Semiconductor Products Sector
Application Note

AN2188

Compatibility considerations between the 0.65µ ROM M68HC12D60 and the 0.5µ Flash M68HC912D60A

1 Introduction

Designs using the 0.65µ M68HC12D60 mask ROM device may require support from a Flash device either as an emulation tool during development and/or to provide flexibility in production where the Flash device may be used to bridge any unexpected short term increases in production demand.

The 0.65µ Flash M68HC912D60 is currently being superseded by the 0.5µ Flash M68HC912D60A.

The purpose of this document is to help designers maintain compatibility when using the 0.5µ Flash M68HC912D60A (referred to as 'Flash device') as an emulation tool / socket replacement for the 0.65µ mask ROM M68HC912D60 (referred to as 'ROM device').

This document is intended for use in conjunction with the most current data sheets and mask set errata for each device. At the time of writing these documents were available for download from the Freescale web site* or they can be requested from your Freescale distributor. Please check these documents for any new information.

* http://e-www.freescale.com.

Table 1: Documentation Order Numbers

Document Number	For Device
MC68HC912D60/D	MC68HC12D60 (document also covers MC68HC912D60)
MC68HC912D60A/D	MC68HC912D60A

Table 2: MC68HC12D60 Device Ordering Information

Package	Temperature		Voltage	Frequency	Order Number
	Range	Designator			
112-Pin TQFP Single Tray 60 Pcs	0 to +70°C		4.5V–5.5V	8 MHz	XC68HC12D60PV8
	–40 to +85°C	C			XC68HC12D60CPV8
	–40 to +105°C	V			XC68HC12D60VPV8
	–40 to +125°C	M			XC68HC12D60MPV8
80-Pin QFP Single Tray 84 Pcs	0 to +70°C		4.5V–5.5V	8 MHz	XC68HC12D60FU8
	–40 to +85°C	C			XC68HC12D60CFU8
	–40 to +105°C	V			XC68HC12D60VFU8
	–40 to +125°C	M			XC68HC12D60MFU8

Table 3: MC68HC912D60A Device Ordering Information

Package	Temperature		Voltage	Frequency	Order Number
	Range	Designator			
112-Pin TQFP Single Tray 60 Pcs	–40 to +85°C	C	4.5V–5.5V	8 MHz	MC912D60ACPV8
	–40 to +105°C	V			MC912D60AVPV8
	–40 to +125°C	M			MC912D60AMPV8
80-Pin TQFP Single Tray 84 Pcs	–40 to +85°C	C	4.5V–5.5V	8 MHz	MC912D60ACFU8
	–40 to +105°C	V			MC912D60AVFU8
	–40 to +125°C	M			MC912D60AMFU8

Summary

Most aspects of the two devices are fully compatible. Where there are functional differences simple recommendations are described which should provide compatibility.

The majority of differences relate to the Non-Volatile Memory (NVM) implementation.

The ATD was also modified on the Flash device for improved flexibility when selecting the conversion source or the results location & format. Most additional features default to being compatible with the ROM.

All registers in the ROM device are supported in the Flash device and all other peripheral modules are unchanged.

There are no mask option registers and all of the device parameters are configured by the application software.

On the ROM device, control of ‘BDM lockout’ is located in a shadow EEPROM byte.

On the Flash device, control of ‘BDM lockout’ and the default EEPROM pre-scaler value (Section 3.2) are located in a shadow EEPROM word.

2 Code Storage

On the ROM device, code is stored in static, factory masked, Read Only Memory (ROM).

On the Flash device, code is stored in non-volatile, electrically erasable and programmable memory constructed from split-gate flash technology (Flash EEPROM).

Reading from the ROM or Flash memories is identical. The equivalent array sizes are equivalent (32K and 28K) and the MAPROM, ROMON28 & ROMON32 bits in the MISC control register, for swapping and disabling the memory arrays, function identically.

2.1 Flash Architecture

The Flash memory is ‘5V only’ so no separate external programming voltage (VFP) is required. Programming is carried out on a whole row (64 bytes) at a time whilst erase is carried out as a bulk erase of each entire array (32K or 28K).

