Mask Set Errata for Mask XN33G

This report applies to mask XN33G for these products:

- MC56F82xxx

Table 1. Errata and Information Summary

Errata ID	Errata Title	Mask [1]			
		0N33G	1N33G	2N33G	3N33G
e6279	320psec PWM resolution (PWMA_SMnFRACVALx) doesn't work properly.	X			
e6273	ADC produces incorrect result when used in sequential (any of once/loop/triggered) mode (CTRL1[SMOD]).	X			
e5866	EOSI asertion delayed one conversion time after each loop mode iteration.	X			
e5873	GPIOA2,B2,A3 and B3 will not pull to external pull of 5V in open drain GPIO output mode and gpio_data register 1 mode.	X			
e6070	I2C: Repeat start cannot be generated if the I2Cx_F[MULT] field is set to a non-zero value.	X			
e6753	ANB1/CMPB_IN0 not working as per expectations.	X	X		
e6329	The scan halt interrupt function [ADC_SCHLTEN] is not working as expected.	X	X		
e11484	POR: Residual voltage on VDD may cause POR unsuccessful.	X	X	X	
e9432	eFlexPWM: Fractional delay block may power up with an output of 1 instead of 0.	X	X	X	X

1. 'X' represents the errata present in the mask.

Table 2. Revision History

Revision	Changes
24 Oct 2013	Public release revision.

Table continues on the next page...

Table 2. Revision History (continued)

Revision	Changes
02 Aug 2018	The following errata added: • e9432 • e11484 (fixed in mask 3N33G)
Dec 2020	Workaround description updated in the following errata: • e11484

e6279: 320psec PWM resolution (PWMA_SMnFRACVALx) doesn't work properly.

Description: When PWMA_SMnFRACVALx register is used to configure PWM output for 320psec resolution, the PWM outputs are not stable and produce incorrect resolution and jitter at PWM output.

Workaround: No workaround.

Do not enable fractional value registers (PWMA_SMnFRCTRL), hence minimum PWM resolution for 1.0 device would be 10 ns.

e6273: ADC produces incorrect result when used in sequential (any of once/loop/triggered) mode (CTRL1[SMOD]).

Description: ADC produces incorrect result when used in sequential (any of once/loop/triggered) mode (CTRL1[SMOD]), when input sample switches from one converter to another converter for the first time.

Case1: ADC working at >5Mhz ADC CLOCK (CTRL2[DIV0]): if both converters A and B are used. the first sample (ADC_CLISTx[samplex])after switching from one converter to other produces error needs to be sampled twice. For example,

- example(i): if the required sequence is sample0->ANA0, sample1->ANA1, sample2->ANB0, sample3->ANB1, sample4->ANA2, sample5->ANB2. In the given sequence only sample2->ANB0 will produce incorrect result.
- example(ii):if the required sequence is sample0->ANB0, sample1->ANB1, sample2->ANA0, sample3->ANA1, sample4->ANB2, sample5->ANA2. in the given sequence only sample2->ANA0 will produce incorrect result.

Case 2: ADC working at <5 MHz ADC CLOCK (CTRL2[DIV0]): if both converter A and B are used. Second ADC converter in sequence produces incorrect result.

Workaround: Workaround for Case1:

- example(i): ANB0 need to sampled twice and sample2->ANB0 should be thrown out. Hence the suggested sample sequence should be sample0->ANA0, sample1->ANA1, sample2->ANB0, sample3->ANB0, sample4->ANB1, sample5->ANA2, sample6->ANB2.
- example(ii): ANA0 need to sampled twice and sample2->ANA0 should be thrown out. Hence the suggested sample sequence should be sample0->ANB0, sample1->ANB1, sample2->ANA0, sample3->ANA0, sample4->ANA1, sample5->ANB2, sample6->ANA2.

Workaround for Case2: Use both ADCs in parallel (any of once/loop/triggered) mode (CTRL1[SMOD]).

e5866: EOSI asertion delayed one conversion time after each loop mode iteration.

Description: EOSI asertion delayed one conversion time after each loop mode iteration

Workaround: There are several possible workarounds for this issue if EOSI interrupt is using to read the result registers after each loop iteration.

1. Account for extra latency in ISR

   When EOSI asserts at the end of each loop iteration, the software can read the result registers out of order by starting with the 2nd or 3rd sample of the scan, which will allow the 1st sample extra time to get updated. The drawback is that the first 1-2 samples will need to be thrown out as they will get overwritten by the time the ISR has a chance to read them.

2. Poll the ADC_RDY[RDY] bits for the completion of the last sample of the loop iteration. You may need to experiment to find the right scan length to give you enough time to read the conversion results from the RSLT registers.

e5873: GPIOA2,B2,A3 and B3 will not pull to external pull of 5V in open drain GPIO output mode and gpio_data register 1 mode.

