

## Part 4. "New Advanced Study of Magic Squares and Cubes"

### Chapter 7. New Method of Composing High-Dimensional Extra-Cubic Objects and their Developed Forms: Kanji Setsuda

#### Section 4. Composition of 'Three-Type Simultaneous' MS88 by Down-Converting from the Extra-Cubic Object 2<sup>6</sup>

##### #1. Our Purpose

Let's compose 6-dimensional ECO of order 2 again, and convert it down to 2-dimensional object and transform it into our true goal 'three-type simultaneous' magic squares of order 8: 'Composite', pandiagonal and self-complementary.

We want to compose it to know precisely how many standard solutions it has.

As we want to use the same ECO on our way, we design and compose it just in the same manner as in the previous section. For the next step we want to find any appropriate 'model' solution and invent any method of transformation. Yes. We want to make ECO at first and then transform SC type of MS88 into the final object.

##### #2. How do you compose ECO2<sup>6</sup>?

We take the same definition to the same ECO just as in the previous section: the same 49 simultaneous equations and 'self-complementary conditions'. We also use the same kind of Basic Diagrams modified a little for our purpose.

**BD/2^6**

1 - - - 2	3 - - - 4	33 - - - 34	35 - - - 36
5 - + - 6	7 - + - 8	37 - + - 38	39 - + - 40
9 - -   - 10	11 - -   - 12	41 - -   - 42	43 - -   - 44
13 - - - 14	15 - - - 16	45 - - - 46	47 - - - 48

  

17 - - - 18	19 - - - 20	49 - - - 50	51 - - - 52
21 - + - 22	23 - + - 24	53 - + - 54	55 - + - 56
25 - -   - 26	27 - -   - 28	57 - -   - 58	59 - -   - 60
29 - - - 30	31 - - - 32	61 - - - 62	63 - - - 64

SCMC4/MS88							
1	2	3	4	1	2	3	4
5	6	7	8	9	10	11	12
33	34	11	35	12	36	13	14
10	11	35	12	36	17	18	19
13	37	14	38	15	39	16	40
41	18	42	19	43	20	44	25
21	45	22	46	23	47	24	48
25	49	26	50	27	51	28	52
29	53	30	54	31	55	32	56
57		58		59		60	
61		62		63		64	

##### \*\* Essential 49 Equations for 'Composi te Conditions' \*\*

n1+n2+n3+n4=C	... (1);	n1+n2+n5+n6=C	... (2);	n1+n2+n9+n10=C	... (3);
n1+n2+n17+n18=C	... (4);	n1+n2+n33+n34=C	... (5);	n1+n3+n5+n7=C	... (6);
n1+n3+n9+n11=C	... (7);	n1+n3+n17+n19=C	... (8);	n1+n3+n33+n35=C	... (9);
n1+n5+n9+n13=C	... (10);	n1+n5+n17+n21=C	... (11);	n1+n5+n33+n37=C	... (12);
n1+n9+n17+n25=C	... (13);	n1+n9+n33+n41=C	... (14);	n1+n17+n33+n49=C	... (15);
n2+n4+n6+n8=C	... (16);	n2+n4+n10+n12=C	... (17);	n2+n4+n18+n20=C	... (18);
n2+n4+n34+n36=C	... (19);	n2+n6+n10+n14=C	... (20);	n2+n6+n18+n22=C	... (21);
n2+n6+n34+n38=C	... (22);	n2+n10+n18+n26=C	... (23);	n2+n10+n34+n42=C	... (24);
n2+n18+n34+n50=C	... (25);	n3+n7+n11+n15=C	... (26);	n3+n7+n19+n23=C	... (27);
n3+n7+n35+n39=C	... (28);	n3+n11+n19+n27=C	... (29);	n3+n11+n35+n43=C	... (30);
n3+n19+n35+n51=C	... (31);	n4+n8+n12+n16=C	... (32);	n4+n8+n20+n24=C	... (33);

```

n4+n8+n36+n40=C ... (34); n4+n12+n20+n28=C ... (35); n4+n12+n36+n44=C ... (36);
n4+n20+n36+n52=C ... (37); n5+n13+n21+n29=C ... (38); n5+n13+n37+n45=C ... (39);
n5+n21+n37+n53=C ... (40); n6+n14+n22+n30=C ... (41); n6+n14+n38+n46=C ... (42);
n6+n22+n38+n54=C ... (43); n7+n15+n23+n31=C ... (44); n7+n15+n39+n47=C ... (45);
n7+n23+n39+n55=C ... (46); n8+n16+n24+n32=C ... (47); n8+n16+n40+n48=C ... (48);
n8+n24+n40+n56=C ... (49);
n1+n64=n2+n63=n3+n62=n4+n61=n5+n60=n6+n59=n7+n58=n8+n57=n9+n56
=n10+n55=n11+n54=n12+n53=n13+n52=n14+n51=n15+n50=n16+n49=n17+n48
=n18+n47=n19+n46=n20+n45=n21+n44=n22+n43=n23+n42=n24+n41=n25+n40
=n26+n39=n27+n38=n28+n37=n29+n36=n30+n35=n31+n34=n32+n33=65 ... (50);

```

I found the next 'model' solution (on the right hand side in the following forms) would be suitable for our purpose. According to that model I dictated the additional part of table exchange program for transformation as follows.