To reduce the possibility of accidental modification of the Flash contents, code for programming / erasing the Flash should not be included in any application (as the ROM device can’t be reprogrammed this should not be an issue). Programming should be carried out by downloading a boot loader program (typically via the BDM connection) into RAM which can then be executed to modify the flash – this is how most 3rd party programmers work.

2.2 Flash Control Registers

A 4-byte register block for each module controls the Flash EEPROM operation. At reset, the 4-byte register section starts at address $00F4/$00F8 for the 28K & 32K modules respectively.

On the ROM device these registers ($00F4 to $00FF) are reserved and therefore should not be accessed.

On the Flash device, for best practice, the application may choose to access two of these registers although this is not essential for compatibility with the ROM:

i. Initializing the FEExxLCK (Flash EEPROM Lock Control) Register.

FEE32LCK/FEE28LCK — Flash EEPROM Lock Control Register $00F4/$00F8

0	0	0	0	0	0	0	LOCK

Each Flash EEPROM module has hardware interlocks which protect stored data from accidental corruption. An erase- and program-protected 8-Kbyte ‘boot block’ for protected routines is located at $6000–$7FFF or $E000–$FFFF following reset. Note that the upper two rows (128 bytes) of these memory arrays contains the Reset and interrupt vectors. System critical and fault recovery routines, including

reset and unused interrupt service routines, can be stored in these locations protected from accidental corruption.

Protection of the ‘boot block’ is controlled by the BOOTP bit in the FEExxMCR (Flash EEPROM Module Configuration) register. The reset default is ‘protected’. To avoid this state being accidentally modified, setting the LOCK bit in the ‘write once’ FEExxLCK register prevents any accidental writes to the FEExxMCR register from unprotecting the ‘boot block’.

Summary:

To lock the ‘boot block’ protection mechanism set bit0 (LOCK) in the FEExxLCK register of each Flash module during initialization. (Optional)

ii. Initialising the FEExxLCK (Flash EEPROM Control) Register.

FEE32LCK/FEE28LCK — Flash EEPROM Lock Control Register $00F4/$00F8

0	0	0	FEESWAI	HVEN	0	ERAS	PGM

The clocks to the Flash modules can be optionally disabled on entering WAIT mode to reduce power consumption (all the clock drivers get disabled and the module becomes fully static). In order for this to happen the FEESWAI bit in the FEExxCTL register must be set.

CAUTION: *As this register controls the write / erase function of the flash extreme care must be taken not to write it incorrectly.*

Summary:

To reduce current consumption in WAIT mode set Bit4 (FEESWAI) in the FEExxMCR register of each Flash module during initialization. (Optional)

2.3 Flash Programming Procedure

Programming of the Flash NVM is greatly simplified over previous Flash based HC12s. The read / verify / re-pulse programming algorithm is replaced by a much simpler method. This should only be carried out as an externally controlled process i.e. external from the application.

2.4 Flash Programming Voltage

The Flash does not require an external high voltage supply. All voltages required for programming and erase are generated internally. On early 0.5µ Flash devices Pin 97 (112 QFP) or pin 71 (80 QFP) is used for factory test purposes. It is recommended that this pin is not connected within the application, but it may be connected to VSS or 5.5V max without issue. On later production devices this pin is not bonded out.

On the ROM device this pin is not bonded.

Summary:

For compatibility leave Pin 97 (112 QFP) or pin 71 (80 QFP) unconnected in any application.

3 EEPROM NVM data

3.1 EEPROM Architecture

On the Flash device, the EEPROM is also made from split-gate NVM which is physically constructed in a different way from the EEPROM array on the ROM device.

In order to achieve object code compatibility a state machine is implemented on the Flash device EEPROM module to make the difference transparent. The architecture and basic programming & erase operations are unchanged at the application programming level. However, there is an optional AUTO programming method that allows faster programming of the EEPROM - for compatibility reasons this should not be used. It is included here for completeness and because the additional AUTO bit on the Flash device allows an application to identify between the Flash and the ROM device ‘in circuit’.

