

eTPU Reference Manual Addendum

This document describes features which have been implemented on eTPU2 and which are additional features over eTPU. This document should be read in conjunction with Freescale document ETPURM, *Enhanced Time Processing Unit (eTPU) Preliminary Reference Manual*, which is available from freescale.com. All of the eTPU2 features have been implemented in such a way that legacy object code (for both the host and eTPU microcode) will run on eTPU2 without modification.

TABLE OF CONTENTS

1	Features utilised by accessing a CPU visible register	2
1.1	TCR1 and channels can run at full eTPU clock speed	2
1.2	Enhanced Digital Filter can be disabled	3
1.2.1	Bypass Mode (CDFC = 0b01)	3
1.3	EDF can run at eTPU clock speed	3
1.3.1	Filter Clock Prescaler	3
1.4	Software watchdog	4
1.4.1	ETPUWDSR — eTPU Watchdog Status Register	4
1.4.2	ETPUWDTR — eTPU Watchdog Timer Register	6
1.4.3	Watchdog functional description	7
1.5	MISC complete flag	8
1.6	Real-time performance information	9
1.6.1	ETPUIDLER — eTPU Idle Register	9
1.7	More input to angle clock logic	9
1.8	Engine relative addressing mode	10
1.8.1	Engine Relative Addressing Mode	10
1.9	Disable priority passing	11
2	Features utilised by accessing eTPU2 resources from eTPU code	12
2.1	Testing of flag0 and flag1	12
2.2	Independent TDL negation	13
2.3	User-programmable channel modes	14
2.3.1	Selecting user-defined channel modes	15
2.3.2	UDCM — User-Defined Channel Mode Register	16
2.3.3	Write Channel Match and UDCM Registers	16
2.3.4	Channel mode logic and event flags	17
2.3.5	Changing channel modes	21
2.4	DMA request and interrupt request based on CHAN	22
2.5	Pin sampled at match time	23
2.5.1	PRSS — Pin Request Service Sample	23
2.5.2	PRSS in communications applications	24
2.5.3	Pin states on eTPU2	24
2.6	TDL blocking like TPU	25
2.7	Independent MRLE	26
3	Features where no programming is required	27
3.1	Prescaler reset on gtbe_in transition	27

Appendix A	ETPUECR — eTPU Engine Configuration Register	28
Appendix B	ETPUMCR — eTPU Module Configuration Register	30
Appendix C	ETPUTBCR — eTPU TimeBase Configuration Register	32
Appendix D	Microinstruction Formats	34

1 Features utilised by accessing a CPU visible register

This section describes the new eTPU2 features which are utilised by accessing a CPU (host) visible register.

1.1 TCR1 and channels can run at full eTPU clock speed

TCR1 can be clocked at the eTPU clock frequency by setting the Time Base Configuration register’s TCR1CS bit.

Figure 1 shows how TCR1CS is used to select one of two options: whether ETPU clock, or ETPU clock divided by two, is used to clock the TCR1 prescaler.

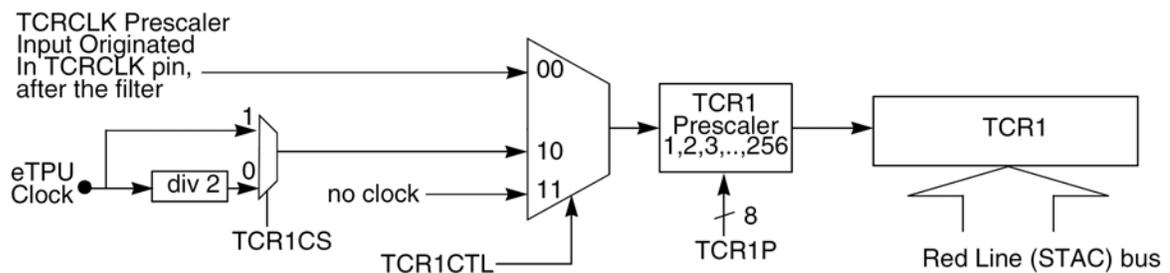


Figure 1. TCR1 Clock Selection

The TCR1CS bit is located in the ETPUTBCR — eTPU Time Base Configuration Register. See “ETPUTBCR — eTPU TimeBase Configuration Register” for details.

1.2 Enhanced Digital Filter can be disabled

eTPU2 supports disabling of the Enhanced Digital Filter (EDF).

The Enhanced Digital Filter can be disabled by writing the appropriate value (0b01) to the CDFC field in the eTPU Engine Configuration Register (ETPUECR) – see Table 1.

CDFC[1:0] — Channel Digital Filter Control

These bits select a digital filtering mode for the channels when configured as inputs for improved noise immunity (refer to Table 1). The eTPU has three digital filtering modes for the channels which provide programmable trade-off between signal latency and noise immunity. Changing CDFC during eTPU normal input channel operation is not recommended since it changes the behavior of the transition detection logic while executing its operation.

Table 1. Channel Digital Filter Control

CDFC	Selected Digital Filter
00	TPU2/3 Two Sample Mode: Using the filter clock which is the eTPU clock divided by (2, 4, 8,..., 256) as a sampling clock (selected by FPSCK field in ETPUECR), comparing two consecutive samples which agree with each other sets the input signal state. This is the default reset state.
01	eTPU bypass mode: the input signal is taken unfiltered,
10	eTPU Three Sample Mode: Similar to the TPU2/3 two sample mode, but comparing three consecutive samples which agree with each other sets the input signal state.
11	eTPU Continuous Mode: Signal need to be stable for the whole filter clock period. This mode compares all the values at the rate of eTPU clock (FCSS=1) or eTPU clock divided by two (FCSS=0), between two consecutive filter clock pulses. Signal needs to be continuously stable for the entire period. If all the values agree with each other, input signal state is updated.

