MOTOROLA
SEMICONDUCTOR
APPLICATION NOTE

Order this document
as AN452/D

AN452

Using the MC68HC11K4 Memory Mapping Logic

By Steven McAslan
MCU Applications Engineering
Motorola Ltd,
East Kilbride, Scotland

INTRODUCTION

The MC68HC11K4 is provided with memory expansion logic which allows the 64K Byte addressing range of the 68HC11 CPU to be extended to more than 1Mbyte. This application note discusses the operation of this logic and provides examples of memory maps and possible hardware configurations.

THE MC68HC11K4 MEMORY EXPANSION LOGIC

The memory expansion logic extends the addressing range of the 68HC11 CPU by providing two new on-chip blocks. The first new block implements additional address lines which are only made active when required by the CPU. The second block eases interfacing to external memory chips by providing chip select signals. Both of these blocks are fully user programmable.

ADDITIONAL ADDRESS CAPABILITY

If the addressing capability of the 68HC11 CPU is to be extended then the first step involved is to provide additional address lines. The CPU itself provides 16 lines (A0 to 15) which allow up to 64K Bytes of memory to be accessed. Each new address line provided will double that total.

To maintain compatibility with other members of the 68HC11 family the CPU used in the K4 is not functionally changed. The extra addressing capability is provided in such a way that the CPU itself is unable to distinguish more than 64K Bytes of memory. The extra capacity is provided by switching banks of the extra memory in and out of the 64K Byte range provided. To maximise the flexibility of this approach the size and number of the switched banks is user programmable.

To use extended memory, the programmer must first allocate a range (called a *window*) within the CPU 64K Byte addressing range which will be used when banks are switched in and out. The memory expansion logic allows windows of 8K, 16K or 32K Bytes to be defined and placed at programmable points in the CPU 64K Byte memory. At any time only one bank may be displayed in the defined window and therefore the CPU may only have access to the memory contents of one bank at a time. The bank which is displayed in the window is selected by the additional address lines provided by the memory expansion logic. The process of replacing the active bank with a new bank is called *bank-switching*.

A useful analogy is that of photographic transparencies and a projector. The viewer may have many transparencies but is only able to view one at a time in the projector. If the photographs are numbered then the viewer is able to select precisely which one to view without having to go through all of them. Here the memory banks are akin to the transparencies – many are available but only one is accessible at any time. The number on the transparency can be thought of here as an address. Any transparency which is not being displayed at any one time is still accessible but only when the current one is removed and it is put in its place.

To extend the addressing capability of the CPU, six address lines were added. These are termed XA13, XA14, XA15, XA16, XA17 and XA18. To allow flexibility in the size of the windows, the lower three address lines are only used when determined by the size of the window and replace the CPU's equivalent addresses.

To understand the need for the XA13 to XA15 addresses consider the case when an 8K Byte window is being used. Address lines XA16 to XA18 are only active when the CPU is accessing memory within the window and they provide an extra 8 (2^3) times the memory provided by the CPU within that window. If an 8K Byte window is used then a maximum of 8 x 8K Bytes is available. However, for the CPU to uniquely distinguish each location in the 8K Byte window it only requires to use addresses A0 to A12. Changes in address lines A13 to A15 take it outside of the window and are therefore never valid within the window except to identify the starting address of the window. The three new addresses XA13 to XA15 are provided so that a full 512K Byte (8 x 8 x 8K Byte) range is realised. For a 16K Byte window only addresses XA14 and XA15 can be used and for 32K Byte only XA15 is usable. Note that the size of the memory window to be used need not necessarily be defined at the hardware design stage since the additional addresses XA13 to XA15 can be programmed to carry the CPU A13 to A15 signals. The situation may be complicated by the need to use differently sized windows (see example 2).

The memory expansion logic actually allows the user to define two independent windows and so more than 1M Byte of memory is accessible.

The hardware required to implement such a large memory range is greatly simplified by the use of the memory expansion's chip select block.

CHIP SELECTS

The memory expansion chip selects are provided to help the user interface the K4 with external memories. Four are provided but only three are of direct importance to the memory expansion logic. These are the two General Purpose Chip Selects and the Program Chip Select. The fourth, I/O Chip Select, is used to simplify the addition of external peripheral chips.

The basic function of a chip select is to provide a logic signal to indicate that the CPU is accessing a certain area of memory. For example, the Program Chip Select is intended to be active in the range of memory where the main program exists. Other chip selects will be active when their respective memory areas are used.

The General Purpose Chip Selects are the most flexible of those provided and their function is closely linked to the memory expansion logic. They can be programmed to be active on an area either within the CPU 64K Byte memory or within either window's 512K Byte range. In both cases the size of memory selected is fully programmable from 2K Bytes to 512K Bytes.