3.2 Flash device EEPROM Clock Source and Pre-scaler

A key difference on the Flash device is that the EEPROM module requires a constant time base to ensure correct programming and erase operations. To cater for the large range of input clock frequencies found in applications, this time base (around 28.6KHz) is derived from the external clock signal (EXTALi) divided by a pre-scaler. The divide ratio for this source must be configured by programming a 10-bit time base pre-scalar value into two new registers, EEDIVH and EEDIVL (referred to as a single word register 'EEDIV') to produce a 35µs +/- 2µs timebase.

EEDIVH — EEPROM Modulus Divider $00EE

0	0	0	0	0	0	EEDIV9	EEDIV8

EEDIVL — EEPROM Modulus Divider $00EF

EEDIV7	EEDIV6	EEDIV5	EEDIV4	EEDIV3	EEDIV2	EEDIV1	EEDIV0

As a result of the EEPROM clock requirement the resonator / crystal frequency tolerance should be better than 2% total for < 2MHz, 3% total for >= 2MHz. This is well within the requirements for most applications and certainly for those using the CAN communications bus.

EEDIVH/L values are determined by the following formula:

$$EEDIVH/L = INT[\text{reference clock (Hz)} \times 35 \times 10^{-6} + 0.5]$$

INT[x] denotes the round down integer value of x.

For example if the Reference Frequency is 4.0MHz, the EEDIV value from the above formula will be INT[140.5] = 140 and hence the module clock will be $4.0x10^6$ / 140 = 28.571KHz in frequency equivalent to a period of 35.00µs .

EXTALi Freq	Clock Period	DivideFactor	EEDIV value
8 MHz	125ns	280	$0118
4 Mhz	250ns	140	$008C
2 Mhz	500ns	70	$0046
1 Mhz	1µs	35	$0023
500 Khz	2µs	18	$0012

The appropriate value can easily be calculated by the 'C' pre-processor e.g.

```
#define XTAL_FREQ    8.0
        /* macro value for calculating EEDIV value */
        /* rounded down by preprocessor and only   */
        /* used as unsigned integer */

#define EEDIVIDER_VALUE ((XTAL_FREQ * 35.0) + 0.5)
```

The EEDIV register is volatile. However, it is loaded upon reset by the contents of the non-volatile SHADOW word in much the same way as the NOBDML bit in the EEPROM module control register (EEMCR) interacts with the SHADOW word for locking out the BDM on the ROM.

Summary:

One (or both) of the following 2 methods for setting the EEDIV value MUST be implemented before programming or erasing any EEPROM location as an incorrect value will damage the EEPROM.

1. program the shadow word with an appropriate divider value (proportional to the application's XTAL value) at the same time as the Flash is programed. Care must be taken to ensure that the other bits of the shadow word are programmed to correct values.
2. over-write the reset value of the EEDIV 'write once only' register with an appropriate divider value in the application initialization routine.

