GigaDevice Semiconductor Inc.

Arm® Cortex®-M3/4/23/33 32-bit MCU

Application Note
AN013

Table of Contents

List of Figures

List of Table

1. Introduction

The configurable logic array provides 256 programmable digital logic operations for external pins, ADC and timers without intervention from the CPU. Eight independent CLA units are implemented in this module. Each CLA unit supports configurable asynchronous or asynchronous output for GPIO pins. This paper presents a design scheme of using CLA to build square wave filter.

2. CLA square wave filter scheme

2.1. CLA square wave filter principle

The square wave filter architecture constructed by CLA is shown in ***Figure 2-1. CLA square wave*** **filter architecture**. ClA0 ~ ClA4 and CLA6 are used as filtering units to realize filtering function through logical combination among CLAs; CLA5 is used as trigger signal converter of filtering unit, CLA5 can convert TIMER2_TRGO signal to CLA5_ASYNC_OUT, CLA5_ASYNC_OUT is used as synchronous trigger clock source of filtering unit, and only CLA3 use HCLK as trigger clock source.

The user can set the filter width to filter out the maximum burr by configuring the TIMER2_TRGO.

Figure 2-1. CLA square wave filter architecture

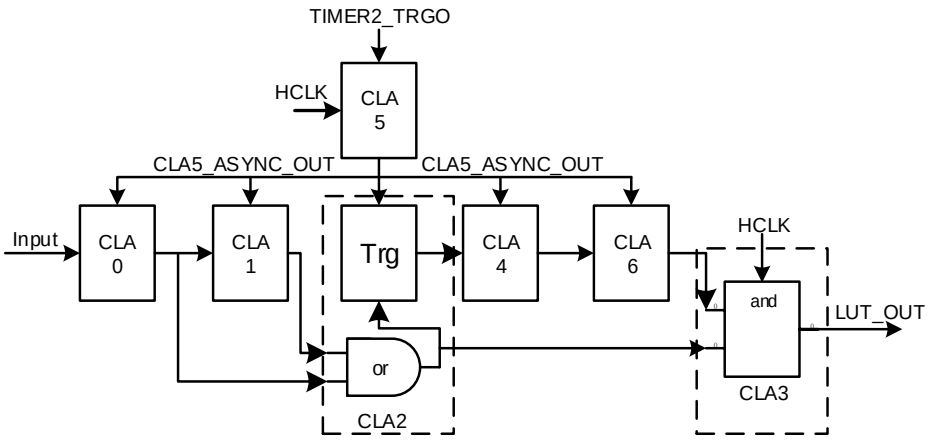
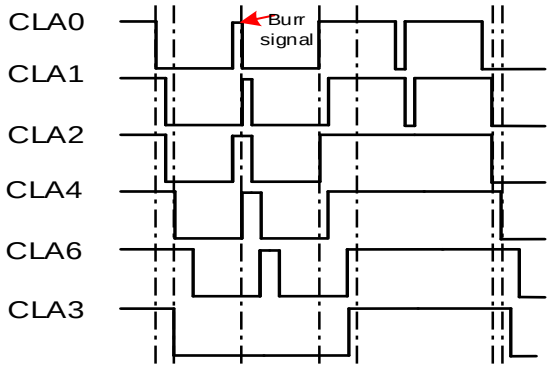


Figure 2-2. The principle of CLA square **wave filter** shows the principle of filtering square wave signal by CLA square wave filter.

Figure 2-2. The principle of CLA square wave filter



2.1. CLA software configuration

Square wave filter sets each CLA as a specific logic, and then combines them to achieve the purpose of filtering. The detailed configuration of CLA0 ~ CLA6 is as ***Table 2-1. CLA configuration in GD project***.