Description: When using the pins muxed on GPIOA2,A3,B2,B3 as GPIO in open drain mode and output enabled but floating from Design (Open drain mode, output enable and write 1 to GPIO Data register) with an external pull up of 5V, then the voltage coming on those pads would limit to around 3.9V. This is related to design internal connection to ADC which when enabled keep inputs limited to less than 3.3V and hence does not allow design to pull to external voltage.

Workaround: These particular GPIOs the functionality is guaranteed only till VDDA. For 5V operation, user should use the other IOs.

e6070: I2C: Repeat start cannot be generated if the I2Cx_F[MULT] field is set to a non-zero value.

Description: If the I2Cx_F[MULT] field is written with a non-zero value, then a repeat start cannot be generated.

Workaround: There are two possible workarounds:

1. Configure I2Cx_F[MULT] to zero if a repeat start has to be generated.
2. Temporarily set I2Cx_F [MULT] to zero immediately before setting the Repeat START bit in the I2C C1 register (I2Cx_C1[RSTA]=1) and restore the I2Cx_F [MULT] field to the original value after the repeated start has occurred.

e6753: ANB1/CMPB_IN0 not working as per expectations.

Description: For GPIOB1(ANB1/CMPB_IN0), when used as an ADC channel, ANB1 will be degraded as compared to the other analog channels; further when used as CMPB_IN0, reduced performance will be seen at output of comparator.

Workaround: Suggest to use another analog pin for the same purpose.

If must use, use offset value for ANB1 conversion (the offset value might be different for this channel for the same input voltage).

e6329: The scan halt interrupt function [ADC_SCHLTEN] is not working as expected.

Description:

1. When user intends to "trigger scan of sample0 ~ sample2 first and then halt, generating an interrupt, waiting for the next trigger signal to trigger scan of sample3 ~ sample7". To achieve the same for example, the user configures SC[3] bit of ADC_SCTRL to make the scan halt after completing scan of sample0 to sample2 and SCHLTEN[2] of ADC_SCHLTEN is set to get interrupt when channel 2 is halted.

   In above case, the scan is halted at Channel2 but no interrupt is generated at end of channel2 scanning.

2. To achieve the same for example, the user configures SC[3] bit of ADC_SCTRL to make the scan halt after completing scan of sample0 to sample2 and in case user sets SCHLTEN[2] and SCHLEN[3] of ADC_SCHLTEN is set to get interrupt when channel 2 is halted.

   In above case, the scan is halted at Channel2 but no interrupt is generated at end of channel2 scanning.

   Once channel 3 is retriggered after being halted, the scan halt interrupt is generated at end of channel3 conversion, which is unexpected.

Workaround: No workaround.

e9432: eFlexPWM: Fractional delay block may powerup with an output of 1 instead of 0.

Description: Any of the eFlexPWM outputs may be set to 1 upon powering up the fractional delay block. Because there is no reset signal to the flops in the fractional delay block to force a specific reset state, the output must be cleared by creating a pulse on the PWM. This issue only occurs when using the fractional delay block and only lasts until the first time that PWM channel transitions.

Workaround: After powering up the fractional delay block of the eFlexPWM by setting FRCTRL[FRAC_PU] and waiting the required power up time, program the VAL2-5 registers in all submodules to create a PWM pulse (>0% duty cycle) and run for at least one PWM period to clear the state of the registers in the analog block. This can be done prior to enabling the PWM outputs so that external circuitry is not affected.

e11484: POR: Residual voltage on VDD may cause POR unsuccessful.

Description: When powered on with residual voltage larger than 0.2 V on VDD, there is a chance that POR signal releases when VDD rises to a value around 2.1 V, and VCAP has not reached 1.2 V yet. It may cause system to work abnormally, such as some internal peripherals cannot be reset successfully, or the core cannot run. The residual voltage may be caused by an external clamping diode to VDD with the anode powered up earlier than VDD, or VDD not fully dropped to below 0.2 V from last time powered off.

Workaround: To avoid formalizing a residual voltage, VDD ramp-up shall be monotonic increase, with ramp-up rate during 0 to 0.5 V not slower than the rest of ramp period, and the whole ramp-up time is between 1 ms and 200 ms.

How to Reach Us:

Home Page:
nxp.com

Web Support:
nxp.com/support

Information in this document is provided solely to enable system and software implementers to use NXP products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document. NXP reserves the right to make changes without further notice to any products herein.

NXP makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does NXP assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters that may be provided in NXP data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including “typicals,” must be validated for each customer application by customer’s technical experts. NXP does not convey any license under its patent rights nor the rights of others. NXP sells products pursuant to standard terms and conditions of sale, which can be found at the following address: nxp.com/SalesTermsandConditions.

While NXP has implemented advanced security features, all products may be subject to unidentified vulnerabilities. Customers are responsible for the design and operation of their applications and products to reduce the effect of these vulnerabilities on customer's applications and products, and NXP accepts no liability for any vulnerability that is discovered. Customers should implement appropriate design and operating safeguards to minimize the risks associated with their applications and products.