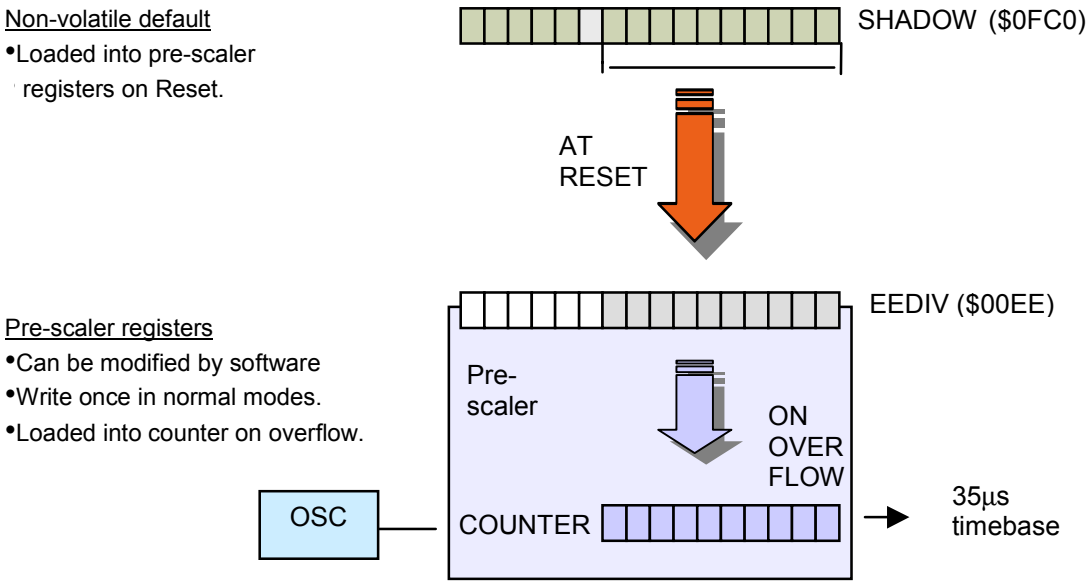

Figure 1: Loading the EEDIV value

3.3 EEPROM AUTO bit – Detecting whether its a ROM or Flash device

Another difference on the Flash device EEPROM module is the inclusion in the EEPROG (EEPROM control) register of an additional AUTO bit (Bit-5).

EEPROG — EEPROM Control $00F3

BULKP	0	**AUTO**	BYTE	ROW	ERASE	EELAT	EEPGM

On the ROM device this bit is unused.

As the AUTO functionality is not available on the ROM, the AUTO bit should not be set in order to ensure compatible program or erase.

However, the AUTO bit can be tested in the application to detect whether its running on a ROM or a Flash device.

On the Flash device this (AUTO) bit can be set and cleared in any mode, hence attempts to set and clear it will be successful on the Flash device where it will read as ‘1’ or ‘0’ as appropriate.

On the ROM, Bit-5 of the EEPROG register is unused and will always read as ‘0’.

Summary:

The application software should ensure that bit-5 of the EEPROG register is always cleared during normal EEPROM functions.

3.3.1 ‘C’ Code example for determining the device type

This function returns TRUE if Flash, FALSE if ROM

```
int
testFlashDevice(void)
{
                /* attempt to set Bit-5 of $00F3 */
     Eeprom.eeprog.bit.autobit = 1;
     if(Eeprom.eeprog.bit.autobit == 1){
                /* the bit set so  */
                /* it’s a 0.5u Flash device */
                /* clear the AUTO bit again */
          Eeprom.eeprog.bit.autobit = 0;
          return( TRUE );
     } else
                /* the bit failed to set so */
                /* it’s a 0.65u Flash / ROM device */
          return( FALSE );
}
```

This is useful for determining whether to write the EEDIV value in an application e.g.

```
     If( testFlashDevice() == TRUE )
                /* write EEDIVIDER at $00EE */
          Eeprom.eediv = (unsigned int)EEDIVIDER_VALUE & 0x03FF;
                /* else its a ROM so do nothing */
```

In this case the EEDIV register is only written if the application is running on a 912D60A (but not on the ROM). This routine can also be extended to initialize the flash registers as described in Section 2.2.

3.4 When Entering Limp Home Mode

On the Flash device, if Limp Home is entered during a program or erase cycle, i.e. with EEPGM bit = 1, the state machine will complete the cycle with EEDIV forced to a nominal value of $0023 (reflecting fVCOMIN typical = 1MHz).

Program and erase of the EEPROM is not guaranteed in Limp home mode. This is a valid restriction for both ROM and Flash devices as the Limp Home Frequency (f_{VCOMIN}) is not tightly specified. On the Flash device it is very important not to program or erase the EEPROM with an undefined clock (outside the specification described in Section 3.2) in order to avoid damaging the EEPROM.

Summary:

It is recommended to abort any EEPROM modify operations immediately on entry to Limp Home Mode by clearing the EEPGM bit and then EELAT bit in the EEPROM Control Register.

3.5 Clock Monitor

The Clock Monitor (CME bit) should be enabled during program/erase in order to avoid damaging the EEPROM. On a clock failure, in the case of a Clock Monitor reset sequence any EEPROM modify in progress will be aborted (as the EEPGM bit and the EELAT bits in the EEPROM Control Register are cleared by reset) thus protecting the cell(s). This is a valid strategy for both Flash and ROM devices.

