NXP Semiconductors
Application Note

Document Number: AN5284
Rev. 0, 05/2016

Using FCCU on MPC5744P

By: Peter Vlna

1. Introduction

This document describes the configuration, restrictions, principles, and correct usage of the FCCU module implemented in the MPC5744P devices.

Contents

2. Important FCCU Notes for MPC5744P Devices

When the FCCU state is changed, the FCCU internal watchdog starts. After the watchdog expires, the FCCU operation run bit field in the FCCU_CTRL register is changed to ABORT. Therefore, it is not possible to perform the “STEP” operations in the debugger.

3. FCCU Faults Reading

The FCCU_NCF_Sx register contains the latched fault indication collected from the non-critical fault sources.

Write to the FCCU control register (FCCU_CTRL) to read the FCCU faults stored in the NCF_Sx registers on the MPC5744P devices:

1. Write a proper OPR to the FCCU_CTRL[OPR] = 10.
2. Wait for a successful operation run (OPR) mode change.
3. Read the faults from the NCF_S[x] registers.

Here is the example code:

```
void FCCU_read_faults(void)
{
  /* To read faults OP10 must be set in FCCU_CFG[OPR] field */
  FCCU.CTRL.R = 0xA;  Read the NCF status register [OP10]

  /* wait for the  operation run (OPR) mode change */
  while (FCCU.CTRL.B.OPS != 0x3);//operation status

  /* Read all NCFS registers to variables */
  FCCU_status[0] = FCCU.NCF_S[0].R;     //read FCCU.NCF_S0 register
  FCCU_status[1] = FCCU.NCF_S[1].R;     //read FCCU.NCF_S1 register
  FCCU_status[2] = FCCU.NCF_S[2].R;     //read FCCU.NCF_S2 register

}//FCCU_read_faults
```

4. FCCU Faults Clearing

To clear the latched FCCU faults, perform these steps:

1. Write a proper key to the FCCU_NCFK register.
2. Clear the NCF_Sx status (flag) bit, and the OP12 opcode is automatically set in the FCCU_CTRL.OPR field.
3. Wait for the operation to complete (FCCU_CTRL.OPS field).

4. Read the FCCU_NCF_Sx register to verify that the deletion was successful, and repeat the sequence when a failure occurs.

Here is the example code:

```
void FCCU_clear_faults(void)
{
  /* 1. Write a proper key into the FCCU_NCFK register */
  //Non-critical fault key = AB34_98FEh
  FCCU.NCFK.R = 0xAB3498FE;

  /* 2. Clear the status (flag) bit NCFSx => the opcode OP12 is automatically
  /* Read all NCFS registers to clear all faults.*/
  /* For details which faults can be cleared see Table 7-36. FCCU Non-Critical Faults Mapping
in RM */
  FCCU.NCF_S[0].R = 0xFFFFFFFF;          // read FCCU.NCF_S0 register

  /* Verify if state change was successful */
  while (FCCU.CTRL.B.OPS != 0x3);//Operation status successful

  /* NCFS_1 register clear */
  FCCU.NCFK.R = 0xAB3498FE;              //Non-critical fault key = AB34_98FEh
  FCCU.NCF_S[1].R = 0xFFFFFFFF;          // clear FCCU.NCF_S1 register
  /* Verify if state change was successful */
  while (FCCU.CTRL.B.OPS != 0x3);//Operation status successful

  /* NCFS_2 register clear */
  FCCU.NCFK.R = 0xAB3498FE;              //Non-critical fault key = AB34_98FEh
  FCCU.NCF_S[2].R = 0xFFFFFFFF;          // clear FCCU.NCF_S2 register
  /* Verify if state change was successful */
  while (FCCU.CTRL.B.OPS != 0x3);//Operation status successful
}//FCCU_clear_faults
```

5. FCCU Fault Injection

Before injecting a fault, keep in mind that not all faults can be injected. Some of the faults can be injected through the peripherals (such as ADC or PMC), and some cannot be injected at all. See Table 7-37 in the *MPC5744P Reference Manual* (document MPC5744PRM) to distinguish between them. This example code represents a fault injection with no reaction set on the injected fault:

```
void FCCU_Fake_faults_inject(void)
{
```

```
  /* Inject a NCF[7] -> STCU2 fault */
  FCCU.NCFF.R = 0x7;
  /* Fault is now injected and user can read FCCU NCFSx register to see it is
     latched in NCFSx registers. But by default no reaction is set on this fault
     so it is only informative */
}//FCCU_Fake_faults_inject
```

6. CONFIG State

The default configuration can be modified only in the CONFIG state. The subset of the FCCU registers to define the FCCU configuration (global configuration, reactions to fault, timeout, non-critical fault masking) can be accessed in the write mode only in the CONFIG state.