Table 2-1. CLA configuration in GD project

```
/*!
    \brief        configure the CLA interface
    \param[in]    none
    \param[out] none
    \retval       none
*/
void cla_config(void)
{
    /* configure the CLA0 */
    /* select CLA5_ASYNC_OUT as input of MUX0 */
    cla_multiplexer_input_config(CLA0,MUX0,CLA0MUX0_CLA5_ASYNC_OUT);
    /* select CLAIN11(PA8) as input of MUX1 */
    cla_multiplexer_input_config(CLA0,MUX1,CLA0MUX1_CLAIN11);
    /* IN1 as CLA0 output */
    cla_lut_control_config(CLA0,0xCC);
    /* select flip-flop result as CLA output */
    cla_output_config(CLA0,FLIP_FLOP_OUTPUT);
    cla_flip_flop_clocksource_config(CLA0,MUX0_OUTPUT);
    cla_flip_flop_clockpolarity_config(CLA0,CLA_CLOCKPOLARITY_POSEDGE);
    cla_flip_flop_output_reset(CLA0);

    /* configure the CLA1 */
    /* select CLA5_ASYNC_OUT as input of MUX0 */
    cla_multiplexer_input_config(CLA1,MUX0,CLA1MUX0_CLA5_ASYNC_OUT);
    /* select CLA0_ASYNC_OUT as input of MUX1 */
    cla_multiplexer_input_config(CLA1,MUX1,CLA1MUX1_CLA0_ASYNC_OUT);
    /* IN0 as CLA1 output */
    cla_lut_control_config(CLA1,0xCC);
    /* select flip-flop result as CLA output */
    cla_output_config(CLA1,FLIP_FLOP_OUTPUT);
    cla_flip_flop_clocksource_config(CLA1,MUX0_OUTPUT);
    cla_flip_flop_clockpolarity_config(CLA1,CLA_CLOCKPOLARITY_POSEDGE);
    cla_flip_flop_output_reset(CLA1);

    /* configure the CLA2 */
    /* select CLA0_ASYNC_OUT as input of MUX0 */
```

```
cla_multiplexer_input_config(CLA2,MUX0,CLA2MUX0_CLA0_ASYNC_OUT);
/* select CLA1_ASYNC_OUT as input of MUX0 */
cla_multiplexer_input_config(CLA2,MUX1,CLA2MUX1_CLA1_ASYNC_OUT);
/* IN0|IN1 */
cla_lut_control_config(CLA2,0xFC);
/* select flip-flop result as CLA output */
cla_output_config(CLA2,FLIP_FLOP_OUTPUT);
cla_flip_flop_clocksource_config(CLA2,TIMER_TRGO);
cla_flip_flop_clockpolarity_config(CLA2,CLA_CLOCKPOLARITY_POSEDGE);
cla_flip_flop_output_reset(CLA2);
cla_output_enable(CLA2);

/* configure the CLA3 */
/* select CLA5_ASYNC_OUT as input of MUX0 */
cla_multiplexer_input_config(CLA3,MUX0,CLA3MUX0_CLA5_ASYNC_OUT);
/* select CLA4_ASYNC_OUT as input of MUX1 */
cla_multiplexer_input_config(CLA3,MUX1,CLA3MUX1_CLA6_ASYNC_OUT);
/* IN0|IN3 */
cla_lut_control_config(CLA3,0x88);
/* select LUT result as CLA output */
cla_output_config(CLA3,LUT_RESULT);
cla_output_enable(CLA3);

/* configure the CLA4 */
/* select CLA5_ASYNC_OUT as input of MUX0 */
cla_multiplexer_input_config(CLA4,MUX0,CLA4MUX0_CLA5_ASYNC_OUT);
/* select CLA2_ASYNC_OUT as input of MUX1 */
cla_multiplexer_input_config(CLA4,MUX1,CLA4MUX1_CLA2_ASYNC_OUT);
/* IN0 */
cla_lut_control_config(CLA4,0xCC);
/* select flip-flop result as CLA output */
cla_output_config(CLA4,FLIP_FLOP_OUTPUT);
cla_flip_flop_clocksource_config(CLA4,MUX0_OUTPUT);
cla_flip_flop_clockpolarity_config(CLA4,CLA_CLOCKPOLARITY_POSEDGE);
cla_flip_flop_output_reset(CLA4);
cla_output_enable(CLA4);

/* configure the CLA5 */
/* select TIMER2_TRGO as input of MUX0 */
cla_multiplexer_input_config(CLA5,MUX0,CLA5MUX0_TIMER2_TRGO);
/* IN0 */
cla_lut_control_config(CLA5,0xF0);
/* select LUT result as CLA output */
```

```
    cla_output_config(CLA5,LUT_RESULT);

    /* configure the CLA6 */
    /* select CLA5_ASYNC_OUT as input of MUX0 */
    cla_multiplexer_input_config(CLA6,MUX0,CLA6MUX0_CLA5_ASYNC_OUT);
    /* select CLA2_ASYNC_OUT as input of MUX1 */
    cla_multiplexer_input_config(CLA6,MUX1,CLA6MUX1_CLA4_ASYNC_OUT);
    /* IN0&IN1 */
    cla_lut_control_config(CLA6,0xCC);
    /* select flip-flop result as CLA output */
    cla_output_config(CLA6,FLIP_FLOP_OUTPUT);
    cla_flip_flop_clocksource_config(CLA6,MUX0_OUTPUT);
    cla_flip_flop_clockpolarity_config(CLA6,CLA_CLOCKPOLARITY_POSEDGE);
   cla_flip_flop_output_reset(CLA6);

    /* enable CLA0~CL6 */
    cla_enable(CLA5);
    cla_enable(CLA0);
    cla_enable(CLA1);
    cla_enable(CLA2);
    cla_enable(CLA4);
    cla_enable(CLA3);
    cla_enable(CLA6);
}
```

