Freescale Semiconductor
Application Note

Document Number: AN4756
Rev. 0, 10/2013

eTPU2 Time Base Running at Full Speed

Aspects of T2/T4 Channel Timing

by: Milan Brejl

1 Introduction

The Enhanced Time Processing Unit (eTPU) is a programmable I/O controller with its own core and memory system, allowing it to perform complex timing and I/O management independently of the CPU. The eTPU module is a peripheral on 32-bit MPC5xxx Power Architecture® Automotive MCUs.

The eTPU2, introduced with the MPC56xx family, enables the internal eTPU time base TCR1 to run at the same speed as the eTPU core, so that the TCR1 increments every clock. As the eTPU instruction cycle always takes two clocks, the TCR1 increments twice per microcycle (microinstruction cycle). In this case the eTPU channels run in T2/T4 Channel Timing mode. Enabling this new feature affects all channels and may require software updates.

In principle, the T2/T4 Channel Timing mode enables two updates of channel Action Units during one eTPU core microcycle. Therefore, a channel flag cleared by a microinstruction in T2 phase could be set back during the same microinstruction in T4 phase. If T2/T4 mode is used on the eTPU2, it may require a change to eTPU function codes.

Contents

2 Configuring the eTPU2 for T2/T4 Channel Timing

The eTPU microcycle, or in other words microinstruction cycle, takes two clock cycles and consists of four phases: T1, T2, T3, and T4. In the T2 Channel Timing mode, the TCR1 timebase increments on microcycle phase T2. In the T2/T4 mode, the TCR1 timebase increments on microcycle phases T2 and/or T4. The following list summarizes the Channel Timing modes that are available for individual eTPU module versions.

- eTPU module (MPC55xx devices)
  - T2 Channel Timing mode
- eTPU2 module (MPC56xx devices)
  - T2 Channel Timing mode
  - T2/T4 Channel Timing mode on all channels
- eTPU2+ module (MPC57xx devices)
  - T2 Channel Timing mode
  - T2/T4 Channel Timing mode on all channels
  - T2 Force mode on selected channels

The T2 Force mode (eTPU2+ only) enables to safely run eTPU functions which are not ready for T2/T4 Channel Timing mode.

The following table lists all configuration bits which affect the resulting channel timing mode:

Table 1. Configuration items affecting the Channel Timing mode

Field	Field name	Register	Register name	Description
TCR1CS	TCR1 Clock Source	TBCR	Time Base Configuration Register	This field enables to select the same speed of both the eTPU core and the clock source for TCR1 prescaler.
FCSS	Filter Clock Source Selection	ECR	Engine Configuration Register	This bit selects the same speed of both the eTPU core and the clock source of Enhanced Digital Filter prescaler.
CDFC	Channel Digital Filter Control	ECR	Engine Configuration Register	Value 01 selects the input signal and bypasses the Enhanced Digital Filter.
T4DIS (eTPU2+ only)	T4 channel action Disables	CxCR	Channel x Configuration Register	Forces the channel to run in T2 timing mode regardless of other settings.

The following table summarizes configuration bit values and resulting channel timing mode:

Table 2. Configuration values and resulting Channel Timing mode

TCR1CS	FCSS	CDFC	F4DIS (eTPU2+)	Channel Timing mode
1[1]	–	–	0	**T2/T4**
–	1	–	0	**T2/T4**
–	–	01	0	**T2/T4**
0	0	00, 10, 11	0	**T2**
–	–	–	1	**T2**

3 eTPU Channel operations and microcycle phases

Channel operations, latched either by the channel hardware (Action Units) or by a microinstruction, execute on specific microcycle phases. For the scope of this application note is limited to the following operations:

- Transition Detection Latch (TDL) update: notification of an input transition detected by the channel Action Unit.
- Match Recognition Latch (MRL) update: notification of match recognized by the channel Action Unit.
- Match Recognition Latch Enable (MRLE) set or clear: a microinstruction enables or disables the match recognitions.
- Capture register update: the current value of TCR1/2 clock counter is stored to channel capture register on a transition or a match.

These channel operations execute on microcycle phase T2 and/or T4, depending on the Channel Timing mode, as detailed in the following tables:

Table 3. Channel operations in T2 Channel Timing mode

T2 mode	Microcycle phase T2	Microcycle phase T4
Channel operation	pin state update	
	capture register update	
	TDL, MRL update	
	MRLE clear	MRLE set + match register write

1 The TBCR[TCR1CS] = 1 does not affect the eTPU Angle Counter. The tick rate value have to be calculated using the rate of **eTPU clock divided by 2** and divided by TCR1 prescaler, independently of the TBCR[TCR1CS] setting.

Table 4. Channel operations in T2/T4 Channel Timing mode

T2/T4 mode	Microcycle phase T2	Microcycle phase T4
Channel operation	pin state update	pin state update
	capture register update	capture register update
	TDL, MRL update	TDL, MRL update
	MRLE clear	MRLE set + match register write

A special care is required in T2/T4 Channel Timing mode, where clearing MRL or TDL on T2 enables to recognize a match, update pin state and update capture register value, already on T4 of the same microcycle.

4 Examples

The eTPU function consists of threads. If a channel match happens during the thread execution, the match event recognition by the thread can be enabled or disabled:

- match_disable(): matches are disabled during the thread (default).
- match_enable(): matches are enabled during the thread.

If matches are disabled in a thread, there are no issues with the T2/T4 timing.