Summary:

Use the Clock Monitor to detect when the clock goes out of spec in order to terminate EEPROM functions either due to a reset process or on entry into Limp Home mode.

3.6 EERC bit

On the Flash device, the EEPROM module RC oscillator is not implemented and the EERC bit is replaced with a dummy bit (DMY) with no functionality.

EEMCR – EEPROM Module Configuration - ROM Device $00F0

NOBDML	NOSHW	rsv	rsv	1	EESWAI	PROTLCK	**EERC**

EEMCR – EEPROM Module Configuration - Flash Device $00F0

NOBDML	NOSHW	rsv	rsv	1	EESWAI	PROTLCK	**DMY**

On the ROM device, for bus frequencies < f_{PROG} (i.e. <1MHz in the current spec = 2MHz EXTALi without PLL) the EERC bit has to be set to enable the RC oscillator in the EEPROM module for program and erase.

On the Flash device there is no RC as the new EEPROM cell and state machine can program and erase at lower frequencies. However, in order to meet the EEPROM time base specification, program/erase should not be performed with an input clock frequency EXTALi <250 KHz.

Setting the replacement DMY bit on the Flash device has no effect so there are no software compatibility issues.

Summary:

No compatibility issue.

3.7 Flash device EEPROM Module Selective Write More Zeros

The new split-gate construction allows an additional technique referred to as ‘Selective Write More Zeros’, described in the M68HC912D60A data book. As this functionality is not specified on the ROM this technique should not be used.

Summary:

All programming and erase cycles should be considered independent and successive writes to any EEPROM location should ALWAYS be preceded by an erase cycle.

4 STOP mode

The Flash device will exit STOP mode without having to synchronize the start of STOP with the RTI clock.

On the ROM device this is an errata.

Summary:

Implement the work around for the ROM errata. This will not cause any problems on the 912D60A.

5 WAIT mode

The Flash device will correctly exit WAIT mode using short XIRQ or IRQ inputs.

On the ROM device this is an errata.

Summary:

Implement the work around for the ROM errata. This will not cause any problems on the 912D60A.

6 KWU Filter

On the Flash device the KWU filter will now ignore pulses shorter than 2 microseconds.

On the ROM device this is an errata.

Summary:

Implement the work around for the ROM errata. This will not cause any problems on the 912D60A.

7 ATD

CAUTION: *On both the ROM and the Flash, power must be applied to VDDA at all times even if the ADC is not being used. This is necessary for port AD0 and port AD1 to function correctly as digital inputs and is in line with the specification of both devices.*

7.1 Operation of CA, CB & CC bits in the ATDxCTL5 Control Register

There is a difference in the operation of the ATD module between the ROM device and the Flash device when performing multichannel conversions (MULT bit is set and on the Flash Device the new S1C bit is cleared - see below), i.e. with 8 or 4 conversions making up a conversion sequence.

ATD0CTL5/ATD1CTL5 — ATD Control Register 5 $0065/$01E5

0	S8CM	SCAN	MULT	**CD**	CC	CB	CA

ATD0CTL5/ATD1CTL5 — ATD Control Register 5 $0065/$01E5

0	S8CM	SCAN	MULT	**SC**	CC	CB	CA

On the ROM, the value of the CC, CB & CA bits are masked depending on the value of the S8CM & CD bits. The value of the CC, CB & CA bits doesn't affect where the conversion results are stored. The CD bit selects the internal channels for conversion. See **Table 4**.

Table 4: Multichannel Mode Result Register Assignment

S8CM	CD	CC	CB	CA	Channel Signal	Result in ADRxx if MULT = 1
0	0	0	0	0	AN0	ADRx0
			0	1	AN1	ADRx1
			1	0	AN2	ADRx2
			1	1	AN3	ADRx3