The above paragraphs outline the method by which the memory expansion logic extends the addressing range of the CPU. A detailed description of the internal registers used to implement the new logic is now required. Finally a series of examples are considered.

MEMORY EXPANSION AND CHIP SELECT REGISTERS

This discussion describes the functionality of the internal registers relating to the memory expansion logic and chip selects. Further details on their addresses and specific bit operations may be found in the Technical Summary [1].

The following registers perform the memory expansion function.

The **MMWBR** register allows the starting address of each of the two windows within the CPU 64K Byte address range to be defined. The windows will normally start on a boundary related to their size, for example an 8K Byte window may start on any 8K Byte boundary starting at $0000, that is, $2000 $4000 ... $E000. A 16K Byte window can only start on 16K Byte boundaries, $0000 $4000 ... $C000. An exception is made for the 32K Byte window. This would normally start at either $0000 or $8000. However, the window is also allowed to start at $4000.

The **MMSIZ** register sets the size of the windows in use and selects whether the chip selects are active for only CPU addresses or for extended addresses.

Each General Purpose Chip Select has two registers called **GP1CSC, GP1CSA** and **GP2CSC** and **GP2CSA**.

The *control* register (**GP?CSC**) determines the logical output required when an area of memory is selected (with possible logic combinations with other chip selects) and the range of memory over which the chip select is to be active. Each chip select can be programmed to become active whenever the CPU address enters an memory expansion window (regardless of the actual bank selected); this is known as *following* a window.

The *address* register (**GP?CSA**) allows the starting address of the chip select to be programmed. The bits in this register which are active are determined by the size of the chip select range selected by the control register.

The program and I/O chip selects are programmable via the **CSCTL** register.

Two window registers, **MM1CR** and **MM2CR**, are used to indicate which bank is active in a window. Each contains the values of XA13 to XA18 to be output when the CPU selects addresses within the extended memory window. To change banks the user writes the address of the new bank into the appropriate window register.

The actual memory expansion address lines are multiplexed with PORT G I/O pins. Selecting an address line on one of these pins means that a PORT G pin is lost. For this reason the user need only select those address lines which are needed by the expansion logic. This allows unused lines to be used as general purpose I/O. The register which defines which extended address lines are used is **PGAR**. If an address line is not required then the appropriate bit in **PGAR** should be cleared to 0. A special case exists for two address lines which overlap the CPU address lines (XA13 and XA14). If XA13 or XA14 are selected as address lines in **PGAR**, but are not used in either window, then the appropriate CPU address line will be output on the port.

EXTENDED MEMORY EXAMPLES

The best way to grasp the implications of the K4 extended memory function is to consider some examples. Each example consists of two figures. Figures a are the logical arrangement of the memory and Figures b are a *possible* hardware configuration.

Example 1 shows one window in use. This is an 8K Byte window scheme and provides 8 banks from a single 64K Byte EPROM chip. Note that the logical address of each bank is derived from address lines A0 to A12 then XA13 to XA15. Chip select 1 is used.

Example 2 shows two windows in use. The first window is of 8K Bytes and is organised as in example 1. The second window is of 16K Bytes and is organised as 16 banks in two 128K Byte RAM chips. The logical addresses of the Window 2 banks are determined by A0 to A13 then XA14 to XA17 (XA18 determines start address). Chip select 2 is used for window 2.

Example 3 shows the same two windows as in example 2 except that the logical addressing of the windows are changed. Now Window 1's logical address is determined by A0 to A12 then XA15 to XA17 (XA13 and XA14 ignored) and Window 2's logical address is determined by A0 to A13 then XA15 to XA18 (XA14 ignored). Note that in both windows every bank will be duplicated due to the lack of decoding on certain address lines. In Window 1 each bank is duplicated four times. In Window 2 each bank is duplicated twice.

Example 4 shows the maximum 1Mbyte extended memory possibility in use. Window 1 is 16K Bytes starting at $0000 and Window 2 is 32K Bytes starting at $4000. Note that the internal RAM and registers of the K4 are echoed in every page of window 1, but that the internal EPROM in window 2 only occurs in the first 64K Bytes of extended memory. That is, for addresses below $10000, the internal EPROM is present in the memory map at the relevant address. This currently applies to all window sizes and configurations, however, the user should avoid referring to EPROM at any address beyond page 0 to ensure compatibility with any future upgrades.

Logical addresses of both windows are given by A0 to A13 then XA14 to XA18. Window 2 uses XA14 because it does not start on a 32K Byte boundary and so requires that the XA14 is inverted. Both chip selects are used and follow a window each.

