



ORGANIZATION OF MEMORY CHIPS IN MEMORY SYSTEMS THAT HAVE WORD SIZE WIDER THAN 8-BIT

Omar W. Abdul-Wahab
University of Baghdad
College of Engineering
Computer Engineering Department

ABSTRACT

This paper presents a method to organize memory chips when they are used to build memory systems that have word size wider than 8-bit. Most memory chips have 8-bit word size. When the memory system has to be built from several memory chips of various sizes, this method gives all possible organizations of these chips in the memory system. This paper also suggests a precise definition of the term “memory bank” that is usually used in memory systems. Finally, an illustrative design problem was taken to illustrate the presented method practically.

الخلاصة

هذا البحث يقدم طريقة لتنظيم رقائق الذاكرة المستخدمة لبناء أنظمة الذاكرة التي يكون عرض الكلمة فيها أكبر من 8 خانات. معظم رقائق الذاكرة يكون عرض كلمتها 8- خانات. عندما يراد بناء نظام ذاكرة باستخدام عدة رقائق ذاكرة ذات أحجام مختلفة فإن هذه الطريقة تعطي كل التنظيمات الممكنة لهذه الرقائق في نظام الذاكرة. هذا البحث يقترح أيضاً تعريفاً دقيقاً لمصطلح (صف الذاكرة) الذي يستخدم عادةً في أنظمة الذاكرة. أخيراً تم أخذ مسألة تصميم توضيحية لشرح الطريقة المقدمة عملياً.

KEYWORDS

Data bank, address bank, p -tuple (n_1, n_2, \dots, n_p) , combinations with repetition.

INTRODUCTION

Microprocessors are categorized into 4-bit, 8-bit, 16-bit, 32-bit, and 64-bit microprocessors. There are two points of views in identifying the category of a microprocessor. The first one, which is adopted in (Brey 2009) (Antonakos 2006) (Rafiqzaman 1995) (Rafiqzaman 2008) (Tischer and Jennrich 1996) (Triebel 1996) considers the internal architecture, i.e. the width of the internal data bus of the microprocessor, while the second one, which is adopted in (Protopapas 1988) (Godse 2007) (Antonakos 1999) considers the width of its external data bus. The difference becomes clear for microprocessors which have internal data bus wider than their external data bus, as in Intel 8088 and Motorola 68000. From the first point of view the 8088 is a 16-bit microprocessor and the 68000 is a 32-bit microprocessor, while from the second point of view the 8088 is an 8-bit microprocessor and the 68000 is a 16-bit microprocessor. Some references, such as (Mathur 1989), comprises between the two points of views so that they categorize the 8088 as a hybrid 8/16-bit microprocessor and the 68000 as a

hybrid 16/32-bit microprocessor. To avoid any confusion, the phrase “microprocessor with d -bit external data bus” is used throughout this paper.

In many memory applications the required memory capacity or word size cannot be satisfied by one memory chip. Instead, several memory chips must be combined to provide the capacity and/or the word size (Tocci and Widmer 2001). The concept of building a memory device from smaller memories is called *memory banking* (Page 2009) or *memory array design* (Rafiqzaman 2008), and the collection of smaller memories is called *memory bank*. There are two main advantages of memory banking. Firstly, from the memory designers' point of view, it prevents scale from becoming a problem. Secondly, the use of banked memory allows improving memory access performance for microprocessors that have external data bus wider than 8-bit (Page 2009). The second advantage is achieved by implementing the physical address space of the microprocessor as a number of independent byte-wide banks and each bank supplies just eight lines of the microprocessor's data

bus (Triebel 1996). In this way, all the bytes of a d -bit aligned word that are scattered amongst the banks can be accessed in parallel.

Memory locations are still numbered byte-wise either from the low order bank to the high order bank (little endian) such as the Intel family of microprocessors or from the high order bank to the low order bank (big endian) such as the Motorola family of microprocessors (Tanenbaum 2006) (Berger 2005) (Balch 2003). Since contiguous memory addresses represent memory locations lying in contiguous banks, the sizes of the memory banks must be equal.

