

V9968 Programmer's Manual

## **Attention!**

This document has been translated from Japanese to English from its original source at:

[https://github.com/hra1129/V9968\\_Cartridge/blob/main/fpga/V9968\\_Cartridge\\_TangNano20K/src/th9958/manual/v9968\\_programmers\\_manual\\_vdp\\_command.pdf](https://github.com/hra1129/V9968_Cartridge/blob/main/fpga/V9968_Cartridge_TangNano20K/src/th9958/manual/v9968_programmers_manual_vdp_command.pdf)

This documentation was translated on March 6, 2026, and it is important to note that it is part of a Work in Progress documentation on the development of V9968 and information could change or it can contain errors.

For the most up-to-date and accurate information, please refer to the original documentation and repository:

[https://github.com/hra1129/V9968\\_Cartridge](https://github.com/hra1129/V9968_Cartridge)

V9968 Programmer's Manual

# VDP Command

**table of contents**

1. Introduction.....3  
2. What are VDP commands?.....3  
3. How to execute commands.....3  
4. VDP Command Types.....4

## 1. Introduction

This document describes the VDP commands available on the V9968 processor.

## 2. What are VDP commands?

In the bitmap graphics modes of Graphic4 to 7, the amount of information to be handled increases enormously. Processing this information one by one by the CPU is inefficient, so VDP commands exist as dedicated hardware for drawing. When instructions are received from the CPU, they automatically perform processes such as drawing lines and rectangles, or block transfers. There are also commands that support block transfers to and from the CPU. In any case, it is hardware that supports drawing.

For modern programmers, it will be easiest to understand it as "the part that handles so-called GPU-like processing."

The VDP commands have been included since the V9938. The V9958 did not modify the commands, but was equipped with a mode that allowed VDP commands to be used in modes other than Graphic4 to 7. This will be discussed later.

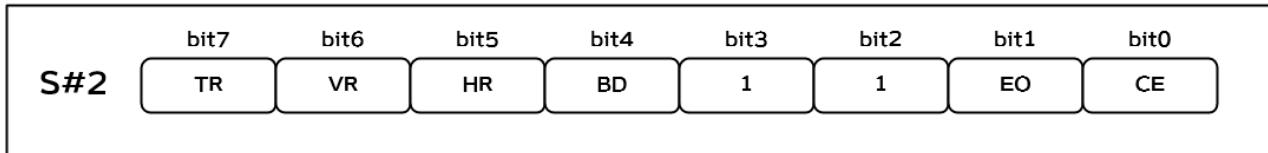
## 3. How to execute commands

Registers R#32 to R#58 are for VDP commands. Of these, R#47 to R#58 are new for the V9968.

Of the registers for VDP commands, all except R#46 are registers for setting command parameters. After setting the necessary parameters in these, writing them to R#46 begins execution of the command.

## V9968 Programmer's Manual

While the command is being executed, the least significant bit CE of status register S#2 is 1, and when it becomes 0, it signals that the command has been executed.



With the V9938 and V9958, this was the only way to know when a VDP command had completed, so in order to wait for completion when it was not known whether it had, it was necessary to repeatedly read S#2 and check whether the least significant bit was 0. For large block transfers, the command itself can take a considerable amount of time, so it is more efficient to complete another calculation while waiting.

The V9968 supports VDP command completion interrupts. You can use the same method as the V9938/V9958, but you can also check for completion using the interrupt that occurs when the command is complete.

We will discuss both control methods, waiting for completion by polling and by interrupt, later.

Before we get into the detailed control, let's introduce the types of commands.

## 4. VDP Command Types

There are 16 VDP commands, as summarized in the table below.

*Table 1: List of VDP commands*

Command name	Destination	Source	Transfer Unit	Mnemonic	R#46 CMD			
High speed move	VRAM	CPU	byte	HMMC	1	1	1	1
	VRAM	VRAM	byte	YMMM	1	1	1	0
	VRAM	VRAM	byte	HMMM	1	1	0	1
	VRAM	VDP	byte	HMMV	1	1	0	0
Logical move	VRAM	CPU	dot	LMMC	1	0	1	1
	CPU	VRAM	dot	LMCM	1	0	1	0
	VRAM	VRAM	dot	LMMM	1	0	0	1
	VRAM	VDP	dot	LMMV	1	0	0	0
Line	VRAM	VDP	dot	LINE	0	1	1	1
Search	VRAM	VDP	dot	SRCH	0	1	1	0
Pset	VRAM	VDP	dot	PSET	0	1	0	1
Point	VDP	VRAM	dot	POINT	0	1	0	0
Logical move	VRAM	VRAM	dot	LRMM	0	0	1	1
Logical draw font	CPU	VRAM	dot	LFMC	0	0	1	0
	VRAM	VRAM	dot	LFMM	0	0	0	1
Stop	-	-	-	STOP	0	0	0	0

Among these, LRMM, LFMC, and LFMM are newly added commands in the V9968. Normally, they function as STOP commands, just like in the V9958, but by setting ECOM in R#20 to 1, LRMM, LFMC, and LFMM become available.

The specific usage of each command is explained below.

#### **4.1. HMMC**

#### **4.2. YMMM**

#### **4.3. HMMM**

#### **4.4. HMMV**

#### **4.5. LMMC**

#### **4.6. LMCM**

#### **4.7. LMMM**

#### **4.8. LMMV**

#### **4.9. LINE**

**4.10. SRCH**

**4.11. PSET**

**4.12. POINT**

**4.13. LRMM**

**4.14. LFMC**

**4.15. LFMM**

**4.16. STOP**