**Abstract** - This paper focuses on study of NOR based Flash Memory Controller. A flash memory controller manages the data stored on flash memory and communicates with a computer or electronic device. When the system or device needs to read data from or write data to the flash memory, it will communicate with the flash memory controller. In order to simplify the process of programming or erasing the memory along with taking care of all special operations required to update the memory contents, controller can be used.

**Keywords** - NOR flash memory, Flash memory controller, Commands

I. INTRODUCTION

There is an extensive range of NOR and NAND Flash storage products available, including several code and data Flash memory solutions from serial and parallel NOR, to raw and managed NAND, to solid state drives (SSDs). Selecting the right solution requires an understanding of each technology, including attributes related to I/O performance, pin count, data integrity differences dependent on the system usage model, and manufacturing longevity requirements. This guide describes the various Flash technologies and provides a systematic way for the system designer to select the optimal nonvolatile memory solution based on key design considerations. The application requirements will ultimately dictate the right solution.[1]

NOR Flash devices are primarily used for reliable code storage (boot, application, OS, and execute-in-place (XIP) code in an embedded system), with some limited data storage. They are available in low densities up to 2Gb. NOR devices operate in high-speed burst or page mode for use in XIP environments where code runs directly from the device to minimize system boot time. NOR is easy to implement and requires minimal ongoing management due to the underlying cell structure. This eliminates the need for external error correction code (ECC), bad block management, and wear leveling. NOR flash provides a convenient location for storage of the setting. A method of operating a NOR flash memory controller may include sequentially writing to a first block of each sector of the NOR flash memory, followed by the memory controller sequentially. When all blocks of all sectors of the NOR flash memory have been written, the memory controller erases a first sector sequentially and then writes to the sequentially first block of the first sector.[1,3] The rest of this paper is organized as follows. Section II describes flash memory controller architecture. Bus operations are described in section III. Standard command definitions are discussed in section IV. Applications of flash memory controller are described in section V. Section VI concludes the paper.

II. FLASH MEMORY CONTROLLER

As shown in figure 1, PROGRAM and ERASE commands are written to the command interface of the memory to perform different operations. Controller simplifies the process of programming, erasing, reading the memory by taking care of all special operations required to update the memory contents. The end of a PROGRAM or ERASE

---

**Figure 1**: Block diagram of Flash Memory Controller

**Figure 2**: Block diagram of Parallel NOR Flash Memory (M29W128GH)
operation can be detected and any error condition can be identified.

Data and input commands are first stored in FIFO. After that command decoder checks whether it received data/command sequence or any garbage data. Then it separates data and commands from input sequence. Then the operation is selected according input command i.e. whether it is READ, WRITE or PROGRAM operation. According to condition READ data condition or WRITE data condition sub codes of commands containing address and data cycles are generated. At this all signals are generated with the timing specifications which is required by flash memory.[2]

In reference to figure 2, various signals are shown which are required by Parallel NOR Flash Memory. All these signals are to be generated by flash memory controller.

III. BUS OPERATION

Table 1 : Bus Operation Of Memory

<table>
<thead>
<tr>
<th>Operation</th>
<th>AL Mode</th>
<th>SL Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>READ</td>
<td>L L H H</td>
<td>X</td>
</tr>
<tr>
<td>WRITE</td>
<td>L H L H</td>
<td>X</td>
</tr>
<tr>
<td>STANDBY</td>
<td>K K H H</td>
<td>X</td>
</tr>
<tr>
<td>DISABLE</td>
<td>A A A A</td>
<td>X</td>
</tr>
<tr>
<td>ENABLE</td>
<td>X X X X</td>
<td>X</td>
</tr>
</tbody>
</table>

As shown in Table no.1, different bus operations of flash memory and their respective conditions are discussed below.

A. READ
Bus READ operations read from the memory cells, registers, or CFI space. To accelerate the READ operation, the memory array can be read in page mode where data is internally read and stored in a page buffer. The page size is 8 words (16 bytes) and is addressed by address inputs A[2:0] in x16 bus mode and A[2:0] plus DQ15/A-1 in x8 bus mode. The extended memory blocks and CFI area do not support page read mode.

A valid READ operation requires setting the appropriate address on the address inputs, taking CE# and OE# LOW, and holding WE# HIGH. Data I/O signals output the value.

B. WRITE
Bus WRITE operations write to the command interface. A valid WRITE operation requires setting the appropriate address on the address inputs. These are latched by the command interface on the falling edge of CE# or WE#, whichever occurs last. Values on data I/O signals are latched by the command interface on the rising edge of CE# or WE#, whichever occurs first. OE# must remain HIGH during the entire operation.

