M16C/26

Using Vdet3 Vcc Voltage Monitoring

1.0 Abstract

The following article discusses the Vdet3 Vcc voltage monitoring circuits of the M16C/26 MCU chip. A sample program is provided for evaluating the Vdet3 behavior on the MSV30262 SKP board. A variable power supply, connected to the board, is required to make the MCU Vcc voltage adjustable for verification.

2.0 Introduction

The Mitsubishi M30262 is a 16-bit MCU based on the M16C/60 series CPU core. The MCU features include up to 64KB of Flash ROM, 2KB of RAM, and 4KB of virtual EEPROM. The peripheral set includes 10-bit A/D, UARTs, Timers, DMA, and GPIO. The voltage detection circuit has monitoring circuits to check the input voltage of the Vcc pin. These circuits monitor the input voltage at Vdet3 and Vdet4 (see Figure 1). VC26 to VC27 of VCR2 register is used to enable/disable these monitoring circuits (see Figure 2). This application note discusses the use of the Vdet3 circuit to monitor the Vcc input voltage on the M16C/26 MCU.

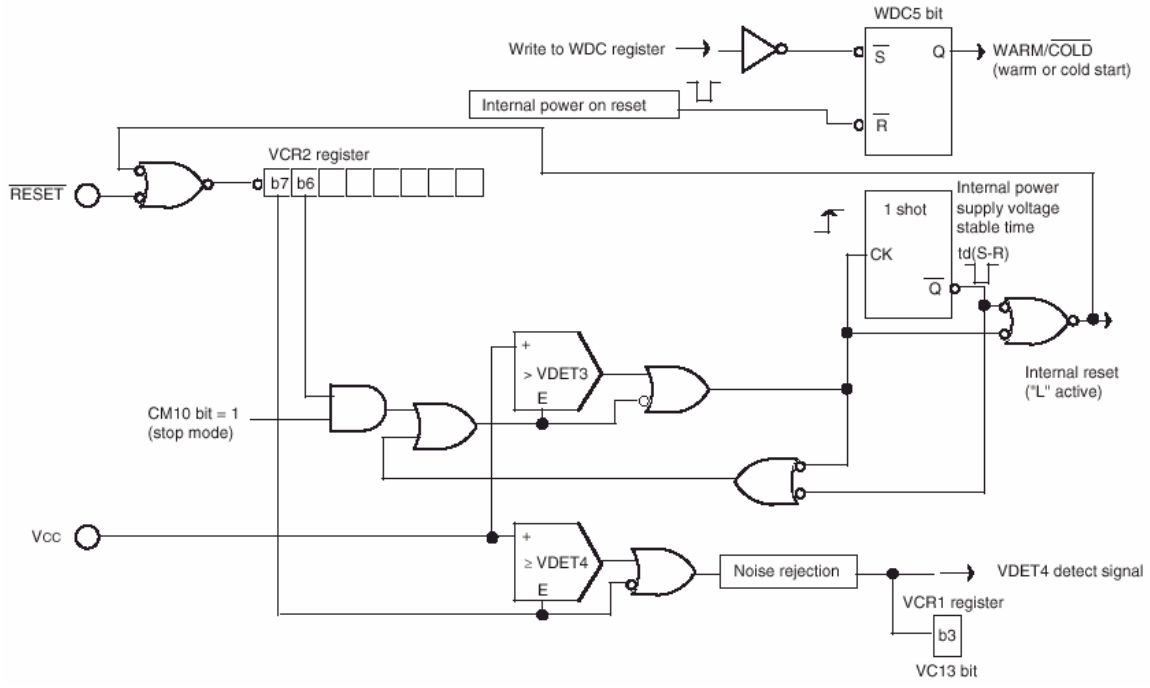

Figure 1 Reset circuit block

Power supply detection register 1

b7	b6	b5	b4	b3	b2	b1	b0
0	0	0	0		0	0	0

Symbol: VCR1 — Address: 001916 — After reset: 000010002

Bit symbol	Bit name	Function	RW
——	Reserved bit	Set to "0"	RW
VC13	V_{DET4} power supply monitor flag (Note)	0: $V_{CC} < V_{DET4}$ 1: $V_{CC} \geq V_{DET4}$	RO
——	Reserved bit	Set to "0"	RW

