HT8 True EEPROM & Emulated EEPROM Application Difference Comparisons

Introduction

EEPROM, which stands for Electrically Erasable Programmable Read-Only Memory, is by its nature a non-volatile form of re-programmable memory, experiencing no data lost even when its power supply has been removed. This kind of memory is often used to store user data, product ID and other information. Holtek provides different capacities of True EEPROM integrated within the HT8 Flash MCUs to meet the needs of data storage and simplified circuit design. Holtek has also developed an Emulated EEPROM type which is also integrated within the HT8 Flash MCUs and which is an alternative solution for low cost applications.

This document will describe the operation methods and relevant points to note when using both True EEPROM and Emulated EEPROM, and compare the differences between these two memory types.

Functional Description

True EEPROM

True EEPROM Operation Methods

General True EEPROM always supports byte operation mode while some True EEPROM with larger capacities also provide a page operation mode. The operating methods and points to consider for the byte operation mode have already been introduced in the AN0408E application note. For the page operation mode, the following summarises the differences between the page operation and byte operation modes.

- Byte write & Page write

<table>
<thead>
<tr>
<th>Comparison Item</th>
<th>Byte Write</th>
<th>Page Write</th>
</tr>
</thead>
<tbody>
<tr>
<td>Write unit</td>
<td>Byte</td>
<td>Page(2)</td>
</tr>
<tr>
<td>Write address</td>
<td>A single location indicated by EEA</td>
<td>A page indicated by EEA</td>
</tr>
<tr>
<td>After writing data into EED</td>
<td>EEA is not changed</td>
<td>EEA is incremented by one by hardware(2)</td>
</tr>
<tr>
<td>Write data</td>
<td>Data into EED</td>
<td>Data into page buffer</td>
</tr>
<tr>
<td>Write sequence</td>
<td>No differences</td>
<td></td>
</tr>
<tr>
<td>Write cycle time</td>
<td>No differences</td>
<td></td>
</tr>
</tbody>
</table>

Table 1
### Byte read & Page read

<table>
<thead>
<tr>
<th>Comparison Item</th>
<th>Byte read</th>
<th>Page read</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read unit</td>
<td>Byte (3)</td>
<td></td>
</tr>
<tr>
<td>Read address</td>
<td>A single location indicated by EEA</td>
<td></td>
</tr>
<tr>
<td>Read sequence</td>
<td>No differences</td>
<td></td>
</tr>
<tr>
<td>Read cycle time</td>
<td>No differences</td>
<td></td>
</tr>
<tr>
<td>Read data</td>
<td>Data from EED (3)</td>
<td>EEA is incremented by one by hardware(2)</td>
</tr>
<tr>
<td>After reading data</td>
<td>EEA is not changed</td>
<td>EEA is incremented by one by hardware(2)</td>
</tr>
</tbody>
</table>

**Table 2**

Note: 1. The page buffer size depends on the specific MCU EEPROM specification.

2. The EEA content is incremented by one automatically until its value reaches the last address of the current page and then remains unchanged.

3. Because there is only one EED register, even for the Page read operation, only one-byte of data is actually read out after executing a read operation.

### True EEPROM Operation Considerations

1. Ensure that the fSUB clock is stable before executing a write operation.

2. The WREN and WR bits or the RDEN and RD bits should not both be set high at the same time in a single instruction to avoid unintended results.

3. A write sequence that successively sets the WREN and WR bits high must be completed within two consecutive instruction cycles, otherwise the write operation will fail.
   - To avoid the write sequence being interrupted, the global interrupt bit EMI should first be cleared before setting the WREN bit to 1, and then set high again after the WR bit has been set to 1.
   - For extended instructions such as LSET, one instruction requires at least two instruction cycles to implement. Therefore, the write sequence, which must be completed within two consecutive instruction cycles, cannot be performed using extended instructions.
   - For EEPROM write operations, the EEC register settings, which is generally located in Bank 1, must be implemented using an indirect addressing method.

4. To enable a read operation and if the RDEN bit is zero, the RDEN bit should be set high first after which the RD bit should be set high. Otherwise the read operation will fail.

5. If no more read operations are required, the RDEN bit should be cleared to zero by S/W.

### Emulated EEPROM

**Emulated EEPROM Erase/Write/Read Format**

Emulated EEPROM supports Erase, Write and Read operations. The Erase/Write/Read Formats may differ with different MCU specifications. The following table lists several examples with different formats.
**Emulated EEPROM Register**

The Emulated EEPROM related registers fall into three categories, which are the address register, EAR, the data registers, EDH/EDL, and a control register, ECR. The register definitions for different MCU specifications may be different. For example, different write formats require different numbers of data register pairs and different capacities lead to different actual valid bits in the data registers.

**Emulated EEPROM Operation Methods**

All the Emulated EEPROM operations are implemented by setting the related registers as shown in the following operation flowcharts.

