Freescale Semiconductor
Application Note

Document Number:AN4422
Rev. 1, 12/2012

Qorivva MPC5675K and PXS30 Test and Shadow Flash Parameters for ADC Self Test, MBIST, and LBIST

by: Curt Hillier
Applications Engineering
Austin, Texas USA

1 Introduction

The Qorivva MPC5675K microcontroller offers a set of features to support functional safety applications. Memory, digital logic, and analog circuit integrity checking are part of the MPC5675K functional safety feature set.

The Self-Test Control Unit (STCU) controls the device's self-test sequence before the primary user application starts running. The self-test goal is to detect physical defects in the digital logic and embedded memories with enough coverage to meet the required Safety Integrity Level (SIL) of the system. The STCU controls two types of self-test: Memory Built-In Self-Test (MBIST) and Logic Built-In Self-Test (LBIST).

The ADC Self-Test feature checks to verify that the ADC is functioning correctly at regular intervals. The ADC Self-Tests use analog watchdogs to verify the result of self-test conversions. The thresholds of these watchdogs are programmed in the test flash memory at the factory.

Test flash and shadow flash memory contain parameters defining operation of MBIST, LBIST, and ADC Self-Test features. For MBIST and LBIST, you can program an entry into shadow flash memory to disable BIST execution. For ADC Self-Test, the user software reads Analog Watchdog Enable (AWDE), Watchdog Timer Enable (WDTE), and threshold high / threshold low settings from test flash memory and programs the values to ADC Self-Test registers.

Contents

1 Introduction....1
2 Shadow flash contents—factory programmed, user read / write / erase....2
3 Test flash contents for TSENSE and ADC Self-Test—user read-only....3
4 Test flash settings for MBIST and LBIST —user read-only....5
5 Revision history....8

This application note details the contents of test flash (TF) and shadow flash (SF) memory as they relate to MBIST, LBIST, and ADC Self-Test settings. It does not detail the Nonvolatile User Options Register (NVUSRO) or provide details on programming and erasing flash memory. Please refer to MPC5675KRM, *MPC5675K Microcontroller Reference Manual* for additional information.

2 Shadow flash contents—factory programmed, user read / write / erase

Pre-production settings: the following table shows SF contents for devices with MBIST and LBIST bypassed.

Table 1. Pre-production settings: SF contents for MBIST and LBIST bypassed

Offset from base address[1]	Offset 0x0	Offset 0x4	Offset 0x8	Offset 0xC	Description
0x0010	0x05AA_55AF	0x0000_0000	0xFFFF_FFFF	0xFFFF_FFFF	Valid shadow scan data (start word)
0x0020	0x0000_0100	0x0008_000C	0xFFFF_FFFF	0xFFFF_FFFF	STCU bypass MBIST and LBIST
0x0030	0xFFFF_FFFF	0xFFFF_FFFF	0xFFFF_FFFF	0xFFFF_FFFF	Stop word (stop scanning flash for test parameters)

1. SF base address = 0x0020_0000

NOTE

To enable MBIST and LBIST for pre-production devices, you must erase the SF area and allow STCU MBIST / LBIST to be driven from TF contents.

Production settings: The table below details SF contents for MBIST / LBIST driven from TF settings; these settings apply to production devices. The key change from the pre-production settings is to remove the bypass BIST setting—STCU MBIST / LBIST are configured and enabled per the TF settings.

Table 2. Production settings: SF contents for MBIST / LBIST driven from TF settings

Offset from base address[1]	Offset 0x0	Offset 0x4	Offset 0x8	Offset 0xC	Description
0x0010	0x05AA_55AF	0x0000_0000	0xFFFF_FFFF	0xFFFF_FFFF	Valid shadow scan data (start word)
0x0020	0xFFFF_FFFF	0xFFFF_FFFF	0xFFFF_FFFF	0xFFFF_FFFF	Stop word (stop scanning flash for test parameters)

1. SF base address = 0x0020_0000

NOTE

To disable MBIST and LBIST for production setting devices, you must program a single Device Configuration File (DCF) entry starting at offset 0x0020 as shown below.

