXT J
940   NEXT I
950   '--------- DEFINE XYZF( X, Y, Z ), YZF( X, Y, Z ), XZF( X, Y, Z ), XYF
      (X, Y, Z), ZF (X, Y, Z),
960   '--------- YF (X,Y,Z),XF (X,Y,Z)
970   DEF FNZ1 (X, Y, Z) =FNM (X, FNM (Y, FNM (Z, F (X, Y, Z) ) ) )
980   DEF FNZ2 (X, Y, Z) =FNM (Y, FNM (Z, F (X, Y, Z) ) )
990   DEF FNZ3 (X, Y, Z) =FNM (X, FNM (Z, F (X, Y, Z) ) )
1000  DEF FNZ4 (X, Y, Z) =FNM (X, FNM(Y, F (X, Y, Z) ) )
1010  DEF FNZ5 (X, Y, Z) =FNM (Z, F (X, Y, Z) )
1020  DEF FNZ6 (X, Y, Z) =FNM (Y, F (X, Y, Z) )
1030  DEF FNZ7 (X, Y, Z) =FNM (X, F (X, Y, Z) )
1040  FOR I=0 TO 2
1050         FOR J=0 TO 2
1060             FOR K=0 TO 2
```

```
1070                    B (19) =FNA (B(19),FNZ1 (I,J,K))            'A(1,1,1)
1080                    B (20) =FNA (B(20),FNZ2 (I,J,K))            'A(2,1,1)
1090                    B (21) =FNA (B(21),FNZ3 (I,J,K))            'A(1,2,1)
1100                    B (22) =FNA (B(22),FNZ4 (I,J,K))            'A(1,1,2)
1110                    B (23) =FNA (B(23),FNZ5 (I,J,K))            'A(2,2,1)
1120                    B (24) =FNA (B(24),FNZ6 (I,J,K))            'A(2,1,2)
1130                    B (25) =FNA (B(25),FNZ7 (I,J,K))            'A(1,2,2)
1140                    B (26) =FNA (B(26),F(I,J,K))                'A(2,2,2)
1150            NEXT K
1160      NEXT J
1170 NEXT I
1180 '---------- OUTPUT THE FORMULA FOR F (X,Y,Z)
1190 LPRINT"  F (X, Y, Z) = " ; F (0, 0, 0) ; " - " ; B (1) ; " X- " ; B (3) ;
     " Y- " ; B (5) ; " Z- " ; B (2) ; " XX- " ; B (4) ; " YY- " ; B (6) ; " ZZ+ "
     ; B (7) ; " XY+ " ; B (11) ; " XZ+ "
1200 PRINT"  F (X, Y, Z) = " ; F (0, 0, 0) ; " - " ; B (1) ; " X- " ; B (3) ;
     " Y- " ; B (5) ; " Z- " ; B (2) ; " XX- " ; B (4) ; " YY- " ; B (6) ; " ZZ+
     " ; B (7) ; " XY+ " ; B (11) ; " XZ+ "
1210 LPRINT B (15) ; " YZ- " ; B (19) ; " XYZ+ " ; B (8) ; " XXY+ " ; B (12)
     ; " XXZ+ " ; B (9) ; " XYY+ " ; B (16) ; " YYZ+ " ; B (13) ; " XZZ+ " ; B
     (7) ; " YZZ+ "
1220 PRINT B (15) ; " YZ- " ; B (19) ; " XYZ+ " ; B (8) ; " XXY+ " ; B (12) ;
     " XXZ+ " ; B (9) ; " XYY+ " ; B (16) ; " YYZ+ " ; B (13) ; " XZZ+ " ; B (7)
     " YZZ+ "
1230 LPRINT B (10) ; " XXYY+ " ; B (14) ; " XXZZ+ " ; B (18) ; " YYZZ- " ;
     B (20) ; " XXYZ- " ; B (21) ; " XYYZ- " ; B (22) ; " XYZZ- " ; B (23) ; " X
     XYYZ- " ; B (24) ; " XXYZZ- "
1240 LPRINT B (25) ; " XYYZZ- " ; B (26) ; " XXYYZZ. "
1250 PRINT B (10) ; " XXYY+ " ; B (14) ; " XXZZ+ " ; B (18) ; " YYZZ- " ; B
     (20) ; " XXYZ- " ; B (21) ; " XYYZ- " ; B (22) ; " XYZZ- " ; B (23) ; " XX
     YYZ- " ; B (24) ; " XXYZZ- "
1260 PRINT B (25) ; " XYYZZ- " ; B (26) ; " XXYYZZ. "
1270 PRINT STRING$ (79, " - ")
1280 LPRINT STRING$ (79, " - ")
1290 PRINT : PRINT
1300 '--------- CONTINUE?
1310 PRINT"  CONT=RETURN, STOP=0 " ; : A$ =INPUT$ (1)
1320 IF A$ <> CHR$ (13) AND A$ <>" 0 " THEN 1310
1330 IF A$ =CHR$ (13) THEN 60
1340 END
```

References

[HK] Hachimine, G and Kimura, M, *A method for the logical design of three valued logic functions,* Trans. IECE (Japanese), 1976, pp.711−718.

[NZK] Nakao, Z, Zukeran, C and Kyan, S, *Quaternary logic functions over Galois field* GF(4) (Pre-print).

[T] Takahashi, I. *Combinatorics and its applications* (Japanese), Iwanami 1979, pp. 199-204.