1.2.1 Bypass Mode (CDFC = 0b01)

In bypass mode the signal that feeds the edge detection comes directly from the output of the synchronizer and is not filtered.

1.3 EDF can run at eTPU clock speed

This is an eTPU2 enhancement which allows the Enhanced Digital Filter to be clocked at eTPU clock frequency. The feature is activated by use of the FCSS bit in the ETPUECR. See

ETPUECR — eTPU Engine Configuration Register.

1.3.1 Filter Clock Prescaler

The TCRCLK signal and each channel configured as an input have an associated synchronizer followed by a digital filter connected to the signal that samples signal transitions. After reset, the digital filter filters out high and low pulse widths smaller than the period of two eTPU clocks with ETPUECR bit FCSS=0, or one eTPU clock with FCSS=1, preventing these transitions from being input to the transition detect logic.

For FPSCK = 0 and FCSS = 0, the synchronizer and digital filter are guaranteed to pass pulses that are as wide as or wider than four eTPU clocks, meaning a minimum period of eight eTPU clocks. These

figures are halved by setting FCSS = 1. By changing the FPSCK field in register ETPUECR, the user can select a lower clock rate for the filter signal to define wider valid pulses and filter out wider noise pulses. The filter prescaler clock control is a division of the eTPU clock. To guarantee pulse detection by the digital filter, the pulse must cover at least the stated number of samples at the filter clock rate. For example, a two sample digital filter must sample two points in the pulse to detect it. Table 2 shows the minimum guaranteed detected pulse width and the maximum filtered noise pulse width. The table refers only to the digital filter operation. The external pulses may have to be wider (to ensure detection) or narrower (to ensure filtering) depending on the rise/fall delay differences in the MCU receivers and internal logic.

Table 2. Pulse Widths and Delays

Filter Control (FPSCK)		Sample on ETPU clock Divided by:	Min. Width Guaranteed Detected / Max. Width filtered (Min. Filter Delay / Max. Filter Delay) ¹	
FCSS = 0	FCSS = 1		Two-Sample or Continuous Mode	Three-Sample or Integrator ² Mode
not available	000	1	2 / 1 (2 / 3)	3 / 2 (3 / 4)
000	001	2	4 / 2 (3 / 3)	6 / 4 (5 / 5)
001	010	3	8 / 4 (5 / 7)	12 / 8 (9 / 11)
010	011	8	16 / 8 (9 / 15)	24 / 16 (17 / 23)
011	100	16	32 / 16 (17 / 31)	48 / 32 (33 / 47)
100	101	32	64 / 32 (33 / 63)	96 / 64 (65 / 95)
101	110	64	128 / 64 (65 / 127)	192 / 128 (129 / 191)
110	111	128	256 / 128 (129 / 255)	384 / 256 (257 / 383)
111	not available	256	512 / 256 (257 / 511)	768 / 512 (513 / 767)

NOTES:

¹ This table shows pulse widths and delays in number of periods of the eTPU clock.

² Integrator mode is available for TCRCLK filtering only.

1.4 Software watchdog

eTPU2 implements a watchdog feature which can stop a thread after a programmable amount of time.

1.4.1 ETPUWDSR — eTPU Watchdog Status Register

ETPUWDSR indicates the watchdog influence in each of the engine channels.

eTPU 1: Base + 0x260 / eTPU 2: Base + 0x264

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	WDS															
W	WDS															
	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C
RESET:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	WDS															
W	WDS															
	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C
RESET:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 2. ETPUWDSR Register

WDS_x — Channel x Watchdog Status

1 = Watchdog forced end of the channel thread and disabled it.

0 = No watchdog force on the channel.

WDSC_x — Channel x Watchdog Status Clear

1 = Clear watchdog status bit.

0 = Keep watchdog status bit unaltered.

1.4.2 ETPUWDTR — eTPU Watchdog Timer Register

This register configures the watchdog timer for the engine.

eTPU 1: Base + 0x060 / eTPU 2: Base + 0x070

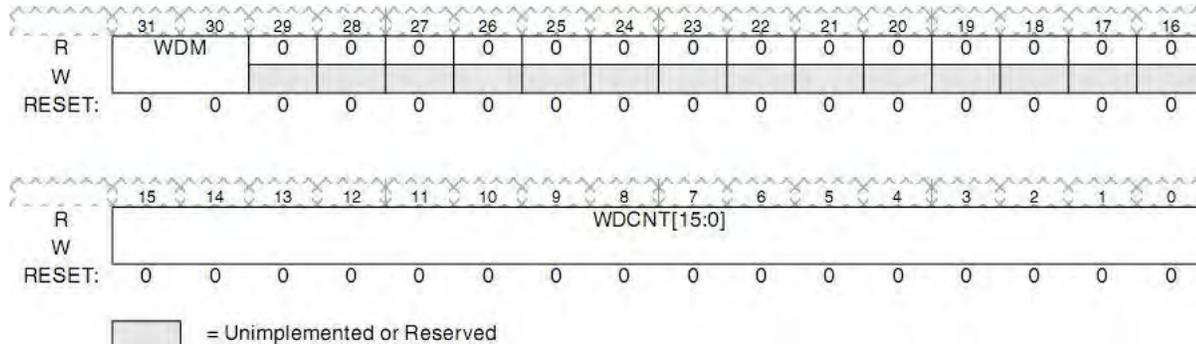


Figure 3. ETPUWDTR Register

WDM — Watchdog Mode

WDM selects the Watchdog operation mode, as shown in Table 3. For more information on the Watchdog operation, see Watchdog functional description.

Table 3. WDM — Watchdog Mode

Value	Watchdog Mode
0 0	Disabled
0 1	Reserved
1 0	Thread length
1 1	Busy length

NOTE

The watchdog must be disabled first before a new mode is configured.