A memory system generally consists of several memory chips. Generally, the word size of the memory system is determined by the width of the external data bus of the microprocessor (Mathur 1989). In (Shiva 2008), a method is given to design a memory system using memory chips. However, this method assumes equal size memory chips and does not handle memory systems that have word size wider than 8-bit. The method presented in this paper is a general method to design memory systems that have 8-bit word size or wider from different size memory chips.

DATA BANK AND ADDRESS BANK

The presented method needs a precise definition of the term *memory bank*. As defined in (Page 2009), a memory bank is a collection of memory devices. The system shown in **Fig. 1** is a 256 KB memory system built from eight 32 KB memory chips and interfaced to the 80486 microprocessor (Brey 2009). From one point of view there are four memory banks: chip collection (U6, U10), chip collection (U5, U9), chip collection (U4, U8), and chip collection (U3, U7). From another point of view there are two memory banks: chip collection (U3, U4, U5, U6) and chip collection (U7, U8, U9, U10). To distinguish between the two points of views, the term bank can be given a more descriptive definition which results in classifying it into *data bank* and *address bank*. Data bank is defined as the collection of memory chips that are connected to the same 8-bit data bus lines. On the other hand, address bank is defined as the collection of memory chips for which the maximum number of the higher address lines in the memory map of the memory system remain constant. Thus, the memory system shown in **Fig. 1** consists of four data banks and two address banks. The data banks are as follows: chip

collection U6 and U10 which are connected to $D_0 - D_7$, chip collection U5 and U9 which are connected to $D_8 - D_{15}$, chip collection U4 and U8 which are connected to $D_{16} - D_{23}$, and chip collection U3 and U7 which are connected to $D_{24} - D_{31}$. The address banks are as follows: chip collection U3, U4, U5, and U6 for which the address lines A_{17} through A_{31} in the memory map of the memory system remain constant, and chip collection U7, U8, U9, and U10 for which the same address lines remain constant. **Fig. 2** shows a 128 KB memory system built from two 32 KB and one 64KB memory chips and interfaced to the 8086 microprocessor. For this memory system, there are two data banks and one address bank. The first data bank is chip collection U1 and U2 which are connected to $D_0 - D_7$ and the second data bank consists of chip U3 only which is connected to $D_8 - D_{15}$. To identify the address bank, memory chips U1 and U2 are considered as one unit so that the sum of their sizes equals to the size of U3. For chips U1 and U2, address lines A_{17} through A_{19} in the memory map remain constant. For chip U3, the same address lines have the same value as that of U1 and U2. Hence, U1 and U2 together with U3 constitute the address bank. This means that the largest memory chip in any data bank is considered as a reference in constructing address banks.

The address of any memory location determines the address bank and the data bank that contain this memory location. While the higher address lines

determine the address bank, the $\log_2 \frac{d}{8}$ lower

address bits (which are encoded internally by the microprocessor) determine the data bank. Therefore, a data bank does not contain memory locations that have arbitrary addresses; rather, it contains memory

locations that have the same value of the $\log_2 \frac{d}{8}$

lower address bits. Thus, for a given memory location, its data bank is determined by:

$$i = (\text{Address of memory location}) \bmod \left(\frac{d}{8}\right) \quad (1)$$

where i is the index of the data bank.



CONSTRUCTION OF DATA BANKS

To design an M KB memory system whose word size is d -bit using N_1 memory chips of size M_1 KB and N_2 memory chips of size M_2 KB up to N_p memory chips of size M_p KB (so that

$$\sum_{i=1}^p N_i \times M_i = M),$$

there may be more than one possible arrangement of these memory chips among data banks. The following procedure can be used to determine these arrangements:

1- Since the size of the data banks in a memory system must be equal, the amount of memory in each data bank is

$$D = \frac{\text{Total amount of memory}}{\text{Word size of the memory subsystem}/8} = \frac{M}{d/8} \quad (2)$$

2- For each data bank, all the possible p -tuples (n_1, n_2, \dots, n_p) must be determined, where n_i are nonnegative integers that represent the number of memory chips of size M_i and satisfy the equation

$$M_1 \times n_1 + M_2 \times n_2 + \dots + M_p \times n_p = D \quad (3)$$

This step results in **Table 1**.