#### \*\* Concept of Transformation from SC-type to 3T Simul-type \*\*

BD/SC								BD/Sml								1/Sol							
1	2	3	4	5	6	7	8	1	2	4	3	6	5	7	8	1	63	4	62	6	60	7	57
9	10	11	12	13	14	15	16	9	10	12	11	14	13	15	16	56	10	53	11	51	13	50	16
17	18	19	20	21	22	23	24	25	26	28	27	30	29	31	32	25	39	28	38	30	36	31	33
25	26	27	28	29	30	31	32	17	18	20	19	22	21	23	24	48	18	45	19	43	21	42	24
33	34	35	36	37	38	39	40	41	42	44	43	46	45	47	48	41	23	44	22	46	20	47	17
41	42	43	44	45	46	47	48	33	34	36	35	38	37	39	40	32	34	29	35	27	37	26	40
49	50	51	52	53	54	55	56	49	50	52	51	54	53	55	56	49	15	52	14	54	12	55	9
57	58	59	60	61	62	63	64	57	58	60	59	62	61	63	64	8	58	5	59	3	61	2	64

/\* Additional Program Part \*/

/\* Transform the SC-Type of MS88 into 3T Simul-Type \*/

```

tn[1]=nn[1]; tn[2]=nn[2]; tn[3]=nn[4]; tn[4]=nn[3];
tn[5]=nn[6]; tn[6]=nn[5]; tn[7]=nn[7]; tn[8]=nn[8];
tn[9]=nn[9]; tn[10]=nn[10]; tn[11]=nn[12]; tn[12]=nn[11];
tn[13]=nn[14]; tn[14]=nn[13]; tn[15]=nn[15]; tn[16]=nn[16];
tn[17]=nn[25]; tn[18]=nn[26]; tn[19]=nn[28]; tn[20]=nn[27];
tn[21]=nn[30]; tn[22]=nn[29]; tn[23]=nn[31]; tn[24]=nn[32];
tn[25]=nn[17]; tn[26]=nn[18]; tn[27]=nn[20]; tn[28]=nn[19];
tn[29]=nn[22]; tn[30]=nn[21]; tn[31]=nn[23]; tn[32]=nn[24];
tn[33]=nn[41]; tn[34]=nn[42]; tn[35]=nn[44]; tn[36]=nn[43];
tn[37]=nn[46]; tn[38]=nn[45]; tn[39]=nn[47]; tn[40]=nn[48];
tn[41]=nn[33]; tn[42]=nn[34]; tn[43]=nn[36]; tn[44]=nn[35];
tn[45]=nn[38]; tn[46]=nn[37]; tn[47]=nn[39]; tn[48]=nn[40];
tn[49]=nn[49]; tn[50]=nn[50]; tn[51]=nn[52]; tn[52]=nn[51];
tn[53]=nn[54]; tn[54]=nn[53]; tn[55]=nn[55]; tn[56]=nn[56];
tn[57]=nn[57]; tn[58]=nn[58]; tn[59]=nn[60]; tn[60]=nn[59];
tn[61]=nn[62]; tn[62]=nn[61]; tn[63]=nn[63]; tn[64]=nn[64];
*/

```

Now let me show you a sample program of mine, the simplest one designed for you to understand easily.

```

/** Make the Extra-Cubic Magic Objects of Order 2^6 and ***/
/** Transform into '3-Type Simultaneous' Magic Squares 8^2 ***/
/** 'Composite', Pan-Diagonal and Self-Complementary Type ***/
/** 'EC026Sml 3T.c' by Kanj i Setsuda ***/
/** on MacOS Xcode 1.5; Apr. 24, 2005 ***/
/**/
#include <stdio.h>
/**/
/* Variables */
short int cnt;

```

```

short CC, SSM;
short nm[65], uflg[65];
short tn[5761][65];
/**/
/* Sub-Routines */
void stp01(void), stp02(void), stp03(void), stp04(void), stp05(void);
void stp06(void), stp07(void), stp08(void), stp09(void), stp10(void);
void stp11(void), stp12(void), stp13(void), stp14(void), stp15(void);
void stp16(void), stp17(void), stp18(void), stp19(void), stp20(void);
void stp21(void), stp22(void), stp23(void), stp24(void), stp25(void);
void stp26(void), stp27(void), stp28(void), stp29(void), stp30(void);
void stp31(void), stp32(void), stp33(void), stp34(void), stp35(void);
void stp36(void), stp37(void), stp38(void), stp39(void), stp40(void);
void recrdans(void);
void prms8(short x, short y);
/**/
/* Main Program */
main(){
    short n;
    printf("\n** Make the Extra-Cubic Magic Objects of Order 2^6 and **\n");
    printf("** Transform into '3-Type Simultaneous' Magic Squares 8^2 **\n");
    printf("** 'Composite', Self-Complementary and Pan-Diagonal M.S. **\n\n");
    for(n=0; n<65; n++){nm[n]=0; uflg[n]=0; }
    CC=65; SSM=130; cnt=0;
    stp01(); /* Begin the Calculations */
    prms8(1, cnt);
    printf("\n [Count = %d]\n", cnt);
    printf("\n OK!\n");
    return 0;
}
/**/
/* Calculations */
/* Set n1 & n64 & n1<n64 */
void stp01(){
    short a, b;
    for(a=1; a<33; a++){b=CC-a;
        if((uflg[a]==0)&&(uflg[b]==0)){
            nm[1]=a; nm[64]=b;
            uflg[a]=1; uflg[b]=1;
            stp02();
            uflg[b]=0; uflg[a]=0;
        }
    }
    /* Set n2 & n63 */
    void stp02(){
        short a, b;
        for(a=64; a>0; a--){b=CC-a;
            if((uflg[a]==0)&&(uflg[b]==0)){
                nm[2]=a; nm[63]=b;
                uflg[a]=1; uflg[b]=1;
                stp03();
                uflg[b]=0; uflg[a]=0;
            }
        }
    }
    /* Set n3 & n62 */
    void stp03(){
        short a, b;
        for(a=64; a>0; a--){b=CC-a;
            if((uflg[a]==0)&&(uflg[b]==0)){
                nm[3]=a; nm[62]=b;
                uflg[a]=1; uflg[b]=1;
                stp04();
                uflg[b]=0; uflg[a]=0;
            }
        }
    }
    /* Set n4=SSM-n1-n2-n3 & n61 */
    void stp04(){
        short a, b;
        a=SSM-nm[1]-nm[2]-nm[3];
        if((0<a)&&(a<65)){b=CC-a;
            if((uflg[a]==0)&&(uflg[b]==0)){
                nm[4]=a; nm[61]=b;
                uflg[a]=1; uflg[b]=1;
                stp05();
                uflg[b]=0; uflg[a]=0;
            }
        }
    }
    /* Set n5 & n60 */
    void stp05(){
        short a, b;
        for(a=64; a>0; a--){b=CC-a;
            if((uflg[a]==0)&&(uflg[b]==0)){
                nm[5]=a; nm[60]=b;
                uflg[a]=1; uflg[b]=1;
                stp06();
                uflg[b]=0; uflg[a]=0;
            }
        }
    }
    /* Set n6=SSM-n1-n2-n5 & n59 */
}

