1. DEFINITION OF TERMS

Bank One row of 8 64k dynamic RAMs on GM862 providing 64k bytes of memory.

Page-Mode. The 80-BUS method of extending the available system memory by selecting one of four `Pages´ of memory. Each Page has a maximum size of 64k.

Extended addressing.
The 80-BUS supports a memory address range of 512k, (19 address lines, A0-A18), but the Z80 can only address a maximum of 64k, (16 address lines, A0-A15). To support this feature the Gemini GM813 CPU card incorporates a `memory-mapping´ RAM on the four high address lines of the Z80´s address bus. This RAM `maps´ (translates) the A12-A15 address lines of the Z80 to the A12-A18 address lines of the 80-BUS. The Z80 can still only address 64k of memory at any one instant, but, by manipulating the contents of the mapping RAM, this 64k may lie anywhere within a physical address space of 512k. (See section 4.2 for a more detailed explanation.)

In this document address lines A16-A18 are referred to as the extended address, or EA.

(More recent Gemini CPU cards, such as the GM888 which contains an 8088 processor, can support this address range directly without requiring mapping-RAMs.)

Further descriptioas of Page-Mode and Extended addressing are given in section 4 of this manual.

Where references are made to a position on the board, (e.g. Upper left corner), it is assumed that the board is lying in front you with the component side uppermost, and the edge connector nearest to you. In this orientation the board type number (GM862) and IC numbering should be the correct way up.

2. GENERAL

The Gemini GM862 provides 256k of expansion read/write memory for 80-BUS based systems. The card supports the two standard 80-BUS memory expansion techniques of memory-mapped Extended Addressing and Page-Mode. In addition, when used in page-mode, GM862 can be configured so that only the lower 56k or 60k of memory is paged, leaving the top 8k or 4k of memory common across the four pages.

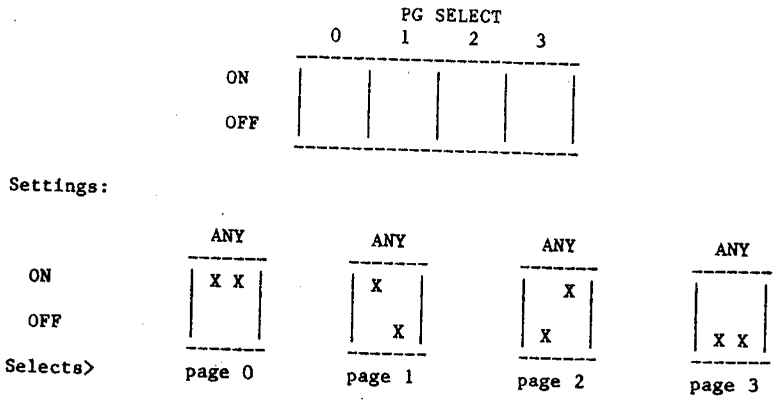
The configuration of the card is totally flexible. Each of the four 64k banks of memory on the card can be assigned to a particular Page and a particular Extended Address. These are defined by the settings of the on-board switches and the contents of a PROM. The standard PROM provides a reasonable sub-set of the possible permutations, and any esoteric requirements can be met by a specialised PROM. The board configuration is defined in the main by two sets of switches. The first set is concerned with the page-mode settings, and the second set with the extended addressing. The address of each 64k bank is defined by the logical AND of these two settings. (i.e. The selected Page must have been selected AND the extended address lines A16-A18 must hold the chosen address.)

3. SWITCH SETTINGS

The on-board DIL switches are divided into three groups which control various aspects of the board´s configuration. Each group is described separately below:

3.1. Page-Mode Settings

These are set by an eight-pole DIL switch located at the upper left of the printed circuit board. The switch is labelled PG SELECT (and the position is also designated IC1). The eight switches are divided into four pairs. Each pair of switches selects the active page for one of the banks of 64k RAMs.



