2.4 — Cursor programming

The cursor position is defined by two counters :

- The X counter (0:5) points to a column, from 00 to 39.
- The Y counter (0:4) points to a row, from 00 to 23. The Y counter value for the service row is 31.

Whenever the cursor is incremented, the X counter is incremented. The Y counter is automatically incremented when the X counter overflows from 39 to 00. The Y counter overflows from 23 or 31 to 00.

Four commands allow the cursor programming :

BEGIN ROW : The cursor is set at the beginning of the row whose value is given by the CRA register contents. The binary value to be loaded into CRB for this command is : 000X XXXX (X = don't care).

LOAD Y : The CRA contents is loaded into the Y counter. The binary value to be loaded into CRB for this command is : 001X XXXX.

LOAD X : The CRA contents is loaded into the X counter. The binary value to be loaded into CRB for this command is : 010X XXXX.

INC C : The cursor is incremented. The binary value to be loaded into CRB for this command is : 011X XXXX.

NOTA : In addition to these four commands, the cursor position may be automatically incremented when data is read or written into the page memory (see M register programming).

2.5 — R register programming

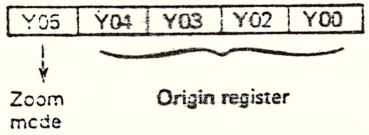
This 8 bit register of VIN defines the display unit operating mode (see EF9340 and EF9341 specifications). The "LOAD R" command loads the R register with the CRA contents when the binary value 101X XXXX is written into the CRB register.

2.6 — Y0 register programming

This 6 bit register of VIN allows roll-up, roll-down and zoom programming. The "LOAD Y0" command loads the Y0 register with the CRA contents when the binary value 110X XXXX is written into CRB.

ORIGIN REGISTER : The 5 LSB of Y0 define the origin register, i.e. the first row to be displayed on the top of the screen, after the service row. Therefore, roll-up or roll-down is made by incrementing or decrementing the origin register.

ZOOM MODE : When bit 5 of the Y0 register is set, the first 12 rows from the one defined by the origin register Y0 (0:4) are displayed in double height. Characters defined with double height attribute are displayed in quadruple height. Since programming the zoom mode is independent of the origin register, roll-up or roll-down operation may be performed in the zoom mode as well as in the normal mode.



NOTA : The service row is never affected by Y0 programming and depends only on bit 3 of the R register (service row on/off).

```
PAGE  001  APPLI   .SA:0

00001                       OPT    LLE=110
00002                 *
00003                 ************************************************
00004                 *  THIS PROGRAMMING EXAMPLE SHOWS TYPICAL DATA
00005                 *  AND COMMAND TRANSFERS BETWEEN THE DISPLAY
00006                 *  UNIT AND THE MICROPROCESSOR.
00007                 ************************************************
00008                 *
00009                 *
00010                 * REGISTER ADDRESS DEFINITION
00011                 *
00012          EC00   A TRA    EQU    $EC00    DATA TRANSFER REGISTERS.
00013          EC01   A TRB    EQU    $EC01
00014          EC02   A CRA    EQU    $EC02    COMMAND TRANSFER REGISTERS.
00015          EC03   A CRB    EQU    $EC03
00016                 *
00017                 * CONSTANT DEFINITION
00018                 *
00019                 *
00020          0020   A LDY    EQU    $20      "LOAD Y" COMMAND
00021          0040   A LDX    EQU    $40      "LOAD X" COMMAND
00022          0000   A BEGROW EQU    $00      "BEGIN ROW" COMMAND
00023          0060   A INCCUR EQU    $60      "INC C" COMMAND
00024          0080   A LDM    EQU    $80      "LOAD M" COMMAND
00025          00A0   A LDR    EQU    $A0      "LOAD R" COMMAND
00026          00C0   A LDY0   EQU    $C0      "LOAD Y0" COMMAND
00027                 *
00028A 4000                    ORG    $4000
00029                 *
00030                 * CHARACTER STRING DEFINITION.
00031                 *
00032A 4000    45     A MESS1  FCC    /EF9340 AND EF9341/
00033          4011   A ENDM1  EQU    *
00034A 4011    41     A MESS2  FCC    /APPLICATION/
00035          401C   A ENDM2  EQU    *
00036                 *
00037                 * EXTENDED CHARACTER DEFINITION
00038                 *
00039A 401C    20     A CAR1   FCB    $20,$38,$3C,$3E,$3F,$3F,$1F,$1F,$0F,$0F
00040A 4026    04     A CAR2   FCB    $04,$1C,$3C,$7C,$FC,$FC,$F8,$F8,$F0,$F0
00041A 4030    07     A CAR3   FCB    $07,$C7,$E3,$F3,$F9,$FC,$FC,$F8,$E0,$80
00042A 403A    E0     A CAR4   FCB    $E0,$E3,$C7,$CF,$9F,$3F,$3F,$1F,$07,$01
00043
00044          5000   A STACK  EQU    $5000    STACK INITIALIZATION.
00045                 *
00046                 *  MEMORY RESERVATION
00047                 *
00048A 4044    0002   A SAV    RMB    2
00049A 4046    0050   A IBUF   RMB    80
```