```

```

void stp06(){
    short a, b;
    a=SSM-nm[1]-nm[2]-nm[5];
    if((0<a)&&(a<65)){b=CC-a;
        if((ufl g[a]==0)&&(ufl g[b]==0)){
            nm[6]=a; nm[59]=b;
            ufl g[a]=1; ufl g[b]=1;
            stp07();
            ufl g[b]=0; ufl g[a]=0; }}
    }
}

/* Set n8=SSM-n2-n4-n6 & n57 & n1<n8 & n1<n57 */
void stp08(){
    short a, b;
    a=SSM-nm[2]-nm[4]-nm[6];
    if((nm[1]<a)&&(a<65)){b=CC-a;
        if((b>nm[1])&&(ufl g[a]==0)&&(ufl g[b]==0)){
            nm[8]=a; nm[57]=b;
            ufl g[a]=1; ufl g[b]=1;
            stp09();
            ufl g[b]=0; ufl g[a]=0; }}
    }

/* Search Level : 2 */
/* Set n9 & n56 & n9<n2 */
void stp09(){
    short a, b;
    for(a=nm[2]-1; a>0; a--) {b=CC-a;
        if((ufl g[a]==0)&&(ufl g[b]==0)){
            nm[9]=a; nm[56]=b;
            ufl g[a]=1; ufl g[b]=1;
            stp10();
            ufl g[b]=0; ufl g[a]=0; }}
    }

/* Set n10=SSM-n1-n2-n9 & n55 */
void stp10(){
    short a, b;
    a=SSM-nm[1]-nm[2]-nm[9];
    if((0<a)&&(a<65)){b=CC-a;
        if((ufl g[a]==0)&&(ufl g[b]==0)){
            nm[10]=a; nm[55]=b;
            ufl g[a]=1; ufl g[b]=1;
            stp11();
            ufl g[b]=0; ufl g[a]=0; }}
    }

/* Set n11=SSM-n1-n3-n9 & n54 */
void stp11(){
    short a, b;
    a=SSM-nm[1]-nm[3]-nm[9];
    if((0<a)&&(a<65)){b=CC-a;
        if((ufl g[a]==0)&&(ufl g[b]==0)){
            nm[11]=a; nm[54]=b;
            ufl g[a]=1; ufl g[b]=1;
            stp12();
            ufl g[b]=0; ufl g[a]=0; }}
    }

/* Set n12=SSM-n2-n4-n10 & n53 */
void stp12(){
    short a, b;
}

/* Set n7=SSM-n1-n3-n5 & n58 */
void stp07(){
    short a, b;
    a=SSM-nm[1]-nm[3]-nm[5];
    if((0<a)&&(a<65)){b=CC-a;
        if((ufl g[a]==0)&&(ufl g[b]==0)){
            nm[7]=a; nm[58]=b;
            ufl g[a]=1; ufl g[b]=1;
            stp08();
            ufl g[b]=0; ufl g[a]=0; }}
    }

/* Set n13=SSM-n1-n5-n9 & n52 */
void stp13(){
    short a, b;
    a=SSM-nm[2]-nm[4]-nm[10];
    if((0<a)&&(a<65)){b=CC-a;
        if((ufl g[a]==0)&&(ufl g[b]==0)){
            nm[12]=a; nm[53]=b;
            ufl g[a]=1; ufl g[b]=1;
            stp13();
            ufl g[b]=0; ufl g[a]=0; }}
    }

/* Set n14=SSM-n2-n6-n10 & n51 */
void stp14(){
    short a, b;
    a=SSM-nm[2]-nm[6]-nm[10];
    if((0<a)&&(a<65)){b=CC-a;
        if((ufl g[a]==0)&&(ufl g[b]==0)){
            nm[13]=a; nm[52]=b;
            ufl g[a]=1; ufl g[b]=1;
            stp14();
            ufl g[b]=0; ufl g[a]=0; }}
    }

/* Set n15=SSM-n3-n7-n11 & n50 */
void stp15(){
    short a, b;
    a=SSM-nm[3]-nm[7]-nm[11];
    if((0<a)&&(a<65)){b=CC-a;
        if((ufl g[a]==0)&&(ufl g[b]==0)){}
    }
}