If eq. (3) has no nonnegative integer solution, then any data bank cannot be constructed using the desired sizes of memory chips and hence the memory system cannot be designed.

Since the amount of memory in each data bank is the same (D KB), step 2 need not be repeated for all data banks. Instead, **Table 1** holds for all data banks.

3- Since there are $\frac{d}{8}$ data banks, therefore, $\frac{d}{8}$ p -tuples (n_1, n_2, \dots, n_p) must be chosen from **Table 1** to implement the total memory system. The number of ways of choosing a set of size $\frac{d}{8}$ from a set of size r that may be repeated any number of times is the combinations with repetition (Stirzaker 2003) (Kreyszig 2006). It is given by

$$C = \binom{r + \frac{d}{8} - 1}{\frac{d}{8}} = \frac{(r + d/8 - 1)!}{\frac{d}{8}! \times (r - 1)!} \quad (4)$$

However, not all of these combinations satisfy the requirements of the total number of memory chips. From these combinations, the only possible ones are those satisfying the following summations

$$\sum_{j=1}^{d/8} n_{1i(j)} = N_1, \quad \sum_{j=1}^{d/8} n_{2i(j)} = N_2, \quad \dots, \quad \text{and} \quad \sum_{j=1}^{d/8} n_{pi(j)} = N_p \quad (5)$$

where $i(j)$ is the index of the ordered tuple in the combinations mentioned above. If there is no combination satisfies the above summations, then the memory system cannot be designed using the desired number of memory chips. **Fig. 3** shows a flow chart of an algorithm that determines these combinations.

CONSTRUCTION OF ADDRESS BANKS

After constructing data banks, address banks must be constructed. A suitable construction of address banks helps in simplifying the design of the decoder of the memory system.

A suitable way for constructing address banks is to sort the memory chips in each data bank in an ascending order according to their sizes. Then, looking at the first memory chip in each data bank, if all of these memory chips are of the same size, they would compose the first address bank; otherwise, the largest memory chip among these memory chips would be considered as a reference and in each data bank the remaining memory chips are collected with the first one so that they compose the same amount of memory as that of the reference one. The resultant memory chips in all data banks would compose the first address bank. This procedure is repeated for constructing other address banks (if any) with the remaining memory chips. The sizes of the address banks need not be equal.

DESIGN OF A 512 KB MEMORY SYSTEM WHOSE WORD SIZE IS 32-BIT

To illustrate the presented method practically, a 512 KB memory system whose word size is 32-bit is to be designed using eight 32 KB and four 64 KB memory chips. Here, $M = 512$, $N_1 = 8$, $M_1 = 32$, $N_2 = 4$, $M_2 = 64$, and $d = 32$. The amount of memory in each data bank, according to eq. (2), is

$$D = \frac{M}{d/8} = \frac{512}{32/8} = 128 \text{ KB.}$$

Substituting $M_1 = 32$ and $M_2 = 64$ in eq. (3) yields

$$32 \times n_1 + 64 \times n_2 = 128 \tag{6}$$

and for each data bank the ordered pair (n_1, n_2) must satisfy eq. (6). Fig. 4 shows the graph of eq. (6) which is a line in the $n_1 n_2$ -plane. However, since n_1 and n_2 are nonnegative integers, only three points on this line satisfy eq. (6), which are (0,2), (2,1), and (4,0). Listing (1) is a C program that determines these solutions. The result is shown in Table 2.