Table 4: Multichannel Mode Result Register Assignment

S8CM	CD	CC	CB	CA	Channel Signal	Result in ADRxx if MULT = 1
0	0	1	0	0	AN4	ADRx0
			0	1	AN5	ADRx1
			1	0	AN6	ADRx2
			1	1	AN7	ADRx3
0	1	0	0	0	Reserved	ADRx0
			0	1	Reserved	ADRx1
			1	0	Reserved	ADRx2
			1	1	Reserved	ADRx3
0	1	1	0	0	V_{RH}	ADRx0
			0	1	V_{RL}	ADRx1
			1	0	$(V_{RH} + V_{RL})/2$	ADRx2
			1	1	TEST/Reserved	ADRx3
1	0	0	0	0	AN0	ADRx0
		0	0	1	AN1	ADRx1
		0	1	0	AN2	ADRx2
		0	1	1	AN3	ADRx3
		1	0	0	AN4	ADRx4
		1	0	1	AN5	ADRx5
		1	1	0	AN6	ADRx6
		1	1	1	AN7	ADRx7
1	1	0	0	0	Reserved	ADRx0
		0	0	1	Reserved	ADRx1
		0	1	0	Reserved	ADRx2
		0	1	1	Reserved	ADRx3
		1	0	0	V_{RH}	ADRx4
		1	0	1	V_{RL}	ADRx5
		1	1	0	$(V_{RH} + V_{RL})/2$	ADRx6
		1	1	1	TEST/Reserved	ADRx7

Shaded bits are “don’t care” if MULT = 1 and the entire block of four or eight channels make up a conversion sequence. When MULT = 0, all four bits (CD, CC, CB, and CA) must be specified and a conversion sequence consists of four or eight consecutive conversions of the single specified channel.

On the Flash device the value of the CC, CB & CA bits is used to provide greater flexibility allowing the application to select which of the eight ATD input channels is sampled first in a sequence. As a result of this they also effectively modify where the result of the conversions are stored – the first conversion in the sequence is stored in the lowest results register. The CD bit has been renamed as SC (Special Conversion) to differentiate its functionality from the Cx bits. The SC bit still selects the internal channels for conversion but note that the CC, CB & CA bits continue to modify which internal channel is converted first. See **Table 5**.

Table 5: MC68HC912D60A Multichannel Mode Result Register Assignment (MULT=1)

4 channel conversion, External channels (S8C = 0, SC = 0)

CC	**0**	0	0	0	**1**	1	1	1
CB	**0**	0	1	1	**0**	0	1	1
CA	**0**	1	0	1	**0**	1	0	1
ADR0	**AN0**	AN1	AN2	AN3	**AN4**	AN5	AN6	AN7
ADR1	**AN1**	AN2	AN3	AN4	**AN5**	AN6	AN7	AN0
ADR2	**AN2**	AN3	AN4	AN5	**AN6**	AN7	AN0	AN1
ADR3	**AN3**	AN4	AN5	AN6	**AN7**	AN0	AN1	AN2

S1C bit must be clear.

4 channel conversion, Internal sources (S8C = 0, SC = 1)

CC	0	0	0	0	**1**	1	1	1
CB	0	0	1	1	**0**	0	1	1
CA	0	1	0	1	**0**	1	0	1
ADR0					**VRH**	VRL	MID	
ADR1				VRH	**VRL**	MID		
ADR2			VRH	VRL	**MID**			
ADR3		VRH	VRL	MID				

Shaded cells are reserved
MID = (VRH + VRL) / 2
S1C bit must be clear.

Table 5: MC68HC912D60A Multichannel Mode Result Register Assignment (MULT=1)

8 channel conversion, External channels (S8C = 1, SC = 0)

CC	**0**	0	0	0	1	1	1	1
CB	**0**	0	1	1	0	0	1	1
CA	**0**	1	0	1	0	1	0	1
ADR0	**AN0**	AN1	AN2	AN3	AN4	AN5	AN6	AN7
ADR1	**AN1**	AN2	AN3	AN4	AN5	AN6	AN7	AN0
ADR2	**AN2**	AN3	AN4	AN5	AN6	AN7	AN0	AN1
ADR3	**AN3**	AN4	AN5	AN6	AN7	AN0	AN1	AN2
ADR4	**AN4**	AN5	AN6	AN7	AN0	AN1	AN2	AN3
ADR5	**AN5**	AN6	AN7	AN0	AN1	AN2	AN3	AN4
ADR6	**AN6**	AN7	AN0	AN1	AN2	AN3	AN4	AN5
ADR7	**AN7**	AN0	AN1	AN2	AN3	AN4	AN5	AN6