3.2. Extended Address Settings:

These are controlled by switches 4-8 on the second DIL switch, (labelled IC2), which can be found adjacent to IC1. The switches are grouped into two fields, switches 4 & 5 selecting the `mode´ of operation, and switches 6-8 selecting a `base´ extended address. Note that the switch settings operate as negative logic. A `0´ is represented by an ON switch, and a `1´ by an OFF switch.

```
                 IC2
                | mode |   base   |
      ----------------------------------
      |    |    |    | 4  | 5  | 6  | 7  | 8  |
      ----------------------------------
```

3.2.1. Mode 0.

This mode selects Page-Mode only. The extended address lines A16-A18 are ignored. This <u>must</u> be used with CPU boards that do NOT support Extended Addressing. e.g. Nascom computers and Gemini GM811 boards.

```
                    | mode |   base   |
           ----------------------------------
    ON     |    |    |    | 4  | 5  |    |    |    |
                                   | 6  | 7  | 8  |
    OFF    |    |    |    |    |    |    |    |    |
           ----------------------------------
                                   don´t care
```

3.2.2. Mode 1

Here the four 64k banks of memory occupy a single EA. Switches 6-8 set the EA address. (This option assumes that the Page-Mode settings will be used to ensure that only one bank is actually enabled during a memory access.)

```
                         | mode |   base   |
           ----------------------------------
    ON     |    |    |    | 4  |    |    |    |    |      ON = `0´
                                   | 6  | 7  | 8  |
    OFF    |    |    |    |    | 5  |    |    |    |      OFF = `1´
           ----------------------------------
                                   A18  A17  A16    <- equivalent addresses
```

Mode 1 example:

```
                    | mode |   base   |
           ----------------------------------
    ON     |    |    |    | 4  |    | 6  | 7  |    |
                                                       (=001)
    OFF    |    |    |    |    | 5  |    |    | 8  |
           ----------------------------------
                         A18  A17  A16      <-  equivalent addresses
```

The switches above are shown set to a base address of 001, and the table below shows how the board responds to the 8 possible EA addresses on the BUS.

A18	A17	A16	
0	0	0	Nothing enabled
0	0	1	Banks 1-4 enabled (=base)
0	1	0	Nothing enabled
0	1	1	Nothing enabled
1	0	0	Nothing enabled
1	0	1	Nothing enabled
1	1	0	Nothing enabled
1	1	1	Nothing enabled

3.2.3. Mode 2

Here the four 64k banks of memory occupy four consecutive EAs. Switches 6-8 set the base address (the first EA address).

```
                 | mode |    base   |
        ------------------------------
ON      |  |  |  |  | 5|  |  |  |       ON = `0´
        |  |  |  |  |  | 6| 7| 8|
OFF     |  |  |  | 4|  |  |  |  |       OFF = `1´
        ------------------------------
                      A18 A17 A16 <-equivalent addresses
```

Mode 2 example:

```
              | mode |    base   |
     ------------------------------
ON   |  |  |  |  | 5| 6| 7|  |
                                      (=001)
OFF  |  |  |  | 4|  |  |  | 8|
     ------------------------------
                  A18 A17 A16
```

The switches above are shown set to a base address of 001, and the table below shows how the board responds to the 8 possible EA addresses on the BUS.

A18	A17	A16	
0	0	0	Nothing enabled
0	0	1	Bank 1 enabled (=base)
0	1	0	Bank 2 enabled (=base + 1)
0	1	1	Bank 3 enabled (=base + 2)
1	0	0	Bank 4 enabled (=base + 3)
1	0	1	Nothing enabled
1	1	0	Nothing enabled
1	1	1	Nothing enabled

Note If switches 6-8 are set to give a base extended address (EA) of 5 or higher, then the banks of memory do NOT `wrap-round´. i.e. if BASE is set to 7, then no bank is enabled for EAs of 0-6, and bank 1 is enabled for an EA of 7. Banks 2-4 are never enabled.

In this mode it would be normal to set all banks to the same page, so that all memory resides in one page. This is not, however, essential and each bank can, if desired, reside in any page.

3.2.4. Mode 3

This mode is reserved for `specials´. Currently one `special´, is defined.

```
          ?  ?  ? | mode | `Special´ |
     ------------------------------
ON   |  |  |  |  |  |  |  |  |
     |  |  |  |  |  | 6| 7| 8|
OFF  |  |  |  | 4| 5|  |  |  |
     ------------------------------
                      option no.
```

option	Switches			Action
0	ON	ON	ON	RAM disk with GM813.
1	ON	ON	OFF	Undefined.
2	ON	OFF	ON	Undefined.
3	ON	OFF	OFF	Undefined.
4	OFF	ON	ON	Undefined.
5	OFF	ON	OFF	Undefined.
6	OFF	OFF	ON	Undefined.
7	OFF	OFF	OFF	Undefined.