Figure 1a. One 8K Byte window

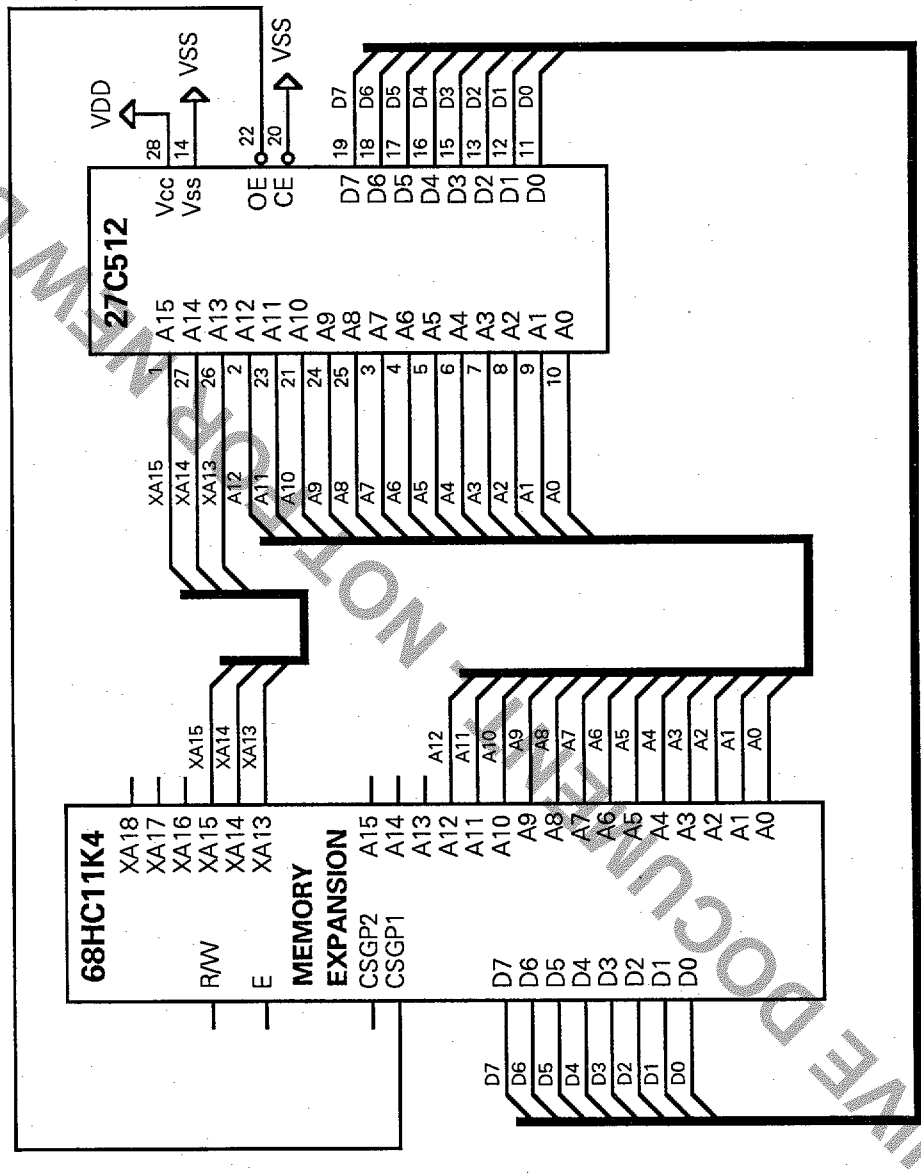

Figure 1b. 64K Bytes in one 8K Byte window

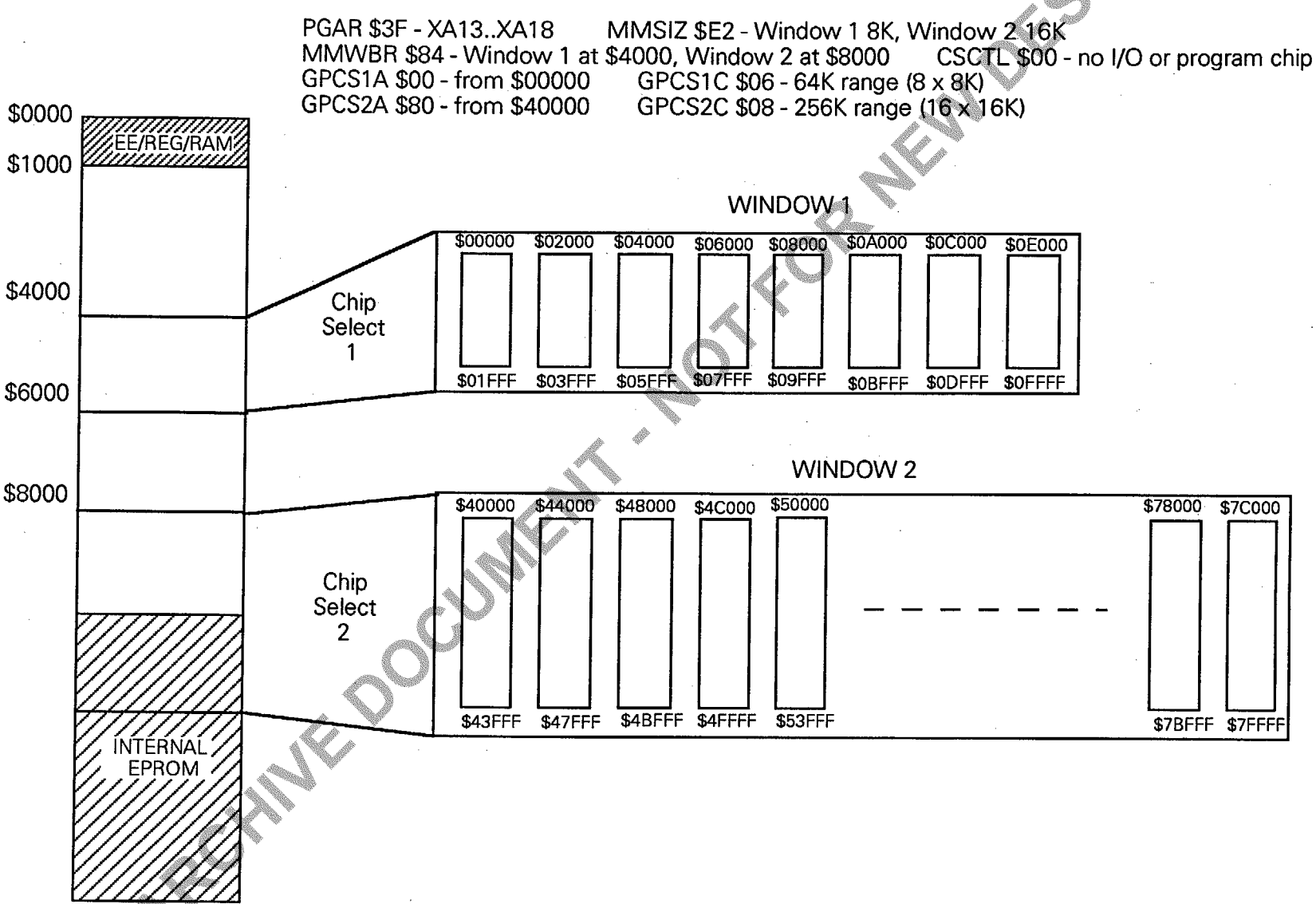

Figure 2a. One 8K Byte window and one 16K Byte window

