OS-9/68K
CC143 PROGRAMMERS MANUAL

VERSION 1.0 MAY 1991

Documentation history

date	version	change / description
91/05/07	1.0	first release

Copyright

Disclaimer

The information in this document has been carefully checked and is believed to be entirely reliable. However, no responsibility is assumed for inaccuracies. Compcontrol B.V. makes no representations or warranties with respect to the contents hereof and specifically disclaims any implied warranties of merchantability or fitness for any particular purpose. Furthermore, Compcontrol B.V. reserves the right to make changes to any product herein to improve reliability, function or design, without obligation of Compcontrol B.V. to notify any person of such revision or changes. Compcontrol B.V. does not assume any liability arising out of applications or use of any product or circuit described herein; neither does it convey any license under its patent rights nor the rights of others.

CC143 PROGRAMMER MANUAL

TABLE OF CONTENTS

CHAPTER 1

General Information

1.1 Introduction

This manual describes how programmers can use the CC143 graphic board. It is focussed on how to program the INMOS G300 graphic controller. Part of this manual is an example program.

CHAPTER 2

Programming the G300B on the CC143

This chapter describes how to initialize the G300 and how to access the CLUT in byte mode.

2.1 Initialize the G300B

The initialization method for the G300B in byte mode is a bit different than specified in the G300 data sheet. This 'byte' mode is necessary if the host CPU cannot access the CC143 in 32-bit mode.
At startup the following initialization must be performed:

- Reset the G300.
- Write (the PLL multiplication factor + 0x20) to the Boot Location at address x683.
- Read a byte from the Control Register at address x583.
- Write the first byte to the Control Register (bits 7-0) at address x583.
- Read a byte from the Control Register at address x583
  After writing the timing parameter registers the VTG can be started by writing a '1' to bit 1 of the Control Register.

2.2 Accessing CLUT in byte mode

If The host CPU cannot access the CC143 in 32-bit mode, the CLUT has to be accessed in byte mode. The G300B Data Sheet states that a byte-access sequence must be aborted and restarted when the G300 performs a Transfer (DMA) cycle before the access sequence has been completed. There is no method to detect if the G300 has made a Transfer cycle, other than performing a read sequence to check the written value. However this read sequence can also be interrupted by a G300 transfer cycle which gives unreliable data. A method that may be used for accessing the CLUT is to wait for the FrameInactive signal to become True and then perform the three consecutive access cycles. During the first part of FrameInactive, the G300 has no need to start a Transfer cycle.

CHAPTER 3

Calculating G300 video parameters

This chapter describes how the video parameters for the G300B can be calculated. It is intended as an addendum for the G300 Data Sheet

remark:
    All calculated parameters for the G300 VTG should be whole numbers.

pixel_frequency = horizontal resolution / horizontal display time
calculate the crystal frequency and the multiplication factor from:
    - pixel_frequency = crystal_frequency * multiplication factor
    - 5 <= multiplication factor <= 15
    - 5 <= crystal frequency <= 9
screenunit = 4 / pixel_frequency
g_linetime = linetime / screenunit (nearest even number)
g_halfsync = horizontal sync / screenunit / 2
g_backporch = backporch / screenunit
g_display   = horizontal resolution / 4

remark:
    g_frontporch is used for calculation purposes only

g_frontporch = g_linetime - 2*g_halfsync - g_backporch - g_display

halfline point rule (condition should be true):
    0 < g_linetime - 2*g_halfsync - g_backporch < g_display

g_shdisplay = g_linetime/2
              - 2*g_halfsync - g_backporch - g_frontporch
g_broadpuls = g_linetime/2 - g_frontporch
g_vdisplay = 2*vertical resolution
g_vsync   = 2*vertical sync / screenunit
g_vblank   = 2*(vertical blanked/linetime) - 3*g_vsync
(vert. blanked = vert. sync + vert. frontporch + vert. backporch)

check for the right screen frequency:
    screen time (us) =
            ((g_vblank + 3*g_vsync + g_vdisplay)/2)*linetime
    screen frequency = 1000000 / screen time

Calculation of Transferdelay:

```
Transferdelay = system DMA latency + VRAM access + 4*SClk
g_transfdel = Transferdelay ( in periods of SClk )
g_transfdel = (1053ns / period of SClk(ns)) + 12
g_meminit   = 512 - g_transfdel
```

check if:

```
    g_transfdel < g_backporch
    g_transfdel < g_shdisplay
```

When (g_transfdel <= g_backporch) adjust g_backporch to obey this rule. Adjust the parameters calculated from g_backporch and see if all conditions are still true.

