
AT07911: Glue Logic Controller (GLOC)

ASF PROGRAMMERS MANUAL

Glue Logic Controller (GLOC)

Driver for the Glue Logic Controller. This driver provides access to the main features of the Glue Logic controller.

The outline of this documentation is as follows:

- [Prerequisites](#)
- [Module Overview](#)
- [Special Considerations](#)
- [Extra Information](#)
- [Examples](#)
- [API Overview](#)

Table of Contents

Glue Logic Controller (GLOC)	1
Software License	3
1. Prerequisites	4
2. Module Overview	5
3. Special Considerations	6
4. Extra Information	7
5. Examples	8
6. API Overview	9
6.1. Structure Definitions	9
6.1.1. Struct gloc_dev_inst	9
6.1.2. Struct gloc_lut_config	9
6.2. Function Definitions	9
6.2.1. Function gloc_disable()	9
6.2.2. Function gloc_enable()	9
6.2.3. Function gloc_init()	10
6.2.4. Function gloc_lut_get_config_defaults()	10
6.2.5. Function gloc_lut_set_config()	10
7. Extra Information for GLOC	11
7.1. Acronyms	11
7.2. Dependencies	11
7.3. Errata	11
7.4. Module History	11
8. Examples for GLOC	12
8.1. Quick Start Guide for the GLOC driver	12
8.1.1. Use Cases	12
8.1.2. GLOC Basic Usage	12
8.1.3. Setup Steps	12
8.1.4. Usage Steps	13
Index	14
Document Revision History	15

Software License

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of Atmel may not be used to endorse or promote products derived from this software without specific prior written permission.
4. This software may only be redistributed and used in connection with an Atmel microcontroller product.

THIS SOFTWARE IS PROVIDED BY ATMEL "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT ARE EXPRESSLY AND SPECIFICALLY DISCLAIMED. IN NO EVENT SHALL ATMEL BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1. Prerequisites

- General Purpose I/O (GPIO) driver
- Power Management Controller (PMC) driver

2. Module Overview

The Glue Logic Controller (GLOC) contains programmable logic which can be connected to the device pins. This allows the user to eliminate logic gates for simple glue logic functions on the PCB.

The GLOC consists of a number of lookup table (LUT) units. Each LUT unit can generate an output as a user programmable logic expression with four inputs. Inputs can be individually masked. The output can be combinatorially generated from the inputs, or filtered to remove spikes.

3. Special Considerations

in order to use this peripheral, other functions on the device also need to be configured.

- The pins used for interfacing the GLOC may be multiplexed with I/O Controller lines. The programmer must first program the I/O Controller to assign the desired GLOC pins to their peripheral function. If I/O lines of the GLOC are not used by the application, they can be used for other purposes by the I/O Controller. It is only required to enable the GLOC inputs and outputs in use. Pullups for pins configured to be used by the GLOC will be disabled.
- The clock for the GLOC bus interface (CLK_GLOC) is generated by the Power Management Controller (PMC). This clock is disabled at reset, and can be enabled in the PMC. It is recommended to disable the GLOC before disabling the clock, to avoid freezing the module in an undefined state. Additionally, the GLOC depends on a dedicated Generic Clock (GCLK). The GCLK can be set to a wide range of frequencies and clock sources, and must be enabled by the System Control Interface (SCIF) before the GLOC filter can be used.
- When an external debugger forces the CPU into debug mode, the GLOC continues normal operation.

4. Extra Information

For extra information, see [Extra Information for GLOC](#). This includes:

- [Acronyms](#)
- [Dependencies](#)
- [Errata](#)
- [Module History](#)

5. Examples

For a list of examples related to this driver, see [Examples for GLOC](#).

6. API Overview

6.1 Structure Definitions

6.1.1 Struct gloc_dev_inst

Device instance structure for a Glue Logic Controller driver instance. This structure should be initialized by the [gloc_init\(\)](#) function to associate the instance with a particular hardware module of the device.

Table 6-1. Members

Type	Name	Description
Gloc *	hw_dev	Base address of the GLOC module

6.1.2 Struct gloc_lut_config

Configuration structure for a GLOC LUT instance. This structure could be initialized by the [gloc_lut_get_config_defaults\(\)](#) function before being modified by the user application.