C. STANDBY AND AUTOMATIC STANDBY
When the device is in read mode, driving CE# HIGH places the device in standby mode and drives data I/Os to High-Z. During PROGRAM or ERASE operations, the device continues to use the program/erase supply current (ICC3) until the operation completes. Automatic standby enables low power consumption during read mode. The memory enters automatic standby as internal supply current is reduced to ICC2. Data I/O signals still output data if a READ operation is in progress.

D. OUTPUT DISABLE
Data I/Os are High-Z when OE# is HIGH.

E. RESET
During reset mode, the device is desellected, and outputs are High-Z. The device is in reset mode when RST# is LOW. Power consumption is reduced to standby level independently from CE#, OE#, or WE# inputs.

IV. STANDARD COMMAND DEFINITIONS

PROGRAM and ERASE commands are written to the command interface of the memory. These commands should be generated by flash memory controller to perform operations on flash memory.[2]

A. READ/RESET Command
The READ/RESET (F0h) command returns the device to read mode and resets the errors in the status register. One or three bus WRITE operations can be used to issue the READ/RESET command. To return the device to read mode, this command can be issued between bus WRITE cycles before the start of a PROGRAM or ERASE operation.

B. PROGRAM Command
The PROGRAM (A0h) command can be used to program a value to one address in the memory array. The command requires four bus WRITE operations, and the final WRITE operation latches the address and data in the internal state machine and starts the program/erase controller. After programming has started, bus READ operations output the status register content.

After the PROGRAM operation has completed, the device returns to read mode, unless an error has occurred. When an error occurs, bus READ
operations to the device continue to output the status register. A READ/RESET command must be issued to reset the error condition and return the device to read mode.

C. PROGRAM SUSPEND Command

Used to interrupt a program operation so that data can be read from any block. When the PROGRAM SUSPEND command is issued during a program operation, the device suspends the operation within the program suspend latency time and updates the status register bits. After the program operation has been suspended, data can be read from any address. However, data is invalid when read from an address where a program operation has been suspended.

D. PROGRAM RESUME Command

Command must be issued to exit a program suspend mode and resume a PROGRAM operation. The controller can use DQ7 or DQ6 status bits to determine the status of the PROGRAM operation.

E. CHIP ERASE command

The CHIP ERASE (80/10h) command erases the entire chip. Six bus WRITE operations are required to issue the command and start the program/erase controller. Protected blocks are not erased. During the CHIP ERASE operation, the device ignores all other commands, including ERASE SUSPEND. It is not possible to abort the operation.

The CHIP ERASE command sets all of the bits in unprotected blocks of the device to 1. The operation is aborted by performing a reset or by powering-down the device. In this case, data integrity cannot be ensured, and the entire chip should be erased again.

F. BLOCK ERASE command

The BLOCK ERASE (80/30h) command erases a list of one or more blocks. It sets all of the bits in the unprotected selected blocks to 1. All previous data in the selected blocks is lost. Six bus WRITE operations are required to select the first block in the list.

After the command sequence is written, a block erase timeout occurs. During the timeout period, additional block addresses and BLOCK ERASE commands can be written. Each additional block must therefore be selected within the timeout period of the last block.

G. ERASE SUSPEND command

The ERASE SUSPEND (B0h) command temporarily suspends a BLOCK ERASE operation. One bus WRITE operation is required to issue the command. The block address is "Don't Care."

The program/erase controller suspends the ERASE operations within the erase suspend latency time of the ERASE SUSPEND command being issued. However, when the ERASE SUSPEND command is written during the block erase timeout, the device immediately terminates the timeout period and suspends the ERASE operation. After the program/erase controller has stopped, the device operates in read mode, and the erase is suspended.

H. ERASE RESUME command

The ERASE RESUME (30h) command restarts the program/erase controller after an ERASE SUSPEND operation. The device must be in read array mode before the RESUME command will be accepted.

V. APPLICATIONS

Flash memory is used for easy and fast information storage in

1) Computers,
   Memory Cards,
   USB Flash drives

2) It is also used in
   - Digital cameras
   - Home video game
   - Solid State drive for transfer of data between
     Computers and other digital products

CONCLUSION

In this paper, working of Flash memory controller and various commands are discussed. Various commands are studied such as READ/RESET command, PROGRAM SUSPEND command, PROGRAM RESEME command, BLOCK/CHIP ERASE command, ERASE SUSPEND command, ERASE RESUME command. Other supported commands can be implemented in flash controller such as UNLOCK BYPASS, PASSWORD PROTECTION commands

REFERENCES