WDCNT[15:0] — Watchdog Count

This field indicates the maximum number of microcycles allowed for a thread (in thread length mode) or a sequence of threads (in busy length mode) before the current running thread is forced to end. For more information on Watchdog operation, see Watchdog functional description.

NOTE

The TST microcycles are also counted by the watchdog.

1.4.3 Watchdog functional description

Each engine has a watchdog mechanism to prevent a thread or a sequence of threads from running too long, impacting the latency of the other channel services. The watchdog is configured through the register ETPUWDTR. When the watchdog is enabled, an internal counter increments on each microcycle when a thread is executing. If the count is greater than the value specified in the ETPUWDTR field WDCNT and a thread is still executing, the watchdog:

- 1) Forces an END of the thread.
- 2) Sets the WDS status bit of the serviced channel in the ETPUWDSR register. The channel is disabled, not initiating any thread until its WDS bit is cleared (its CPR field in ETPUCxCR is not changed, however).
- 3) Issues a Global Exception and sets the ETPUMCR bit WDTO (see ETPUMCR — eTPU Module Configuration Register).

The watchdog can be configured in one of the following modes, defining how the internal watchdog count is reset:

- **Thread Length Mode:** the watchdog count is reset at the end of each thread.
- **Busy Length Mode:** the watchdog count is reset when the microengine goes idle. A sequence of threads, one right after another, keeps the count running. The counter is also reinitialized when a thread is forced to end, so that a new count begins if another TST initiates at the following microcycle.

The following applies to the watchdog mechanism:

- Microcycles during TST and SDM access wait-states (on TST or instruction execution) are counted.
- If the watchdog count equals WDCNT in the last microinstruction (with SDM wait-states or not) of a thread servicing a channel, its WDS bit is not set.
- If the watchdog count expires (gets greater than WDCNT) during the TST, the thread is forced end on its first instruction.

NOTE

Watchdog must not be enabled when the microengine enters halt mode. The counter does not run when the engine is stopped, and resets when the watchdog is disabled.

1.5 MISC complete flag

On the eTPU it is not possible to tell when an SCM (Shared Code Memory) MISC (Multiple Input Signature Calculator) calculation has completed. This has been enhanced on eTPU2 by the addition of an MISC complete flag and a means of clearing it.

The new flag is located in the ETPUMCR — Module Configuration Register (see ETPUMCR — eTPU Module Configuration Register). The flag is called SCMMISC (SCM MISC Complete) and indicates that MISC has completed the evaluation of the SCM signature since reset or since the last time it was cleared. SCMMISC is cleared by writing 1 to SCMMISCC (SCM MISC Complete Clear). SCMMISCC shares the same bit position as SCMMISC. SCMMISC is not cleared when MISC is disabled (SCMMISEN=0). SCMMISC asserts at the end of the SCM memory scan, irrespective of whether the signature matches or not.

1.6 Real-time performance information

A direct way to measure the eTPU2's performance and loading has been added to eTPU2. eTPU2 has a new register called Idle Counter (ETPUIDLER).

The Idle Counter Register ETPUIDLER continuously counts microcycles in which the microengine is not busy. It can be used to measure the microengine utilization by deriving a ratio of the count measured during a period of time to the number of microcycles contained in the period. The Idle counter does not count microcycles when the engine is stopped, is in TST or halt states.

1.6.1 ETPUIDLER — eTPU Idle Register

This register counts the microcycles in which the microengine is idle

eTPU 1: Base + 0x068 / eTPU 2: Base + 0x078

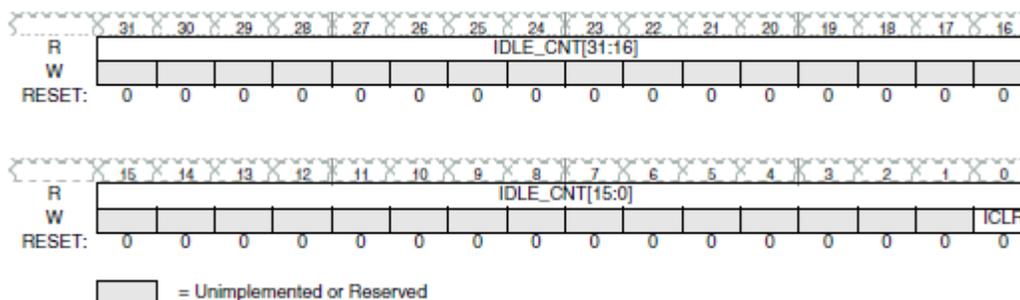


Figure 4. ETPUIDLER Register

IDLE_CNT[31:0] — Idle Count

This is a free-running count of the number of idle microcycles in the microengine.

ICLR — Idle Clear

This write-only bit is used to clear the idle count IDLE_CNT:

1 = Clear the idle count IDLE_CNT

0 = Do not clear idle count IDLE_CNT

1.7 More input to angle clock logic

On eTPU2, Channels 1 and 2 (in addition to channel 0) can be selected to control the eTPU Angle Counter. This allows another signal to be used to drive the Angle Clock if the primary signal input were to fail.

The AM field in the ETPUTBCR — eTPU TimeBase Configuration Register has been extended to allow channels 1 and 2 to be inputs to the EAC. See ETPUTBCR — eTPU TimeBase Configuration Register.

1.8 Engine relative addressing mode

On devices with two eTPU engines, the micro-engine could not determine if a function is running on engine A or B. This means that the compiler could not allocate local heap memory. This has been enhanced on eTPU2 by the addition of an engine relative addressing mode. An Engine Relative Base Address field in the ETPUECR (see

ETPUECR — eTPU Engine Configuration Register) has been added to support this addressing mode.