Table 6-2. Members

Type	Name	Description
bool	filter	True for enable, false for disable
uint8_t	input_mask	Input enable mask
uint16_t	truth_table_value	Truth table value

6.2 Function Definitions

6.2.1 Function gloc_disable()

Disable the GLOC module.

```
void gloc_disable(  
    struct gloc_dev_inst *const dev_inst)
```

Table 6-3. Parameters

Data direction	Parameter name	Description
[in]	dev_inst	Device structure pointer

6.2.2 Function gloc_enable()

Enable the GLOC module.

```
void gloc_enable(  
    struct gloc_dev_inst *const dev_inst)
```

Table 6-4. Parameters

Data direction	Parameter name	Description
[in]	dev_inst	Device structure pointer

6.2.3 Function gloc_init()

Initialize the GLOC module.

```
void gloc_init(  
    struct gloc_dev_inst *const dev_inst,  
    Gloc *const gloc)
```

Table 6-5. Parameters

Data direction	Parameter name	Description
[in]	dev_inst	Device structure pointer
[in]	gloc	Base address of the GLOC instance

6.2.4 Function gloc_lut_get_config_defaults()

Get the default configuration for lookup table (LUT) unit of GLOC.

```
void gloc_lut_get_config_defaults(  
    struct gloc_lut_config *const config)
```

The default configuration is as follows:

- Filter: enable
- Input mask: 0xF, all four inputs are enabled
- Truth table value: 0x00

Table 6-6. Parameters

Data direction	Parameter name	Description
[in, out]	config	Pointer to GLOC LUT configuration

6.2.5 Function gloc_lut_set_config()

Configure a lookup table (LUT) unit of GLOC.

```
void gloc_lut_set_config(  
    struct gloc_dev_inst *const dev_inst,  
    uint32_t lut_id,  
    struct gloc_lut_config *const config)
```

Table 6-7. Parameters

Data direction	Parameter name	Description
[in]	dev_inst	Device structure pointer
[in]	lut_id	LUT ID
[in]	config	Pointer to GLOC LUT configuration

7. Extra Information for GLOC

7.1 Acronyms

Below is a table listing the acronyms used in this module, along with their intended meanings.

Acronym	Definition
LUT	Look Up Table
QSG	Quick Start Guide

7.2 Dependencies

This driver has the following dependencies:

- None

7.3 Errata

There are no errata related to this driver.

7.4 Module History

An overview of the module history is presented in the table below, with details on the enhancements and fixes made to the module since its first release. The current version of this corresponds to the newest version in the table.

Changelog
Initial document release

8. Examples for GLOC

This is a list of the available Quick Start guides (QSGs) and example applications for [Blue Logic Controller \(GLOC\)](#). QSGs are simple examples with step-by-step instructions to configure and use this driver in a selection of use cases. Note that QSGs can be compiled as a standalone application or be added to the user application.

- [Quick Start Guide for the GLOC driver](#)

8.1 Quick Start Guide for the GLOC driver

This is the quick start guide for the [Blue Logic Controller \(GLOC\)](#), with step-by-step instructions on how to configure and use the driver for a specific use case. The code examples can be copied into e.g. the main application loop or any other function that will need to control the GLOC module.

8.1.1 Use Cases

- [GLOC Basic Usage](#)

8.1.2 GLOC Basic Usage

This use case will demonstrate how to initialize the GLOC module to match the truth table for simple glue logic functions.

8.1.3 Setup Steps

8.1.3.1 Prerequisites

This module requires the following service

- Clock Service

8.1.3.2 Setup

- The selected GLOC input/output pins are PIN_PA20 (GLOC_IN0), PIN_PA21 (GLOC_IN1), PIN_PA22 (GLOC_IN2), PIN_PA07 (GLOC_IN3), and PIN_PA24 (GLOC_OUT0).
- GPIO pins used to set/clear GLOC input pins are PIN_PC01 (connected to GLOC_IN0), PIN_PC07 (connected to GLOC_IN1), PIN_PC08 (connected to GLOC_IN2), and PIN_PC13 (connected to GLOC_IN3).
- GPIO pin used to read GLOC output pin is PA24 (connected to GLOC_OUT0)