There are $\frac{d}{8} = \frac{32}{8} = 4$ data banks, therefore, four ordered pairs (n_1, n_2) must be chosen from Table 2 with the possibility of repetition. The number of combinations when repetition is possible, according to eq. (4), is

$$C = \binom{r + \frac{d}{8} - 1}{\frac{d}{8}} = \binom{3 + 4 - 1}{4} = \binom{6}{4} = \frac{6!}{4! \times 2!} = 15$$

From these 15 combinations we pick only those satisfying eq. (5), i.e., those satisfying

$$\sum_{j=1}^4 n_{1i(j)} = N_1 = 8 \quad \text{and} \quad \sum_{j=1}^4 n_{2i(j)} = N_2 = 4.$$

Listing (2) is a C program implementation of the algorithm shown in Fig. 3 with $p = 2$, $r = 3$, and

$d = 32$. Running this program gives the combinations $\{(2,1),(2,1),(2,1),(2,1)\}$, $\{(0,2),(0,2),(4,0),(4,0)\}$, and $\{(0,2),(2,1),(2,1),(4,0)\}$.

Fig.5 through Fig. 7 show the schematic diagrams of the memory system for these three combinations of the data banks after constructing the address banks for each case.

CONCLUSIONS

From the presented method it is obvious that designing a memory system whose word size is wider than 8-bit is not as simple and straightforward as designing a memory system whose word size is 8-bit wide. The existence of more than one data bank raises the problem of distributing the memory chips among them so that they have the same amount of memory. This makes some restrictions on the size and number of the memory chips so that in some cases it is impossible to design a memory system from a given set of memory chips either because of their sizes or because of their number. On the other hand, the existence of more than one data bank makes the design of the decoder of the memory system more complex since the chip select signal must activate all the data banks simultaneously. This illustrates the need to suitably construct the address banks to simplify the design of the decoder. Finally, this method can be included in references that cover the subject of memory interface with advanced microprocessors.

Listing (1)

```
#include<stdio.h>
main()
{
    int n1,n2;
    for(n1=0;n1<=8;n1++)
    for(n2=0;n2<=4;n2++)
    if((32*n1+64*n2)==128)
    printf("%d\t%d\n",n1,n2); }
```

Listing (2)

```
#include<stdio.h>
main()
{
    int i,j,k,m,sumn1,sumn2, n[5][5];
    n[1][1]=0;n[2][1]=2;n[1][2]=2;n[2][2]=1;
    n[1][3]=4;n[2][3]=0;
    for(i=1;i<=3;i++) for(j=1;j<=3;j++)
    for (k=1;k<=3;k++) for (m=1;m<=3,m++)
    {
        if((m>=k)&&(k>=j)&&(j>=i))
```



```
{    sumn1=n[1][i]+n[1][j]+n[1][k]+n[1][m];
    sumn2=n[2][i]+n[2][j]+n[2][k]+n[2][m];
    if((sumn1==8)&&(sumn2==4))
{    printf("\n %d %d",n[1][i],n[2][i]);
    printf("\n %d %d",n[1][j],n[2][j]);
    printf("\n %d %d",n[1][k],n[2][k]);
    printf("\n %d %d",n[1][m],n[2][m]);
} } }
```

REFERENCES

- Antonakos, J. L., "The Intel Microprocessor Family", Delmar Thomson Learning, 2006.
- Antonakos, J. L., "The 68000 Microprocessor", Prentice-Hall, 1999.
- Balch, M., "Complete Digital Design", McGraw-Hill, 2003.
- Berger, A. S., "Hardware and Computer Organization", Elsevier, 2005.
- Brey, B. B., "The Intel Microprocessors", Prentice-Hall, 2009.
- Godse, A. P. and Godse, D. A., "Microprocessor Interfacing Techniques", Technical Publications pune, 2007.
- Kreyszig, E., "Advanced Engineering Mathematics", John Wiley and Sons, 2006.
- Mathur, A. P., "Introduction to Microprocessors", Tata McGraw-Hill, 1989.
- Page, D., "A Practical Introduction to Computer Architecture", Springer, 2009.
- Protopapas, D. A., "Microcomputer Hardware Design", Prentice-Hall, 1988.
- Rafiqzaman, M., "Microprocessors and Microcomputer-Based System Design", 1995.
- Rafiqzaman, M., "Microprocessor Theory", John Wiley & Sons, 2008.
- Shiva, S. G., "Computer Organization, Design and Architecture", Marcel Dekker, 2008.
- Stirzaker, D., "Elementary Probability", Cambridge University Press, 2003.
- Tanenbaum, A. S., "Structured Computer Organization", Pearson Education, 2006.
- Tischer, M. and Jennrich B., "PC Intern", Abacus, 1996.
- Tocci, R. J. and Widmer, N. S., "Digital Systems", Prentice-Hall, 2001.
- Triebel, W. A., "The 80386, 80486, and Pentium Processors", Prentice Hall, 1996.