4.1 Scenario A

Under the following conditions:

- Channel runs in T2/T4 channel timing mode, and
- Matches are enabled in thread

The following scenario A may happen:

- A match is recognized and MRL is set. The match requests a service and consequently a service thread execution is started.
- During the thread execution, there is an instruction which includes both these operations
    - clear MRL (microcode: channel.MRLA = 0 or channel.MRLB = 0)
    - write the match register and set MRLE (microcode: channel.ERWA = 0 or channel.ERWB = 0)

  The MRL is cleared on T2. The match register is written and MRLE is set on T4.

  As the match recognition is enabled, a new match is recognized immediately, because the MRL, pin state and capture register can be updated on T4 in T2/T4 Channel Timing mode. However, the match happens on the old match value, not on the one written in this instruction.

In order to prevent this unwanted behavior, the match recognition should be disabled on T2 and enabled again on T4. The match recognition can be disabled on T2 by MRLE clear (microcode: channel.MRLE

= 0, channel.MRLEA = 0 or channel.MRLEB = 0). The match recognition is enabled on T4 together with match register write (microcode: channel.ERWA = 0 or channel.ERWB = 0).

Examples of microcode, which enables the described scenario A to happen, and microcode which prevents from it, are provided in the following table:

Table 5. Scenario A microcode examples

	Match on old value may happen	Match on old value is prevented

Example 1	`match_enable();`
	`erta = tcr1 + 1000;`
	`channel.MRLA = 0;`
	`channel.ERWA = 0;`
		`match_enable();`
		`erta = tcr1 + 1000;`
		`channel.MRLEA = 0;`
		`channel.MRLA = 0;`
		`channel.ERWA = 0;`

Example 2	`match_enable();`
	`ertb = ertb + 5000;`
	`channel.MRLB = 0;`
	`channel.ERWB = 0;`
		`match_enable();`
		`ertb = ertb + 5000;`
		`channel.MRLEB = 0;`
		`channel.MRLB = 0;`
		`channel.ERWB = 0;`

4.2 Scenario B

Under the same conditions:

- Channel runs in T2/T4 channel timing mode
- Matches are enabled in thread

Also the scenario B may happen:

- A transition is detected, TDL is set and match is blocked. The transition detection requests a service and consequently a service thread execution is started.
- During the thread execution, there is an instruction which includes both operations
    - clear TDL (microcode: channel.TDL = 0, channel.TDLA = 0 or channel.MRLB = 0)
    - write the match register and set MRLE (microcode: channel.ERWA = 0 or channel.ERWB = 0)

  The TDL is cleared on T2. The match register is written and MRLE is set on T4.

  As the TDL clear on T2 unblocks the match recognition, a match may be recognized immediately on T4. However, the match happens on the old match value, not on the one written in this instruction.

Again, in order to prevent this unwanted behavior, the mach recognition should be disabled on T2 by MRLE clear (microcode: channel.MRLE = 0, channel.MRLEA = 0 or channel.MRLEB = 0) and enabled again on T4 together with match register write (microcode: channel.ERWA = 0 or channel.ERWB = 0).

Examples of microcode, which enables the described scenario B to happen, and microcode which prevents from it, are provided in the following table:

Table 7. Scenario B microcode examples

	Match on old value may happen	Match on old value is prevented

Example 1	`match_enable();`
	`erta = erta + 1000;`
	`channel.TDLA = 0;`
	`channel.ERWA = 0;`
		`match_enable();`
		`erta = erta + 1000;`
		`channel.MRLEA = 0;`
		`channel.TDLA = 0;`
		`channel.ERWA = 0;`

Example 2	`match_enable();`
	`ertb = ertb + 5000;`
	`channel.TDLB = 0;`
	`channel.ERWB = 0;`
		`match_enable();`
		`ertb = ertb + 5000;`
		`channel.MRLEB = 0;`
		`channel.TDLB = 0;`
		`channel.ERWB = 0;`

NOTE

The addition of MRLE clear may not increase the size of code.

5 Freescale eTPU Library

The whole Freescale eTPU Library, available through [3], was examined for correct operation in T2/T4 Channel Timing mode. A potential issue was identified in Output Compare (OC) and Queued Output Match (QOM) eTPU functions and has been fixed by modification of macro ConfigMatch_AB() in etpu_common.h file.

6 Conclusion

Running an eTPU2 channel in T2/T4 timing mode, which is typically the case when TCR1 internal counter is configured for full speed, requires a special care of microcoding channel operations. Examples of correct eTPU2 microcode were provided.

7 References

1. eTPU Reference Manual, ETPURM/D, freescale.com
2. eTPU2 Reference Manual Addendum, ETPURMAD, freescale.com
3. eTPU Function Selector web tool: www.freescale.com/webapp/etpu/
4. Freescale eTPU(2) Development Suite: www.freescale.com/webapp/sps/site/prod_summary.jsp?code=eTPU2

8 Revision history

Revision number	Date	Substantial changes
0	10/2013	Initial release

How to Reach Us:

Home Page:
freescale.com

Web Support:
freescale.com/support

Information in this document is provided solely to enable system and software implementers to use Freescale 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.

Freescale reserves the right to make changes without further notice to any products herein. Freescale makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale 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 Freescale 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. Freescale does not convey any license under its patent rights nor the rights of others. Freescale sells products pursuant to standard terms and conditions of sale, which can be found at the following address: freescale.com/SalesTermsandConditions.

Document Number: AN4756
Rev. 0, 10/2013