

## Class Description

This application note describes the class *SamsungNand\_basis*, which has as base class the *HpeJTAG\_basis*. For further information of the *HpeJTAG\_basis* and its methods see the AN-1-001 [1]

The *SamsungNand\_basis* provides general methods for the Nand-Flashes from the Manufacturer Samsung.

## Functional block diagram of Samsung Nand-Flashes

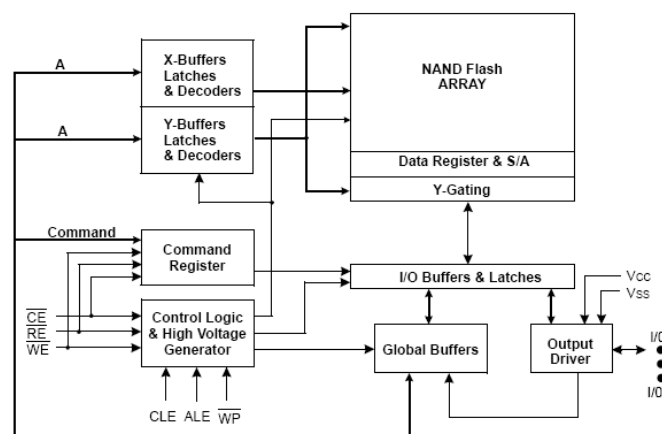


Figure 1: Functional block diagram

The class variables are connected as following:

- self.databus = I/O
- self.cen =  $\overline{CE}$
- self.wen =  $\overline{WE}$
- self.ren =  $\overline{RE}$
- self.cle = CLE
- self.ale = ALE
- self.wpn =  $\overline{WP}$

**The private method `_readFlashData()`:**

This method changes the direction of the Nand-Flash databus (called `self.databus`) to input. Afterwards it sets the ReadEnable-Signal (called `self.ren`). Now the data on the bus will be read out and then the direction will be changed back to output (default).

Return value = read databus value (as binary string, LSB first)

**The private method `_sendCmdToFlash()`:**

This method writes the given parameter `cmd` to the Nand-Flash databus (`self.databus`).

**The private method `_sendAddrToFlash()`:**

This method writes the given parameter `addr` to the Nand-Flash databus (`self.databus`).

**The private method `_sendProgramData()`:**

This method writes the given parameter `data` to the Nand-Flash databus (`self.databus`). After the data has been written to the databus the Write-Enable Signal will be set.

**The private method `_setAddrSignals()`:**

It first sets the Write-Enable Signal to 1. This is important that more than one Address will be sent correctly to the Nand-Flash device. Afterwards `self.ale = 1`, `self.cle = 0` and `self.wen = 0` will be set (see figure 2).

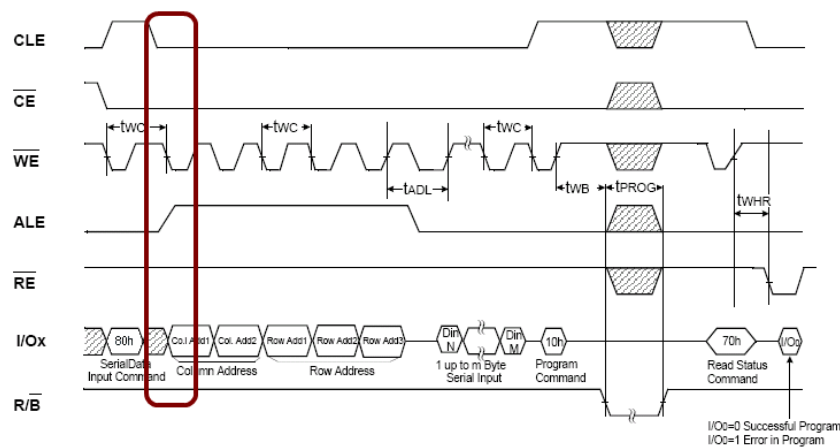


Figure 2: Write an address to a Samsung Flash [3]

**The private method \_setSendCmdSigs():**

Works like the \_setAddrSignals() method (see figure 3).

