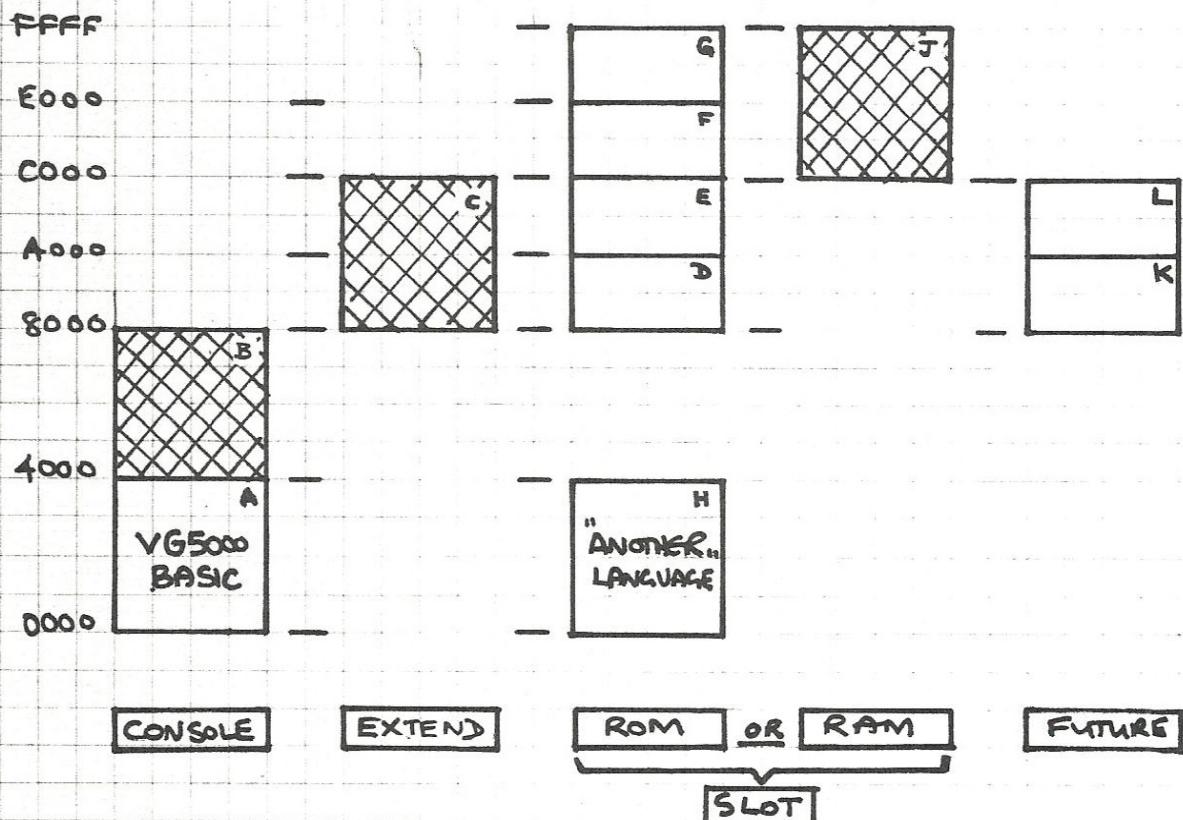


EXTERNAL EPROM / ROMA) MEMORY MAP

1. The various memory combinations are:



= ROM

= RAM

2. Memory controls are as follows:

MEMORY BLOCK

A: VG5000 BASIC
B: 16K RAM
C: EXT. 16K RAM

D:
E:
F:
G:
H: 16K "LANGUAGE"

J: 16K RAM PACK

K: } FUTURE EXP.
L: } ROMS

SELECT

CSROM1 - OPEN
ALWAYS SELECTED
BUSDIR - OPEN
(PIN ON EXT. SLOT)
STATIC OR DYNAMIC
- SEE NOTES
USUALLY ALWAYS
SELECTED
MAKE CSROM1 LOW
- SEE NOTES
ALWAYS SELECTED

MAKE BYSDIR ON
EXT. SLOT LOW

ENABLE

CEO
N/A
CE1/CE2
(EQUIV.)
CE1
CE2
CE3
CE4

CEO
CE3/CE4
(EQUIV.)
CE1/CE2
(OR EQUIV.)