8 channel conversion, Internal Sources (S8C = 1, SC = 1)

CC	**0**	0	0	0	1	1	1	1
CB	**0**	0	1	1	0	0	1	1
CA	**0**	1	0	1	0	1	0	1
ADR0					VRH	VRL	MID	
ADR1				VRH	VRL	MID		
ADR2			VRH	VRL	MID			
ADR3		VRH	VRL	MID				
ADR4	**VRH**	VRL	MID					
ADR5	**VRL**	MID						VRH
ADR6	**MID**						VRH	VRL
ADR7						VRH	VRL	MID

Shaded cells are reserved
MID = (VRH + VRL) / 2

NOTES:
1) For compatibility with the 68HC912D60, CA, CB, CC bits must be '0' where masked on the 68HC912D60. This is shown above in bold text.
2) When MULT = 0, all four bits (SC, CC, CB, and CA) must be specified and a conversion sequence consists of four or eight consecutive conversions of the single specified channel.
3) When S8C = 0 and S1C = 1, all four bits (SC, CC, CB, and CA) must be specified and a conversion sequence consists of one conversion of the single specified channel.

Summary:

For compatibility between Flash and ROM the software should ensure that the relevant Cx bits in the ATDxCTL5 register are cleared when performing ATD conversions with MULT bit set (as shown in **bold text** in **Table 5**).

7.2 Additional ATD features on the Flash device

An extra feature is available on the Flash M68HC912D60A for controlling the justification of conversion results - enabled by an additional bit in the ATDxCTL2 register. This features should not be used to ensure compatibility .

ATD0CTL2/ATD1CTL2 — ATD Control Register 2 $0062/$01E2

ADPU	AAFC	ASWAI	**DJM**	Reserved	Reserved	ASCIE	ASCIF

i. The DJM (Result Register Data Justification Mode) bit controls whether the result registers are left or right justified.
Bit = 0 – Result is Left justified (as per the ROM)

Summary:

The default setting (bit = 0) ensures compatibility between the ROM device and the Flash device. As the other bits in this register are usually written only once at initialization this is not expected to be a problem.

Ensure this additional bit is cleared in any writes to this register for compatibility.

NOTE: *Bit AWAI in ATDxCTL2 of the ROM device is renamed ASWAI on the Flash device compatible with M68HC912DT128A - functionality is unchanged.*

NOTE: I*n initial documentation of the flash device, bit 3 of this register was additionally defined as DSGN, for controlling sign extension of the results. This is incorrect and the bit should be always written as zero*.

7.3 Additional ATD Conversion Sequence Controls on the Flash Device

Extra features are available on the Flash M68HC912D60A for controlling the sequence and location of conversions – enabled by two additional

bits in the ATDxCTL3 register. These features should not be used to ensure compatibility.

ATD0CTL3/ATD1CTL3 — ATD Control Register 3 $0063/$01E3

0	0	0	0	**S1C**	**FIFO**	FRZ1	FRZ0

i. The S1C (Conversion Sequence Length – lsb) bit interacts with control bit S8C in the ATDCTL5 register to determine how many conversions are performed per sequence.
Bit = 0 – Sequence length defined by S8C bit (as per the ROM)
Bit = 1 – if S8C = 0 then Sequence length = 1 conversion

ii. The FIFO (Result Register FIFO Mode) bit controls whether the result registers map to the conversion sequence or not i.e. whether the result register counter is reset at the beginning or ending of a conversion sequence.

NOTE: *The correct bit definition is:*

Bit = 0 – result registers map to the conversion sequence (as per the ROM)

Bit = 1 – result registers do not map to the conversion sequence

In some early device documentation these bit definitions were incorrectly swapped.

Summary:

For both features the default setting (bit = 0) ensures compatibility between the ROM device and the Flash device. As the other bits in this register are usually written only once at initialization this is not expected to be a problem.

Ensure these additional bits are cleared in any writes to this register for compatibility.