```

```

nm[15]=a; nm[50]=b;
ufl g[a]=1; ufl g[b]=1;
stp16();
ufl g[b]=0; ufl g[a]=0; }}
}

/* Set n16=SSM-n4-n8-n12 & n49 */
void stp16(){
short a, b;

/* Search Level : 3 */
/* Set n17 & n48 */
void stp17(){
short a, b;
for(a=64; a>0; a--) {b=CC-a;
if((ufl g[a]==0)&&(ufl g[b]==0)) {
    nm[17]=a; nm[48]=b;
    ufl g[a]=1; ufl g[b]=1;
    stp18();
    ufl g[b]=0; ufl g[a]=0; }
}
}

/* Set n18=SSM-n1-n2-n17 & n47 */
void stp18(){
short a, b;
a=SSM-nm[1]-nm[2]-nm[17];
if((0<a)&&(a<65)) {b=CC-a;
if((ufl g[a]==0)&&(ufl g[b]==0)) {
    nm[18]=a; nm[47]=b;
    ufl g[a]=1; ufl g[b]=1;
    stp19();
    ufl g[b]=0; ufl g[a]=0; }}
}

/* Set n19=SSM-n1-n3-n17 & n46 */
void stp19(){
short a, b;
a=SSM-nm[1]-nm[3]-nm[17];
if((0<a)&&(a<65)) {b=CC-a;
if((ufl g[a]==0)&&(ufl g[b]==0)) {
    nm[19]=a; nm[46]=b;
    ufl g[a]=1; ufl g[b]=1;
    stp20();
    ufl g[b]=0; ufl g[a]=0; }}
}

/* Set n20=SSM-n2-n4-n18 & n45 */
void stp20(){
short a, b;
a=SSM-nm[2]-nm[4]-nm[18];
if((0<a)&&(a<65)) {b=CC-a;
if((ufl g[a]==0)&&(ufl g[b]==0)) {
    nm[20]=a; nm[45]=b;
    ufl g[a]=1; ufl g[b]=1;
    stp21();
    ufl g[b]=0; ufl g[a]=0; }}
}

/* Search Level : 4 */
/* Set n25=SSM-n1-n9-n17 & n40 */
void stp25(){
short a, b;
a=SSM-nm[1]-nm[9]-nm[17];
}

a=SSM-nm[4]-nm[8]-nm[12];
if((0<a)&&(a<65)) {b=CC-a;
if((ufl g[a]==0)&&(ufl g[b]==0)) {
    nm[16]=a; nm[49]=b;
    ufl g[a]=1; ufl g[b]=1;
    stp17();
    ufl g[b]=0; ufl g[a]=0; }}
}

/* Set n21=SSM-n1-n5-n17 & n44 */
void stp21(){
short a, b;
a=SSM-nm[1]-nm[5]-nm[17];
if((0<a)&&(a<65)) {b=CC-a;
if((ufl g[a]==0)&&(ufl g[b]==0)) {
    nm[21]=a; nm[44]=b;
    ufl g[a]=1; ufl g[b]=1;
    stp22();
    ufl g[b]=0; ufl g[a]=0; }}
}

/* Set n22=SSM-n2-n6-n18 & n43 */
void stp22(){
short a, b;
a=SSM-nm[2]-nm[6]-nm[18];
if((0<a)&&(a<65)) {b=CC-a;
if((ufl g[a]==0)&&(ufl g[b]==0)) {
    nm[22]=a; nm[43]=b;
    ufl g[a]=1; ufl g[b]=1;
    stp23();
    ufl g[b]=0; ufl g[a]=0; }}
}

/* Set n23=SSM-n3-n7-n19 & n42 */
void stp23(){
short a, b;
a=SSM-nm[3]-nm[7]-nm[19];
if((0<a)&&(a<65)) {b=CC-a;
if((ufl g[a]==0)&&(ufl g[b]==0)) {
    nm[23]=a; nm[42]=b;
    ufl g[a]=1; ufl g[b]=1;
    stp24();
    ufl g[b]=0; ufl g[a]=0; }}
}

/* Set n24=SSM-n4-n8-n20 & n41 */
void stp24(){
short a, b;
a=SSM-nm[4]-nm[8]-nm[20];
if((0<a)&&(a<65)) {b=CC-a;
if((ufl g[a]==0)&&(ufl g[b]==0)) {
    nm[24]=a; nm[41]=b;
    ufl g[a]=1; ufl g[b]=1;
    stp25();
    ufl g[b]=0; ufl g[a]=0; }}
}

if((0<a)&&(a<65)) {b=CC-a;
if((ufl g[a]==0)&&(ufl g[b]==0)) {
    nm[25]=a; nm[40]=b;
    ufl g[a]=1; ufl g[b]=1;
}
}