As the signal source of CLA5_ASYNC_OUT, TIMER2_TRGO signal adjusts the performance of the square wave filter by modifying the time interval of generating TRGO signal. The specific configuration of TIMER2 is as ***Table 2-2. TIMER2_TRGO configuration in GD project.***

Table 2-2. TIMER2_TRGO configuration in GD project

```
void timer_trgo_config(void)
{
     /* ----------------------------------------------------------------------------
    TIMER Configuration:
    TIMERxCLK = SystemCoreClock/100 = 1MHz.
    TIMER configuration is timing mode, and the timing is 12 microsecond (12/1000000 = 12us).
    ---------------------------------------------------------------------------- */
    timer_oc_parameter_struct timer_ocinitpara;
    timer_parameter_struct timer_initpara;
    /* enable the peripherals clock */
    rcu_periph_clock_enable(RCU_TIMER2);
    /* deinit a TIMER2 */
    timer_deinit(TIMER2);
```

```
    /* initialize TIMER init parameter struct */
    timer_struct_para_init(&timer_initpara);
    /* TIMER2 configuration */
    timer_initpara.prescaler         = 99;
    timer_initpara.alignedmode       = TIMER_COUNTER_EDGE;
    timer_initpara.counterdirection  = TIMER_COUNTER_UP;
    timer_initpara.period            = 12;
    timer_initpara.clockdivision     = TIMER_CKDIV_DIV1;
    timer_init(TIMER2, &timer_initpara);
    timer_master_output_trigger_source_select(TIMER2, TIMER_TRI_OUT_SRC_UPDATE);
    /* enable a TIMER */
    timer_enable(TIMER2);
}
```

3. Square wave filter performance test

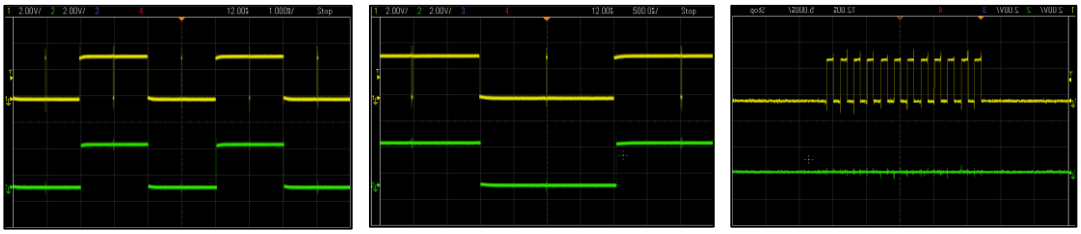
3.1. Test conditions

The test input signal is a square wave with a duty cycle of 4ms and a duty cycle of 50%. Each half cycle has 12 consecutive burrs with a width of 1us.By configuring timer2, the time interval of TIMER2_TRGO signal is set to 12us, and CLA filter can filter out the burr of maximum pulse width to 12us.