Address = 0x0020
Offset 0x0 = 0x0000_0100 // sets the bypass bit in STCU_CFG
Offset 0x4 = 0x0008_000C // STCU Configuration Register (STCU_CFG)
Offset 0x8 = 0xFFFF_FFFF
Offset 0xC = 0xFFFF_FFFF

3 Test flash contents for TSENSE and ADC Self-Test—user read-only

The following table lists the TSENSE and ADC Self-Test parameters contained in TF. The user software must read values from the ADC0, ADC1, ADC2, and ADC3 parameters as shown below, and then program the corresponding registers for each ADC.

NOTE

During production, ADC Self Test parameters may change. Customers can read TF contents and program the values to ADC Self Test registers to keep up to date with any changes.

Table 3. Definition of ADC Self-Test parameters

Offset from TEST_FLASH_BASE 0x0040_0000	Word name	Function	Note	Production data
0x0000	TSENS1_CAL W1/W2	Temperature sensor calibration data W1 and W2	2 ×12 bits word right aligned	Device specific
0x0004	TSENS1_CAL W3/W4	Temperature sensor calibration data W3 and W4	2 ×12 bits word right aligned	Device specific
0x0008	SPARE	Reserved	For future usage	Reserved
0x000C	SPARE	Reserved	For future usage	Reserved
0x0010	ADC0_CAL W1	ADC0 Self-Test calibration—RC algorithm	STAW3RH/L	0xFE20F1E0
0x0014	ADC0_CAL W2	ADC0 Self-Test calibration—C algorithm step 0	STAW4RH/L	0xF856F732
0x0018	ADC0_CAL W3	ADC0 Self-Test calibration—C algorithm step 1 to N	STAW5RH/L	0xF873F732
0x001C	ADC0_CAL W4	ADC0 Self-Test calibration—S algorithm step 0—3.3V	STAW0RH/L	0xF75AF4DF
0x0020	ADC0_CAL W5	ADC0 Self-Test calibration—S algorithm step 0—5.0V	STAW0RH/L	0xF4D0F2DB
0x0024	ADC0_CAL W6	ADC0 Self-Test calibration—S algorithm step 1—integer	STAW1ARH/L	0xF003F002
0x0028	ADC0_CAL W7	ADC0 Self-Test calibration—S algorithm step 1—fractional	STAW1BRH/L	0xF3D9F1E3
0x002C	ADC0_CAL W8	ADC0 Self-Test calibration—S algorithm step 2	STAW2R	0xFFFFFFF9
0x0030	ADC0 Reserved	ADC0 reserved	For future expansion	Reserved
0x0034	ADC1_CAL W1	ADC1 Self-Test calibration—RC algorithm	STAW3RH/L	0xFE20F1E0
0x0038	ADC1_CAL W2	ADC1 Self-Test calibration—C algorithm step 0	STAW4RH/L	0xF856F732
0x003C	ADC1_CAL W3	ADC1 Self-Test calibration—C algorithm step 1 to N	STAW5RH/L	0xF873F732
0x0040	ADC1_CAL W4	ADC1 Self-Test calibration—S algorithm step 0—3.3V	STAW0RH/L	0xF75AF4DF
0x0044	ADC1_CAL W5	ADC1 Self-Test calibration—S algorithm step 0—5.0V	STAW0RH/L	0xF4D0F2DB
0x0048	ADC1_CAL W6	ADC1 Self-Test calibration—S algorithm step 1—integer	STAW1ARH/L	0xF003F002

Table continues on the next page...