**Emulated EEPROM Erase Flowchart**

<table>
<thead>
<tr>
<th>Operation</th>
<th>HT66F0021</th>
<th>HT68F0021</th>
<th>BH67F5235</th>
</tr>
</thead>
<tbody>
<tr>
<td>Erase</td>
<td>1 page - 16 words</td>
<td>1 page - 32 words</td>
<td>1 page - 16 words</td>
</tr>
<tr>
<td>Write</td>
<td>1 word</td>
<td>1 word</td>
<td>4 words</td>
</tr>
<tr>
<td>Read</td>
<td>1 word</td>
<td>1 word</td>
<td>1 word</td>
</tr>
</tbody>
</table>

Table 3

---

**Figure 1**

[Flowchart for Emulated EEPROM Erase Operation]
Emulated EEPROM Write Flowchart

Start

Execute erase operation

EAR = Desired write start address

EDH/L = Data to be written

EMI = 0

EWREN = 1

EWR = 1

EMI = 1

Write finish? (EWR = 0)?

No

Write other data?

Yes

Yes

No

End

Figure 2

Note: The setting of the EAR and EDH/EDL registers is related to the Write format which should refer to the specific MCU specification. The actual available bits in the EAR and EDH/EDL registers may be different in different MCUs.
Emulated EEPROM Read Flowchart

Start

EAR = Desired read address

ERDEN = 1

ERD = 1

Read finish? (ERD = 0)?

No

Yes

Read EDHFL (See Note 1)

Read other address data?

Yes

No

ERDEN = 0

End

Figure 3

Note: for certain MCUs, there are multiple pairs of data registers, refer to the datasheet for their register definitions.

Emulated EEPROM Operation Considerations

1. Ensure that the fSYS clock frequency is equal to or greater than 2MHz and the fSUB clock is stable before executing an erase or write operation.

2. When an Emulated EEPROM erase, write or read operation is successfully activated, the CPU stops. Therefore other operations cannot be executed until the erase, write or read operation has fully completed.

3. The EEREN and EER bits, EWREN and EWR bits or the ERDEN and ERD bits cannot be set high at the same time in the same instruction to avoid unintended results.

4. Ensure that an erase operation has been carried out on the area that data will be written into before executing a write operation.

5. An erase sequence that successively sets the EEREN and EER bits high as well as a write sequence that successively sets the EWREN and EWR bits high must be completed within two consecutive instruction cycles. Otherwise the erase or write operation will fail.

   - To avoid the above-mentioned sequence being affected by interrupts, the global interrupt bit EMI should first be cleared before starting an erase or write sequence. The EMI bit can then be set high again after the erase or write sequence has completed.
For extended instructions such as LSET, here a single instruction requires at least two instruction cycles to execute. Therefore any erase or write sequence which must be completed within two consecutive instruction cycles cannot be performed using extended instructions.

For erase or write operation, any ECR register settings which are not located in Bank 0 must be implemented using an indirect addressing method.

6. To enable a read operation and if the ERDEN bit is zero, the ERDEN bit should be set high first after which the ERD bit should be set high. Otherwise the read operation will fail.

7. If no more read operations are required, the ERDEN bit should be cleared to zero by S/W.

### True EEPROM and Emulated EEPROM Comparisons

<table>
<thead>
<tr>
<th>Comparison Item</th>
<th>True EEPROM</th>
<th>Emulated EEPROM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Structure width</td>
<td>Byte</td>
<td>Word(Note1)</td>
</tr>
<tr>
<td>Allowed Operation</td>
<td>Byte write (including erase which has been hidden) / Byte read / Page writer (including erase which has been hidden) / Page read (Note2)</td>
<td>Erase, Write, Read</td>
</tr>
<tr>
<td>Operation sequence</td>
<td>Setup related registers by S/W</td>
<td>Erase/Write time: selected using EWRTS[1:0] bits</td>
</tr>
<tr>
<td>Operation time</td>
<td>Write time: Max. 4ms; Read time(Note3): Max. 4tSYS(Note4)</td>
<td>Read time: about 8tSYS(Note4)</td>
</tr>
<tr>
<td>Interrupt</td>
<td>Write cycle end interrupt</td>
<td>No interrupts, CPU is stopped when write operation is being executed</td>
</tr>
<tr>
<td>Effects on CPU</td>
<td>Operations to EEPROM have no effects on CPU running</td>
<td>When erase, read or write operation is being executed, CPU is stopped.</td>
</tr>
</tbody>
</table>

Table 4

Note: 1. The number of structure width valid bits should be referred to the specific MCU EEPROM specification.

2. Whether supporting a Page write (including erase which has been hidden) and page read operations are defined by the specific MCU EEPROM specification.

3. Generally the maximum time to read a byte of data from a true EEPROM is 4tSYS. However, the actual required time depends on the specific MCU EEPROM specification.