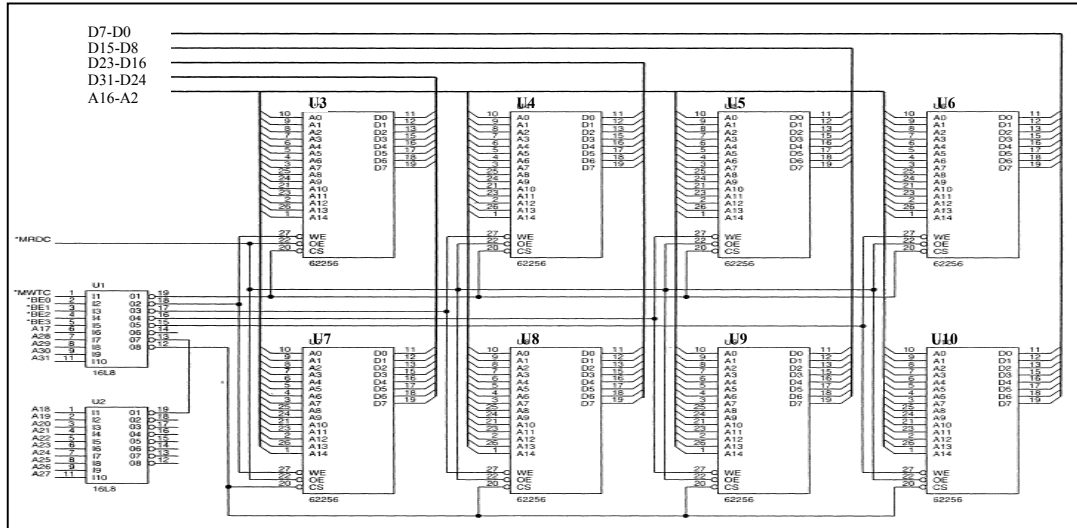


Fig. 1: A 256 KB memory system interfaced to the 80486 microprocessor.

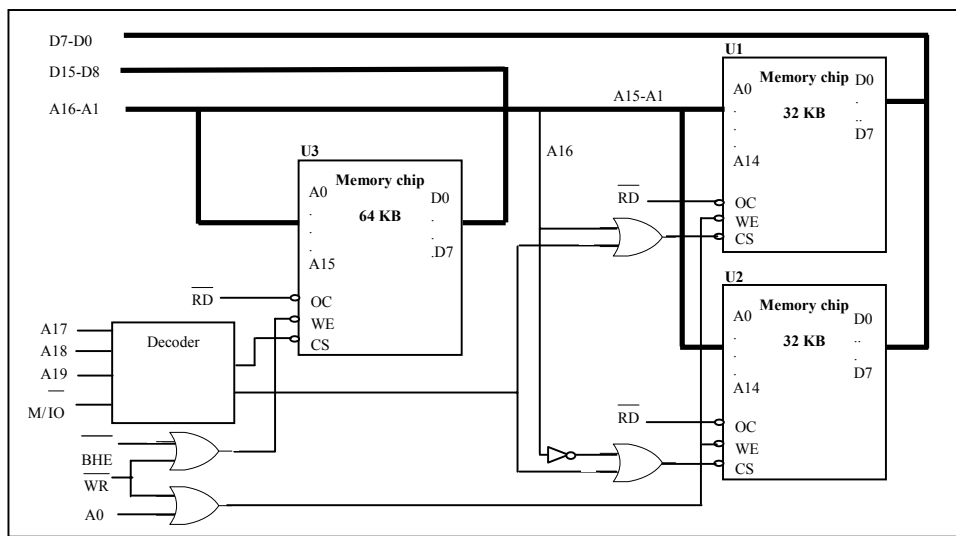


Fig. 2: A 128 KB memory system interfaced to the 8086 microprocessor.