Add this to the main loop or a setup function:

```
#define XOR_TRUTH_TABLE_FOUR_INPUT    0x6996u
#define XOR_LUT 0

// Pin settings for GLOC

#define GLOC_IN0 PIN_PC01
#define GLOC_IN1 PIN_PC07
#define GLOC_IN2 PIN_PC08
#define GLOC_IN3 PIN_PC13

// Output from GLOC
#define GLOC_OUT0 PIN_PA24
```

The four configured inputs act as an index into or bit number of XOR_TRUTH_TABLE_FOUR_INPUT. In this way any combinatorial logic function can be simply realised.

```
struct gloc_dev_inst dev_inst;
struct gloc_lut_config lut_config;
```

```
gloc_init(&dev_inst, GLOC);
gloc_enable(&dev_inst);
gloc_lut_get_config_defaults(&lut_config);
lut_config.truth_table_value = XOR_TRUTH_TABLE_FOUR_INPUT;
gloc_lut_set_config(&dev_inst, XOR_LUT, &lut_config);
```

8.1.3.3 Workflow

1. Initialize the GLOC module.

```
gloc_init(&dev_inst, GLOC);
```

2. Enable the GLOC module.

```
gloc_enable(&dev_inst);
```

3. Configure 4 inputs XOR truth table value in LUT0.

```
gloc_lut_get_config_defaults(&lut_config);
lut_config.truth_table_value = XOR_TRUTH_TABLE_FOUR_INPUT;
gloc_lut_set_config(&dev_inst, 0, &lut_config);
```

8.1.4 Usage Steps

The pin OUT0 will output according to the settings in lookup table 0 (LUT0) when pins IN0 to IN3 change.

```
ioport_set_pin_level(GLOC_IN0, HIGH);
ioport_set_pin_level(GLOC_IN1, LOW);
ioport_set_pin_level(GLOC_IN2, HIGH);
ioport_set_pin_level(GLOC_IN3, HIGH);

delay_us(50); // Ensure a minimum propagation delay to read the port pin.

out = ioport_get_pin_level(GLOC_OUT0);
```

Index

F

Function Definitions

- `gloc_disable`, [9](#)
- `gloc_enable`, [9](#)
- `gloc_init`, [10](#)
- `gloc_lut_get_config_defaults`, [10](#)
- `gloc_lut_set_config`, [10](#)

S

Structure Definitions

- `gloc_dev_inst`, [9](#)
- `gloc_lut_config`, [9](#)

Document Revision History

Doc. Rev.	Date	Comments
42317A	05/2014	Initial document release



Atmel Corporation 1600 Technology Drive, San Jose, CA 95110 USA

T: (+1)(408) 441.0311

F: (+1)(408) 436.4200

| www.atmel.com

© 2014 Atmel Corporation. All rights reserved. / Rev.: 42317A-MCU-05/2014

Atmel®, Atmel logo and combinations thereof, Enabling Unlimited Possibilities®, and others are registered trademarks or trademarks of Atmel Corporation or its subsidiaries. Other terms and product names may be trademarks of others.

Disclaimer: The information in this document is provided in connection with Atmel products. No license, express or implied, by estoppel or otherwise, to any intellectual property right is granted by this document or in connection with the sale of Atmel products. EXCEPT AS SET FORTH IN THE ATMEL TERMS AND CONDITIONS OF SALES LOCATED ON THE ATMEL WEBSITE, ATMEL ASSUMES NO LIABILITY WHATSOEVER AND DISCLAIMS ANY EXPRESS, IMPLIED OR STATUTORY WARRANTY RELATING TO ITS PRODUCTS INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. IN NO EVENT SHALL ATMEL BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, PUNITIVE, SPECIAL OR INCIDENTAL DAMAGES (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS AND PROFITS, BUSINESS INTERRUPTION, OR LOSS OF INFORMATION) ARISING OUT OF THE USE OR INABILITY TO USE THIS DOCUMENT, EVEN IF ATMEL HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Atmel makes no representations or warranties with respect to the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and products descriptions at any time without notice. Atmel does not make any commitment to update the information contained herein. Unless specifically provided otherwise, Atmel products are not suitable for, and shall not be used in, automotive applications. Atmel products are not intended, authorized, or warranted for use as components in applications intended to support or sustain life.