Note: The VC13 bit is useful when the VCR2 register's VC27 bit = 1 (V_{DET4} detection circuit enabled).
The VC13 bit is always 1 ($V_{CC} \geq V_{DET4}$) when the VCR2 register's VC27 bit = 0 (V_{DET4} detection circuit disabled).

Power supply detection register 2 (Note 1)

b7	b6	b5	b4	b3	b2	b1	b0
		0	0	0	0	0	0

Symbol: VCR2 — Address: 001A16 — After reset: 0016

Bit symbol	Bit name	Function	RW
——	Reserved bit	Set to "0"	RW
VC26	Power supply V_{DET3} monitor bit	0: Disables detection circuit 1: Enables detection circuit	RW
VC27	Power supply V_{DET4} monitor bit (Note 2)	0: Disables detection circuit 1: Enables detection circuit	RW

Note 1: Write to this register after the PRCR register's PRC3 bit is set to "1" (write enabled).
Note 2: To use the VCR1 register's VC13 bit or D4INT register's D42 bit, set the VC27 bit to "1" (V_{DET4} detection circuit enabled).

Power supply V_{DET4} detection register (Note 1)

b7	b6	b5	b4	b3	b2	b1	b0
X	X						

Symbol: D4INT — Address: 001F16 — After reset: 0016

Bit symbol	Bit name	Function	RW
D40	V_{DET4} detection interrupt enable bit.	0 : Disable 1 : Enable	RW
D41	STOP mode deactivation control bit (Note 4)	0: Disable (do not use the V_{DET4} detection interrupt to get out of stop mode) 1: Enable (use the V_{DET4} detection interrupt to get out of stop mode)	RW
D42	V_{DET4} up/down detection flag (Note 2)	0: Not detected 1: V_{DET4} up/down detected	RW (Note 3)
D43	WDT overflow detected flag	0: Not detected 1: Detected	RW (Note 3)
DF0	Sampling clock select bit	b5b4 00 : BCLK divided by 8 01 : BCLK divided by 16 10 : BCLK divided by 32 11 : BCLK divided by 64	RW
DF1			RW
—— (b7-b6)	Nothing is assigned. In an attempt to write to these bits, write "0". The value, if read, turns out to be "0".		—

Note 1: Write to this register after the PRCR register's PRC3 bit is set to "1" (write enabled).
Note 2: Useful when the VCR2 register's VC27 bit = 1 (V_{DET4} detection circuit enabled). If the VC27 bit is cleared to 0 (V_{DET4} detection circuit disabled), the D42 bit is set to 0 (Not detected).
Note 3: This bit is cleared to "0" by writing a "0" in a program. (Writing a "1" has no effect.)
Note 4: If the V_{DET4} detection interrupt needs to be used to get out of stop mode again after once used for that purpose, reset the D41 bit by writing a 0 and then a 1.

Figure 2 VCR1, VCR2 and D4INT registers

3.0 Vdet3 and MCU Reset

The Vdet3 Vcc monitoring circuit is used to reset the MCU whenever the supply voltage falls below the Vdet3 trip point. Vdet3 is enabled by setting VC26 bit of the VCR2 register to "1". With Vdet3 enabled, when the Vcc supply voltage falls below the Vdet3 trip point, the MCU is placed and held in RESET. When the Vcc supply voltage goes above Vdet3 trip point, the RESET is released and the MCU re-starts execution of user's application at the address pointed to by the reset vector.