Table 3. Definition of ADC Self-Test parameters (continued)

Offset from TEST_FLASH_BASE 0x0040_0000	Word name	Function	Note	Production data
0x004C	ADC1_CAL W7	ADC1 Self-Test calibration—S algorithm step 1—fractional	STAW1BRH/L	0xF3D9F1E3
0x0050	ADC1_CAL W8	ADC1 Self-Test calibration—S algorithm step 2	STAW2R	0xFFFFFFF9
0x0054	ADC1 reserved	ADC1 reserved	For future expansion	Reserved
0x0058	SPARE	Reserved	For future usage	Reserved
0x005C	SPARE	Reserved	For future usage	Reserved
0x0060	SPARE	Reserved	For future usage	Reserved
0x0064	SPARE	Reserved	For future usage	Reserved
0x0068	SPARE	Reserved	For future usage	Reserved
0x006C	ADC2_CAL W1	ADC2 Self-Test calibration—RC algorithm	STAW3RH/L	0xFE20F1E0
0x0070	ADC2_CAL W2	ADC2 Self-Test calibration—C algorithm step 0	STAW4RH/L	0xF856F732
0x0074	ADC2_CAL W3	ADC2 Self-Test calibration—C algorithm step 1 to N	STAW5RH/L	0xF873F732
0x0078	ADC2_CAL W4	ADC2 Self-Test calibration—S algorithm step 0—3.3V	STAW0RH/L	0xF75AF4DF
0x007C	ADC2_CAL W5	ADC2 Self-Test calibration—S algorithm step 0—5.0V	STAW0RH/L	0xF4D0F2DB
0x0080	ADC2_CAL W6	ADC2 Self-Test calibration—S algorithm step 1—integer	STAW1ARH/L	0xF003F002
0x0084	ADC2_CAL W7	ADC2 Self-Test calibration—S algorithm step 1—fractional	STAW1BRH/L	0xF3D9F1E3
0x0088	ADC2_CAL W8	ADC2 Self-Test calibration—S algorithm step 2	STAW2R	0xFFFFFFF9
0x008C	ADC2 reserved	ADC2 reserved	For future expansion	Reserved
0x0090	ADC3_CAL W1	ADC3 Self-Test calibration—RC algorithm	STAW3RH/L	0xFE20F1E0
0x0094	ADC3_CAL W2	ADC3 Self-Test calibration—C algorithm step 0	STAW4RH/L	0xF856F732
0x0098	ADC3_CAL W3	ADC3 Self-Test calibration—C algorithm step 1 to N	STAW5RH/L	0xF873F732
0x009C	ADC3_CAL W4	ADC3 Self-Test calibration—S algorithm step 0—3.3V	STAW0RH/L	0xF75AF4DF
0x00A0	ADC3_CAL W5	ADC3 Self-Test calibration—S algorithm step 0—5.0V	STAW0RH/L	0xF4D0F2DB
0x00A4	ADC3_CAL W6	ADC3 Self-Test calibration—S algorithm step 1—integer	STAW1ARH/L	0xF003F002
0x00A8	ADC3_CAL W7	ADC3 Self-Test calibration—S algorithm step 1—fractional	STAW1BRH/L	0xF3D9F1E3
0x00AC	ADC3_CAL W8	ADC3 Self-Test calibration—S algorithm step 2	STAW2R	0xFFFFFFF9
0x00B0	ADC3 reserved	ADC3 reserved	For future expansion	Reserved
0x00B4	SPARE	Reserved	For future usage	Reserved
0x00B8	SPARE	Reserved	For future usage	Reserved

4 Test flash settings for MBIST and LBIST—user read-only

The following table details the TF parameters for Self Test Control Unit (STCU) MBIST and LBIST configuration.

When the MPC5675K comes out of reset, the device scans the contents of TF for a valid DCF start word. Once the device encounters a valid start word, it scans subsequent TF entries and programs the STCU based on TF contents. Scanning of TF and writing to STCU continues until the device encounters a valid DCF stop word.

