

To our customers,

Old Company Name in Catalogs and Other Documents

On April 1st, 2010, NEC Electronics Corporation merged with Renesas Technology Corporation, and Renesas Electronics Corporation took over all the business of both companies. Therefore, although the old company name remains in this document, it is a valid Renesas Electronics document. We appreciate your understanding.

Renesas Electronics website: <http://www.renesas.com>

April 1st, 2010
Renesas Electronics Corporation

Issued by: Renesas Electronics Corporation (<http://www.renesas.com>)

Send any inquiries to <http://www.renesas.com/inquiry>.

RENESAS TECHNICAL UPDATE

Nippon Bldg., 2-6-2, Ohte-machi, Chiyoda-ku, Tokyo 100-0004, Japan
Renesas Technology Corp.

Product Category	MPU & MCU	Document No.	TN-16C-A180A/E	Rev.	1.00
Title	Note on Generating STOP Condition in Multi-master I ² C-bus Interface for R32C/116 Group, R32C/117 Group, and R32C/118 Group		Information Category	Technical Notification	
Applicable Products	R32C/116 Group R32C/117 Group R32C/118 Group	Lot No. Contact a Renesas sales office	Reference Document		

This document describes a note on generating a STOP condition in the multi-master I²C-bus interface.

1. Note

In the multi-master I²C-bus interface, when the slave device and/or other master devices drive the MSCL line low, no normal STOP condition is generated. This is because the MSDA line is released while the MSCL line is still driven low.

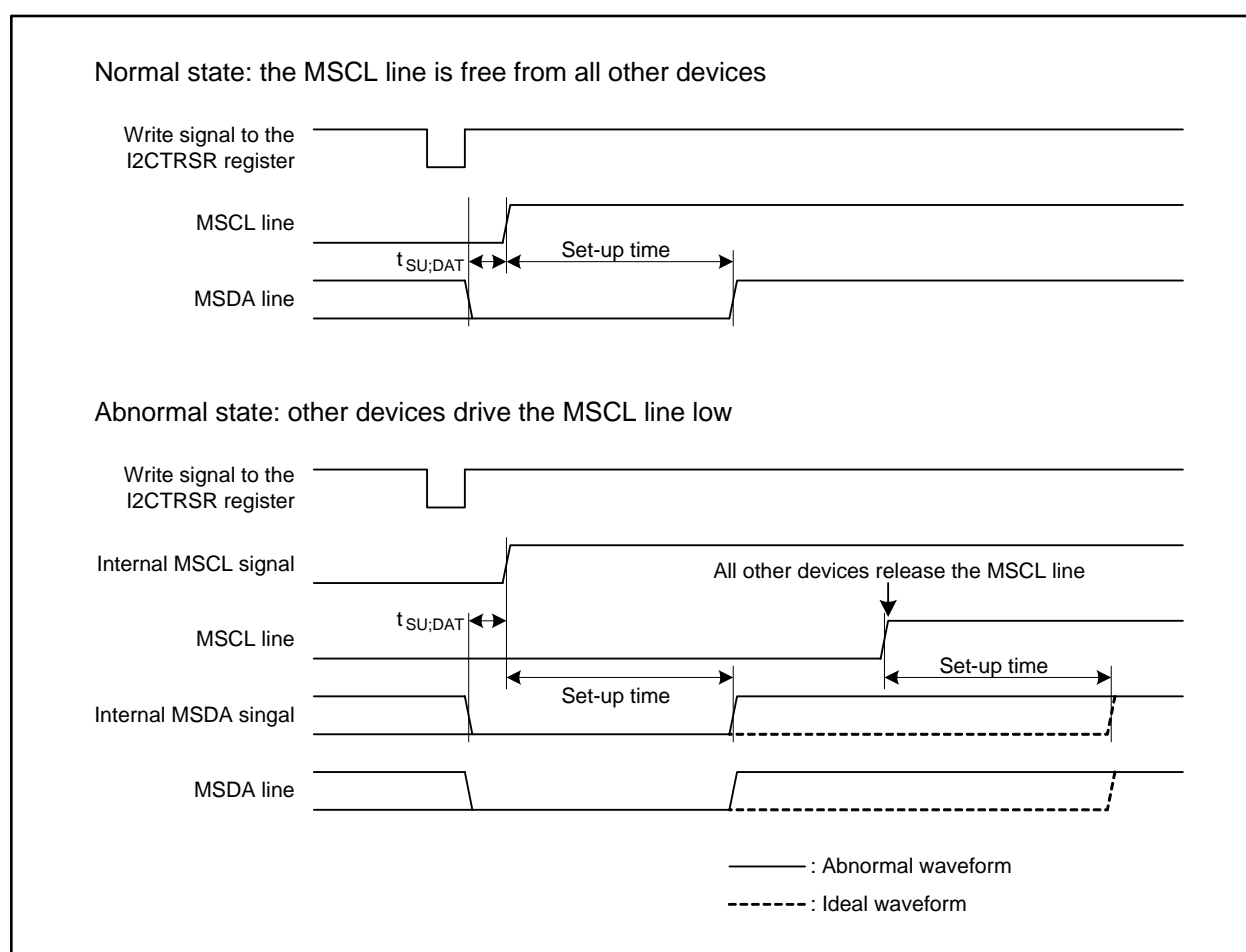


Figure 1. Abnormal Waveform

2. Solution

Use the following flowchart to generate a STOP condition.