4.0 Reference

Renesas Technology Corporation Semiconductor Home Page

http://www.renesas.com

E-mail Support

support_apl@renesas.com

Data Sheets

- M16C/26 datasheets, M30262eds.pdf

User's Manual

- M16C/20/60 C Language Programming Manual, 6020c.pdf
- M16C/20/60 Software Manual, 6020software.pdf
- Writing interrupt handlers in C for the M16C Application Note
- MSV30262-SKP or MSV-Mini26-SKP Quick start guide
- MSV30262-SKP or MSV-Mini26-SKP Users Manual
- MDECE30262 or MSV-Mini26-SKP Schematic

5.0 Software Code

5.1 Application Code Outline

The Vdet3 detection program, written in C and compiled using the KNC30 compiler, will run on the M16C/26 starter kit MSV30262 SKP board. To run the program, follow the steps below.

1. Download the program to the M30262 MCU using the FoUSB Programmer (with USB-ICD).
2. Disconnect USB-ICD from the SKP board. Using a variable voltage power supply, supply 5V to the SKP board using the board's Vcc and GND pins. At this point, the green power LED (D7) should light up, the red LED D3 is blinking, and the green LED D5 is lit up.

3. Every time the pushbutton switch S2 is pressed, the yellow and green LED's are toggled (i.e. if yellow is ON, green is OFF and vice versa).
4. Lower Vcc (e.g. 2.2V) from the supply voltage and as soon as Vcc<Vdet3 the MCU is held in reset and user LED's (D3, D4, & D5) are OFF.
5. Now gradually increase Vcc from the supply voltage and as soon as Vcc>Vdet3 the red LED D3 starts blinking again and the green LED D5 is ON.
6. To verify that the code is running correctly, press pushbutton switch S2 will continue to toggle the yellow and green LED's.

5.2 Software Source Code

```
/***********************************************************************
*
*       File Name: main_vdet3.c
*
*       Content:   This program blinks the red LED (D3) to show that a program (MCU)
*                  is running. Pressing S2 toggles which LED comes on: yellow or green.
*
*                  To test Vdet3, lower the supply voltage under Vdet3 (2.7V).
*                  The MCU goes into reset and LED's are off. After bringing Vcc
*                  above Vdet3, MCU exits reset condition and starts running.
*
*                  To verify the code is still working properly, press S2 to toggle the
*                  yellow and green LEDs. If it toggles, the code is running correctly.
*
*       Date:  5-02-2003
*       This program was written to run on the MSV30262-SKP Board.
*
*       Copyright 2003 Renesas Technology America, Inc.
*       All rights reserved
*
*=======================================================================
*       $Log:$
*=======================================================================*/

#include "..\common\sfr262.h"  /* M16C/26 special function register definitions */

/* LEDs */
#define red_led         p7_0
#define yellow_led      p7_1
#define green_led       p7_2

/* SWITCHES */
#define sw2             p10_5
```

```
void led_init(void);    /* routine that initializes the LEDs */
void Vdet3_init(void);  /* routine that initializes Vdet3 */
void blink_redLED(void);/* blink red LED to show continuation of program execution */

int dly_cnt;            /* delay counter for blinking LED */
int flag=1;             /* start with cold start flag = 1 */

/***********************************************************************
Name:   main
Parameters: None
Returns: None
Description:  main program loop and initialization
***********************************************************************/
main() {

        led_init();                     /* initialize LEDs */
        Vdet3_init();                   /* initialize Vdet3 */
        while(1){                       /* infinite program loop */

                blink_redLED();

                /* press S2 write to toggle green & yellow LED to verify code is
                   still working correctly */
                if(!sw2){
                        if (green_led == 1){
                                yellow_led  = 1;        /* turn OFF yellow LED */
                                green_led = 0;          /* turn ON green LED */
                        }
                        else{
                                yellow_led  = 0;        /* turn ON yellow LED */
                                green_led       = 1;    /* turn OFF green LED */
                        }
                }
        }
}

/***********************************************************************
Name:           led_init
Parameters:     None
Returns:        None
Description: Initialization routine for the user LED's.
***********************************************************************/
void led_init(void) {

        pd7_0 = 1;              /* set LED ports to outputs (connected to LEDs) */
        pd7_1 = 1;
        pd7_2 = 1;
        red_led = 1;            /* turn off Red & Yellow but Green will be turned on */
        yellow_led = 1;
        green_led = 0;
}

/***********************************************************************
```

```
Name:           Vdet3_init
Parameters:     None
Returns:        None
Description: Initialization routine for Voltage Detection Circuit 3 (Vdet3).
***********************************************************************/
void Vdet3_init(void) {
        prc3 = 1;               /* unlock vcr2 and d4int registers */
        vc26 = 1;               /* enable Vdet3 */
        prc3 = 0;               /* lock vcr2 and d4int */
}

/***********************************************************************
Name:           blink_redLED
Parameters:     None
Returns:        None
Description: Routine for blinking red LED (D3) to indicate continuation of
             program execution but RAM was retained.
***********************************************************************/
void blink_redLED(void){

                red_led         = 0;    /* turn red LED ON */
                for (dly_cnt = 0; dly_cnt<0xffff; dly_cnt++);  /* delay */

                red_led         = 1;   /* turn red LED OFF */
                for (dly_cnt = 0; dly_cnt<0xffff; dly_cnt++);  /* delay */
}
```

