**CGRAM ADDRESS MAPPING**

In the HD44780 LCD controller, the CGRAM address format is defined by the internal architecture of the controller. When writing to CGRAM, the address is specified using a specific bit pattern where the first two bits are always set to `0` and `1`. This is part of the command protocol used to differentiate CGRAM addressing from other commands and data operations.

Here is a detailed explanation of why the first two bits are always `0` and `1` when addressing CGRAM:

**HD44780 Command and Data Structure**

The HD44780 uses an 8-bit command and data interface. The commands and addresses sent to the LCD are structured in such a way to allow the controller to understand the type of operation being requested. Here's how it works:

1. **Command/Instruction Code:** Commands or instructions are issued to the LCD controller to perform specific tasks such as setting the CGRAM address, clearing the display, etc.
2. **Address Codes**: When writing to the CGRAM or DDRAM, specific address codes are used. These addresses need to be distinct from regular instructions.

**CGRAM Addressing Format**

To address the CGRAM, the HD44780 controller expects a specific bit pattern in the command byte. The format for CGRAM addressing is as follows:

* **Bits 7 and 6:** Always set to `01` to indicate a CGRAM address operation.
* **Bits 5 to 0:** These bits specify the actual address within the CGRAM.

Thus, the full 8-bit address for CGRAM is formatted like this:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | A5 | A4 | A3 | A2 | A1 | A0 |

Where `A5 A4 A3 A2 A1 A0` represent the 6-bit CGRAM address.

**CGRAM Address Range**

Since CGRAM addresses are 6 bits wide (A5-A0), the address range is from `0x00` to `0x3F` (0 to 63 in decimal). This allows for addressing up to 64 different bytes in the CGRAM. The two most significant bits (MSBs), `01`, indicate that the following bits represent a CGRAM address.

**Example of CGRAM Addressing**

To clarify, let’s look at the CGRAM address for the first custom character (Custom Char 0):

* **Start Address for Custom Char 0:** `0x40` (which is **`01000000`** in binary)
* In the CGRAM address command format, this would be: `01 000000`

Here's how the addressing for the first custom character (Custom Char 0) looks in binary and hexadecimal:

### CGRAM Address Mapping (8 Bits Apart)

|  |  |  |
| --- | --- | --- |
| **Custom Character** | **CGRAM Address** | **Binary Address** |
| Custom Char 0 | 0x40 | 0100 0000 |
| Custom Char 1 | 0x48 | 0100 1000 |
| Custom Char 2 | 0x50 | 0101 0000 |
| Custom Char 3 | 0x58 | 0101 1000 |
| Custom Char 4 | 0x60 | 0110 0000 |
| Custom Char 5 | 0x68 | 0110 1000 |
| Custom Char 6 | 0x70 | 0111 0000 |
| Custom Char 7 | 0x78 | 0111 1000 |

**Explanation**

Each custom character in the CGRAM is defined by 8 bytes (rows). Thus, the address for each subsequent character starts 8 bytes after the previous one. Here’s how it looks:

**Custom Character 0:** Starts at address `0x40` (binary `01000000`).

**Custom Character 1:** Starts at address `0x48` (binary `01001000`).

**Custom Character 2:** Starts at address `0x50` (binary `01010000`).

**Custom Character 3:** Starts at address `0x58` (binary `01011000`).

**Custom Character 4:** Starts at address `0x60` (binary `01100000`).

**Custom Character 5:** Starts at address `0x68` (binary `01101000`).

**Custom Character 6:** Starts at address `0x70` (binary `01110000`).

**Custom Character 7:** Starts at address `0x78` (binary `01111000`).

Each character definition consists of 8 bytes of data, one for each row of the 5x8 dot matrix. The `cmd` function sets the CGRAM address, and the `dat` function writes the data to that address. This way, you can store custom characters in CGRAM and then display them by writing the appropriate character code to DDRAM.

**Summary**

The first two bits of the CGRAM address are always `0` and `1` because this bit pattern is reserved by the HD44780 controller to indicate a CGRAM addressing operation. This design ensures that the controller can distinguish between different types of commands and data operations, providing a clear and structured way to manage custom character data.