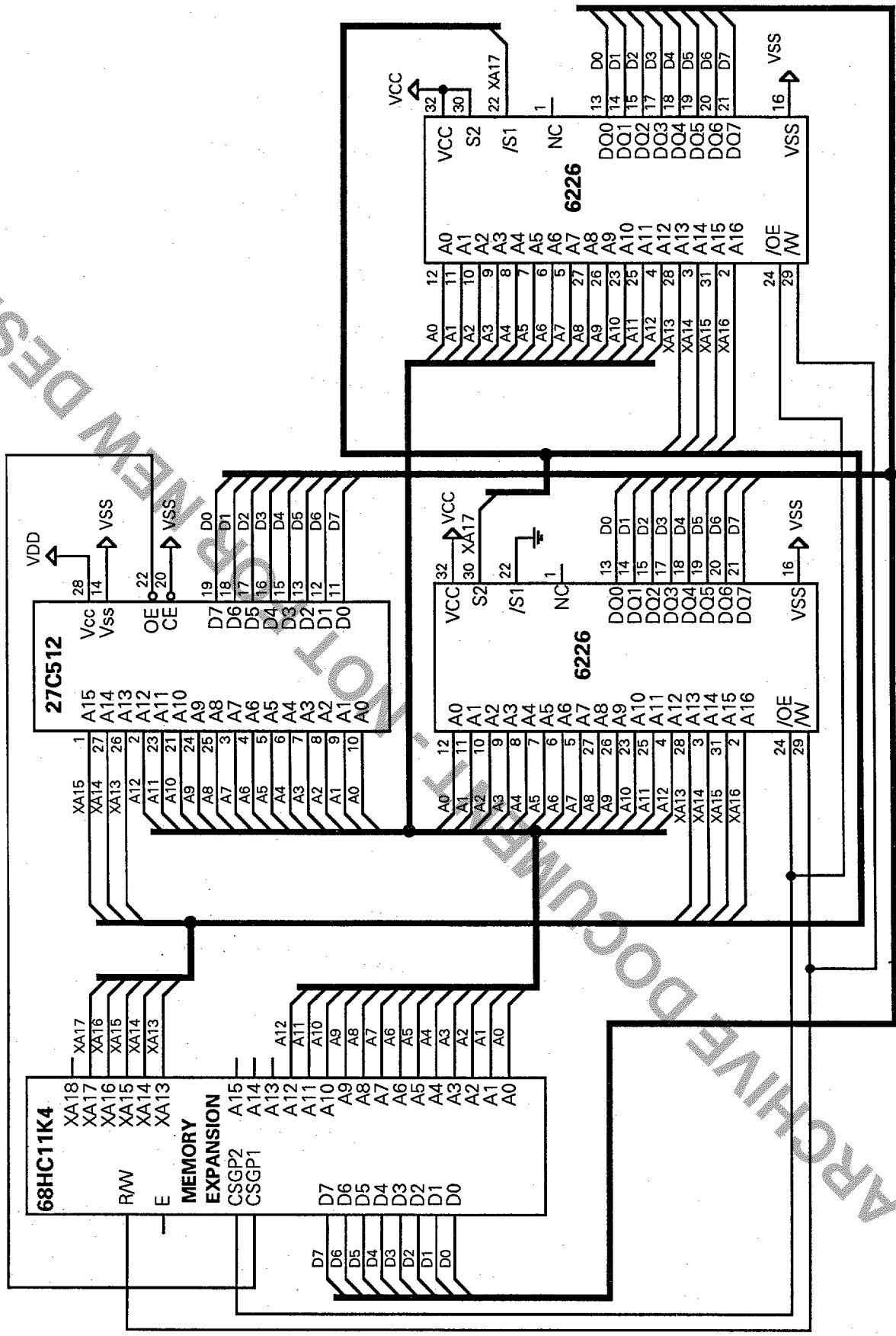

Figure 2b. One 8K Byte window and one 16K Byte window

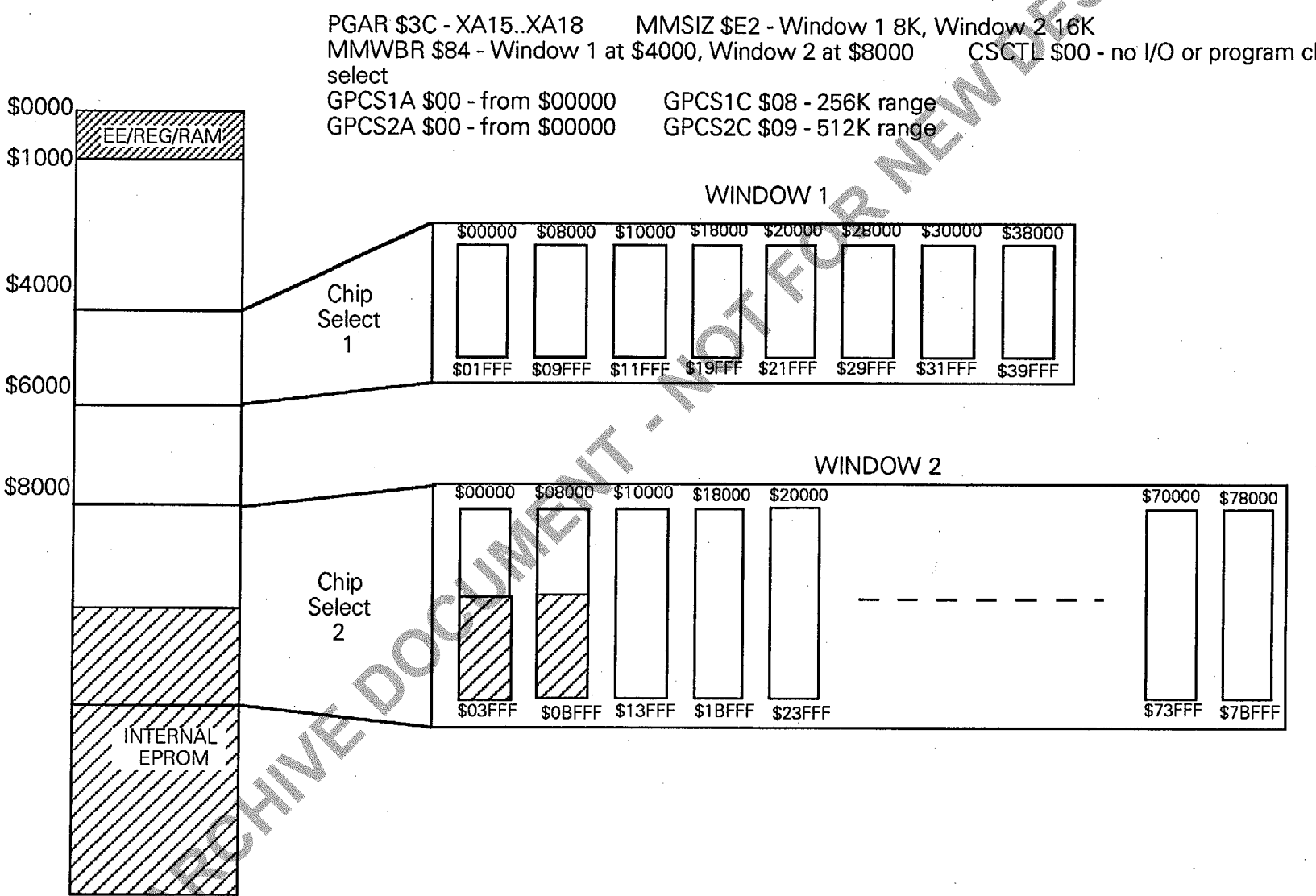

Figure 3a. 8K Byte window and 16K Byte