Keep safety first in your circuit designs!

- Renesas Technology Corporation puts the maximum effort into making semiconductor products better and more reliable, but there is always the possibility that trouble may occur with them. Trouble with semiconductors may lead to personal injury, fire or property damage.
  Remember to give due consideration to safety when making your circuit designs, with appropriate measures such as (i) placement of substitutive, auxiliary circuits, (ii) use of nonflammable material or (iii) prevention against any malfunction or mishap.

Notes regarding these materials

- These materials are intended as a reference to assist our customers in the selection of the Renesas Technology Corporation product best suited to the customer's application; they do not convey any license under any intellectual property rights, or any other rights, belonging to Renesas Technology Corporation or a third party.
- Renesas Technology Corporation assumes no responsibility for any damage, or infringement of any third-party's rights, originating in the use of any product data, diagrams, charts, programs, algorithms, or circuit application examples contained in these materials.
- All information contained in these materials, including product data, diagrams, charts, programs and algorithms represents information on products at the time of publication of these materials, and are subject to change by Renesas Technology Corporation without notice due to product improvements or other reasons. It is therefore recommended that customers contact Renesas Technology Corporation or an authorized Renesas Technology Corporation product distributor for the latest product information before purchasing a product listed herein.
  The information described here may contain technical inaccuracies or typographical errors.
  Renesas Technology Corporation assumes no responsibility for any damage, liability, or other loss rising from these inaccuracies or errors.
  Please also pay attention to information published by Renesas Technology Corporation by various means, including the Renesas Technology Corporation Semiconductor home page (http://www.renesas.com).
- When using any or all of the information contained in these materials, including product data, diagrams, charts, programs, and algorithms, please be sure to evaluate all information as a total system before making a final decision on the applicability of the information and products. Renesas Technology Corporation assumes no responsibility for any damage, liability or other loss resulting from the information contained herein.
- Renesas Technology Corporation semiconductors are not designed or manufactured for use in a device or system that is used under circumstances in which human life is potentially at stake. Please contact Renesas Technology Corporation or an authorized Renesas Technology Corporation product distributor when considering the use of a product contained herein for any specific purposes, such as apparatus or systems for transportation, vehicular, medical, aerospace, nuclear, or undersea repeater use.
- The prior written approval of Renesas Technology Corporation is necessary to reprint or reproduce in whole or in part these materials.
- If these products or technologies are subject to the Japanese export control restrictions, they must be exported under a license from the Japanese government and cannot be imported into a country other than the approved destination.
  Any diversion or reexport contrary to the export control laws and regulations of Japan and/or the country of destination is prohibited.
- Please contact Renesas Technology Corporation for further details on these materials or the products contained therein.