

Organization of Memory: Banks and Chips

This lecture focuses on a standard arrangement for organizing memory into interleaved banks.

After discussing the organization, we shall present the advantages of the banked memory concept.

For this lecture, we shall focus on a memory system that is so small that it is almost ridiculous. The concepts apply to all larger memories.

The memory will have 64 entries at addresses 0 through 63.

Noting that $2^6 = 64$, we solve the equation $2^N = 64$ to get $N = 6$. Each address to this memory will have six bits.

Put another way, the MAR (Memory Address Register) for this memory will have six bits.

Note that I have not specified the size of the addressable unit.

Is it 8 bits for a 64 x 8 memory? Is it 16 bits for a 64 x 16 memory?

I shall say more on that later. For now, I say “**64 word memory**”.

The Basic Idea of Interleaving

The idea of interleaving memory is that memory is divided into banks. Each bank is to be considered as having the same addressable unit as the main memory.

We shall show how to implement a memory bank from different sized memory chips in a future slide.

For the moment, I focus on division of the address space.

Consider our 64-word memory, without specifying the word size.

If the memory is 4-way interleaved, this means that it is implemented using four banks, numbered 0 through 3.

If the memory is 8-way interleaved, this means that it is implemented using eight banks, numbered 0 through 7.

Each bank will deliver a word that is the size of the memory addressable unit.

The Basic Idea of Low–Order Interleaving

In low–order interleaving, consecutive addresses in the memory will be found in different memory banks.

Consider a 64–word memory that is 4–way interleaved. This means that there are four memory banks, each holding 16 words.

If this memory is also low–order interleaved, we have the following allocation of words to banks.

Bank 0: Words 00, 04, 08, 12, 16, 20, 24, ..., 60

Bank 1: Words 01, 05, 09, 13, 17, 21, 25, ..., 61

Bank 2: Words 02, 06, 10, 14, 18, 22, 26, ..., 62

Bank 3 Words 03, 07, 11, 15, 19, 23, 27, ..., 63

Again, we have not yet specified the size of the memory word.

The Address Space for Low-Order Interleaving

When a memory is N -way interleaved, we always find that $N = 2^K$. This is due to the structure of the memory address.

For $K = 1$, we have 2-way interleaving.

For $K = 2$, we have 4-way interleaving.

For $K = 3$, we have 8-way interleaving.

For $K = 4$, we have 16-way interleaving.

For each scheme, the low-order K bits of the address select the bank.

For a 64-word, 4-way low-order interleaved memory, the address structure is as follows. Each address is a 6-bit unsigned binary number.

Bit	5	4	3	2	1	0
Use	Offset of word in a given bank				Bank number	

An Example of Low-Order Interleaving

The first 12 words in the 64 word memory that is 4-way low-order interleaved.

Bank	Word Address	Binary Address	Split Address	Offset in Bank	Bank Number
Bank 0:	00	000000	0000 00	0	0
	04	000100	0001 00	1	0
	08	001000	0010 00	2	0
Bank 1:	01	000001	0000 01	0	1
	05	000101	0001 01	1	1
	09	001001	0010 01	2	1
Bank 2:	02	000010	0000 10	0	2
	06	000110	0001 10	1	2
	10	001010	0010 10	2	2
Bank 3:	03	000011	0000 11	0	3
	07	000111	0001 11	1	3

11 001011 0010 11 2 3

Another Example of Low-Order Interleaving

Consider a 64-word memory that is 8-way low-order interleaved.

Here is the address structure. Here $8 = 2^3$.

Bit	5	4	3	2	1	0
Field	Offset in the bank			Bank number		

Here is the structure seen in the first two banks of this memory.

Bank	Word Address	Binary Address	Split Address	Offset in Bank	Bank Number
Bank 0:	00	000000	000 000	0	0
	08	001000	001 000	1	0
	16	010000	010 000	2	0
Bank 1:	01	000001	000 001	0	1
	09	001001	001 001	1	1
	17	010001	010 001	2	1

Banks and Chips

Suppose a 64–byte memory that is to be implemented using chips that are 16 bytes: a 64 x 8 memory from 16 x 8 memory chips.

Note that $(64 \bullet 8) / (16 \bullet 8) = 4$. This is a 4–way interleaved memory.

It has four banks. Each bank contains one chip.

Because the width of the memory chip is the same as that of the memory that is constructed, this interleaving is simple.

When the width of the entry in the chip does not match that of the main memory, we have to pay a bit more attention to details.

A 64 x 16 memory implemented with 16 x 8 chips.

The chip count is easy: $(64 \bullet 16) / (16 \bullet 8) = 4 \bullet 2 = 8$.

A 64 x 32 memory implemented with 16 x 8 chips.