1.8.1 Engine Relative Addressing Mode

In Engine Relative Addressing mode, the physical address is the concatenation of the ETPUECR field ERBA (see

ETPUECR — eTPU Engine Configuration Register) with the 7-bit AID instruction field. This allows the same function microcode, when running on distinct engines, to access different address spaces, global to the engine only. There is an Engine Relative Base Address field for each engine.

1.9 Disable priority passing

The priority passing scheme on eTPU is not always suitable for all application types as it can lead to increased worst case latency under some conditions. eTPU2 implements a means of disabling priority passing.

Consider the case shown in Figure 5.

The table represents the normal priority passing scheme. In a given slot, if priority passing is enabled and a channel with the first choice priority is not requesting service, then a channel with the second choice priority will be serviced. If a channel with the second choice priority is not requesting service, then a channel with the third choice priority will be serviced.

Consider the case where priority has just been passed from the first choice to the second choice in slot 3 (H to M) and a High priority channel immediately requests service. With priority passing enabled (SPPDIS = 0) the eTPU2 will:

- Service the Medium channel (slot 3)
- Service the Low channel (slot 4)
- Service the High Channel (slot 5)
- (The yellow arrows are followed)

The High channel has waited for almost 2 whole slots before being serviced.

H channel requests service here

Slot	1	2	3	4	5	6	7
Standard order	H	M	H	L	H	M	H
1 st priority pass	M	H	M	H	M	H	M
2 nd priority pass	L	L	L	M	L	L	L

Figure 5. Priority Passing Disabling Example

If priority passing has been disabled (SPPDIS = 1) the eTPU2 will:

- Service the Low channel (slot 4)
- Service the High channel (slot 5)
- (The red arrows are followed)

The High channel has waited for almost one whole slot before being serviced.

In this case, with priority passing disabled the high priority channel gets serviced faster.

Priority passing is disabled by writing a 1 to the SPPDIS bit in the ETPUECR (see

ETPUECR — eTPU Engine Configuration Register). By default, priority passing is enabled from reset.

2 Features utilised by accessing eTPU2 resources from eTPU code

This section describes the new eTPU2 features which are utilised by accessing eTPU2 resources from eTPU code.

2.1 Testing of flag0 and flag1

It was found with C-programming on eTPU that testing of the hardware flags would be beneficial. On eTPU2 BCC encoding has been extended to allow branching based on the value of flag0 and flag1. Encoding is defined in Table 4.

Table 4. Branch Condition Selection — BCC

BCC	Meaning	BCC	Meaning
001110	Flag 0	001111	Flag 1
100000	V ALU flag	110000	PSS channel flag
100001	N ALU flag	110001	PRSS channel flag
100010	C ALU flag	110010	“Less Than” ALU flag combination (signed) ¹
100011	Z ALU flag	110011	“Lower or Equal” ALU flag combination (unsigned) ²
100100	MV MDU flag	110100	P[24]
100101	MN MDU flag	110101	P[25]
100110	MC MDU flag	110110	P[26]
100111	MZ MDU flag	110111	P[27]
101000	TDLA channel flag	111000	P[28]
101001	TDLB channel flag	111001	P[29]
101010	MRLA channel flag	111010	P[30]
101011	MRLB channel flag	111011	P[31]
101100	LSR channel flag	111100	PSTO channel flag
101101	MB flag MDU flag	111101	PSTI channel flag
101110	FM[1] channel flag	111110	SMLCK semaphore flag
101111	FM[0] channel flag	111111	false
all other values reserved			

NOTES:

¹ “less than” is a signed comparison, equal to the xor between ALU flags V and N; for example, $0 < 0xFFFFFFFF$ tests as false ($0 < -1$).

² “lower equal” is an unsigned comparison, equal to Z or C; for example, $0 < 0xFFFFFFFF$ tests as true.

2.2 Independent TDL negation

Flags TDLA and TDLB indicate the state of transitions detected in the selected channel, and it is possible to clear those flags using the microcode field TDL (one or two bits, depending on the format). These microcode fields clear the actual channel flags, and also the ones sampled into the branch logic. TDL can be one or two bits wide, depending on the microinstruction format (see Microinstruction Formats). Two-bit TDL allows independent clearing of TDLA and/or TDLB. Table 5 defines the one-bit TDL field and Table 6 defines the two-bit TDL field.

Table 5. Clear Transition/Match Event Registers - MRLA/B, TDL

Field	Meaning
TDL (1 bit)	0 = clear TDLA and TDLB flags 1 = don't change

Table 6. Independent TDLA/B clear – two bit TDL

Value	Meaning
0 0	clear TDLA
0 1	clear TDLB
1 0	clear both TDLA and TDLB
1 1	do not clear TDLA or TDLB

In addition the transitions A and B now can be unordered, so that TDLB can set before TDLA. Unordered transitions are supported using User Programmable channel modes, setting bit T1ET2=0 (see User-programmable channel modes).

On eTPU programmers have to rely on the current pin state at service time in order to determine which edge happened first, when dealing with a double transition situation. Here both action units would be programmed to capture any transition (any_trans).