3.2.4.1. Option 0:

This option allows GM862 to be used in conjunction with GM813 as a RAM-disk using certain versions of Gemini CP/M software. This software supports either multiple GM833s (RAM-disk boards), or `paged´ system RAM. A GM862 set to mode 0 cannot be used in conjunction with GM813 to provide the paged RAM, as one of the banks of memory will conflict with the 64k of RAM present on GM813. (See your Gemini documentation on how to set up CP/M for a `Memory´ drive using the utility CONFIG.)

Option 0 prevents this conflict by only enabling three of the banks of memory when EA is 0. To enable users´ software to access the full 256k, all four banks are enabled if the EA=1.

The page-mode switches must also be set, and the correct settings are shown below:

```
          PG SELECT
       0     1     2     3                      IC2
     ------------------------        ------------------------------
ON   | 1  2| 3   |    6|    |        |  |  | 3|  |  | 6| 7| 8|
                                     | X|
OFF  |     |    4| 5   | 7 8|        |  | 2|  | 4| 5|  |  |  |
     ------------------------        ------------------------------
               IC1
```

3.3. Auxilliary Setting

This allows a common block of memory to exist within a Page-Mode environment. It may also be used in conjunction with extended addressing, but an understanding of how the common area is implemented is necessary in order to be able to predict the behaviour of the card.

On the GM862 an IC is used to monitor address lines A12-A15 from the 80-BUS. Whenever the address is found to be:

```
             A18...............A0
              XXX1111XXXXXXXXXXXX   (4k common area)
   or        XXX111XXXXXXXXXXXXX  (8k common area)   whereX=don't care
```

then the `write enable´ control lines from the Page-Mode control latch are forced ON. If Page-Mode only is being used, then this will result in any write to the top 4k (or 8k) of memory occuring in all four banks simultaneously. Thus the same information will be contained in the top 4k (or 8k) of all four banks, and any attempt to change the contents of this area of the currently selected Page will result in an identical change being made to the other three pages as well. If extended addressing is enabled, then a write will only occur in a bank of memory if the EA is valid as well. (Remember that a particular RAM bank is enabled by the logical AND of the Page-Mode control and the extended address.)

The `Common Area´ feature of GM862 is controlled by switches 2 & 3 of DIL switch IC2. Switch 1 is unused.

```
                   IC2
      ---------------------------------
ON    |   |   |   |   |   |   |   |   |
      |   | 2 | 3 |   |   |   |   |   |
OFF   |   |   |   |   |   |   |   |   |
      ---------------------------------
          4k  Dis
          8k  En
```

Switch 3 Enables or disables the `Common Area´ feature.
Switch 2 selects between the 4k or 8k options.

The possible combinations are shown below:

```
      ---------------------------------
ON    |   | 2 | 3 |   |   |   |   |   |    INVALID
OFF   |   |   |   |   |   |   |   |   |   DO NOT USE
      ---------------------------------

      ---------------------------------
ON    |   |   | 3 |   |   |   |   |   |
                                            No Common Area
OFF   |   | 2 |   |   |   |   |   |   |
      ---------------------------------

      ---------------------------------
ON    |   | 2 |   |   |   |   |   |   |
                                            4k Common Area
OFF   |   |   | 3 |   |   |   |   |   |
      ---------------------------------

      ---------------------------------
ON    |   |   |   |   |   |   |   |   |
                                            8k Common Area
OFF   |   | 2 | 3 |   |   |   |   |   |
      ---------------------------------
```

Note that, as shown above, the combination of switches 2 and 3 both being ON is invalid, and this combination must NOT be used, regardless of what other combination of switches is chosen.

This concludes the description of all of the switch options of the GM862 board. Note that, as described earlier, switch 1 of DIL switch IC2 is unused, and consequently may be left in either position.

4. SOFTWARE IMPLICATIONS

NOTES.

1 - On RESET the Page-Mode latch of GM862 is cleared. Thus the on-board memory of GM862 will be disabled until a Page is specifically selected by writing a control word to port 0FFh. Page 0 is always specifically selected by the intialisation code within the standard Gemini monitors for GM811 or GM813. (The standard CP/M auto-boot EPROM, or RP/M version 2.0 or greater.) Anyone replacing these EPROMs by their own custom version should ensure that the appropriate initialisation software is included.