Table 4. TF settings for MBIST and LBIST

Offset from base address[1]	STCU register name	Function	Offset 0x0	Offset 0x4	Offset 0x8	Offset 0xC
0x0400	—	Start Word for DCF protocol	0x05AA55AF	0x00000000	0xFFFFFFFF	0xFFFFFFFF
0x0410-0x500	—	Reserved—internal use	Reserved	Reserved	Reserved	Reserved
0x0510	SKC	Unlock STCU	0xABFC1893	0x00080008	0xFFFFFFFF	0xFFFFFFFF
0x0520	SKC	Unlock STCU	0x319A6C2F	0x00080008	0xFFFFFFFF	0xFFFFFFFF
0x0530	LB0_CTRL	RUN LBIST 1 in parallel with LBIST0	0x81000005	0x00080080	0xFFFFFFFF	0xFFFFFFFF
0x0540	LB0_PCS[2]	Pattern Count of LBIST0 is 8000	0x00001F3F	0x00080084	0xFFFFFFFF	0xFFFFFFFF
0x0550	LB0_MISREL	Low word of LBIST0 MISR	0x3E8AC3B0	0x00080088	0xFFFFFFFF	0xFFFFFFFF
0x0560	LB0_MISREH	High word of LBIST0 MISR	0xC36E2624	0x0008008C	0xFFFFFFFF	0xFFFFFFFF
0x0570	LB1_CTRL	RUN LBIST 2 in parallel with LBIST1	0x82000005	0x000800A0	0xFFFFFFFF	0xFFFFFFFF
0x0580	LB1_PCS	Pattern Count of LBIST1 is 8000	0x00001F3F	0x000800A4	0xFFFFFFFF	0xFFFFFFFF
0x0590	LB1_MISREL	Low word of LBIST1 MISR	0x5E3B1B6D	0x000800A8	0xFFFFFFFF	0xFFFFFFFF
0x05A0	LB1_MISREH	High word of LBIST1 MISR	0xB96BF646	0x000800AC	0xFFFFFFFF	0xFFFFFFFF
0x05B0	LB2_CTRL	No more LBISTS to RUN	0x7F000005	0x000800C0	0xFFFFFFFF	0xFFFFFFFF
0x05C0	LB2_PCS	Pattern Count of LBIST2 is 8100	0x00001FA3	0x000800C4	0xFFFFFFFF	0xFFFFFFFF
0x05D0	LB2_MISREL	Low word of LBIST2 MISR	0x96C7BBA9	0x000800C8	0xFFFFFFFF	0xFFFFFFFF
0x05E0	LB2_MISREH	High word of LBIST2 MISR	0x5950F4D9	0x000800CC	0xFFFFFFFF	0xFFFFFFFF
0x05F0	MB_CTRL_0	MBIST*n* configuration	0x91030000	0x00080300	0xFFFFFFFF	0xFFFFFFFF
0x0600	MB_CTRL1	MBIST*n* configuration	0x92030000	0x00080304	0xFFFFFFFF	0xFFFFFFFF
0x0610	MB_CTRL2	MBIST*n* configuration	0x93090000	0x00080308	0xFFFFFFFF	0xFFFFFFFF
0x0620	MB_CTRL3	MBIST*n* configuration	0x94090000	0x0008030C	0xFFFFFFFF	0xFFFFFFFF
0x0630	MB_CTRL4	MBIST*n* configuration	0x95090000	0x00080310	0xFFFFFFFF	0xFFFFFFFF
0x0640	MB_CTRL5	MBIST*n* configuration	0x96090000	0x00080314	0xFFFFFFFF	0xFFFFFFFF
0x0650	MB_CTRL6	MBIST*n* configuration	0x97090000	0x00080318	0xFFFFFFFF	0xFFFFFFFF
0x0660	MB_CTRL7	MBIST*n* configuration	0x98090000	0x0008031C	0xFFFFFFFF	0xFFFFFFFF
0x0670	MB_CTRL8	MBIST*n* configuration	0x99090000	0x00080320	0xFFFFFFFF	0xFFFFFFFF
0x0680	MB_CTRL9	MBIST*n* configuration	0x9A090000	0x00080324	0xFFFFFFFF	0xFFFFFFFF
0x0690	MB_CTRL10	MBIST*n* configuration	0x9B030000	0x00080328	0xFFFFFFFF	0xFFFFFFFF

Table continues on the next page...