TABLE 3 – CHARACTER CODES

WINDOW CODE IN PAGE MEMORY																COMMENTS			
Type and character code field									Attribute field							Type	Implicite Attribute	Ad - hoc Serial Attribute	Remark
B7	B6	B5	B4	B3	B2	B1	B0	A7	A6	A5	A4	A3	A2	A1	A0				
0	X	X	X	X	X	X	X	0	N	L	H	S	B_1	G_1	R_1	α_0 (128)		Underlined, C_0	in GEN
1	0	0	–	–	s	↑	m	0	B_0	G_0	R_0	–	B_1	G_1	R_1	DEL			
1	0	1																	EXTENSION
1	1	0	X	X	X	X	X	0	N	L	H	S	B_1	G_1	R_1	α_1 (96)		C_0	
1	1	1																	
0	0	X	X	X	X	X	X	1	B_0	G_0	R_0	S	B_1	G_1	R_1	γ_S (64)	Normal size, positive		in GEN
0	1	X	X	X	X	X	X	1	B_0	G_0	R_0	S	B_1	G_1	R_1	γ_M (64)			
1	0	0	–	–	–	–	–	1	–	–	–	–	–	–	–	ILLEGAL			
1	0	1																	EXTENSION
1	1	0	X	X	X	X	X	1	B_0	G_0	R_0	S	B_1	G_1	R_1	γ_1 (96)	Normal size		
1	1	1																	

Note : Extension for α 1 and γ 1 may be mapped in only one 1 K x 8 RAM or ROM.

GLOSSARY :

α	:	Alphanumeric	C_0 (R_0, G_0, B_0)	:	Background colour
γ	:	Semi-graphic	C_1 (R_1, G_1, B_1)	:	Foreground colour
γS	:	Separated semi-graphic	H	:	Double height
γM	:	Mosaic semi-graphic	L	:	Double width
DEL	:	Delimitor	N	:	Reverse video (negative)
m	:	Blanking	S	:	Stable (non blinking)
i	:	Boxing	X	:	Character code
s	:	Underlining	–	:	Don't care