2 - The `Common Area´ feature is unique to GM862. If GM862 is used in conjunction with any other RAM board (e.g. GM802, or GM813 CPU+RAM) then the top 4k or 8k of the non GM862 RAM will remain unique irrespective of the switch settings on GM862.

4.1. Page-Mode Memory expansion

With page-mode an entire 64K memory bank can be switched into or out of the memory map under software control. One particular I/O port, port 0FFH, is reserved to control this function. The bits of the port are divided into two halves, the upper four bits are used to write-enable a bank, and the lower four bits to read enable a bank. To simplify the amount of logic necessary on the memory boards to implement this feature the four-bit fields are used directly and are not decoded further. The functions of the bits are shown below:

Bit	Function if set
7	Write enable page 3
6	" " " 2
5	" " " 1
4	" " " 0
3	Read enable page 3
2	" " " 2
1	" " " 1
0	" " " 0

As there is no subsequent decoding of the data fields the onus lies on the programmer not to set up an invalid condition. For instance if the page-mode system is being used with all four pages in use, in general only one bank can be read enabled at any one time. If a value such as 1FH is written into port 0FFH then when the Z80 attempts to read data from memory all four banks will attempt to put a byte on the data bus resulting in conflict between their data bus buffers and garbled data into the CPU. However it is perfectly possible to write-enable all of the banks simultaneously (by writing say 0F1H into port 0FFH) so that the same information can be written into each bank. (In fact the GM862 `common area´ is implemented this way - see section 3.3.)

In use the page-mode system requires a little thought to utilise in an application as the entire memory of the processor is switched. Either a common area of memory which is not switched has to be used to effect the transfer, or the identical program has to be written into the all the memory banks so that when the switch occurs the control program continues to run correctly.

4.2. Extended addressing - Memory mapping

The Gemini GM813 CPU board supports an alternative method for extending the addressing capability of the Z80 - memory mapping. GM813 supplies 19 address lines to the 80-BUS rather than the 16 of the Z80. This means that the board can directly address 256k bytes of memory, although the Z80 can only "see" 64k of it at any one time. The extension in the addressing capability is achieved by putting mapping registers on the top four address lines. These registers are used to translate or "map" the top four address lines to seven address lines.

In GM813 the mapping is done by two 74LS189 TTL RAMs which are connected to give a 16 word x 8 bit register array. These RAMs, which are addressed by the top four address lines (A12-A15) of the Z80, supply the seven high address lines (A12-A19) to the 80-BUS from their data outputs. As any value may be written into the RAM registers it is possible to arrange for any 4k segment of the Z80´s 64k physical address space to access any 4k segment in the 256k address space of the 80-BUS.

On Reset the system monitor initialises each register of the 74LS189s with its own address, thus the 16 registers 0-F contain the data bytes 00-0F. This means that initially there is no difference between the Z80´s output address, and the address put out on the 80-BUS. However if we program say 2E into mapping register 0, then whenever the Z80 accesses an address in the range 0XXX the actual physical address put onto the bus will be 2EXXX.

This approach has a distinct advantage over page mode in that the system´s memory can be re-allocated in amounts of 4k at a time, not entire memory boards. In fact the mapping scheme can work quite successfully with just the standard 64k of memory. Consider a real-time task that has to handle say seven identical machines. Let us assume that the control program is contained in the bottom 12k of memory and requires additional workspace for each machine controlled. When switching tasks the program has to switch workspaces. Without memory mapping this would have to be done by either saving the current data and copying in the new, or by writing the program so that all data was accessed indirectly via one of the index registers which could then be changed to point to the new data area. However with memory mapping, the program can be written to use a fixed area of memory at say 3000H. The data areas for the tasks can then be allocated to 4000H, 5000H, 6000H, 7000H, and so on. Then to switch to task 3 it is only necessary to write a 6 into mapping register 3. Thereafter any memory reference in the range 3XXX will actually be mapped to a physical address of 6XXX. To switch to the next task it is only necessary to write a 7 into mapping register 3, and so on.

4.2.1. Writing to the mapping Registers

To simplify the hardware required to implement the memory-mapping feature of GM813, use is made of the special Z80 OUT instruction "OUT r,(C)", where the ouput port is indirectly addressed by register C. When this instruction is executed, the port address (register C) is placed on the low address lines (A0-A7) of the address bus, and register B appears on the high address lines (A8-A15). This means that the four most significant bits of register B will address the memory-mapping rams. If an OUT instruction is performed to port 0FEH then a write strobe will be generated for the 74LS189s to latch whatever is on the data bus into the memory-mapping register selected by A12-A15.