Adjusting the screen

When the screen is not horizontal centered adjust backporch. Adjust the parameters calculated from g_backporch and see if all conditions are still true.

When the the vertical blanked period is to short decrease g_vsync and recalculate g_vblank.

CHAPTER 4

Example video parameters for the G300

This chapter contains values for initializing the G300 VTG. They are calculated for several screen resolutions from the given monitor parameters. The monitor parameters are based on the values from the MultiSync 5D User's Manual (NEC) and were tested with a CC143 module (rev. 1.0) on a NEC MultiSync 5D monitor.

Values marked with *) are calculated back from the register values.

All calculated values except for g_transfdel and g_meminit are calculated with the program 'g300calc'.
For the calculation of g_transfdel and g_meminit refer to the documentation on the calculation of TransferDelay.

	monitor parameters (us)		g300-VTG register values
screen resolution	**1024x768**		
crystal frequency	**5MHz**		
multiplication factor			13
screenunit	0.062		
linetime	20.675	g_linetime	336
horizontal sync	1.0	g_halfsync	8
backporch	2.0	g_backporch	33
hor. resolution	1024	g_display	256
frontporch	1.9 *)	g_frontporch	31 *)
short display		g_shdisplay	88
broadpulse		g_broadpulse	137
vert. resolution	768	g_vdisplay	1536
vertical sync	80	g_vsync	8
total blanked	80+540+20	g_vblank	38
screenfrequency	60.53 Hz *)		
		g_transfdel	30
		g_meminit	482

	monitor parameters (us)		g300-VTG register values
screen resolution	1024x768		
crystal frequency	8MHz		
multiplication factor			8
screenunit	0.063		
linetime	20.675	g_linetime	330
horizontal sync	1	g_halfsync	8
backporch	1.9	g_backporch	30
hor. resolution	1024	g_display	256
frontporch	1.75 *)	g_frontporch	28 *)
short display		g_shdisplay	91
broadpulse		g_broadpulse	137
vert. resolution	768	g_vdisplay	1536
vertical sync	80	g_vsync	8
total blanked	80+540+20	g_vblank	38
screenfrequency	60.68Hz *)		
		g_transfdel	29
		g_meminit	483

	monitor parameters (us)		g300-VTG register values
screen resolution	1280x1024		
crystal frequency	8MHz		
multiplication factor			13
screenunit	0.038		
linetime	15.62	g_linetime	408
horizontal sync	1	g_halfsync	13
backporch	2	g_backporch	52
hor. resolution	1280	g_display	320
frontporch	0.3 *)	g_frontporch	8 *)
short display		g_shdisplay	117
broadpulse		g_broadpulse	195
vert. resolution	1024	g_vdisplay	2048
vertical sync	90	g_vsync	12
total blanked	90+580+20	g_vblank	52
screenfrequency	59.96Hz *)		
		g_transfdel	472
		g_meminit	40

	monitor parameters (us)		g300-VTG register values
screen resolution	640x480		
crystal frequency	5MHz		
multiplication factor			5
screenunit	0.160		
linetime	31.77	g_linetime	198
horizontal sync	2.5	g_halfsync	8
backporch	3.2	g_backporch	20
hor. resolution	640	g_display	160
frontporch	0.32 *)	g_frontporch	2 *)
short display		g_shdisplay	61
broadpulse		g_broadpulse	97
vert. resolution	480	g_vdisplay	960
vertical sync	64	g_vsync	4
total blanked	64+1020+350	g_vblank	78
screenfrequency	60.125Hz *)		
		g_transfdel	19
		g_meminit	493