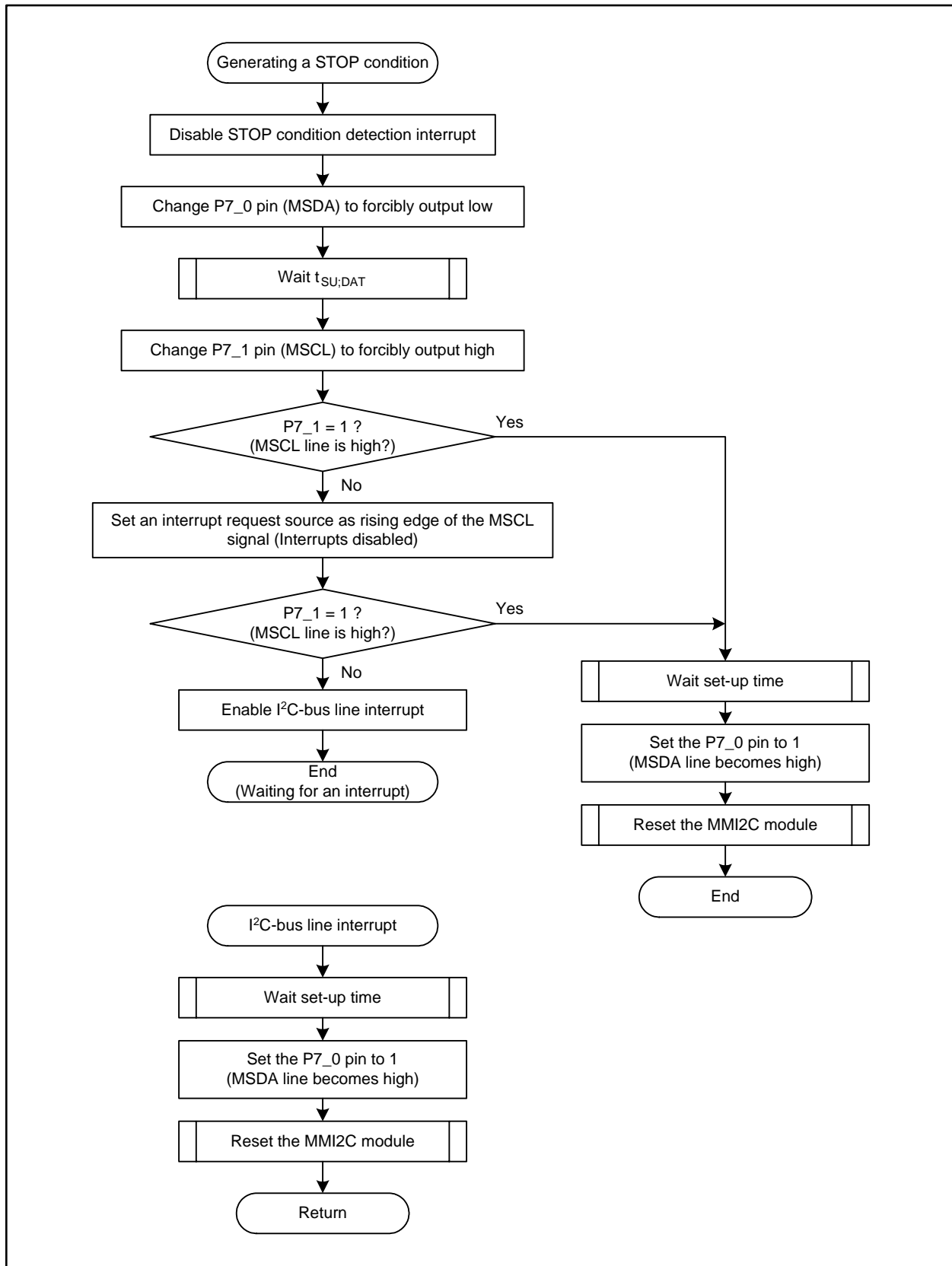


Figure 2. Generating a STOP Condition

```

/* Generate STOP Condition */
void gen_stop_cond (void) {
    stie_i2ccr1 = 0;           // Disable STOP condition detect int.
    p7_0 = 0;                 // Drive MSDA line to low
    p7_0s = 0x40;

    soft_wait(TSU_DAT);      // wait tsu/dat

    pd7_1 = 0;               // Release MSCL line

    if (p7_1 == 1) {
        soft_wait(SETUP_TIME);
        P7_0 = 1;           // Release MSDA line
        reset_mmi2c();      // Reset MMI2C
    } else {
        sip_i2csscr = 1;    // Select rising edge
        sis_i2csscr = 1;    // Select MSCL line
        i2cllic = 0x00;     // Clear IR bit of MMI2C line int.

        if (p7_1 == 1) {
            soft_wait(SETUP_TIME);
            P7_0 = 1;       // Release MSDA line
            reset_mmi2c();  // Reset MMI2C
        } else {
            i2cllic |= 0x01; // Enable MMI2C line int.
        }
    }
}

/* Software Wait Routine */
void soft_wait(long time){
    while (time > 0)
        time--;
}

/* Reset MMI2C and Pin Setting */
void reset_mmi2c(void){
    rst_i2ccr0 = 1           // Reset I2C-bus interface
    while (rst_i2ccr0 == 1)
        ;
    p7_0s = 0x43;           // Select MSDA output
    pd7_1 = 1;              // Set MSCL line to output
    stie_i2ccr1 = 1;        // Enable STOP condition detect int.
}

/* MSCL Line Interrupt Routine */
#pragma INTERRUPT mmi2c_line_int;
void mmi2c_line_int(void){
    soft_wait(SETUP_TIME);
    P7_0 = 1;               // Release MSDA line
    reset_mmi2c();          // Reset MMI2C
}

```

Figure 3. Reference Program