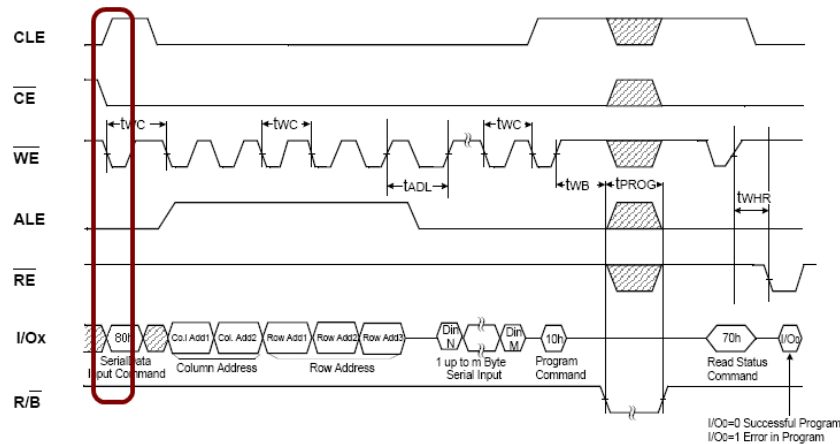


Figure 3: Write a command to a Samsung Flash [3]

**The private method \_sendColRowAddrToFlash():**

This method writes the needed Column and Row Addresses to the Nand-Flash using the \_setAddrSignals() and \_sendAddrToFlash() for sending the row addresses defined in the row\_dict (python dictionary) and the col\_dict (python dictionary).

First all column addresses and then all row addresses will be send. If there is an address which should not be sent to the Nand-Flash enter "no" in the corresponding dictionary position.

An example for a possible row\_dict: [{"0",row1}, {"1",row2}, {"2",row3+"000000"}]

An example for a possible col\_dict: [{"0","no"}, {"1","no"}]

The signals, cmd, address and data will be set from this method as shown in figure 4.

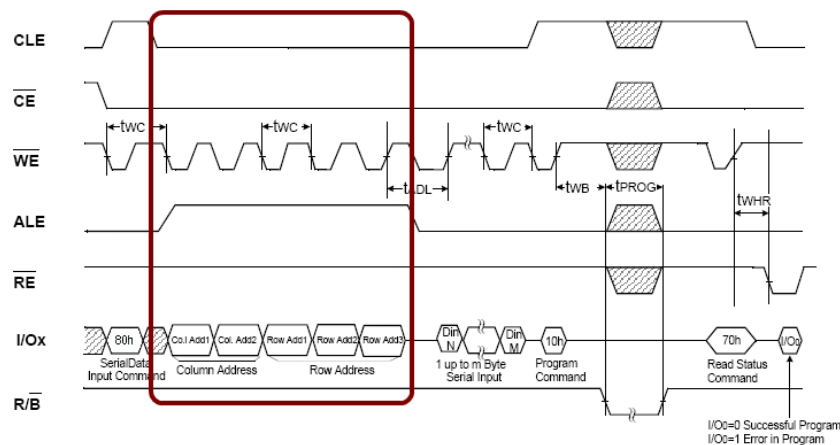


Figure 4: Write a column and row addresses to a Samsung Flash [3]

**The private method `__readStatus()`:**

This method reads out the status information of the Nand-Flash.

`status_cmd` is the command which will be sent to the Nand-Flash. This class attribute must be set in the specific Samsung Nand Flash file. For example see AN-1-003 [2].

The signals, `cmd` and `address` will be set from this method as shown in figure 5.

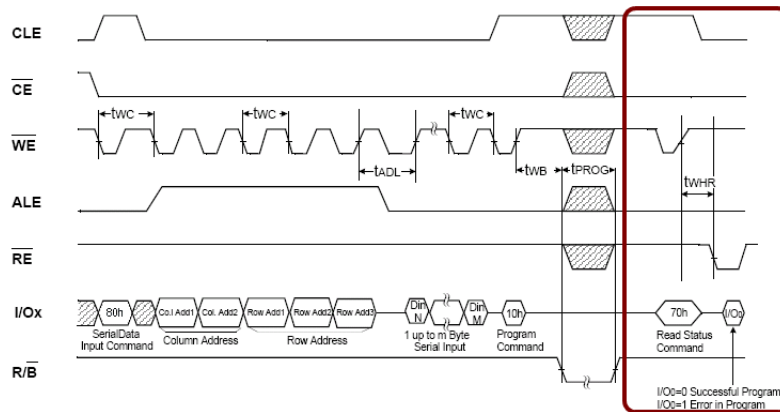


Figure 5: Read out the status information of a Samsung Flash [2]

**The private method `__readIdCode()`:**

The parameter `cycle_dict` is a python dictionary where the number of cycles and their names must be defined. (like `[("0", "Master Code"), ("1", "Device Code"), ("2", "3rd Cycle"), ("3", "4th Cycle"), ("4", "5th Cycle")]`)

`id_cmd` (specific Nand-Flash class attribute) is the command which will be sent to the Nand-Flash.

For reading out the Nand-Flash data the `__readFlashData()`-method is used.

The signals, `cmd` and `address` will be set from this method as shown in figure 6.