	monitor parameters (us)		g300-VTG register values
screen resolution	800x600		
crystal frequency	5		
multiplication factor			7
screenunit	0.114		
linetime	28.44	g_linetime	248
horizontal sync		g_halfsync	9
backporch	2.6	g_backporch	23
hor. resolution	800	g_display	200
frontporch	0.8 *)	g_frontporch	7 *)
short display		g_shdisplay	76
broadpulse		g_broadpulse	112
vert. resolution	600	g_vdisplay	1200
vertical sync	80	g_vsync	6
total blanked	80+600+30	g_vblank	32
screenfrequency	56.452Hz *)		
		g_transfdel	22
		g_meminit	490

	monitor parameters (us)		g300-VTG register values
screen resolution	800x600		
crystal frequency	a)8 b)5		
multiplication factor			a)5 b)8
screenunit	0.1		
linetime	28.44	g_linetime	284
horizontal sync	2	g_halfsync	10
backporch	3.56	g_backporch	30
hor. resolution	800	g_display	200
frontporch	3.4 *)	g_frontporch	34 *)
short display		g_shdisplay	58
broadpulse		g_broadpulse	108
vert. resolution	600	g_vdisplay	1200
vertical sync	80	g_vsync	6
total blanked	60+600+30	g_vblank	32
screenfrequency	56.338Hz *)		
		g_transfdel	23
		g_meminit	489