### CGRAM Address Mapping in 8-bit Binary Format

|  |  |  |  |
| --- | --- | --- | --- |
| **CGRAM Address** | **Binary Address** | **Character Row** | **Custom Character** |
| 0x40 | 1000000 | Row 0 | Custom Char 0 |
| 0x41 | 1000001 | Row 1 | Custom Char 0 |
| 0x42 | 1000010 | Row 2 | Custom Char 0 |
| 0x43 | 1000011 | Row 3 | Custom Char 0 |
| 0x44 | 1000100 | Row 4 | Custom Char 0 |
| 0x45 | 1000101 | Row 5 | Custom Char 0 |
| 0x46 | 1000110 | Row 6 | Custom Char 0 |
| 0x47 | 1000111 | Row 7 | Custom Char 0 |
| 0x48 | 1001000 | Row 0 | Custom Char 1 |
| 0x49 | 1001001 | Row 1 | Custom Char 1 |
| 0x4A | 1001010 | Row 2 | Custom Char 1 |
| 0x4B | 1001011 | Row 3 | Custom Char 1 |
| 0x4C | 1001100 | Row 4 | Custom Char 1 |
| 0x4D | 1001101 | Row 5 | Custom Char 1 |
| 0x4E | 1001110 | Row 6 | Custom Char 1 |
| 0x4F | 1001111 | Row 7 | Custom Char 1 |
| 0x50 | 1010000 | Row 0 | Custom Char 2 |
| 0x51 | 1010001 | Row 1 | Custom Char 2 |
| 0x52 | 1010010 | Row 2 | Custom Char 2 |
| 0x53 | 1010011 | Row 3 | Custom Char 2 |
| 0x54 | 1010100 | Row 4 | Custom Char 2 |
| 0x55 | 1010101 | Row 5 | Custom Char 2 |
| 0x56 | 1010110 | Row 6 | Custom Char 2 |
| 0x57 | 1010111 | Row 7 | Custom Char 2 |
| 0x58 | 1011000 | Row 0 | Custom Char 3 |
| 0x59 | 1011001 | Row 1 | Custom Char 3 |
| 0x5A | 1011010 | Row 2 | Custom Char 3 |
| 0x5B | 1011011 | Row 3 | Custom Char 3 |
| 0x5C | 1011100 | Row 4 | Custom Char 3 |
| 0x5D | 1011101 | Row 5 | Custom Char 3 |
| 0x5E | 1011110 | Row 6 | Custom Char 3 |
| 0x5F | 1011111 | Row 7 | Custom Char 3 |
| 0x60 | 1100000 | Row 0 | Custom Char 4 |
| 0x61 | 1100001 | Row 1 | Custom Char 4 |
| 0x62 | 1100010 | Row 2 | Custom Char 4 |
| 0x63 | 1100011 | Row 3 | Custom Char 4 |
| 0x64 | 1100100 | Row 4 | Custom Char 4 |
| 0x65 | 1100101 | Row 5 | Custom Char 4 |
| 0x66 | 1100110 | Row 6 | Custom Char 4 |
| 0x67 | 1100111 | Row 7 | Custom Char 4 |
| 0x68 | 1101000 | Row 0 | Custom Char 5 |
| 0x69 | 1101001 | Row 1 | Custom Char 5 |
| 0x6A | 1101010 | Row 2 | Custom Char 5 |
| 0x6B | 1101011 | Row 3 | Custom Char 5 |
| 0x6C | 1101100 | Row 4 | Custom Char 5 |
| 0x6D | 1101101 | Row 5 | Custom Char 5 |
| 0x6E | 1101110 | Row 6 | Custom Char 5 |
| 0x6F | 1101111 | Row 7 | Custom Char 5 |
| 0x70 | 1110000 | Row 0 | Custom Char 6 |
| 0x71 | 1110001 | Row 1 | Custom Char 6 |
| 0x72 | 1110010 | Row 2 | Custom Char 6 |
| 0x73 | 1110011 | Row 3 | Custom Char 6 |
| 0x74 | 1110100 | Row 4 | Custom Char 6 |
| 0x75 | 1110101 | Row 5 | Custom Char 6 |
| 0x76 | 1110110 | Row 6 | Custom Char 6 |
| 0x77 | 1110111 | Row 7 | Custom Char 6 |
| 0x78 | 1111000 | Row 0 | Custom Char 7 |
| 0x79 | 1111001 | Row 1 | Custom Char 7 |
| 0x7A | 1111010 | Row 2 | Custom Char 7 |
| 0x7B | 1111011 | Row 3 | Custom Char 7 |
| 0x7C | 1111100 | Row 4 | Custom Char 7 |
| 0x7D | 1111101 | Row 5 | Custom Char 7 |
| 0x7E | 1111110 | Row 6 | Custom Char 7 |
| 0x7F | 1111111 | Row 7 | Custom Char 7 |