window with new addressing

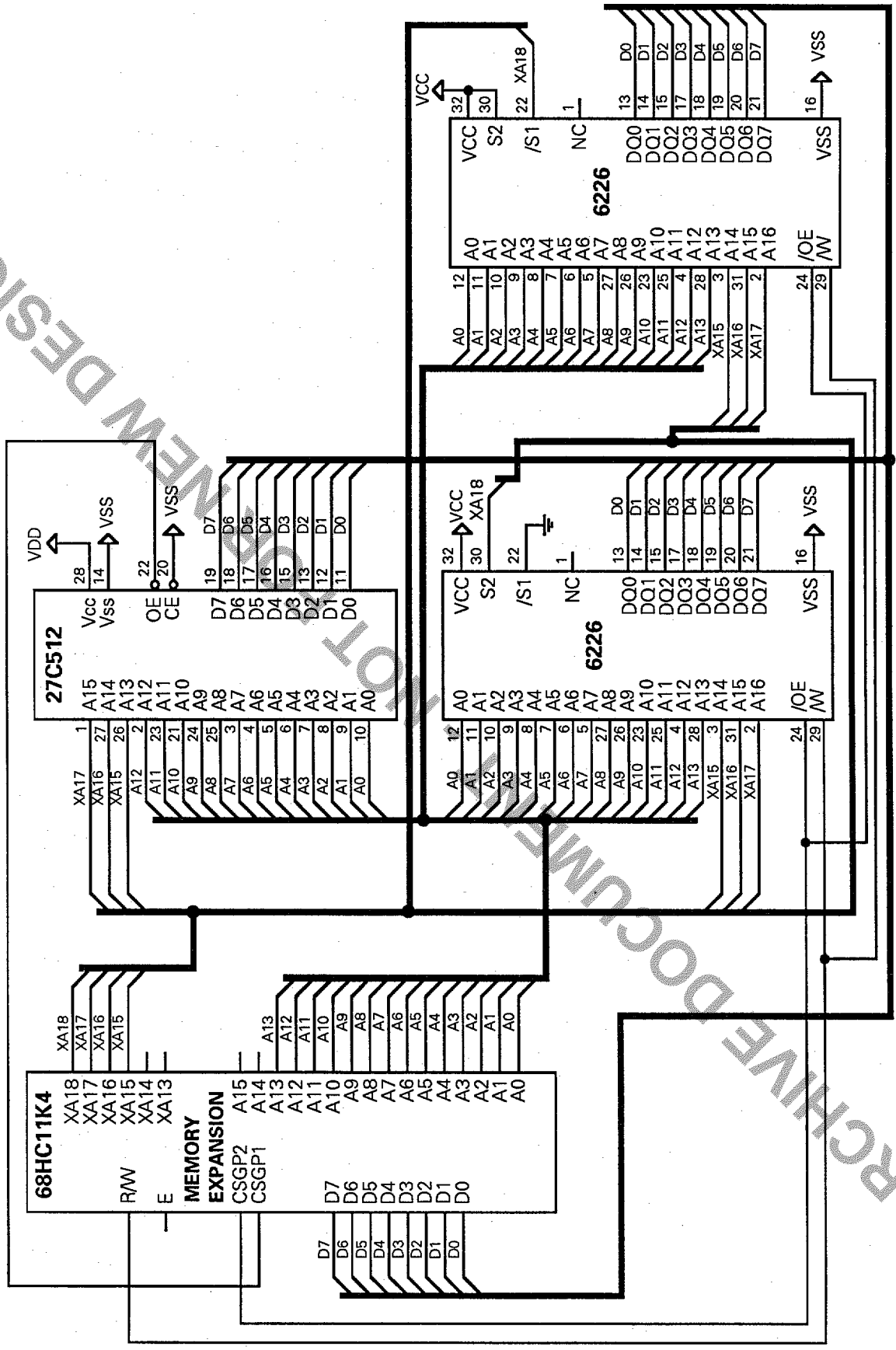

Figure 3b. Alternative 8K Byte window and 16K Byte window

PGAR $3E - XA14..XA18 MMSIZ $F2 - Window 1 16K, Window 2 32K
MMWBR $40 - one window at $0000, one at $4000 CSCTL $00 - no I/O or program chip select
GPCS1A $00 - not relevant GPCS1C $0A - follow window 1
GPCS2A $00 - not relevant GPCS2C $0B - follow window 2