For example to write 2EH into memory mapping register 7 the following sequence of instructions should be executed:

```
        LD      A,2EH          ;Data to be written
        LD      B,70H          ;B high nibble = register address
        LD      C,0FEH         ;C = memory-mapping port
        OUT     (C),A          ;Write data into the selected register
```

5. HARDWARE IMPLICATIONS FOR NASCOMs

NOTE The GM862 RAM board does NOT provide three signals, /NASIO, /NASMEM and DBDR, that may be required by a Nascom 1 or Nascom 2 CPU board - see your Nascom manuals.

/NASIO is required due to the limited I/O decoding on the Nascom CPU boards. It may already be provided by other cards in the system (e.g. Nascom RAM B, Gemini IVC or Gemini FDC boards).

/NASMEM is required due to the limited memory decoding on the Nascom 1 and is used to determine which 4K block of memory the Nascom itself occupies.

DBDR is required by the Nascom 1 buffer board to control the data-bus buffers on that board.

For those with Nascom 1s it may be worthwhile to implement the additional circuitry on the buffer board, as this will allow other 80-BUS cards that do not provide the necessary signals to be used with the system without further modifications.

6. APPLICATIONS

The following sections give some possible configurations for GM862.

6.1. Use with GM811 or a Nascom 2

Neither GM811 nor the Nascom 2 support the 80-BUS extended addresses, and so any memory expansion has to be via the Page-Mode option*. This restricts the possible memory expansion to a single GM862 as only a maximum of 256k is supported by Page-Mode. Each bank of memory should be set to a different page, and the Extended Addressing option should be disabled. (See below.)

With both the Nascom and the GM811 any on-board memory will overlay the RAM memory of GM862 and will thus appear as common memory across all four pages. This may restrict the amount of RAM available to application programs. However, with GM811, the on-board memory can be totally disabled thus freeing any areas of memory that were being overlayed. For those applications that require a small amount of common memory (e.g. CP/M plus) the `common area' feature of GM862 can be enabled to provide this.

GM811 does offer the possibility of a total available memory size of greater than 256k as the mechanism for enabling/disabling the on-board byte-wide sockets is independent of the Page-Mode control port. Thus, for example, 32k of system firmware in the byte-wide sockets could access the full 256k of GM862 with suitable control software.

Sample settings:

Example 1: Page-Mode with no Common Area.

```
               PG SELECT
          0       1       2       3                            IC2
      ---------------------------------       ------------------------------------
ON    | 1  2 | 3      |      6 |      |       |    |    | 3  | 4  | 5  |    |    |    |
      |      |        |        |      |       | X  |    |    |    |    | X  | X  | X  |
OFF   |      |      4 | 5      | 7  8 |       |    | 2  |    |    |    |    |    |    |
      ---------------------------------       ------------------------------------
                     IC1
```

One application for example 1 above is to provide the Gemini/Nascom owner with (upto) 64k system RAM plus (upto) 192 RAM-DISK. Most Gemini CP/Ms for the Nascom and Multiboard support this. The Gemini supplied CP/M utility CONFIG should be used to set up the `M' drive, with the number of pages of RAM set to 3, and the size to 64K. (Note Nascom owners will be restricted to 60k unless they have modified their system so that the Nascom onboard memory is disabled).

*N.B. If the GM888 8088 CPU board is co-resident in the system then EA may be used in setting up the GM862. However the Z80 processor can only access additional memory in EA 0, and then only by Page-Mode. See the GM888 manual for further details.

Example 2a: Page-Mode-with 4k Common Area.

PG SELECT	0		1		2		3	
ON	1	2	3			6		
OFF				4	5		7	8

IC1

IC2

ON	X	2		4	5	X	X	X
OFF			3					

Example 2b: Page-Mode with 8k Common Area.

PG SELECT	0		1		2		3	
ON	1	2	3			6		
OFF				4	5		7	8

IC1

IC2

ON	X			4	5	X	X	X
OFF		2	3					

where `X´ represents a `don´t care´ state.

6.2. Use with GM813 (or similar)

For those CPU cards that support the Extended Addressing feature of the 80-BUS a total of 2Mbytes of memory (8 GM862s) can be supported by combining the EA method with Page-Mode. For those requiring a limited amount of expansion memory (<= 256k total) a choice can be made between using Extended Addressing or Page-Mode. For amounts greater than 512k some combination of the two methods will be required.