Table 1: Nonnegative integer solutions of eq. (3)

$(n_1$	n_2	\dots	$n_p)$
$(n_{11}$	n_{21}	\dots	$n_{p1})$
$(n_{12}$	n_{22}	\dots	$n_{p2})$
\cdot	\cdot	\dots	\cdot
\cdot	\cdot	\dots	\cdot
\cdot	\cdot	\dots	\cdot
$(n_{1r}$	n_{2r}	\dots	$n_{pr})$

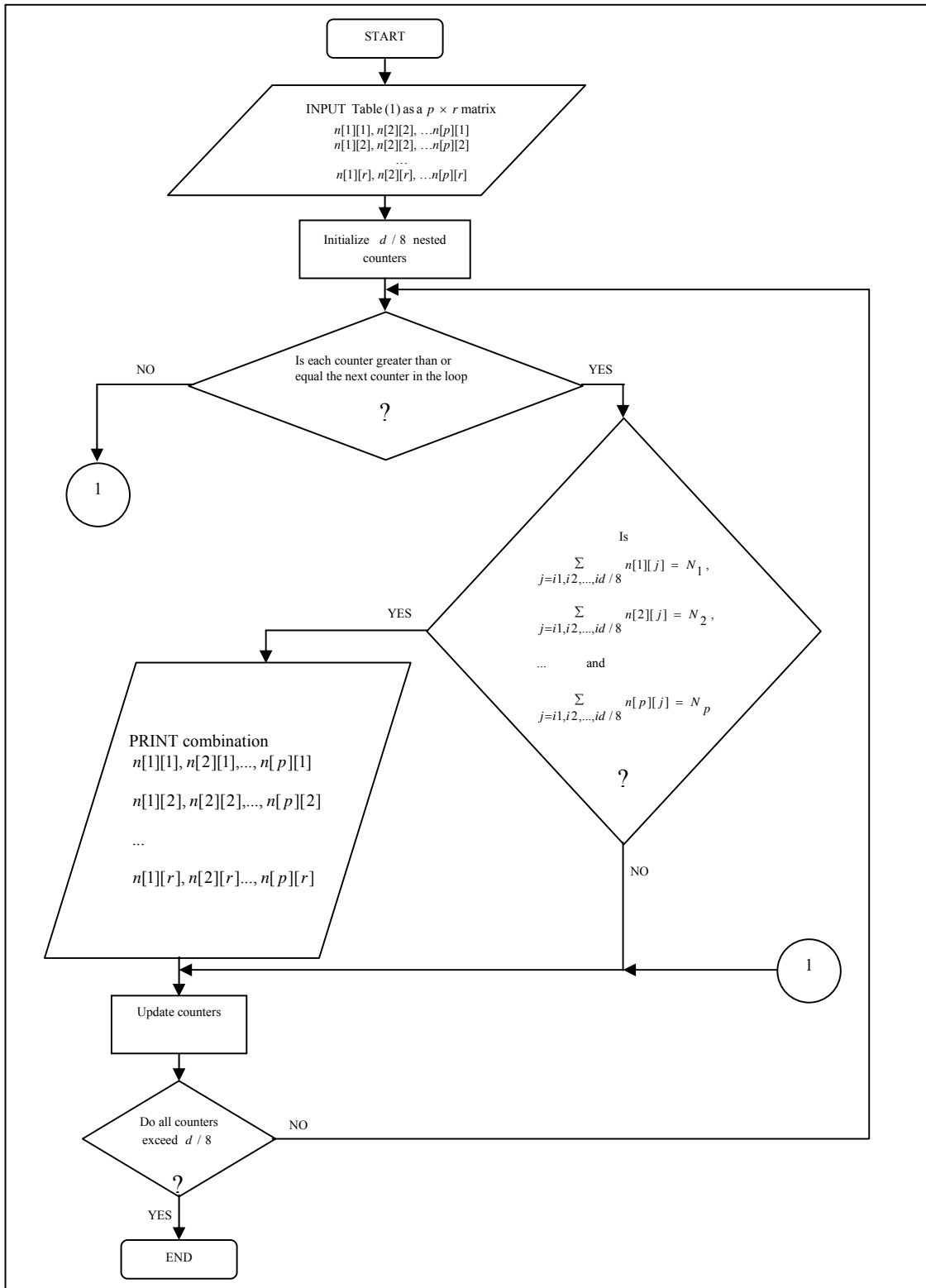


Fig. 3: Flow chart of an algorithm that determines the combinations that represents all the possible arrangements of memory chips among data banks

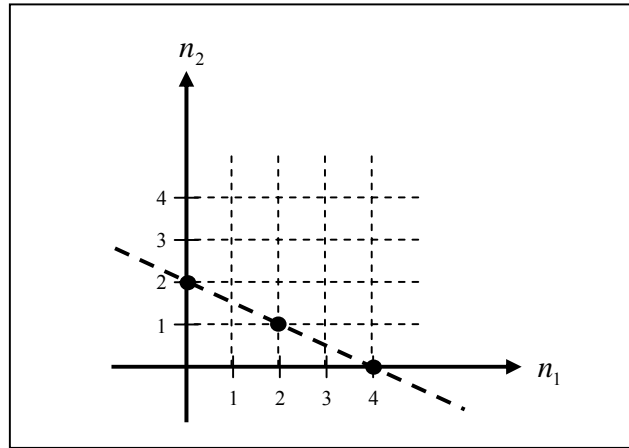


Fig. 4: Graph of eq. (6) with its nonnegative integer solutions.

Table 2: Nonnegative integer solutions of eq. (6).

$(n_1$	$n_2)$
(0	2)
(2	1)
(4	0)