CHAPTER 5

Initialization program for the CC143

```
/*
* program: ig300bb
* Can be used to initialize the G300B on the CC143 module.
*
* The following parameters can be used to initialize the G300B
* fora 1024x768 screen with a 8MHz crystal.
*
* G_HalfSync  = 8
* BackPorch   = 30
* Display     = 256
* ShortDisplay= 91
* BroadPulse  = 137
* VSync       = 8
* VBlank      = 38
* VDisplay    = 1536
* LineTime    = 330
* MemInit     = 483
* TransDel    = 29
* G_LineStar  = 0
* MaskReg     = 0xff
* ControlReg  = 0x70081
*/

#include <stdio.h>

#define SHORTIO  0x5000f000      /* start of CC143 short I/O map */

/* structure defining the G300 Register Map */
struct G300 {
        unsigned int G_Palet[256];  /* Color Lookup Table */
        unsigned int G_ByteCnt;     /* Byte Counter */
        unsigned int G_Res[32];     /* not used (gap in map) */
        unsigned int G_HalfSync;    /* HalfSync Register */
        unsigned int G_BackPorch;   /* BackPorch Register */
        unsigned int G_Display;     /* Display Register */
        unsigned int G_ShortDisplay;/* ShortDisplay Register */
        unsigned int G_BroadPulse;  /* BroadPuls Register */
        unsigned int G_VSync;       /* VSync Register */
        unsigned int G_VBlank;      /* VBlank Register */
        unsigned int G_VDisplay;    /* VDisplay Register */
        unsigned int G_LineTime;    /* LineTime Register */
        unsigned int G_LineStart;   /* LineStart Register */
        unsigned int G_MemInit;     /* MemInit Register */
        unsigned int G_TransDel;    /* TransferDelay Register */
        unsigned int G_HIncr;       /* HIncrementer Register */
        unsigned int G_VIncr;       /* VIncrementer Register */
        unsigned int G_TBIncr;      /* TBIncrementer Register */
        unsigned int G_Res2[16];    /* not used (gap in map) */
```

```
        unsigned int G_MaskReg;     /* Mask Register */
        unsigned int G_Res3[31];    /* not used (gap in map) */
        unsigned int G_ControlReg;  /* Control Register */
        unsigned int G_Res4[31];    /* not used (gap in map) */
        unsigned int G_TopScreen;   /* Top of Screen Register */
        unsigned int G_Res5[31];    /* not used (gap in map) */
        unsigned int G_BootLoc;     /* Boot Location */
};

unsigned char *A3128;
unsigned char *A2724;
unsigned char *A2320;
unsigned char *ADDRM1;
unsigned char *ADDRM2;

unsigned char *CMR2B;
unsigned char *OMRB;

unsigned short *ramstart;
unsigned short *ramstop;

struct G300 *p;
unsigned short *wptr;

main()
{
    unsigned int n;

/* initialize pointers to CC143 Address Registers */
    A3128  = (unsigned char *)(SHORTIO+0xa01);
    A2724  = (unsigned char *)(SHORTIO+0xa03);
    A2320  = (unsigned char *)(SHORTIO+0xa05);

/* initialize pointers to CC143 AM-code Registers */
    ADDRM1 = (unsigned char *)(SHORTIO+0xb01);
    ADDRM2  = (unsigned char *)(SHORTIO+0xb03);

/* initialize pointers to CC143 DUSCC Registers */
    CMR2B  = (unsigned char *)(SHORTIO+0xd43);
    OMRB   = (unsigned char *)(SHORTIO+0xd57);

/* initialize pointers to the start and end of the part of the */
/* CC143 Video RAM used for displaying a 1024x768 screen */
    ramstart = (unsigned short *)0x70000000;
    ramstop  = (unsigned short *)0x700c0000;

/* call routine for resetting the G300 */
    resetg300();

/* Set VMEbus address of the CC143 for standard
    or extended addressing. */

/* This is the start of the CC143 Video RAM */
    *A3128 = 0;
    *A2724 = 0;
    *A2320 = 0;
```

```
/* set am-code to SUP NP PROG DATA */
    *ADDRM1 = 0x0f;
/* set am-code to STD EXT */
    *ADDRM2 = 0x03;

/* initialize a pointer to 'struct G300' */
    p = (struct G300 *) SHORTIO;

/*  set Clock source to PPL mode and
    set multiplication factor to '8' */

    wg300byte( &p->G_BootLoc,     0x28 );
/*  the following read is necessary due to a bug in the G300B */
    n = rg300byte( &p->G_ControlReg );
/*  initialize for byte mode with VTG disabled */
    wg300byte( &p->G_ControlReg , 0x80);
/* now abort the byte wide write cycle before completion */
/* (see data sheet G300B) */

/* write screen parameters to G300 registers */
    wg300( &p->G_HalfSync       , 8);
    wg300( &p->G_BackPorch      , 30);
    wg300( &p->G_Display        , 256);
    wg300( &p->G_ShortDisplay   , 91);
    wg300( &p->G_BroadPulse     , 137);
    wg300( &p->G_VSync          , 8);
    wg300( &p->G_VBlank         , 38);
    wg300( &p->G_VDisplay       , 1536);
    wg300( &p->G_LineTime       , 330);
    wg300( &p->G_MemInit        , 483);
    wg300( &p->G_TransDel       , 29);
    wg300( &p->G_LineStart      , 0);       /* write zero */
    wg300( &p->G_MaskReg        , 0xff);    /* no mask */
    wg300( &p->G_ControlReg     , 0x70081); /* 8bit/pixel,bytemode */
                                            /* VTG enabled */

  /* set clut location 0 to grey */
    wg300( &p->G_Palet[0], 0x808080 );
  /* fill video RAM with first CLUT entry */
    for( wptr=ramstart; wptr<ramstop; wptr++ )
        *wptr=0x0000;
}

resetg300()
{
    /* initialize CMR2B register */
    *(unsigned char *) CMR2B = 0x38;
    /* reset G300 */
    *(unsigned char *) OMRB  = 0x02;
    tsleep(10);
    *(unsigned char *) OMRB  = 0x00;
    tsleep(10);
}

wg300(ptr,value)
```

```
unsigned char * ptr;
unsigned int value;
{
    unsigned char thisb;

    ptr = ptr + 3;
    thisb = (unsigned char) (value & 0xff);
    *ptr = thisb;
    thisb = (unsigned char) ((value>>8) & 0xff);
    *ptr = thisb;
    thisb = (unsigned char) ((value>>16) & 0xff);
    *ptr = thisb;
}

wg300byte(ptr,value)
unsigned char * ptr;
unsigned int value;
{
    unsigned char thisb;

    ptr = ptr + 3;
    thisb = (unsigned char) (value & 0xff);
    *ptr = thisb;
}

rg300byte( ptr )
unsigned char *ptr;
{
    register unsigned int value;
    unsigned char thisb1;

    ptr = ptr + 3;
    thisb1 = *ptr;

    value = thisb1;
    return( value );
}
```

CHAPTER 5

Data sheet G300B colour video controller