When a choice has to be made on which way to configure one or more memory cards the following points should be born in mind:

GM813 includes 64k of on-board RAM. This is located in Page 0, EA 0.

EA is far more flexible than Page-Mode, allowing memory to be re-allocated in units of 4k (rather than 64k).

A common area implemented via EA (by keping one or more mapping registers fixed) does not waste memory as all the remaining memory can be accessed by manipulating the mapping registers. However with page-mode a common area overlays memory. (i.e. a 4K page-mode common area will give a total available memory of 4K + 4*60K = 244K, where an EA approach offers 4K + 60K + 3*64K = 256K.)

Page-Mode, although cumbersome, provides the fastest possible context switch for large areas of memory (a single OUT instruction) rather than the small subroutine required by EA to modify the table in the memory-mapping RAM.

Example 3: GM813 + GM862. (EA Only.)
This uses only Extended Addressing, all the memory resides in Page 0. (N.B. See also `Mode 3, Option 0´.)

PG SELECT	0		1		2		3	
ON	1	2	3	4	5	6	7	8
OFF								

IC1

IC2

ON	X		3		5	6	7	
OFF		2		4				8

Example 4: GM813 + GM862. (Paging + EA.)
In time-critical environments paging may be a more attractive proposition. In view of the 64k already on GM813 (in Page 0, EA 0), GM862 should be located at another EA, (e.g. EA 1). In taking this approach the mapping RAM on GM813 will only require changing in order to switch between the 64k on GM813 (EA 0), and the 256k on GM862 (EA 1). (N.B. See also `Mode 3, Option 0´.)

PG SELECT	0		1		2		3	
ON	1	2	3			6		
OFF				4	5		7	8

IC1

IC2

ON	X		3		5	6	7	
OFF		2		4				8

Example 5: GM813 + 2 x GM862.
Here there is a total of 64k(GM813) + 512k(2 x GM862) available. There are various ways that the 64k memory banks can be allocated, but a straight forward way is to place the two GM862 cards in the second Page. On Reset the system will start up in the GM813 on-board memory located in Page 0, EA 0. System software can then switch Pages to Page 1, where the two GM862s occupy the full EA address range of 512K. This approach uses EA (within Page 1) as the main memory expansion mechanism, and the 64k of GM813 remains available through a Page-switch to Page 0.

PG SELECT	0		1		2		3	
ON	1		3		5		7	
OFF		2		4		6		8

IC1

IC2

ON	X		3		5	6	7	8
OFF		2		4				

Board 1. Page 1, EA 0-3

```
          PG SELECT
      0     1     2     3                            IC2
    -------------------------          ---------------------------------
ON  | 1   | 3   | 5   | 7   |          |   |   | 3 |   | 5 |   | 7 | 8 |
    |     |     |     |     |          | X |   |   |   |   |   |   |   |
OFF |   2 |   4 |   6 |   8 |          |   | 2 |   | 4 |   | 6 |   |   |
    -------------------------          ---------------------------------
             IC1
```

Board 2. Page 1, EA 4-7

Example 6:

An alternative approach is to setup both GM862s for Page Mode, and to place one at EA 1, and the other at EA 2. This approach may offer the possibility of a faster `memory switch´ than the above example, a factor that might be important for time-critical applications.

```
          PG SELECT
      0     1  .  2     3                            IC2
    -------------------------          ---------------------------------
ON  | 1 2 | 3   |   6 |     |          |   |   | 3 | 4 |   | 6 | 7 |   |
    |     |     |     |     |          | X |   |   |   |   |   |   |   |
OFF |     |   4 | 5   | 7 8 |          |   | 2 |   |   | 5 |   |   | 8 |
    -------------------------          ---------------------------------
             IC1
```

Board 1. Pages 0-3, EA 1

```
          PG SELECT
      0     1     2     3                            IC2
    -------------------------          ---------------------------------
ON  | 1 2 | 3   |   6 |     |          |   |   | 3 | 4 |   | 6 |   | 8 |
    |     |     |     |     |          | X |   |   |   |   |   |   |   |
OFF |     |   4 | 5   | 7 8 |          |   | 2 |   |   | 5 |   | 7 |   |
    -------------------------          ---------------------------------
             IC1
```

Board 2. Pages 0-3, EA 2