Taking the HT66F40 as an example, the maximum time to read a byte of data is 90μs.

4. tSYS = 1/fSYS.

### Data Write Protection

Refer to the following notes before executing EEPROM operations to ensure that the data can be written into the EEPROM correctly and that any stored data will not be inadvertently modified.

### Erase / Write Data Timing

1. Ensure that the fSUB clock is stable before executing any erase/write operations. For Emulated EEPROM the fSYS clock frequency must be equal to or greater than 2MHz.

2. It is recommended that the MCU provides an appropriate delay after power on to ensure that VDD has stabilised before executing any EEPROM operations.
3. When a data save operation is implemented during power down, ensure that the remaining power and time is adequate for completion of the data erase and write operations as \( V_{DD} \) falls to the minimum EEPROM operating voltage.

**Write Operation**

Ensure that the write operation has fully completed before changing any EEPROM related registers.

**Data Read Back**

It is recommended to execute a read operation to read out data that has just been written, and then to compare the data with the expected data. If the data is different, which means the data has not been written correctly, the write operation should be re-executed.

**Clear Write Enable Bit Regularly**

After a write operation, the write enable bit WREN (for True EEPROM) or EWREN bit (for Emulated EEPROM) should be cleared to zero regularly.

**Clear Erase Enable Bit Regularly**

After an erase operation, the erase enable bit EEREN (for Emulated EEPROM) should be cleared to zero regularly.

**RAM Pointer Settings**

After an erase or write operation has completed, the RAM Bank Pointer, MP1H or MP2H, and the address register, MP1L or MP2L, whose values were configured for indirectly addressing the EEC register (for True EEPROM) or ECR register (for Emulated EEPROM), should be set to other values to prevent the program from starting a spurious write operation.

**Execute HALT Instruction Timing**

Before executing the HALT instruction, ensure that all the EEPROM operations including erase, write and read operation have completed.

**Data Backup / Check**

If the EEPROM capacity is large enough, the EEPROM can be divided into several blocks to implement a multiple data backup by writing the same data into specified addresses of several blocks. When reading the data, the data stored in different blocks will be read out for comparison and appropriate operations performed under a preset arbitration mechanism.

In addition, the data when being stored can be suffixed by a check code, such as a checksum or CRC check. When the whole string of data has been read, a preset check calculation is executed to generate a check result which is compared with the check code that has been read out to determine the data correctness.
Tips for prolonging Service Life

Except for the influence of the operating environment including temperature, humidity and electromagnetic fields, an EEPROM service life mainly depends on the number of erase/write cycle times. For data that needs to be updated frequently, it is recommended to divide the EEPROM into several blocks and then store the data into a different block for each data update, thus prolonging the EEPROM service life.

Conclusion

This document has introduced the operation procedures and considerations of the HT8 True EEPROM & Emulated EEPROM and has summarised some differences between these two types. This document is provided for users to further understand how to use the HT8 True EEPROM & Emulated EEPROM.

Reference Material

Reference document: AN0408E.
For more details consult the Holtek website at www.holtek.com.

Version and Modification Information

<table>
<thead>
<tr>
<th>Date</th>
<th>Author</th>
<th>Issue release</th>
</tr>
</thead>
<tbody>
<tr>
<td>2019.07.22</td>
<td>Yu Feng, Liao (廖裕峰)</td>
<td>First version</td>
</tr>
</tbody>
</table>
Disclaimer

All information, trademarks, logos, graphics, videos, audio clips, links and other items appearing on this website (‘Information’) are for reference only and is subject to change at any time without prior notice and at the discretion of Holtek Semiconductor Inc. and its related companies (hereinafter ‘Holtek’, ‘the company’, ‘us’, ‘we’ or ‘our’). Whilst Holtek endeavors to ensure the accuracy of the Information on this website, no express or implied warranty is given by Holtek to the accuracy of the Information. Holtek shall bear no responsibility for any incorrectness or leakage. Holtek shall not be liable for any damages (including but not limited to computer virus, system problems or data loss) whatsoever arising in using or in connection with the use of this website by any party. There may be links in this area, which allow you to visit the websites of other companies. These websites are not controlled by Holtek. Holtek will bear no responsibility and no guarantee to whatsoever Information displayed at such sites. Hyperlinks to other websites are at your own risk.

Limitation of Liability

In no event shall Holtek Limited be liable to any other party for any loss or damage whatsoever or howsoever caused directly or indirectly in connection with your access to or use of this website, the content thereon or any goods, materials or services.

Governing Law

The Disclaimer contained in the website shall be governed by and interpreted in accordance with the laws of the Republic of China. Users will submit to the non-exclusive jurisdiction of the Republic of China courts.

Update of Disclaimer

Holtek reserves the right to update the Disclaimer at any time with or without prior notice, all changes are effective immediately upon posting to the website.