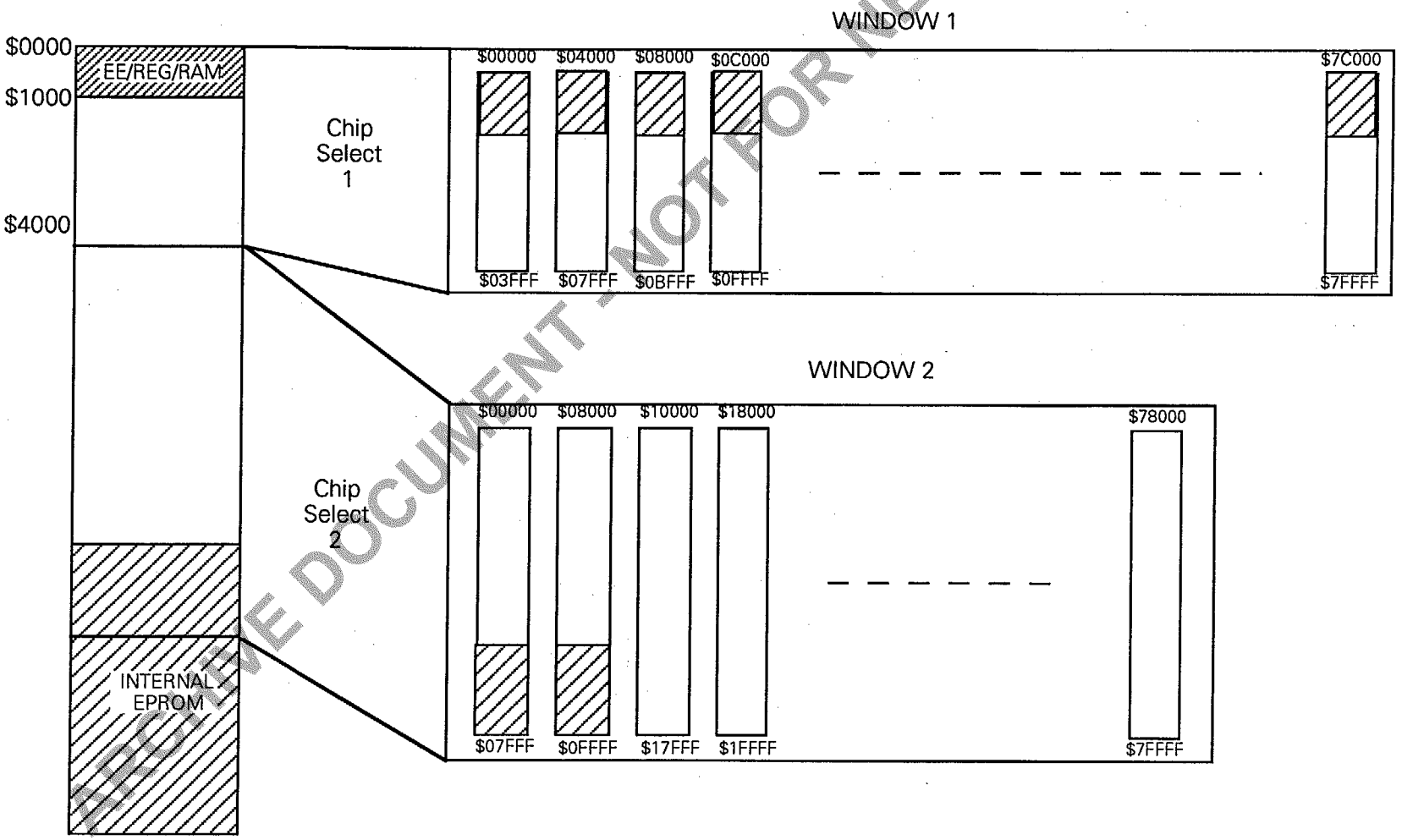

Figure 4a. Possible 1MByte addressing organisation

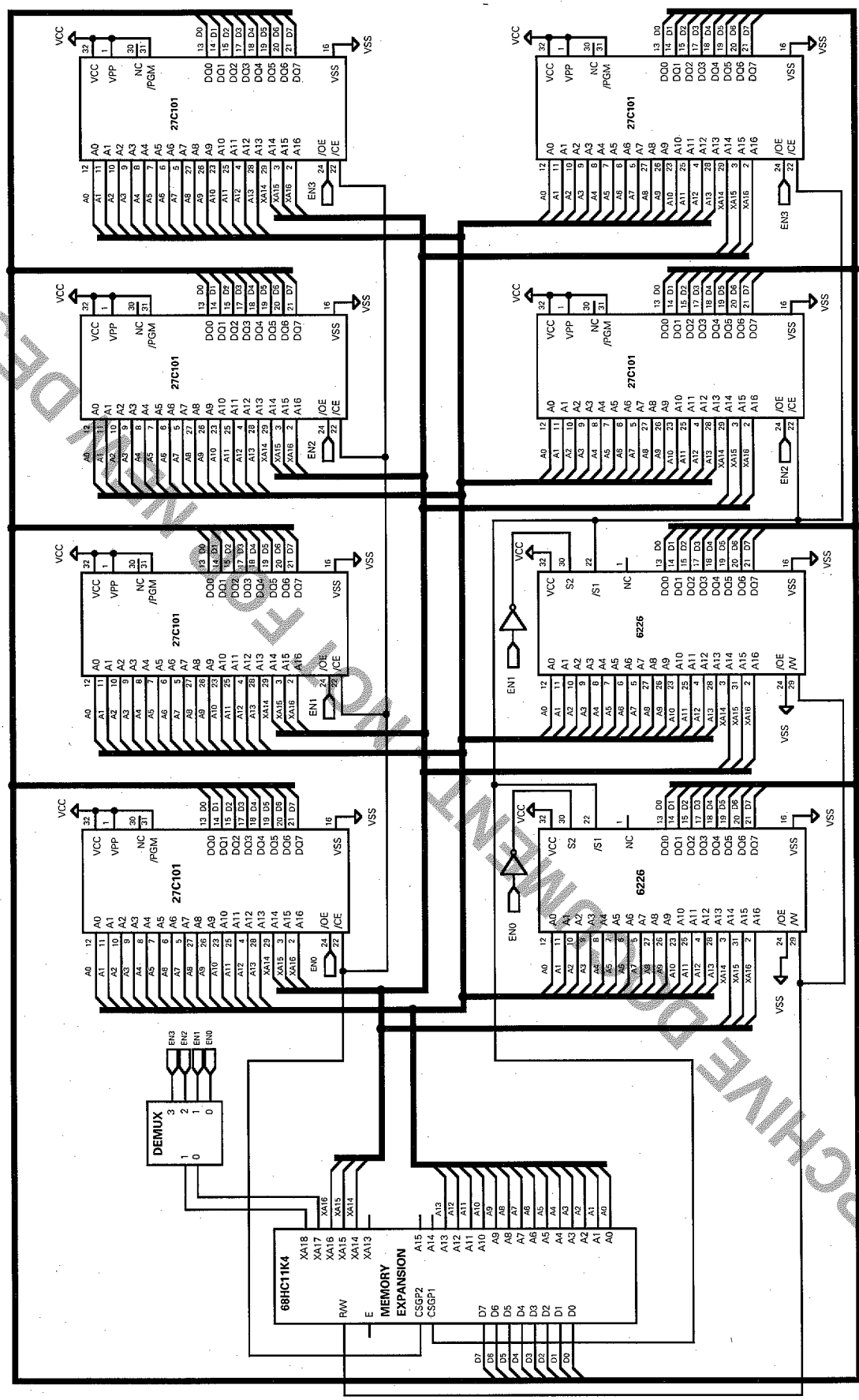

Figure 4b. One MByte in two windows

CONCLUSION

The standard 64K Byte addressing range of the M68HC11 family is easily extended using the MC68HC(7)11K4 memory expansion. This logic also provides programmable chip selects to allow easy interfacing with external memory chips. A similar extended memory system may be implemented on other M68HC11 products by following other systems as described in [2].

REFERENCES

[1] MC68HC711K4 Technical Summary, Reference BR751/D

[2] "128K byte addressing with the M68HC11", Reference AN432/D

Literature Distribution Centres:
USA: Motorola Literature Distribution; P.O. Box 20912; Phoenix, Arizona 85036.
EUROPE: Motorola Ltd.; European Literature Centre; 88 Tanners Drive, Blakelands, Milton Keynes, MK14 5BP, England.
ASIA PACIFIC: Motorola Semiconductors (H.K.) Ltd.; Silicon Harbour Center, No. 2, Dai King Street, Tai Po Industrial Estate, Tai Po, N.T., Hong Kong.
JAPAN: Nippon Motorola Ltd.; 4-32-1, Nishi-Gotanda, Shinagawa-ku, Tokyo 141, Japan.