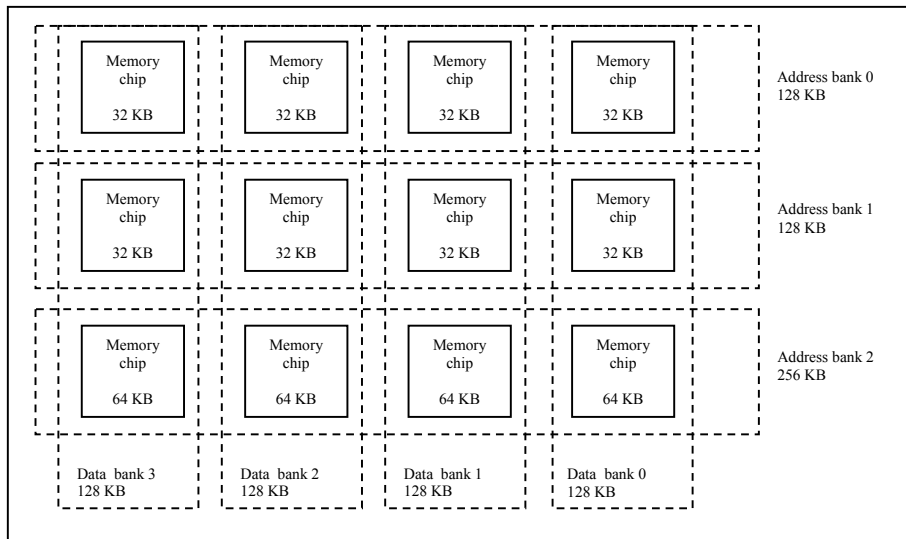


Fig. 5: Data banks arranged as $\{(2,1), (2,1), (2,1), (2,1)\}$

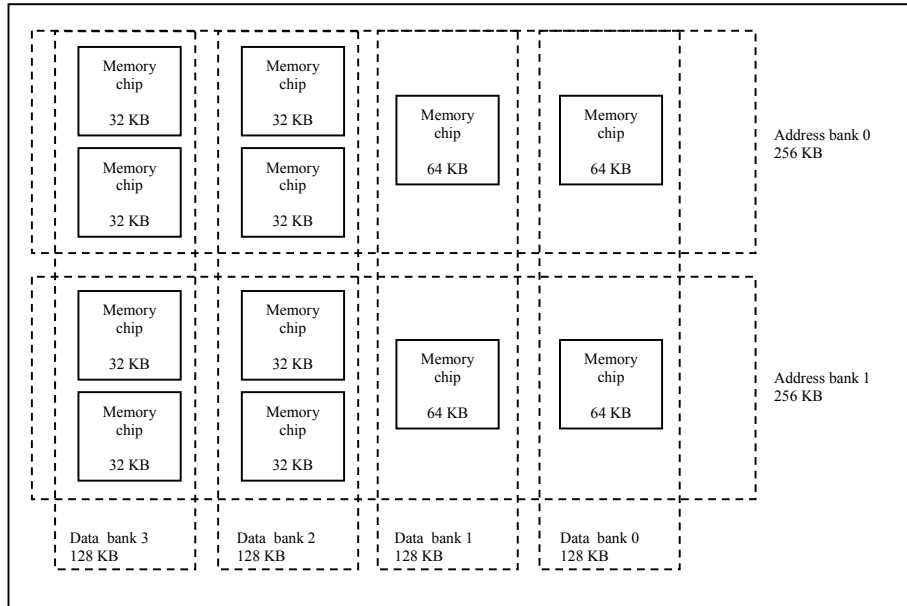


Fig. 6: Data banks arranged as $\{(0,2), (0,2), (4,0), (4,0)\}$

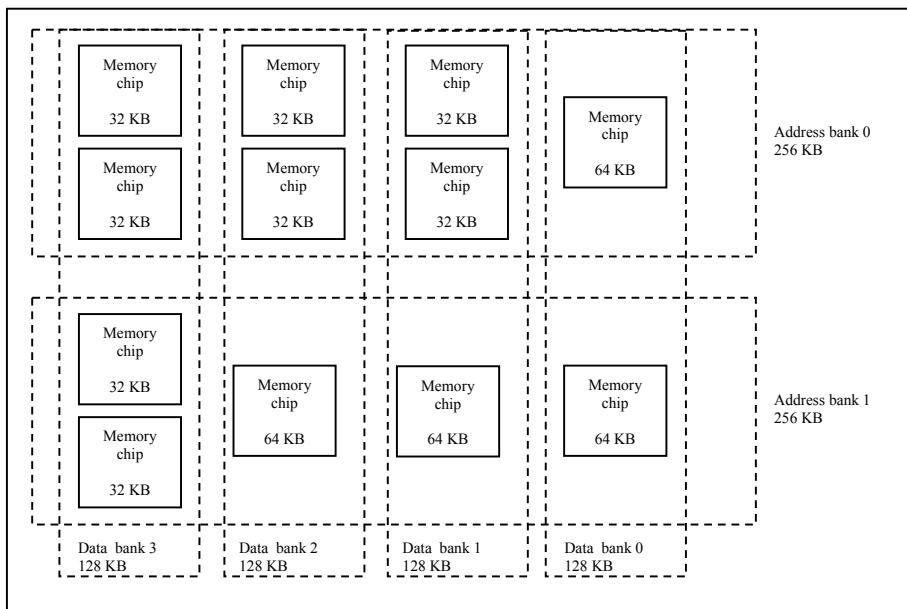


Fig. 7: Data banks arranged as $\{(0,2), (2,1), (2,1), (4,0)\}$