

## Executing commands of the class *K9F8G08U0M*

This application note shows how a possible execution script for the class *K9F8G08U0M* (see *AN-1-003* [1]) may look like.

### Main parts which should be included in an execution script

#### Import of the needed scripting modules

The module *HpeJTAG* contains basic functions for general use and the class *SamsungNand\_basis* and it's methods

The module *K9F8G08* contains the class *K9F8G08U0M* and it's methods

```
import JtagExampleLib.HpeJTAG ← module
import JtagExampleLib.SamsungNands.K9F8G08
      ↑           ↑
      directory  module
```

Figure 1: Import of scripting modules

More information about the *jtag* methods can be found in the *Hpe®\_JTAG* manual [2].

#### Reloading needed scripting modules

The reloading of modules which contain scripting classes is a very usefull feature.

All in this execution python file used modules will be reloaded as shown in figure 2.

This has the advantage that the *Hpe®\_JTAG* always uses the newest class definitions for new created scripting class objects

```
try:
    reload(JtagExampleLib.HpeJTAG)
    reload(JtagExampleLib.SamsungNands.SamsungNand)
    reload(JtagExampleLib.SamsungNands.K9F8G08)
except NameError:
    pass
```

Figure 2: Reload scripting modules

## Initialization of the JTAG-Chain

```

board = "ALC-AS3"

# avoid that the JTAG gui is used while the script is running.
#jtag.lockGUI(True)
#initialize the JTAG interface

try:
    boardConfig
except:
    if board == "ALC-AS3":
        jtag.initialize("ALTERA_INTERFACE", "EPM1270F256, EP3SL50F780, EP3SL50F780")
    else:
        apputil.Msg("Unknown board: %s" % board)

# get signal ids for sample and extest
JTAG_DEVICE_IDX = 1
boardConfig = jtag.getBoardConfiguration()
devConfig = boardConfig.getDeviceConfigurations()[JTAG_DEVICE_IDX]

apputil.Msg("Prepared for board: " + board)

```

Figure 3: Initialization of the JTAG-Chain

## Creation of an object from the class *K9F8G08U0M* and the later on pin assignment

flash0 is the created object of the class *K9F8G08U0M*.

The pin assignment is done with the method registerPinInfos()

The strings "cen, ren, wen, cle, ale, wpn and rbn" are variables of the class *K9F8G08U0M* to which the actual pinning on the FPGA will be assigned to.

```

flash0 = JtagExampleLib.SamsungNands.K9F8G08.K9F8G08U0M(1,jtag) ← creating an object of the class K9F8G08U0M
apputil.Msg("\nTesting Flash 1\n")
if board == "ALC-AS3":
    devConfig.registerPinInfos([("AE23", "ale", "OUT"),
                                ("AE24", "AF24", "AH22", "AE21", "AF21", "AE20", "AF20", "AH25"), "databus[0:7]", "OUT"),
                                ("AF23", "cle", "OUT"),
                                ("AC28", "ren", "OUT"),
                                ("AH23", "wen", "OUT"),
                                ("AH24", "wpn", "OUT"),
                                ("AD27", "cen", "OUT"),
                                ("AD24", "rbn", "IN")],
                                flash0) ← actual pin assignment
ExecNandFunctions(jtag, flash0, [{"0", [{"0", JtagExampleLib.HpeJTAG.Denary2Binary(1)},
{"1", JtagExampleLib.HpeJTAG.Denary2Binary(5)},
{"2", JtagExampleLib.HpeJTAG.Denary2Binary(10)}]},
{"1", [{"0", JtagExampleLib.HpeJTAG.Denary2Binary(2)},
{"1", JtagExampleLib.HpeJTAG.Denary2Binary(4)},
{"2", JtagExampleLib.HpeJTAG.Denary2Binary(8)},
{"3", JtagExampleLib.HpeJTAG.Denary2Binary(16)},
{"4", JtagExampleLib.HpeJTAG.Denary2Binary(32)},
{"5", JtagExampleLib.HpeJTAG.Denary2Binary(64)},
{"6", JtagExampleLib.HpeJTAG.Denary2Binary(128)}]},
{"2", [{"0, JtagExampleLib.HpeJTAG.Denary2Binary(144)}]})

```

Figure 4: Import of scripting modules

The *execute\_K9F8G08U0M* python-file can be found in the same directory as the module *K9F8G08* python-file.

The execution and the results are shown in AN-1-003 [1].

## **Revision History**

Version 1.0:            Initial Version

## **Bibliography**

- [1] Gleichmann Electronics Research. *AN-1-003: K9F8G08U0M* class description. <http://www.ge-research.com/downloads.html>, March 2007.
- [2] Gleichmann Electronics Research. *Hpe<sup>®</sup>\_JTAG* manual. <http://www.ge-research.com/downloads.html>, March 2009.

## 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.