There are 2 possible scenarios which would cause a double capture:

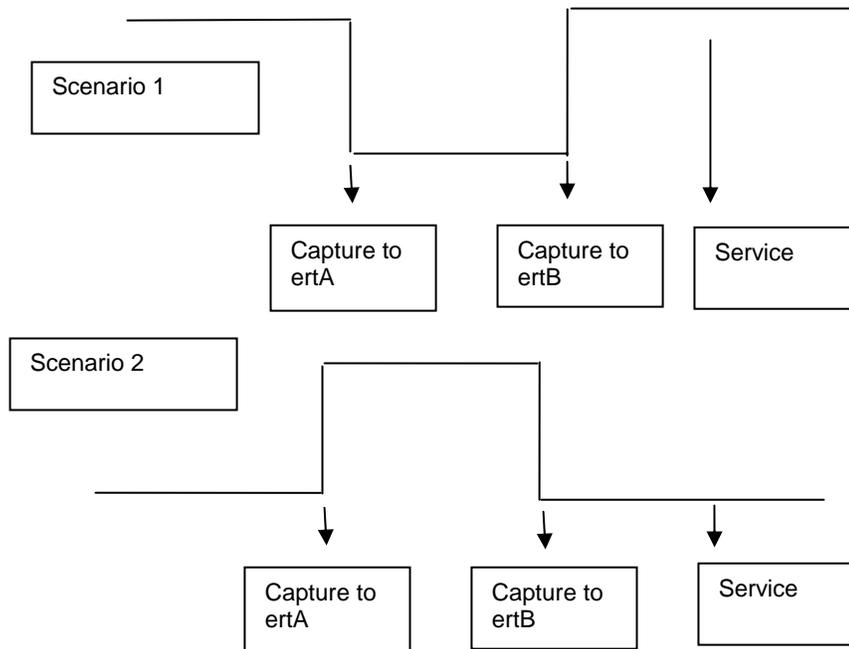


Figure 6. Captures during double transitions using any_trans on eTPU

During service the pin state must be interrogated to establish the direction of the pulse which caused service. If there were a noise edge between the second edge and service, then the pin state would give the wrong information about the transition direction.

On eTPU2 the problem described above is solved using unordered transitions. Action Unit A can be programmed to capture on rising edge and action unit B can be programmed to capture falling edges. On service, the values in ertA and ertB can be examined to determine which edge happened first, and hence the direction of the pulse will be known.

2.3 User-programmable channel modes

eTPU2 has been enhanced by the addition of user-programmable channel modes. These allow users to have more control over the channel hardware and to deal with additional timing requirements that could not be dealt with by the predefined channel modes.

User-defined channel modes are available on eTPU2 in addition to the predefined channel modes. Each channel can be programmed with a predefined channel mode.

2.3.1 Selecting user-defined channel modes

An additional encoding has been added to the PDCM register encoding. User-defined channel modes are selected by writing 0b1010 to the PDCM 4 bit register. See Table 7. If this selection is made, the channel behavior is defined by the settings of the UDCM register (see UDCM — User-Defined Channel Mode).

Table 7. PDCM encoding

PDCM	Channel mode
0000	em_b_st
0001	em_b_dt
0010	em_nb_st
0011	em_nb_dt
0100	m2_st
0101	m2_dt
0110	bm_st
0111	bm_dt
1000	m2_o_st
1001	m2_o_dt
1010	User-defined channel mode
1011	reserved
1100	sm_st ¹
1101	sm_dt
1110	sm_st_e
1111	n.a. ²

NOTES:

1. This is the reset value, also compatible with TPU channel behavior.
2. This value is used as a neutral (do not change) value in the PDCM microinstruction field.

2.3.2 UDCM — User-Defined Channel Mode Register

UDCM holds the control signals that define the channel logic behavior in terms of Match and Transition blocking and enabling, captures and service requests triggered by events. The register layout is defined in Figure 7. The individual fields are defined in Channel mode logic and event flags. There is one UDCM register for each channel, which can be independently programmed.

UDCM can only be written into the Match register by microcode, through ERTA/B microengine registers (see Write Channel Match and UDCM Registers).

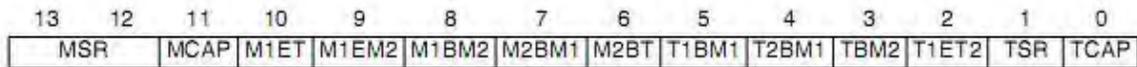


Figure 7. UDCM Register

2.3.3 Write Channel Match and UDCM Registers

Match registers can have their values changed using ERWA and ERWB fields (one bit each). They also set their respective MRLE. In eTPU2, ERWA can also be used to program the UDCM register (see UDCM — User-Defined Channel Mode Register). The new field CMW selects where the content of ERTA is copied when ERWA is active (see Table 8).

Table 8. Write MatchA/B - ERWA/B

Field	CMW	Value	Action
ERWA	1	0	Write ERTA value in MatchA. Enable matches for MatchA register (MRLEA=1)
	0	0	Write ERTA value into UDCM
	1	1	Don't change UDCM, MatchA and MRLEA
	0	1	reserved
ERWB	1	0	Write ERTB value in MatchB. Enable matches for MatchB register (set MRLEB=1)
	1	1	Don't change MatchB and MRLEB
	0	1	reserved
	0	0	reserved

If ERTA or ERTB is a destination of an ALU operation and, at the same time, the respective ERWA/B field is active, the new ERTA value is the one written into the MatchA/B register or the UDCM register.

2.3.4 Channel mode logic and event flags

Figure 8 shows a diagram of channel mode logic. The logic shown here is not necessarily identical to the actual channel logic implementation, but is equivalent with respect to conditions for event blocking, enabling, capture, and service requests.