Table 4. TF settings for MBIST and LBIST (continued)

Offset from base address[1]	STCU register name	Function	Offset 0x0	Offset 0x4	Offset 0x8	Offset 0xC
0x06A0	MB_CTRL11	MBIST*n* configuration	0x9C250000	0x0008032C	0xFFFFFFFF	0xFFFFFFFF
0x06B0	MB_CTRL12	MBIST*n* configuration	0x9D250000	0x00080330	0xFFFFFFFF	0xFFFFFFFF
0x06C0	MB_CTRL13	MBIST*n* configuration	0x9E250000	0x00080334	0xFFFFFFFF	0xFFFFFFFF
0x06D0	MB_CTRL14	MBIST*n* configuration	0x9F250000	0x00080338	0xFFFFFFFF	0xFFFFFFFF
0x06E0	MB_CTRL15	MBIST*n* configuration	0xA0250000	0x0008033C	0xFFFFFFFF	0xFFFFFFFF
0x06F0	MB_CTRL16	MBIST*n* configuration	0xA1250000	0x00080340	0xFFFFFFFF	0xFFFFFFFF
0x0700	MB_CTRL17	MBIST*n* configuration	0xA2250000	0x00080344	0xFFFFFFFF	0xFFFFFFFF
0x0710	MB_CTRL18	MBIST*n* configuration	0xA3250000	0x00080348	0xFFFFFFFF	0xFFFFFFFF
0x0720	MB_CTRL19	MBIST*n* configuration	0xA4250000	0x0008034C	0xFFFFFFFF	0xFFFFFFFF
0x0730	MB_CTRL20	MBIST*n* configuration	0xA5250000	0x00080350	0xFFFFFFFF	0xFFFFFFFF
0x0740	MB_CTRL21	MBIST*n* configuration	0xA6250000	0x00080354	0xFFFFFFFF	0xFFFFFFFF
0x0750	MB_CTRL22	MBIST*n* configuration	0xA7250000	0x00080358	0xFFFFFFFF	0xFFFFFFFF
0x0760	MB_CTRL23	MBIST*n* configuration	0xA8250000	0x0008035C	0xFFFFFFFF	0xFFFFFFFF
0x0770	MB_CTRL24	MBIST*n* configuration	0xA9250000	0x00080360	0xFFFFFFFF	0xFFFFFFFF
0x0780	MB_CTRL25	MBIST*n* configuration	0xAA250000	0x00080364	0xFFFFFFFF	0xFFFFFFFF
0x0790	MB_CTRL26	MBIST*n* configuration	0xAB250000	0x00080368	0xFFFFFFFF	0xFFFFFFFF
0x07A0	MB_CTRL27	MBIST*n* configuration	0xAC030000	0x0008036C	0xFFFFFFFF	0xFFFFFFFF
0x07B0	MB_CTRL28	MBIST*n* configuration	0xAD030000	0x00080370	0xFFFFFFFF	0xFFFFFFFF
0x07C0	MB_CTRL29	MBIST*n* configuration	0xAE090000	0x00080374	0xFFFFFFFF	0xFFFFFFFF
0x07D0	MB_CTRL30	MBIST*n* configuration	0xAF090000	0x00080378	0xFFFFFFFF	0xFFFFFFFF
0x07E0	MB_CTRL31	MBIST*n* configuration	0xB0090000	0x0008037C	0xFFFFFFFF	0xFFFFFFFF
0x07F0	MB_CTRL32	MBIST*n* configuration	0xB1090000	0x00080380	0xFFFFFFFF	0xFFFFFFFF
0x0800	MB_CTRL33	MBIST*n* configuration	0xB2090000	0x00080384	0xFFFFFFFF	0xFFFFFFFF
0x0810	MB_CTRL34	MBIST*n* configuration	0xB3090000	0x00080388	0xFFFFFFFF	0xFFFFFFFF
0x0820	MB_CTRL35	MBIST*n* configuration	0xB4090000	0x0008038C	0xFFFFFFFF	0xFFFFFFFF
0x0830	MB_CTRL36	MBIST*n* configuration	0xB5090000	0x00080390	0xFFFFFFFF	0xFFFFFFFF
0x0840	MB_CTRL37	MBIST*n* configuration	0xB6030000	0x00080394	0xFFFFFFFF	0xFFFFFFFF
0x0850	MB_CTRL38	MBIST*n* configuration	0xB7030000	0x00080398	0xFFFFFFFF	0xFFFFFFFF
0x0860	MB_CTRL39	MBIST*n* configuration	0xB8030000	0x0008039C	0xFFFFFFFF	0xFFFFFFFF
0x0870	MB_CTRL40	MBIST*n* configuration	0xB9040000	0x000803A0	0xFFFFFFFF	0xFFFFFFFF
0x0880	MB_CTRL41	MBIST*n* configuration	0xBA040000	0x000803A4	0xFFFFFFFF	0xFFFFFFFF
0x0890	MB_CTRL42	MBIST*n* configuration	0xBB040000	0x000803A8	0xFFFFFFFF	0xFFFFFFFF
0x08A0	MB_CTRL43	MBIST*n* configuration	0xBC040000	0x000803AC	0xFFFFFFFF	0xFFFFFFFF
0x08B0	MB_CTRL44	MBIST*n* configuration	0xBD020000	0x000803B0	0xFFFFFFFF	0xFFFFFFFF
0x08C0	MB_CTRL45	MBIST*n* configuration	0xBE020000	0x000803B4	0xFFFFFFFF	0xFFFFFFFF
0x08D0	MB_CTRL46	MBIST*n* configuration	0xBF020000	0x000803B8	0xFFFFFFFF	0xFFFFFFFF
0x08E0	MB_CTRL47	MBIST*n* configuration	0xC0020000	0x000803BC	0xFFFFFFFF	0xFFFFFFFF