The chip count is easy: $(64 \bullet 32) / (16 \bullet 8) = 4 \bullet 4 = 16$.

But how does the bank count relate to the chip count.

Multiple Chips per Bank

Consider a main memory of size $L1 \times W1$ built from chips that are $L2 \times W2$.

The number of banks is $L1 / L2$.

The number of chips per bank is $W1 / W2$.

The total number of chips in the memory is $(L1 \bullet W1) / (L2 \bullet W2)$.

Technical note: This applies only when the main memory width ($W1$) is an integer multiple of the chip width ($W2$). For us, that is always the case.

A 64 x 8 memory implemented with 16 x 8 chips.

$L1 = 64, W1 = 8, L2 = 16, W2 = 8. (L1/L2) = 4$ and $(W1/W2) = 1$.

There are four banks, each with one chip. There are four chips.

A 64 x 16 memory implemented with 16 x 8 chips.

$L1 = 64, W1 = 16, L2 = 16, W2 = 8. (L1/L2) = 4$ and $(W1/W2) = 2$.

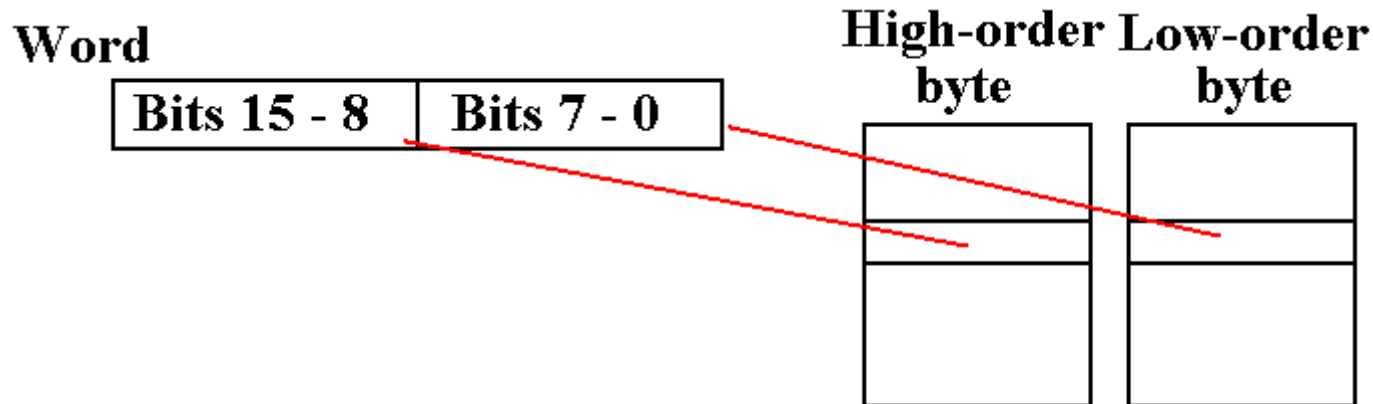
There are four banks, each with 2 chips. There are 8 chips total.

Multiple Chips per Bank: Word Allocation

Consider a 32 x 16 memory, implemented with 8 x 8 chips. Each bank must appear to be 8 x 16: 8 entries of 16 bits each.

There will be $32 / 8 = 4$ memory banks, each with $16 / 8 = 2$ chips.

In this arrangement, a 16-bit word will be split over two chips.



More on Multiple Chips per Bank

Consider bank 1 in this memory holding words 1, 5, 9, 13, 17, 21, 25, and 29. As noted above, this 8 word bank comprises two chips, each with 8 bytes.

Here is the allocation of bytes to the chips.