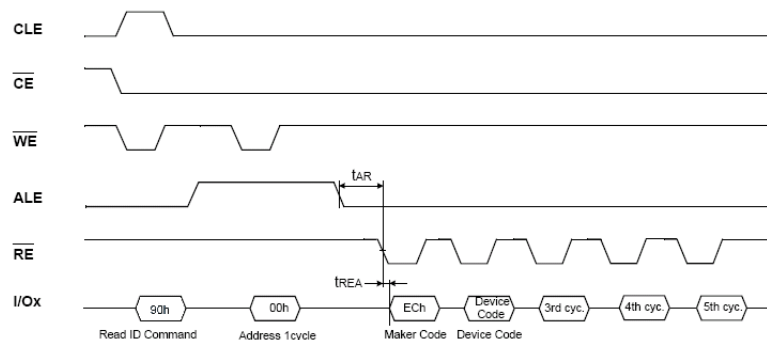


Figure 6: Read out the ID-code of a Samsung Flash [2]

**The private method `__checkOnfi()`:**

It will be checked if a Nand-Flash supports ONFI [4].

**The private method \_resetFlash():**

reset\_cmd (specific Nand-Flash class attribute) is the command which will be sent to the Nand-Flash.

The signals and cmd will be set from this method as shown in figure 7.

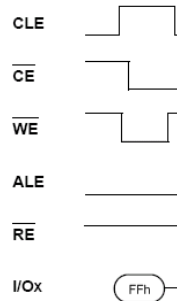


Figure 7: Reset a Samsung Flash [2]

**The private method \_programFlash():**

The paramter write\_dict must be filled with the data, which should be written into the Nand-Flash.

(like [("0","00000000"),("1","01010101"),("2","10101010"),("3","11111111"),("4","00001111")])

The parameters row\_dict and col\_dict must be filled as described in the method \_sendColRowAddrToFlash() (see page 3)

start\_cmd (specific Nand-Flash class attribute) is the command which will be sent to the Nand-Flash.

stop\_cmd (specific Nand-Flash class attribute) is the command which will be sent to the Nand-Flash.

The signals, cmd, address and data will be set from this method as shown in figure 8.

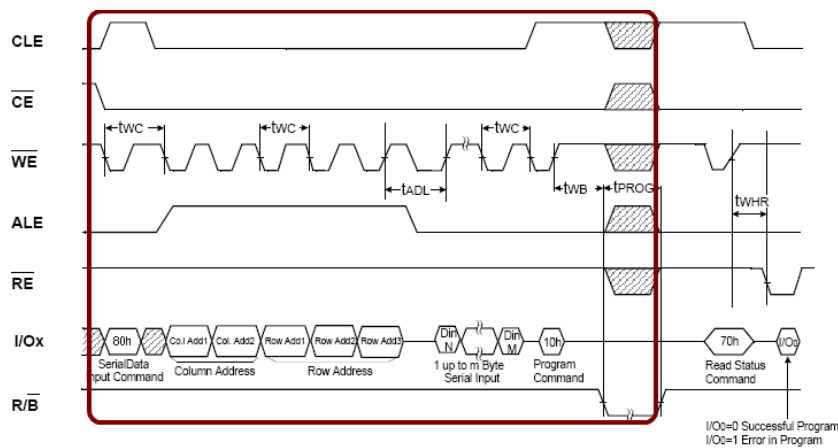


Figure 8: Write data into a Samsung Flash [2]

**The private method `_blockEraseFlash()`:**

This method is used for erasing one page of a Samsung Nand-Flash

Therefore only the row\_dict is important.

start\_cmd (specific Nand-Flash class attribute) is the command which will be sent to the Nand-Flash.

stop\_cmd (specific Nand-Flash class attribute) is the command which will be sent to the Nand-Flash.

The signals, cmd and rowaddress will be set from this method as shown in figure 9.

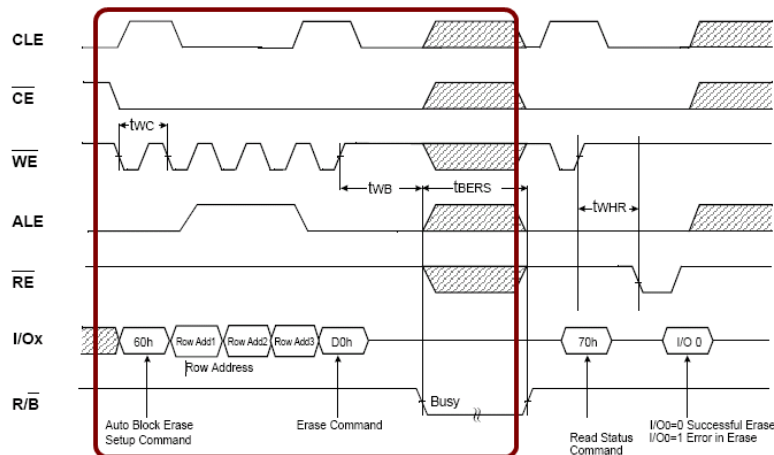


Figure 9: Erase one page of a Samsung Flash [2]

**The private method `_checkBusy()`:**

This method checks if the Nand-Flash is busy,

**The private method `_readFromFlash()`:**

start\_cmd (specific Nand-Flash class attribute) is the command which will be sent to the Nand-Flash.

stop\_cmd (specific Nand-Flash class attribute) is the command which will be sent to the Nand-Flash.