Before entering the CONFIG state of the FCCU, perform these steps:

1. Disable the SWT.
2. Clear the pending FCCU faults latched in the NCF_Sx registers.

To enter the CONFIG state, perform these steps:

1. Unlock the configuration by providing the transition unlocking key value (0xBC).
2. Provide the control register key for the CONFIG state (0x913756AF).
3. Enter the CONFIG state.
4. Verify that the state transition was successful.

The FCCU is now prepared for the user-defined configuration.

To exit the CONFIG state, perform these steps:

1. Provide the control register key for the NORMAL state (0x825A132B).
2. Put the FCCU into the NORMAL state (OP2).
3. Verify that the state transition was successful.

NOTE

The FCCU implements an internal watchdog. The watchdog starts when the FCCU enters the CONFIG state. If the FCCU is not configured and returns to the NORMAL state in the watchdog time, the CONFIG state window is aborted and no changes are taken into account. As a result, the OPS bit in the control register is set to the ABORT value. Therefore, it is not possible to debug the FCCU while it is in the CONFIG state.

Here is the example code:

```
void FCCU_CONFIG (void)
{
  /* Unlock configuration */
  FCCU.TRANS_LOCK.R = 0xBC;
```

```
  /* provide Config state key */
  FCCU.CTRLK.R = 0x913756AF;            //key for OP1
  /* enter config state - OP1 */
  FCCU.CTRL.R = 0x1;                    //set OP1 - set up FCCU into the CONFIG mode
  /* wait for successful state transition */
  while (FCCU.CTRL.B.OPS != 0x3);//operation status successful

  /*************************************/
  /* Insert here the FCCU configuration */
  /*************************************/

  /* set up the NOMAL mode of FCCU */
  FCCU.CTRLK.R = 0x825A132B;            //key for OP2
  FCCU.CTRL.R = 0x2;                    //set the OP2 - set up FCCU into the NORMAL mode
  while (FCCU.CTRL.B.OPS != 0x3);//operational status successful

}//FCCU_CONFIG
```

7. ALARM State

The FCCU state machine provides the possibility to enter the ALARM state when a fault is detected. There is a possibility to trigger the ALARM interrupt on the FCCU fault event, and solve the issue (fault source) in the configured timeout window.

Before entering the FCCU CONFIG state, perform these steps:

1. Disable the SWT.
2. Clear the pending FCCU faults latched in the NCF_Sx registers.

Enter the CONFIG state:

1. Unlock the configuration by providing the transition unlocking key value (0xBC).
2. Provide the control register key for the CONFIG state (0x913756AF).
3. Enter the CONFIG state.
4. Verify that the state transition was successful.

The FCCU is now prepared for the user-defined configuration.

1. Enable the ALARM timeout in the NCF_TOEn register.
2. Enable the reaction on fault in the NCF_En register.
3. Enable the alarm interrupt in the IRQ_ALARM_ENn register.

Exit the CONFIG state:

1. Provide the control key for the NORMAL state (0x825A132B).

2. Put the FCCU into the NORMAL state (OP2).
3. Verify that the state transition was successful.

NOTE

Make sure to properly configure the INTC controller and to set the priority for the ALARM interrupt.

Here is the example code:

```
void FCCU_CONFIG (void)
{
  /* Unlock configuration */
  FCCU.TRANS_LOCK.R = 0xBC;

  /* provide Config state key */
  FCCU.CTRLK.R = 0x913756AF;            //key for OP1

  /* enter config state - OP1 */
  FCCU.CTRL.R = 0x1;                    //set OP1 - set up FCCU into the CONFIG
  /* wait for successful state transition */
  while (FCCU.CTRL.B.OPS != 0x3);       //operation status successful

  /* FCCU moves into the ALARM state if the respective fault is enabled */
  FCCU.NCF_TOE[0].R = 0xFFFFFFFF;       //ALARM Timeout Enable

  FCCU.NCF_E[0].B.NCFE7 = 0x1;          //Enable reaction on fault NCF[7]

  FCCU.IRQ_ALARM_EN[0].R = 0x80;        //Enable ALARM interrupt on fault NCF[7]

  /* set up the NOMAL mode of FCCU */
  FCCU.CTRLK.R = 0x825A132B;            //key for OP2
  FCCU.CTRL.R = 0x2;                    //set the OP2 - set up FCCU into the NORMAL mode
  while (FCCU.CTRL.B.OPS != 0x3);//operational status successful

}//FCCU_CONFIG
```

8. Revision History

This table summarizes the changes done to this document since the initial release:

Table 1. Revision history

Revision number	Date	Substantive changes
0	05/2016	Initial release.

How to Reach Us:

Home Page:
nxp.com

Web Support:
nxp.com/support

Document Number: AN5284
Rev. 0
05/2016