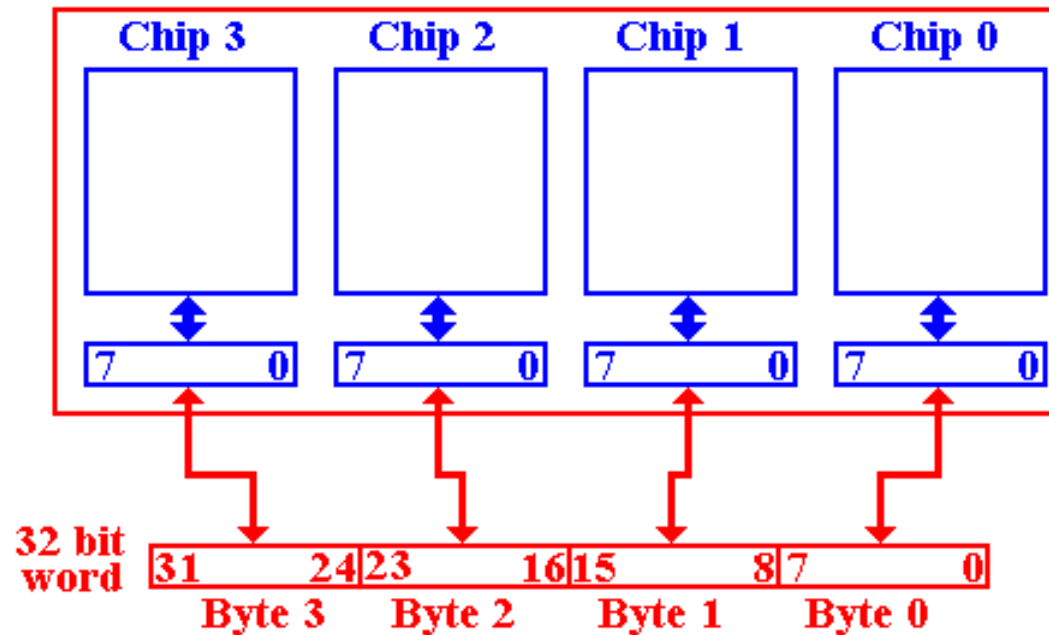
Offset	High-Order Chip	Low-Order Chip
0	Bits 15 – 8 of word 1	Bits 7 – 0 of word 1
1	Bits 15 – 8 of word 5	Bits 7 – 0 of word 5
2	Bits 15 – 8 of word 9	Bits 7 – 0 of word 9
3	Bits 15 – 8 of word 13	Bits 7 – 0 of word 13
4	Bits 15 – 8 of word 17	Bits 7 – 0 of word 17
5	Bits 15 – 8 of word 21	Bits 7 – 0 of word 21
6	Bits 15 – 8 of word 25	Bits 7 – 0 of word 25
7	Bits 15 – 8 of word 29	Bits 7 – 0 of word 29

Another Look at Multiple Chips per Bank

Consider a 128K by 32 memory, fabricated from 4K by 8 memory chips.

The memory has 32 banks, each of which must appear to be 4K by 32.

How is this done?



Each of the four 4K by 8 chips holds one of the four bytes in the 32-bit word that is held in the main memory.

Sample Problem from the Textbook

Suppose that a 2M x 16 main memory is built using 256K x 8 RAM chips and memory is word addressable.

a) **How many RAM chips are necessary?**

Answer: $2\text{M} = 2^{21}$, $256\text{K} = 2^{18}$, $16 = 2^4$, and $8 = 2^3$.

The number is $(2\text{M} \cdot 16) / (256\text{K} \cdot 8) = (2^{21} \cdot 2^4) / (2^{18} \cdot 2^3)$
 $= 2^{25} / 2^{21} = 2^4 = \mathbf{16}$.

b) **How many RAM chips per memory word?**

Answer: This question is not worded well. The only way it can make sense is to note that each chip holds one byte for each addressable item.

The design calls for **two chips per 16-bit word**.
One chip holds the high-order byte for the word.
The other chip holds the low-order byte for the word.

Sample Problem from the Textbook

Suppose that a 2M x 16 main memory is built using 256K x 8 RAM chips and memory is word addressable.

c) How many address bits are needed for each RAM chip?

Answer: $256K = 2^{18}$, so **18 address bits for each chip.**

d) How many banks will this memory have?

Answer: The number of banks depends only on the number of addressable items in each of the main memory and the RAM chips.
It does not depend on the size of the addressable item.

$$\text{Bank count} = 2M / 256K = 2^{21} / 2^{18} = 2^3 = \mathbf{8}.$$

e) How many address bits are needed for all of the memory?

Answer: $2M = 2^{21}$, so **21 address bits are needed.**

Of the 21, 18 bits go to each chip, leaving 3 bits to select the bank.
There are then $2^3 = 8$ banks.

Sample Problem from the Textbook

Suppose that a 2M x 16 main memory is built using 256K x 8 RAM chips and memory is word addressable.

f) If high-order interleaving is used, where would address 14 (0x0E) be located.

ANSWER: If high-order interleaving is used, then the first 256 K words will be found in **bank 0**. This is where address 14 is found.

Sample Problem from the Textbook

g) If low-order interleaving is used, where would address 14 (0x0E) be located.

ANSWER: If there are eight memory banks, then we have 8-way interleaving. We examine the address structure for 8-way low order interleaving.

Hexadecimal address 0xE is binary 1110.

The memory calls for 21-bit addressing. The address lines are labeled $A_{20} - A_0$.

Address bits $A_2 - A_0$ will specify the bank.

All other address bits, here $A_{20} - A_3$, specify the offset within the bank.

Bits	20 through 4	3	2	1	0
Address	All 0	1	1	1	0
Field	Offset within the bank		Bank number		

The bank number is binary 110, or decimal 6 (also hexadecimal 6).