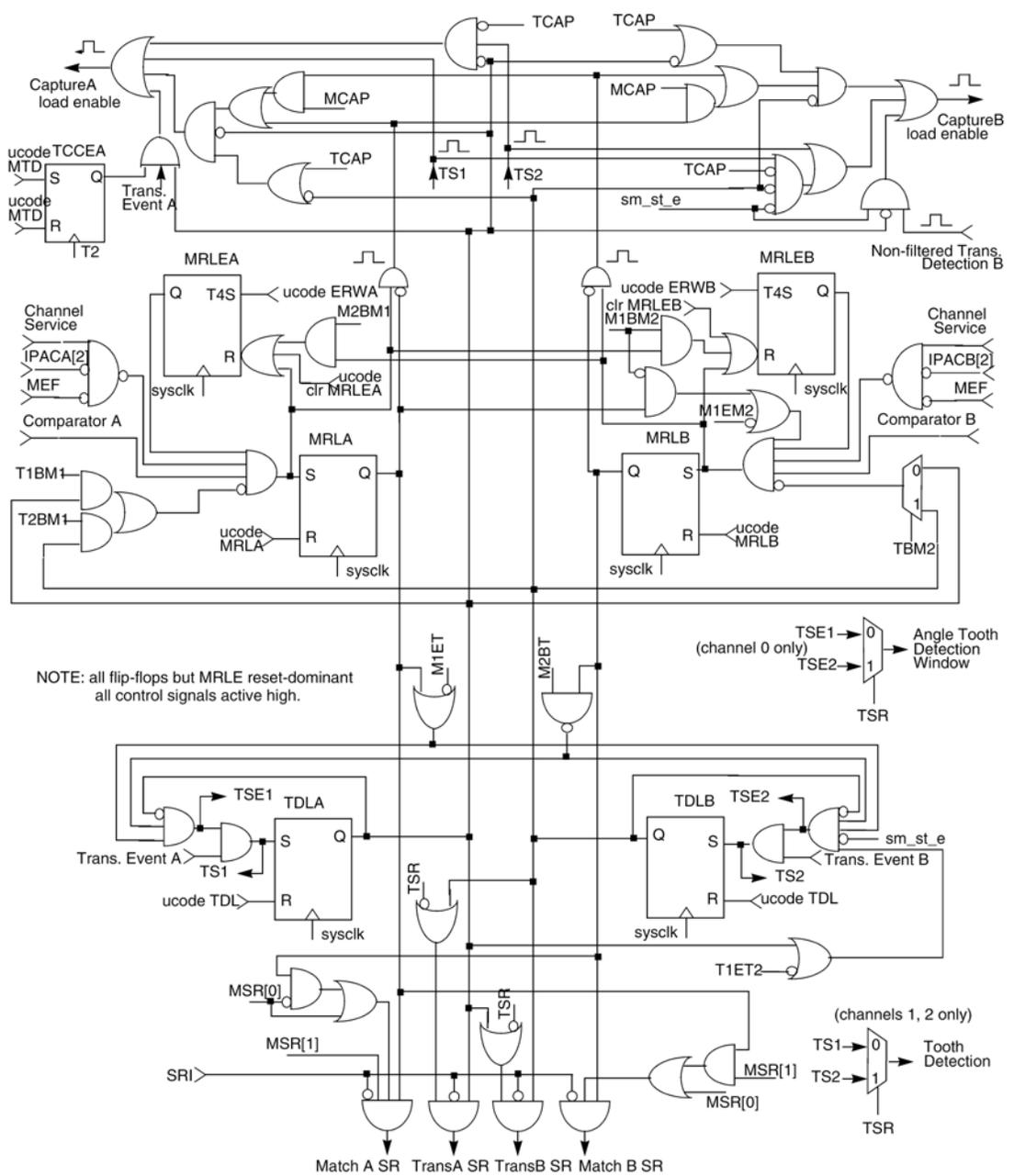


Figure 8. Channel mode logic and event flags

Signals MSR, TSR, MCAP, TCAP, M1ET, M1EM2, M1BM2, M2BM1, M2BT, T1BM1, T2BM1, TBM2, and T1ET2 are decoded from programmed channel mode PDCM in predefined modes, and come directly from the UDCM register when user-defined mode is selected:

- TSR (1 bit) defines service requests issued by transitions, as shown in Table 10.
- MSR (2 bits) defines service requests issued by matches, as shown in Table 9.
- TCAP (1 bit) defines time base captures caused by transitions, as shown in Table 10.
- MCAP (1 bit) defines time base captures caused by matches, as shown in Table 11.
- M1ET, M1EM2, M1BM2, M2BM1, M2BT, T1BM1, T2BM1, TBM2, T1ET2 (1 bit each) define match.
- Transition reciprocal blocking and enabling, as well as transition ordering, as shown in Table 12 and Table 13.

Table 14 shows the decoded values of those control signals for each predefined channel mode. The first column shows the mnemonic reference for the predefined channel modes.

Changing PDCM or the UDCM when user mode is selected may set or reset any of the channel flags, or issue captures and service requests, so it is advisable to switch channel modes only in a quiescent channel state, with channel flags MRLEA/B, TDLA/B, and MRLA/B cleared. Furthermore, an event flag asserted in one mode may remain asserted after the mode programming change, even if it is impossible to set the flag in the new mode.

Table 9. MSR[1:0] signals — Match Service Requests

Value	MSR
00	Issue no service requests on matches
01	Issue service request on match 2 only
10	Issue service request on 2nd ¹ match
11	Issue service request on both matches

NOTES:

1. 2nd match means the match that happens after the 1st match in time, either match 1 or match 2.

Table 10. TCAP and TSR signals — Transition Captures and Service Requests

Value	TCAP	TSR
0	1 st Transition captures Time Bases corresponding to Transition 1 and Transition 2 ² Transition 1 captures corresponding Time Base if it is the second transition; Transition 2 captures corresponding Time Base if it is the second transition.	issue Service Request on the 1st ¹ Transition
1	Transition 1 captures corresponding Time Base. Transition 2 captures corresponding Time Base.	issue Service Request on the 2nd ³ Transition

NOTES:

1. ¹ 1st Transition means the Transition that happens first in time, either Transition 1 or Transition 2.
2. ² Match capture(s) never overwrites a Transition capture. Transition captures can always override a Match capture.
3. ³ 2nd Transition means the Transition that happens second in time, either Transition 1 or Transition 2.