```

```

    stp26();
    ufl g[b]=0; ufl g[a]=0; } }

}

/* Set n26=SSM-n2-n10-n18 & n39 */
void stp26(){
    short a, b;
    a=SSM-nm[2]-nm[10]-nm[18];
    if((0<a)&&(a<65)){b=CC-a;
        if((ufl g[a]==0)&&(ufl g[b]==0)){
            nm[26]=a; nm[39]=b;
            ufl g[a]=1; ufl g[b]=1;
            stp27();
            ufl g[b]=0; ufl g[a]=0; } }
    }

/* Set n27=SSM-n3-n11-n19 & n38 */
void stp27(){
    short a, b;
    a=SSM-nm[3]-nm[11]-nm[19];
    if((0<a)&&(a<65)){b=CC-a;
        if((ufl g[a]==0)&&(ufl g[b]==0)){
            nm[27]=a; nm[38]=b;
            ufl g[a]=1; ufl g[b]=1;
            stp28();
            ufl g[b]=0; ufl g[a]=0; } }
    }

/* Set n28=SSM-n4-n12-n20 & n37 */
void stp28(){
    short a, b;
    a=SSM-nm[4]-nm[12]-nm[20];
    if((0<a)&&(a<65)){b=CC-a;
        if((ufl g[a]==0)&&(ufl g[b]==0)){
            nm[28]=a; nm[37]=b;
            ufl g[a]=1; ufl g[b]=1;
            stp29();
            ufl g[b]=0; ufl g[a]=0; } }
    }

/* Set n29=SSM-n5-n13-n21 & n36 */
void stp29(){
    short a, b;
    a=SSM-nm[5]-nm[13]-nm[21];
    if((0<a)&&(a<65)){b=CC-a;

    /* Check some Composi te Sums */
void stp33(){
    short sm1, sm2, sm3;
    sm1=nm[1]+nm[3]+nm[33]+nm[35];
    sm2=nm[1]+nm[5]+nm[33]+nm[37];
    sm3=nm[1]+nm[9]+nm[33]+nm[41];
    if((sm1==SSM)&&(sm2==SSM)&&(sm3==SSM)){stp34(); }

}

/**/
/* Check some Composi te Sums */
void stp34(){
    short sm1, sm2, sm3;
    sm1=nm[1]+nm[17]+nm[33]+nm[49];
    sm2=nm[2]+nm[4]+nm[34]+nm[36];
    sm3=nm[2]+nm[6]+nm[34]+nm[38];
    if((sm1==SSM)&&(sm2==SSM)&&(sm3==SSM)){stp35(); }

}

/**/
    if((ufl g[a]==0)&&(ufl g[b]==0)){
        nm[29]=a; nm[36]=b;
        ufl g[a]=1; ufl g[b]=1;
        stp30();
        ufl g[b]=0; ufl g[a]=0; } }

}

/* Set n30=SSM-n6-n14-n22 & n35 */
void stp30(){
    short a, b;
    a=SSM-nm[6]-nm[14]-nm[22];
    if((0<a)&&(a<65)){b=CC-a;
        if((ufl g[a]==0)&&(ufl g[b]==0)){
            nm[30]=a; nm[35]=b;
            ufl g[a]=1; ufl g[b]=1;
            stp31();
            ufl g[b]=0; ufl g[a]=0; } }

}

/* Set n31=SSM-n7-n15-n23 & n34 */
void stp31(){
    short a, b;
    a=SSM-nm[7]-nm[15]-nm[23];
    if((0<a)&&(a<65)){b=CC-a;
        if((ufl g[a]==0)&&(ufl g[b]==0)){
            nm[31]=a; nm[34]=b;
            ufl g[a]=1; ufl g[b]=1;
            stp32();
            ufl g[b]=0; ufl g[a]=0; } }

}

/* Set n32=SSM-n8-n16-n24 & n33 */
void stp32(){
    short a, b;
    a=SSM-nm[8]-nm[16]-nm[24];
    b=SSM-nm[1]-nm[2]-nm[34];
    if((0<a)&&(a<65)&&(a+b==CC)){
        if((ufl g[a]==0)&&(ufl g[b]==0)){
            nm[32]=a; nm[33]=b;
            ufl g[a]=1; ufl g[b]=1;
            stp33();
            ufl g[b]=0; ufl g[a]=0; } }

}

/**/

```

```

/* Check some Composite Sums */
void stp35(){
    short sm1, sm2, sm3;
    sm1=nm[2]+nm[10]+nm[34]+nm[42];
    sm2=nm[2]+nm[18]+nm[34]+nm[50];
    sm3=nm[3]+nm[7]+nm[35]+nm[39];
    if((sm1==SSM)&&(sm2==SSM)&&(sm3==SSM)){stp36(); }
}
/**/
/* Check some Composite Sums */
void stp36(){
    short sm1, sm2, sm3;
    sm1=nm[3]+nm[11]+nm[35]+nm[43];
    sm2=nm[3]+nm[19]+nm[35]+nm[51];
    sm3=nm[4]+nm[8]+nm[36]+nm[40];
    if((sm1==SSM)&&(sm2==SSM)&&(sm3==SSM)){stp37(); }
}
/**/
/* Check some Composite Sums */
void stp37(){
    short sm1, sm2, sm3;
    sm1=nm[4]+nm[12]+nm[36]+nm[44];
    sm2=nm[4]+nm[20]+nm[36]+nm[52];
    sm3=nm[5]+nm[13]+nm[37]+nm[45];
    if((sm1==SSM)&&(sm2==SSM)&&(sm3==SSM)){stp38(); }
}
/**/
/* Check some Composite Sums */
void stp38(){
    short sm1, sm2, sm3;
    sm1=nm[5]+nm[21]+nm[37]+nm[53];
    sm2=nm[6]+nm[14]+nm[38]+nm[46];
    sm3=nm[6]+nm[22]+nm[38]+nm[54];
    if((sm1==SSM)&&(sm2==SSM)&&(sm3==SSM)){stp39(); }
}
/**/
/* Check some Composite Sums */
void stp39(){
    short sm1, sm2, sm3;
    sm1=nm[7]+nm[15]+nm[39]+nm[47];
    sm2=nm[7]+nm[23]+nm[39]+nm[55];
    sm3=nm[8]+nm[16]+nm[40]+nm[48];
    if((sm1==SSM)&&(sm2==SSM)&&(sm3==SSM)){stp40(); }
}
/**/
/* Last Check of Composite Sums */
void stp40(){
    short sm1, sm2, sm3;
    sm1=nm[8]+nm[24]+nm[40]+nm[56];
    sm2=nm[17]+nm[18]+nm[49]+nm[50];
    sm3=nm[31]+nm[32]+nm[63]+nm[64];
    if((sm1==SSM)&&(sm2==SSM)&&(sm3==SSM)){
        recrdans(); }
}
/**/
/* Transform the SC-Type into 3-T Simul MS88 and Record it to the Table */
void recrdans(void){
    tn[cnt][0]=cnt+1;
    tn[cnt][1]=nm[1];      tn[cnt][2]=nm[2];      tn[cnt][3]=nm[4];      tn[cnt][4]=nm[3];
    tn[cnt][5]=nm[6];      tn[cnt][6]=nm[5];      tn[cnt][7]=nm[7];      tn[cnt][8]=nm[8];
}