Table continues on the next page...

Table 4. TF settings for MBIST and LBIST (continued)

Offset from base address[1]	STCU register name	Function	Offset 0x0	Offset 0x4	Offset 0x8	Offset 0xC
0x08F0	MB_CTRL48	MBIST*n* configuration	0xC1020000	0x000803C0	0xFFFFFFFF	0xFFFFFFFF
0x0900	MB_CTRL49	MBIST*n* configuration	0xC2020000	0x000803C4	0xFFFFFFFF	0xFFFFFFFF
0x0910	MB_CTRL50	MBIST*n* configuration	0xC3030000	0x000803C8	0xFFFFFFFF	0xFFFFFFFF
0x0920	MB_CTRL51	MBIST*n* configuration	0xC4020000	0x000803CC	0xFFFFFFFF	0xFFFFFFFF
0x0930	MB_CTRL52	MBIST*n* configuration	0xC5030000	0x000803D0	0xFFFFFFFF	0xFFFFFFFF
0x0940	MB_CTRL53	MBIST*n* configuration	0xC6100000	0x000803D4	0xFFFFFFFF	0xFFFFFFFF
0x0950	MB_CTRL54	MBIST*n* configuration	0xC7100000	0x000803D8	0xFFFFFFFF	0xFFFFFFFF
0x0960	MB_CTRL55	MBIST*n* configuration	0xC8100000	0x000803DC	0xFFFFFFFF	0xFFFFFFFF
0x0970	MB_CTRL56	MBIST*n* configuration	0xC9040000	0x000803E0	0xFFFFFFFF	0xFFFFFFFF
0x0980	MB_CTRL57	MBIST*n* configuration	0x00040000	0x000803E4	0xFFFFFFFF	0xFFFFFFFF
0x0990	CFG	No cut selected	0x10000000	0x0008000C	0xFFFFFFFF	0xFFFFFFFF
0x09A0	WDGG	Set MBIST and LBIST max time	0x0000020B	0x00080010	0xFFFFFFFF	0xFFFFFFFF
0x09B0	CRCE	Expected CRC	0xB3C5BF31	0x00080014	0xFFFFFFFF	0xFFFFFFFF
0x09C0	LBFMK	LBIST fault mapping key	0xFE35AB20	0x00080038	0xFFFFFFFF	0xFFFFFFFF
0x09D0	LBCFM	LBIST CF mapping	0x00000007	0x00080030	0xFFFFFFFF	0xFFFFFFFF
0x09E0	LBSFM	LBIST stay in reset fault mapping	0x00000000	0x00080034	0xFFFFFFFF	0xFFFFFFFF
0x09F0	MBFMK	MBIST fault mapping key	0x751AC490	0x00080060	0xFFFFFFFF	0xFFFFFFFF
0x0A00	MBCFML	MBIST CF mapping for 31–0	0xFFFFFFFF	0x00080050	0xFFFFFFFF	0xFFFFFFFF
0x0A10	MBCFMH	MBIST CF mapping for 57–32	0x03FFFFFF	0x00080054	0xFFFFFFFF	0xFFFFFFFF
0x0A20	MBSFML	MBIST stay in reset fault mapping for 31-0	0x00000000	0x00080058	0xFFFFFFFF	0xFFFFFFFF
0x0A30	MBSFMH	Mbist stay in reset fault mapping for 57–32	0x00000000	0x0008005C	0xFFFFFFFF	0xFFFFFFFF
0x0A40	RUN	STCU RUN	0x00000001	0x00080000	0xFFFFFFFF	0xFFFFFFFF
0x0A50	—	STOP test flash scan, DCF STOP	0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF	0xFFFFFFFF