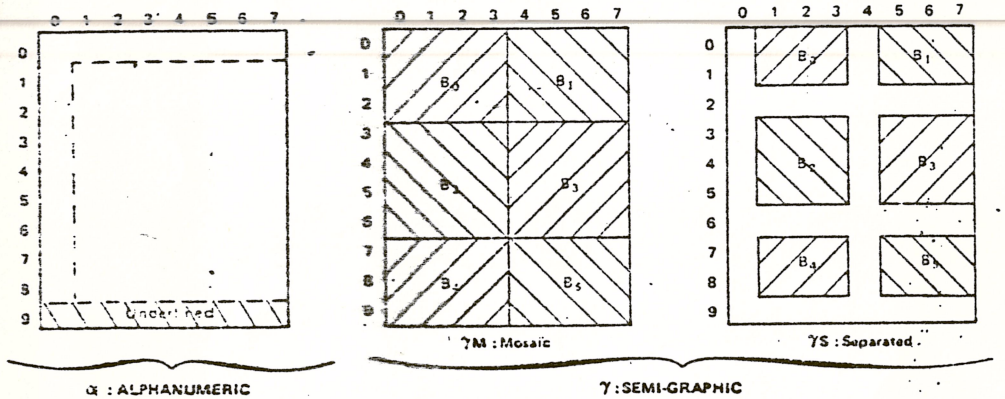


TABLE 1 – COMMANDS

COMMAND CODE																NAME	OPER
B7	B6	B5	B4	B3	B2	B1	B0	A7	A6	A5	A4	A3	A2	A1	A0		
0	0	0									K (0 : 4)					Begin Row	X (0 : 5) ⟶ 0 Y (0 : 4) ⟵ K
0	0	1									K (0 : 4)					Load Y	Y (0 : 4) ⟶ K
0	1	0									K (0 : 5)					Load X	X (0 : 5) ⟵ K
0	1	1														INC C	C ⟵ C
1	0	0							K (0 . 7)							Load M	M (0 : 7) ⟵ K
1	0	1							K (0 : 7)							Load R	R (0 : 7) ⟵ K
1	1	0									K (0 : 5)					Load Y0	Y0 (0 : 5) ⟵ K
1	1	1														ILLEGAL	

TABLE 2 – ACCESS MODES : M REGISTER CONTENT

ACCESS MODE REG.								ACCESS MODE	SUBSEQUENT DATA
M7	M6	M5	M4	M3	M2	M1	M0		
0	0	0						Write	MP (C) ⟵ T ; C ⟵ C+
0	0	1						Read	T ⟵ MP (C) ; C ⟵ C+
0	1	0						Write without INC	MP (C) ⟵ T
0	1	1						Read without INC	T ⟵ MP (C)
1	0	0			N T			Write slice	GC (MP(C), NT) ⟵ T ; NT+
1	0	1			N T			Read slice	T ⟵ GC (MP(C), NT) ; NT+

NT : Slice Number T : Mail Box C : Cursor MP : Page Memory

2.7 — M register programming

This 8 bit register of VIN defines a data transfer mode on the internal data bus. The "LOAD M" command loads the M register with CRA contents when the binary value 100X XXXX is written into CRB. There are two main data transfer modes within the display unit :

— read or write into the page memory
— read or write into a character generator.

When acceding to a character generator, the 4 LSB of the M register define a slice number.
Table 2 summarizes the data transfer modes defined by the M register.

PAGE MEMORY READ

In the page memory read mode, a two byte character code addressed by the cursor is loaded into the TRA and TRB data transfer registers.
Then the cursor position is incremented or not, according to the M register contents :
— M = 001X XXXX for page memory read with cursor incrementation
— M = 011X XXXX for page memory read without cursor incrementation.

PAGE MEMORY WRITE

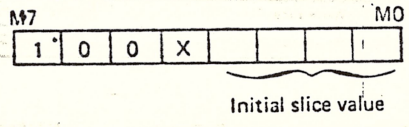
The TRA and TRB register contents is loaded into the page memory at the location addressed by the cursor. Then the cursor is incremented or not, according to the M register contents :
— M = 000X XXXX for page memory write with cursor incrementation.
— M = 010X XXXX for page memory write without cursor incrementation.

WRITING INTO AN EXTENDED CHARACTER GENERATOR

This transfer mode allows the user to load a RAM memory used as an extended character generator. The address of the data to be written is given by :

- NT : a character slice number.
- MP(C) : a character code in page memory pointed by the cursor, which should be an extended character code : B7 = 1 and B5 - B6 = 1. (see table 3).

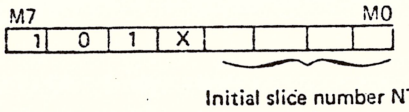
The initial slice number is given by the 4 LSB of the M register :



After the M register is loaded with the initial slice number value, whenever TRA then TRB is written by the microprocessor, the TRA register contents is loaded into the extended character generator location addressed by (MP(C), NT). Then NT is automatically incremented modulo 10.

READING A CHARACTER GENERATOR

This data transfer mode is used to read slices from a character generator. The value to be loaded by a command transfer into the M register to set this mode is :



As in the previous mode, the slice address is given by (MP(C), NT) and the initial slice number is given by the 4 LSB of the M register. The slice number NT is automatically incremented modulo 10 after a read operation.

REMARKS :

- Programming the M register by a "LOAD M" command sets only a data transfer mode within the display unit. Subsequent data transfers are not performed until access either to TRA then TRB, or to TRB. Specially, when programming the M register in a read mode from the page memory or a character generator, a first access to TRB is necessary to load GEN data transfer registers with the correct data to be read.
- When reading or writing into a character generator, a page memory word must be initialized with a character code which is used to address the character generator. This initialization may be done by writing into page memory without cursor incrementation.