```

```

tn[cnt][9]=nm[9]; tn[cnt][10]=nm[10]; tn[cnt][11]=nm[12]; tn[cnt][12]=nm[11];
tn[cnt][13]=nm[14]; tn[cnt][14]=nm[13]; tn[cnt][15]=nm[15]; tn[cnt][16]=nm[16];
tn[cnt][17]=nm[25]; tn[cnt][18]=nm[26]; tn[cnt][19]=nm[28]; tn[cnt][20]=nm[27];
tn[cnt][21]=nm[30]; tn[cnt][22]=nm[29]; tn[cnt][23]=nm[31]; tn[cnt][24]=nm[32];
tn[cnt][25]=nm[17]; tn[cnt][26]=nm[18]; tn[cnt][27]=nm[20]; tn[cnt][28]=nm[19];
tn[cnt][29]=nm[22]; tn[cnt][30]=nm[21]; tn[cnt][31]=nm[23]; tn[cnt][32]=nm[24];
tn[cnt][33]=nm[41]; tn[cnt][34]=nm[42]; tn[cnt][35]=nm[44]; tn[cnt][36]=nm[43];
tn[cnt][37]=nm[46]; tn[cnt][38]=nm[45]; tn[cnt][39]=nm[47]; tn[cnt][40]=nm[48];
tn[cnt][41]=nm[33]; tn[cnt][42]=nm[34]; tn[cnt][43]=nm[36]; tn[cnt][44]=nm[35];
tn[cnt][45]=nm[38]; tn[cnt][46]=nm[37]; tn[cnt][47]=nm[39]; tn[cnt][48]=nm[40];
tn[cnt][49]=nm[49]; tn[cnt][50]=nm[50]; tn[cnt][51]=nm[52]; tn[cnt][52]=nm[51];
tn[cnt][53]=nm[54]; tn[cnt][54]=nm[53]; tn[cnt][55]=nm[55]; tn[cnt][56]=nm[56];
tn[cnt][57]=nm[57]; tn[cnt][58]=nm[58]; tn[cnt][59]=nm[60]; tn[cnt][60]=nm[59];
tn[cnt][61]=nm[62]; tn[cnt][62]=nm[61]; tn[cnt][63]=nm[63]; tn[cnt][64]=nm[64];
cnt++;
}

/**/
/* Print 4 Pieces of MS88 */
void prms8(short x, short y){
    short l, l8, m, n, p;
    for(m=x-1; m<y; m=m+4){
        for(n=0; n<4; n++) {printf("%25d/", tn[m+n][0]); }
        printf("\n");
        for(l=0; l<8; l++) {l8=l *8;
            for(n=0; n<4; n++) {
                printf("   ");
                for(p=1; p<9; p++) {printf("%3d", tn[m+n][l8+p]); }
            }
            printf("\n");
        }
    }
}
/**/

```

### #3. Composition of 'Three-Type Simultaneous' MS88

The next long list shows part of the execution result. It was printed out in a very short time, about a tenth of our old method. I made the monitor list of solution correspondences between the old list and new one, and I must report that the two lists completely prove to have the same solution set with each other.

```

** Make the Extra-Cubic Magic Objects of Order 2^6 and **
** Transform into '3-Type Simultaneous' Magic Squares 8^2 **
** 'Composite', Pan-Diagonal and Self-Complementary **

```

1/	2/	3/	4/
1 63 4 62 6 60 7 57 56 10 53 11 51 13 50 16 25 39 28 38 30 36 31 33 48 18 45 19 43 21 42 24 41 23 44 22 46 20 47 17 32 34 29 35 27 37 26 40 49 15 52 14 54 12 55 9 8 58 5 59 3 61 2 64	1 63 4 62 6 60 7 57 56 10 53 11 51 13 50 16 41 23 44 22 46 20 47 17 32 34 29 35 27 37 26 40 48 18 45 19 43 21 42 24 32 34 29 35 27 37 26 40 49 15 52 14 54 12 55 9 8 58 5 59 3 61 2 64	1 63 4 62 6 60 7 57 48 18 45 19 43 21 42 24 25 39 28 38 30 36 31 33 56 10 53 11 51 13 50 16 49 15 52 14 54 12 55 9 32 34 29 35 27 37 26 40 41 23 44 22 46 20 47 17 8 58 5 59 3 61 2 64	1 63 4 62 6 60 7 57 48 18 45 19 43 21 42 24 49 15 52 14 54 12 55 9 32 34 29 35 27 37 26 40 56 10 53 11 51 13 50 16 41 23 44 22 46 20 47 17 41 23 44 22 46 20 47 17 8 58 5 59 3 61 2 64
5/	6/	7/	8/
1 63 4 62 6 60 7 57 32 34 29 35 27 37 26 40 41 23 44 22 46 20 47 17 56 10 53 11 51 13 50 16 49 15 52 14 54 12 55 9 48 18 45 19 43 21 42 24 25 39 28 38 30 36 31 33 8 58 5 59 3 61 2 64	1 63 4 62 6 60 7 57 32 34 29 35 27 37 26 40 49 15 52 14 54 12 55 9 21 43 24 42 30 36 31 33 48 18 45 19 39 25 38 28 37 27 40 26 46 20 47 17 32 34 29 35 23 41 22 44 12 54 9 55 3 61 2 64	1 63 4 62 10 56 11 53 60 6 57 7 51 13 50 16 37 27 40 26 46 20 47 17 32 34 29 35 23 41 22 44 48 18 45 19 39 25 38 28 21 43 24 42 30 36 31 33 49 15 52 14 58 8 59 5 12 54 9 55 3 61 2 64	1 63 4 62 10 56 11 53 60 6 57 7 51 13 50 16 37 27 40 26 46 20 47 17 32 34 29 35 23 41 22 44 21 43 24 42 30 36 31 33 48 18 45 19 39 25 38 28 49 15 52 14 58 8 59 5 12 54 9 55 3 61 2 64