The parameter bytes defines the number of datapakets which should be read out of the Nand-Flash

The signals, cmd and rowaddress will be set from this method as shown in figure 10.

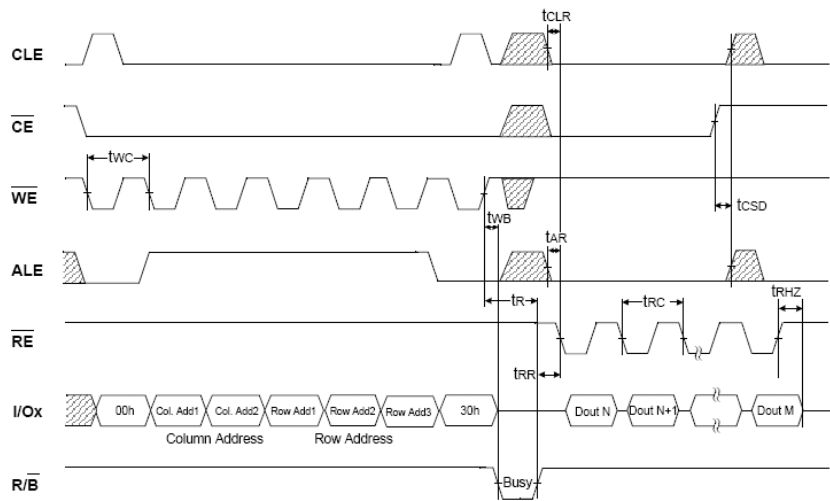


Figure 10: Read data out of a Samsung Flash [2]

## **Revision History**

Version 1.0:            Initial Version

## **Bibliography**

- [1] Gleichmann Electronics Research. *Hpe<sup>®</sup>\_JTAG* basis class and basic functions. <http://www.ge-research.com/downloads.html>, April 2009.
- [2] Gleichmann Electronics Research. *Hpe<sup>®</sup>\_JTAG* script for the samsung nand flash k9f8g08u0m. <http://www.ge-research.com/downloads.html>, April 2009.
- [3] Samsung. Nand-flash k9f8g08u0m. [http://www.samsung.com/global/system/business/semiconductor/product/2007/6/11/NANDFlash/SLC\\_LargeBlock/8Gbit/K9F8G08U0M/ds\\_k9f8g08x0m\\_rev10.pdf](http://www.samsung.com/global/system/business/semiconductor/product/2007/6/11/NANDFlash/SLC_LargeBlock/8Gbit/K9F8G08U0M/ds_k9f8g08x0m_rev10.pdf), March 2007.
- [4] ONFI Industry Workgroup. Open nand flash interface. <http://www.onfi.org>, 2006.

## Disclaimer

**Disclaimer.** The information in this document is provided in connection with Gleichmann Electronics Research (GE Research) products. GE Research makes no representations or warranties with respect to the accuracy or completeness of the document and reserves the right to make changes to specification and product description at any time without notice. GE Research does not accept any liability to the accuracy of the design completed by the developer, whether original design, copied from schematics and/or application notes, supplied by GE Research and/or otherwise. The final circuit integrity and testing is solely the responsibility of the design authority. GE Research accepts no responsibility and/or liability.

This document includes links to both internal (GE Research) and external (non-GE Research) websites to support the users of GE Research products. External links are selected and reviewed whenever a new set of CD/DVD or manuals is published. However, GE Research is not responsible for the content of external websites. Furthermore, GE Research is not responsible for changes made to the products and datasheets that are used in the GE Research product.

If you have any improvements and/or suggestions concerning this manual, please feel free to let us know.

## Copyright Notice

This document is copyrighted, 2009, by Gleichmann Electronics Research (Austria) GmbH & Co KG. All rights are reserved. Gleichmann Electronics Research (Austria) GmbH & Co KG reserves the right to make improvements to the products described in this manual at any time without notice. No part of this manual may be reproduced, copied, translated or transmitted in any form or by any means without the prior written permission of Gleichmann Electronics Research (Austria) GmbH & Co KG. Information provided in this manual is intended to be accurate and reliable. However, Gleichmann Electronics Research (Austria) GmbH & Co KG assumes no responsibility for its use, nor for any infringements upon the rights of third parties which may result from its use.