3.2. Test results

The performance test results of the square wave filter are shown in ***Figure 3-1. Performance test of CLA square*** **wave filter**, in which the yellow-signal represents the original signal and the blue-signal represents the filtered signal. The picture shows the amplified signal details from left to right. The test shows that 12 burrs with a continuous width of 1us have been filtered out by CLA filter.

Figure 3-1. Performance test of CLA square wave filter



3.3. CLA square wave filter precautions

The following points need to be noted when using CLA square wave filters.

1. The relationship between the trigger interval of TIMER2_TRGO of CLA square wave filter and the burr width of signal. The TIMER2_TRGO signal determines the single maximum burr width that can be filtered out by CLA filter. If the burr signal is continuous, the time interval of TIMER2_TRGO should be greater than the sum of the width of burr signal. If the burr signal cannot be completely filtered out, try to increase the time interval of TIMER2_TRGO.
2. For the square wave signal with discontinuous burr signal, the CLA square wave filter configures TIMER2_TRGO as the maximum time width in the burr signal.
3. CLA7 can be added between CLA6 and CLA3, with the same function configuration as CLA6, which can improve the filtering performance of CLA square wave filter when the burr signal width is large.

4. Revision history

Table 4-1. Revision history

Revision No.	Description	Date
1.0	Initial Release	Dec.13 2021

Important Notice

This document is the property of GigaDevice Semiconductor Inc. and its subsidiaries (the "Company"). This document, including any product of the Company described in this document (the “Product”), is owned by the Company under the intellectual property laws and treaties of the People’s Republic of China and other jurisdictions worldwide. The Company reserves all rights under such laws and treaties and does not grant any license under its patents, copyrights, trademarks, or other intellectual property rights. The names and brands of third party referred thereto (if any) are the property of their respective owner and referred to for identification purposes only.

The Company makes no warranty of any kind, express or implied, with regard to this document or any Product, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The Company does not assume any liability arising out of the application or use of any Product described in this document. Any information provided in this document is provided only for reference purposes. It is the responsibility of the user of this document to properly design, program, and test the functionality and safety of any application made of this information and any resulting product. Except for customized products which has been expressly identified in the applicable agreement, the Products are designed, developed, and/or manufactured for ordinary business, industrial, personal, and/or household applications only. The Products are not designed, intended, or authorized for use as components in systems designed or intended for the operation of weapons, weapons systems, nuclear installations, atomic energy control instruments, combustion control instruments, airplane or spaceship instruments, transportation instruments, traffic signal instruments, life-support devices or systems, other medical devices or systems (including resuscitation equipment and surgical implants), pollution control or hazardous substances management, or other uses where the failure of the device or Product could cause personal injury, death, property or environmental damage ("Unintended Uses"). Customers shall take any and all actions to ensure using and selling the Products in accordance with the applicable laws and regulations. The Company is not liable, in whole or in part, and customers shall and hereby do release the Company as well as it’s suppliers and/or distributors from any claim, damage, or other liability arising from or related to all Unintended Uses of the Products. Customers shall indemnify and hold the Company as well as it’s suppliers and/or distributors harmless from and against all claims, costs, damages, and other liabilities, including claims for personal injury or death, arising from or related to any Unintended Uses of the Products.

Information in this document is provided solely in connection with the Products. The Company reserves the right to make changes, corrections, modifications or improvements to this document and Products and services described herein at any time, without notice.