Table 11. MCAP signal — Match Capture

Value	MCAP
0	Match 1 captures corresponding Time Base. Match 2 captures corresponding Time Base.
1	Either Match captures Time Bases corresponding to Match 1 and Match 2 ¹

NOTES:

1. Match capture(s) never overrides a Transition capture. Transition captures can always override a Match capture.

Table 12. TBM2 signal — Transition Blocks Match 2

Value	TBM2
0	Transition 1 Blocks Match 2
1	Transition 2 Blocks Match 2

Table 13. M1ET, M1EM2, M1BM2, M2BM1, M2BT signals

Signal	Active value meaning
M1ET (Match 1 Enable Transitions)	Transitions are initially blocked. Match 1 enables Transitions.
M1EM2 (Match 1 Enables Match 2)	Match 2 is initially blocked ¹ . Match 1 enables Match 2.
M1BM2 ² (Match 1 Blocks Match 2)	Match 2 is initially enabled ¹ . Match 1 blocks Match 2 ³ .
M2BM1 ² (Match 2 Blocks Match 1)	Match 1 is initially enabled. Match 2 blocks Match 1 ³ .
M2BT (Match 2 Blocks Transitions)	Match 2 blocks Transitions.
T1BM1 (Transition 1 Blocks Match 1)	Transition 1 blocks Match 1.
T2BM1 (Transition 2 Blocks Match 1)	Transition 2 blocks Match 1.
T1ET2 (Transition 1 Enables Transition 2)	Transition 2 is initially blocked, and Transition 1 enables Transition 2.

NOTES:

1. The initial condition of M1EM2 prevails over M1BM2, while M1BM2 blocking prevails over M1EM2 enabling, so that Match 2 stays always blocked when both M1BM2 and M1EM2 are active. This combination is used in single-match modes (sm_*).
2. Blocking of one Match by the other is done through MRLEs.
3. Matches always block themselves by resetting their own MRLEs (Match 1 always blocks Match 1, Match 2 always blocks Match 2).

Table 14. Predefined Channel Mode control signals decoding

Predefined Mode	MSR	MCAP	M1ET	M1EM2	M1BM2	M2BM1	M2BT	T1BM1	T2BM1	TBM2 ¹	T1ET2	TSR ¹	TCAP ¹
em_nb_st	11	0	off	off	off	off	off	On	off	0	On	0	0
em_nb_dt	11	0	off	off	off	off	off	On	off	1	On	1	1
em_b_st	11	1	off	off	On	On	off	On	off	0	On	0	0
em_b_dt	11	1	off	off	On	On	off	On	off	1	On	1	1
bm_st	10	0	off	off	off	off	off	On	off	0	On	0	0
bm_dt2	10	0	off	off	off	off	off	off	On	1	On	1	1
m2_st	01	0	On	off	off	On	off	On	off	0	On	0	0
m2_dt	01	0	On	off	off	On	off	On	off	1	On	1	1
m2_o_st	01	0	On	On	off	off	On	On	off	0	On	0	0
m2_o_dt	01	0	On	On	off	off	On	On	off	1	On	1	1
sm_st	11	1	off	On	On	On	off	On	off	0	On	0	0
sm_dt ²	11	1	off	On	On	On	off	off	On	1	On	1	1
sm_st_e ³	11	0	off	On	On	On	off	On	off	0	On	0	0

NOTES:

1. signals TSR, TCAP and TBM2 replace the signal DTM used in previous eTPU versions.
2. bm_dt and sm_dt are exceptions in the match blocking logic by transitions
3. sm_st_e is an exception in the capture scheme.

2.3.5 Changing channel modes

It is advisable to reset channel flags MRLA/B, TDLA/B, and MRLEA/B before writing to UDCM.

2.4 DMA request and interrupt request based on CHAN

Microcode can issue interrupt requests, data transfer requests, and global exception through the CIRC field. CIRC encoding is defined in Table 15. For eTPU2, four new encodings have been added so that the CHAN field can be used to select the channel on which a DMA or interrupt request appears. Additionally it is possible on eTPU2 to select whether a DMA request occurs, an interrupt request occurs, or both a DMA request and an interrupt request occur for the selected channel.

Table 15. Channel and Data Transfer Requests – CIRC

CIRC	Meaning
000	Channel Interrupt Request from selected channel
001	Data Transfer Request from selected channel
010	Channel Interrupt and Data Transfer requests from selected channel
011	Channel Interrupt and Data Transfer requests from serviced channel
100	Channel Interrupt Request from service channel
101	Data Transfer Request from service channel
110	Global Exception
111	Don't request interrupt

2.5 Pin sampled at match time

On eTPU2, a new branch condition named PRSS allows branching according to the pin state at the time when a channel (match or transition) service request occurred (as opposed to the pin state when the event causing the service request occurred).

2.5.1 PRSS — Pin Request Service Sample

Channel logic can, depending on its state and programmed mode, request service to the eTPU microengine. When the channel logic issues a service request, the filtered input signal PSTI is sampled into an internal channel flag, PRSS. There is one such PRSS flag for each channel (see Figure 9). Channel PRSS keeps its value until all its service request sources are cleared and a new service request rises. The channel PRSS flag is sampled into the branch logic as the PRSS flag (see Table 4) during the time slot transition period, or whenever the CHAN register is written by microcode.