153/	154/	155/	156/
1 62 7 60 11 56 13 50 32 35 26 37 22 41 20 47 34 29 40 27 44 23 46 17 63 4 57 6 53 10 51 16 49 14 55 12 59 8 61 2 48 19 42 21 38 25 36 31 18 45 24 43 28 39 30 33 15 52 9 54 5 58 3 64	1 62 7 60 11 56 13 50 32 35 26 37 22 41 20 47 49 14 55 12 59 8 61 2 48 19 42 21 38 25 36 31 34 29 40 27 44 23 46 17 63 4 57 6 53 10 51 16 18 45 24 43 28 39 30 33 15 52 9 54 5 58 3 64	1 62 7 60 19 48 21 42 56 11 50 13 38 25 36 31 10 53 16 51 28 39 30 33 63 4 57 6 45 18 43 24 41 22 47 20 59 8 61 2 32 35 26 37 14 49 12 55 34 29 40 27 52 15 54 9 23 44 17 46 5 58 3 64	1 62 7 60 19 48 21 42 56 11 50 13 38 25 36 31 41 22 47 20 59 8 61 2 32 35 26 37 14 49 12 55 10 53 16 51 28 39 30 33 63 4 57 6 45 18 43 24 34 29 40 27 52 15 54 9 23 44 17 46 5 58 3 64
157/	158/	159/	160/
1 62 7 60 19 48 21 42 32 35 26 37 14 49 12 55 34 29 40 27 52 15 54 9 63 4 57 6 45 18 43 24 41 22 47 20 59 8 61 2 56 11 50 13 38 25 36 31 10 53 16 51 28 39 30 33 23 44 17 46 5 58 3 64	1 62 7 60 19 48 21 42 32 35 26 37 14 49 12 55 41 22 47 20 59 8 61 2 56 11 50 13 38 25 36 31 34 29 40 27 52 15 54 9 63 4 57 6 45 18 43 24 10 53 16 51 28 39 30 33 23 44 17 46 5 58 3 64	1 62 7 60 35 32 37 26 56 11 50 13 22 41 20 47 10 53 16 51 44 23 46 17 63 4 57 6 29 34 27 40 25 38 31 36 59 8 61 2 56 11 50 13 22 41 20 47 34 29 40 27 52 15 54 9 10 53 16 51 28 39 30 33 39 28 33 30 5 58 3 64	1 62 7 60 35 32 37 26 56 11 50 13 22 41 20 47 25 38 31 36 59 8 61 2 48 19 42 21 14 49 12 55 10 53 16 51 44 23 46 17 63 4 57 6 29 34 27 40 18 45 24 43 52 15 54 9 39 28 33 30 5 58 3 64
161/	162/	163/	164/
1 62 7 60 35 32 37 26 48 19 42 21 14 49 12 55 18 45 24 43 52 15 54 9 63 4 57 6 29 34 27 40 25 38 31 36 59 8 61 2 56 11 50 13 22 41 20 47 10 53 16 51 44 23 46 17 39 28 33 30 5 58 3 64	1 62 7 60 35 32 37 26 48 19 42 21 14 49 12 55 25 38 31 36 59 8 61 2 56 11 50 13 22 41 20 47 18 45 24 43 52 15 54 9 63 4 57 6 29 34 27 40 10 53 16 51 44 23 46 17 39 28 33 30 5 58 3 64	1 62 11 56 4 63 10 53 60 7 50 13 57 6 51 16 21 42 31 36 24 43 30 33 63 4 57 6 29 34 27 40 49 14 59 8 52 15 58 5 32 35 22 41 29 34 23 44 21 42 31 36 24 43 30 33 60 7 50 13 57 6 51 16 37 26 47 20 40 27 46 17 12 55 2 61 9 54 3 64	1 62 11 56 4 63 10 53 60 7 50 13 57 6 51 16 37 26 47 20 40 27 46 17 21 42 31 36 24 43 30 33 48 19 38 25 45 18 39 28 49 14 59 8 52 15 58 5 12 55 2 61 9 54 3 64
165/	166/	167/	168/
1 62 11 56 4 63 10 53 48 19 38 25 45 18 39 28 21 42 31 36 24 43 30 33 60 7 50 13 57 6 51 16 49 14 59 8 52 15 58 5 32 35 22 41 29 34 23 44 37 26 47 20 40 27 46 17 12 55 2 61 9 54 3 64	1 62 11 56 4 63 10 53 48 19 38 25 45 18 39 28 49 14 59 8 52 15 58 5 32 35 22 41 29 34 23 44 21 42 31 36 24 43 30 33 60 7 50 13 57 6 51 16 37 26 47 20 40 27 46 17 12 55 2 61 9 54 3 64	1 62 11 56 4 63 10 53 32 35 22 41 29 34 23 44 37 26 47 20 40 27 46 17 60 7 50 13 57 6 51 16 49 14 59 8 52 15 58 5 48 19 38 25 45 18 39 28 21 42 31 36 24 43 30 33 12 55 2 61 9 54 3 64	1 62 11 56 4 63 10 53 60 7 50 13 57 6 51 16 37 26 47 20 40 27 46 17 32 35 22 41 29 34 23 44 21 42 31 36 24 43 30 33 48 19 38 25 45 18 39 28 49 14 59 8 52 15 58 5 12 55 2 61 9 54 3 64
169/	170/	171/	172/
1 62 11 56 7 60 13 50 48 19 38 25 42 21 36 31 18 45 28 39 24 43 30 33 63 4 53 10 57 6 51 16 49 14 59 8 55 12 61 2 32 35 22 41 26 37 20 47 34 29 44 23 40 27 46 17 15 52 5 58 9 54 3 64	1 62 11 56 7 60 13 50 48 19 38 25 42 21 36 31 49 14 59 8 55 12 61 2 32 35 22 41 26 37 20 47 18 45 28 39 24 43 30 33 63 4 53 10 57 6 51 16 34 29 44 23 40 27 46 17 15 52 5 58 9 54 3 64	1 62 11 56 7 60 13 50 32 35 22 41 26 37 20 47 34 29 44 23 40 27 46 17 63 4 53 10 57 6 51 16 49 14 59 8 55 12 61 2 48 19 38 25 42 21 36 31 18 45 28 39 24 43 30 33 15 52 5 58 9 54 3 64	1 62 11 56 7 60 13 50 32 35 22 41 26 37 20 47 49 14 59 8 55 12 61 2 48 19 38 25 42 21 36 31 34 29 44 23 40 27 46 17 63 4 53 10 57 6 51 16 18 45 28 39 24 43 30 33 15 52 5 58 9 54 3 64