ATD0CTL4/ATD1CTL4 — ATD Control Register 4 $0064/$01E4

RES10	SMP1	SMP0	PRS1	PRS1	PRS1	PRS1	PRS0

Bit S10BM (Select 10 Bit Mode) in ATDxCTL4 of the ROM device is renamed RES10 (Resolution 10 bits) on the Flash device, compatible with M68HC912DT128A - functionality is unchanged.

ATD0CTL5/ATD1CTL5 — ATD Control Register 5 $0065/$01E5

0	**S8C**	SCAN	MULT	**SC**	**CC**	**CB**	**CA**

On the Flash device any channel can be selected for the first conversion of a multiple channel conversion. Bits CA, CB & CC in ATDxCTL5 do not get masked, as they do on the ROM device, but are used to select which channel is used to start the sequential conversion sequence. For compatibility ensure that the appropriate bits are cleared in the software. See Section 7.1.

Bit CD in ATDxCTL5 of the ROM device is renamed SC (Special Conversion) on the Flash device to differentiate it from the extended functionality of Bits CA, CB & CC. Functionality is unchanged as it still selects conversion from internal reference sources but when doing a multiple channel scan bits CA, CB & CC must be cleared as appropriate for compatible reference selection. See Section 7.1.

Bit S8CM (Select 8 Channel Mode) in ATDxCTL5 of the ROM device is renamed S8C (Conversion Sequence Length) on the Flash device to highlight the extended functionality of the new S1C bit in ATDxCTL3 (see Section 9.1). Functionality is compatible with S8CM if the new S1C bit = 0 (its default value).

On the Flash device, in SCAN mode (SCAN bit = 1 in ATDxCTL5) the Sequence Complete Flag (SCF bit in ATDSTATx) is set after completion of each conversion sequence. On the ROM device it is only set at the end of the first conversion sequence. To ensure compatibility the application should not rely on this flag being set only once per SCAN mode.

7.4 Starting and Aborting of ATD Conversion Sequences on Register writes

On the ROM device writing to ATDxCTL2 or ADTxCTL3 aborts any ongoing sequence leaving the ATD in idle mode (no conversion sequence being processed). On the Flash device, writing these registers does not abort an ongoing conversion sequence.

On the ROM device writing to the ATDxCTL4 register aborts any ongoing sequence leaving the ATD in idle mode (no conversion sequence being processed). On the Flash device, writing this register aborts any ongoing conversion sequence and initiates a new conversion sequence.

This is unlikely to be a compatibility issue as applications mostly write these registers to configure the ATD, closely followed by a write to the ATDxCTL5 register to initiate a new conversion sequence which aborts any ongoing conversion sequence and resets appropriate flags.

Summary:

To ensure compatibility the application should not rely on ongoing conversions being aborted by writes to registers other than ATDxCTL5. Any interrupts from the completion of any sequence should be masked and/or handled correctly to allow for the fact that writing ATDxCTL4 may have started the conversion.

7.5 Other ATD Changes

Reading the ATDTESTx register in normal modes returns the value of the Successive Approximation Register (SAR).

Previously it always read as zero.

Summary:

To ensure compatibility this register should not be relied upon to read as zero.

The RST bit in the ATDTESTx register can be written in normal modes (in order to reset the ATD). Previously it was read only.

Summary:

To ensure compatibility this register should not be read or written to.

This Page Has Been Intentionally Left Blank

How to Reach Us:

Home Page:
www.freescale.com

E-mail:
support@freescale.com

USA/Europe or Locations Not Listed:
Freescale Semiconductor
Technical Information Center, CH370
1300 N. Alma School Road
Chandler, Arizona 85224
+1-800-521-6274 or +1-480-768-2130
support@freescale.com

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)
support@freescale.com

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 Hong Kong Ltd.
Technical Information Center
2 Dai King Street
Tai Po Industrial Estate
Tai Po, N.T., Hong Kong
+800 2666 8080
support.asia@freescale.com

For Literature Requests Only:
Freescale Semiconductor Literature Distribution Center
P.O. Box 5405
Denver, Colorado 80217
1-800-441-2447 or 303-675-2140
Fax: 303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com