1. TF base address = 0x0040_0000
2. 90% LBIST coverage is supported by: STCU_LB0_PCS = 8000 STCU_LB1_PCS = 8000 STCU_LB2_PCS = 8100

NOTE

The TF checksum is stored at offset 0x3E00. Given a TF base address of 0x0040_0000, this results in a TF checksum location of 0x0040_3E00.

NOTE

Regarding DCF Stop word programming: the DCF protocol states a valid STOP WORD contains the LSB = 1. This means that 0x0000_0001, 0x0000_000F, and 0xFFFF_FFFF are all valid DCF STOP WORDs since the LSB is set in all cases. MPC5675K devices typically use 0xFFFF_FFFF to designate the DCF STOP WORD. To understand the benefits of this situation, consider the following example: you begin with a device programmed at the factory for a configuration in SF, but then you want to add more DCF

entries in the SF. Because Freescale has programmed 0xFFFF_FFFF for the STOP WORD (and not programmed a STOP WORD of 0x0000_0001), you can simply add more lines of code starting at the first free location in the record. Specifically, it is possible to overwrite the 0xFFFF_FFFF entry with a valid DCF word. On the contrary, if Freescale had programmed a STOP WORD of 0x0000_0001, you would first have to erase the SF and reprogram everything back into the SF again.

5 Revision history

Table 5. Revision History

Revision	Date	Description
Rev 0	February 29, 2012	Initial release
Rev 0.1	March 26, 2012	Added note to Section 3 Test Flash Contents for TSENSE and ADC Self Test, clarifying ADC Self Test parameters can change during production. Clarified need for customer to read values from Test Flash (do not use immediate data #defined in SW) Changed '0xFFFFFFFF' to 'Reserved' for reserved locations in Table 3 and Table 4.
Rev 1	December 7, 2012	Changed title to include PXS30.

How to Reach Us:

Home Page:
www.freescale.com

Web Support:
http://www.freescale.com/support

USA/Europe or Locations Not Listed:
Freescale Semiconductor
Technical Information Center, EL516
2100 East Elliot Road
Tempe, Arizona 85284
+1-800-521-6274 or +1-480-768-2130
www.freescale.com/support

Europe, Middle East, and Africa:
Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
www.freescale.com/support

Japan:
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064
Japan
0120 191014 or +81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:
Freescale Semiconductor China Ltd.
Exchange Building 23F
No. 118 Jianguo Road
Chaoyang District
Beijing 100022
China
+86 10 5879 8000
support.asia@freescale.com

Document Number: AN4422
Rev. 1, 12/2012