173/	170/	171/	172/
1 62 11 56 19 48 25 38 60 7 50 13 42 21 36 31 6 57 16 51 24 43 30 33 63 4 53 10 45 18 39 28 37 26 47 20 55 12 61 2 32 35 22 41 14 49 8 59 34 29 44 23 52 15 58 5 27 40 17 46 9 54 3 64	1 62 11 56 19 48 25 38 60 7 50 13 42 21 36 31 37 26 47 20 55 12 61 2 32 35 22 41 14 49 8 59 6 57 16 51 24 43 30 33 63 4 53 10 45 18 39 28 34 29 44 23 52 15 58 5 27 40 17 46 9 54 3 64	1 62 11 56 19 48 25 38 32 35 22 41 14 49 8 59 34 29 44 23 52 15 58 5 63 4 53 10 45 18 39 28 37 26 47 20 55 12 61 2 60 7 50 13 42 21 36 31 6 57 16 51 24 43 30 33 27 40 17 46 9 54 3 64	1 62 11 56 19 48 25 38 32 35 22 41 14 49 8 59 37 26 47 20 55 12 61 2 60 7 50 13 42 21 36 31 34 29 44 23 52 15 58 5 63 4 53 10 45 18 39 28 6 57 16 51 24 43 30 33 27 40 17 46 9 54 3 64
177/	178/	179/	180/
1 62 11 56 35 32 41 22 60 7 50 13 26 37 20 47 6 57 16 51 40 27 46 17 63 4 53 10 29 34 23 44 21 42 31 36 55 12 61 2 48 19 38 25 14 49 8 59 18 45 28 39 52 15 58 5 43 24 33 30 9 54 3 64	1 62 11 56 35 32 41 22 60 7 50 13 26 37 20 47 21 42 31 36 55 12 61 2 48 19 38 25 14 49 8 59 6 57 16 51 40 27 46 17 63 4 53 10 29 34 23 44 18 45 28 39 52 15 58 5 43 24 33 30 9 54 3 64	1 62 11 56 35 32 41 22 48 19 38 25 14 49 8 59 18 45 28 39 52 15 58 5 63 4 53 10 29 34 23 44 21 42 31 36 55 12 61 2 60 7 50 13 26 37 20 47 6 57 16 51 40 27 46 17 43 24 33 30 9 54 3 64	1 62 11 56 35 32 41 22 48 19 38 25 14 49 8 59 21 42 31 36 55 12 61 2 60 7 50 13 26 37 20 47 18 45 28 39 52 15 58 5 63 4 53 10 29 34 23 44 6 57 16 51 40 27 46 17 43 24 33 30 9 54 3 64
181/	182/	183/	184/
1 62 19 48 4 63 18 45 60 7 42 21 57 6 43 24 13 50 31 36 16 51 30 33 56 11 38 25 53 10 39 28 37 26 55 12 40 27 54 9 32 35 14 49 29 34 15 52 41 22 59 8 44 23 58 5 20 47 2 61 17 46 3 64	1 62 19 48 4 63 18 45 60 7 42 21 57 6 43 24 37 26 55 12 40 27 54 9 32 35 14 49 29 34 15 52 13 50 31 36 16 51 30 33 56 11 38 25 53 10 39 28 37 26 55 12 40 27 54 9 20 47 2 61 17 46 3 64	1 62 19 48 4 63 18 45 56 11 38 25 53 10 39 28 13 50 31 36 16 51 30 33 60 7 42 21 57 6 43 24 41 22 59 8 44 23 58 5 32 35 14 49 29 34 15 52 37 26 55 12 40 27 54 9 20 47 2 61 17 46 3 64	1 62 19 48 4 63 18 45 56 11 38 25 53 10 39 28 41 22 59 8 44 23 58 5 32 35 14 49 29 34 15 52 13 50 31 36 16 51 30 33 60 7 42 21 57 6 43 24 37 26 55 12 40 27 54 9 20 47 2 61 17 46 3 64





















#### #4. Discussion

Our goal objects could successfully be taken out of ECO<sup>2</sup><sup>6</sup>, overcoming any differences of dimensions or of types, in one action, that means transforming after the only one 'model' solution, as you have seen.

Though the solutions of our goal objects are quite different from the ones of ECO, the result shows our two solution sets could be connected by the only one method of transformation. There exists the one-to-one correspondence between them.

What does this result mean?

The 'origin' of 3-type simultaneous MS88 is also supposed to point at the ECO<sup>2</sup><sup>6</sup>.

There is certainly a deep, inner structural relationship among various phenomena. Another phase of 'great mysteries' is revealed to us clearly. We are sure to be able to touch it now. How exciting it is!

(Written in English on MacOS X & Xcode 1.5; May 11, 2005; Kanji Setsuda)

---

\*\*\* E-Mail Address:<jag12001@nifty.ne.jp>