3. Bank switching:

Memory block A (VGS000 BASIC) is automatically available at power-on, unless Memory block H (16K "LANGUAGE") is fitted. In this latter case, block H will get control, and block A is disabled. This alternation of control is automatic at power-on, and can be static (block A disabled / block H enabled) or dynamic, if the cartridge contains a flip-flop to control line CSROM1 and the block H chip-select line. Such a flip-flop would be driven from I/O port 01.

Memory block B (16K RAM) is always present. Memory block C (EXTENSION 16K RAM) is automatically available at power-on, unless:

- a ROM pack is fitted mapped to the same block (ROM blocks D and E) that pulls BUSDIR on the extension slot low)
 - a slot expander is in use and other software in blocks K and L need to get control
- This alternation of control is automatic at power-on and can be static (block C disabled / blocks D and E enabled) or dynamic, as for blocks K and L. The dynamic control is driven from I/O port 01. An OUT instruction to port 01 of 1xxxxxxx will disable block C; an OUT to port 01 of 0xxx xxxx will enable block C, provided that BUSDIR on the extension slot is simultaneously raised. This would be done by a flip-flop on the cartridge or expander also decoded to port 01.

Memory blocks F and G are the usual 8-16K games cartridge ROM, controlled by CE3 and CE4. If the 16K RAM cartridge J is fitted, games cartridges cannot be used in blocks F and G. The 48K RAM would presuppose software from another resource (e.g.: cassette).

B) EPROM / ROM FORMAT

1. Unlike MSX, there is only one format for all ROM content types. The treatment of the contents is left to the programmer to determine from his initialisation routine.
2. To recognise a ROM, the first byte (low memory address of 0) must be hex 41 (ASCII "A") and the second byte (low memory address of 1) must be hex 42 (ASCII "B"). These bytes can be mapped so as to appear at locations:

8000 + 8001
A000 + A001
C000 + C001
E000 + E001

3. The BASIC program searches from 8000 to E001, in that order. Thus these addresses are prioritised. When an "AB" is found, a CALL is made to the address given at the 16 bits following the "AB"; i.e.: addresses are taken from:

8002 + 8003
A002 + A003
C002 + C003
E002 + E003

This address would usually be an address within the cartridge memory space.

4. Once a cartridge has control it can do one of the following:

- POP-off the RET address and go into machine code
- Change hook vectors (to point them into the ROM) then RET to reenter BASIC
- Alter BASIC pointers, etc. and automatically run a tokenised BASIC text from ROM

Note that if a RET is issued, BASIC signs-on - there is no further searching for ROMs. One "hit" satisfies the initialisation routine.

See other documentation for hook vector details.
To run tokenised BASIC text, the first byte must be hex 00, and it must end with 3 bytes of hex 00.
The initialisation routine should do the following:

- point (txttab) to the character after the initial hex 00
- point (ft+rbleg) to the first character of a command stream - 1
- reset (getflg) to value of hex FF to execute the command stream
- do anything else required
- issue a RET

BASIC will now sign on, but run automatically.
(txttab) is at 488E/488F
(ft+rbleg) is at 4828/4829
(getflg) is at 4870

The command stream contains BASIC immediate commands in ASCII, with each command ended with hex '13'. To terminate the stream, put hex '03' after the last hex '13'. Note that the stream should be closed before any cassette I/O is used.

E.g.: RUN 13?Bye! 13 3 is a command stream, that will run the text then prints "Bye!" on the screen.

If input is required during a command stream, the (getflg) should be altered to hex 00 beforehand, and reset to hex FF afterwards.