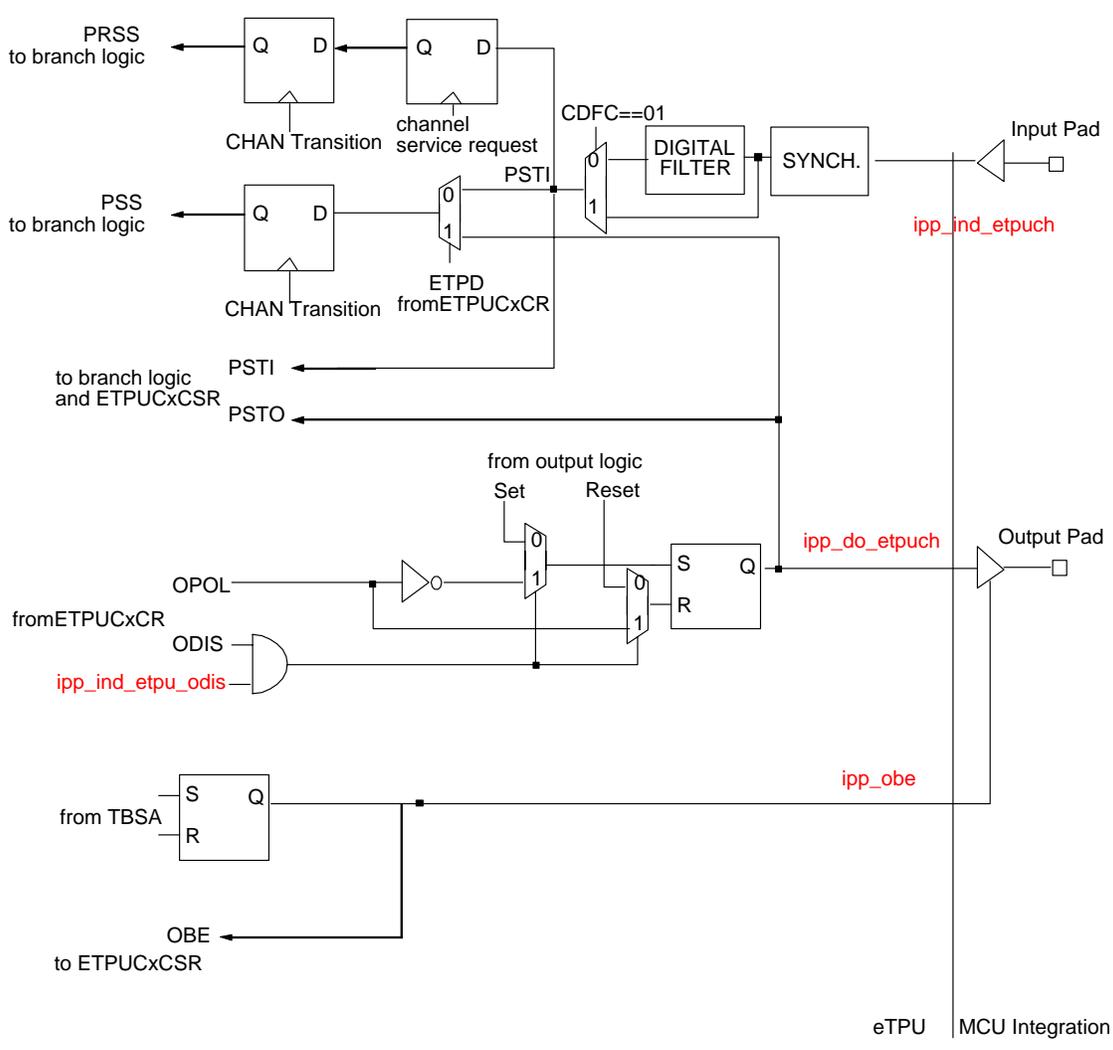


Figure 9. Pin State Input/Output Logic

2.5.2 PRSS in communications applications

PRSS is used to sample a pin at a specific time (match event). Irrespective of any service latency or subsequent noise, the pin state at match time will be known by the thread which services the match. This is useful in communications applications.

2.5.3 Pin states on eTPU2

There are three pin states which can be used on eTPU2. These are called PRSS, PSS, and PIN. These are shown in Figure 10.

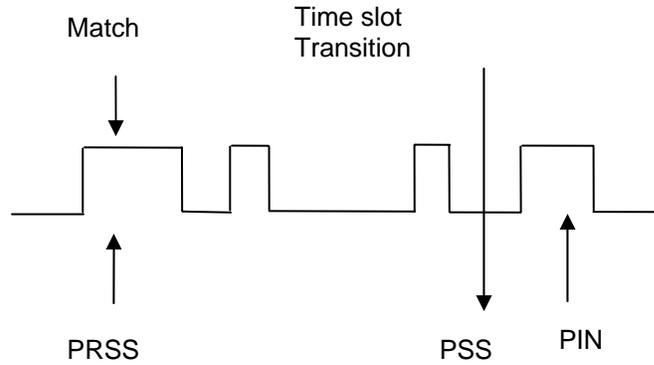


Figure 10. Pin States

- PRSS is sampled at match time.
- PSS is sampled at Time Slot Transition.
- PIN is the pin state during a thread. It may be re-sampled at any time by a thread.

2.6 TDL blocking like TPU

A new channel TCCEA flag allows continuous capture even after TDLA is set, making it fully compatible with TPU behavior.

TCCEA enables capture from transitions after the TDLA flag is set. TCCEA is negated on reset, so that a capture occurs only when TDLA asserts. TCCEA can be set and reset by microcode only, through the instruction field MTD. It can only be set together with inhibiting of the channel service requests (SRI = 1)¹. When TCCEA is asserted, the transition events specified by IPACA that occur after TDLA is set also cause captures, into the CaptureA register only. However, output actions related to transition events are still blocked by TDLA.

The logic associated with TCCEA is shown in Figure 9.

Figure 11 shows that captures can be continuous, depending upon TCCEA. When TCCEA = 0 the first active edge generates a service request. If other edges happen before service, subsequent edges are ignored. When TCCEA = 1 these subsequent edges are captured. In this case the event register will contain the time of the last active edge before service occurred.

¹ TCCEA provides compatibility with TPU when service request is inhibited.

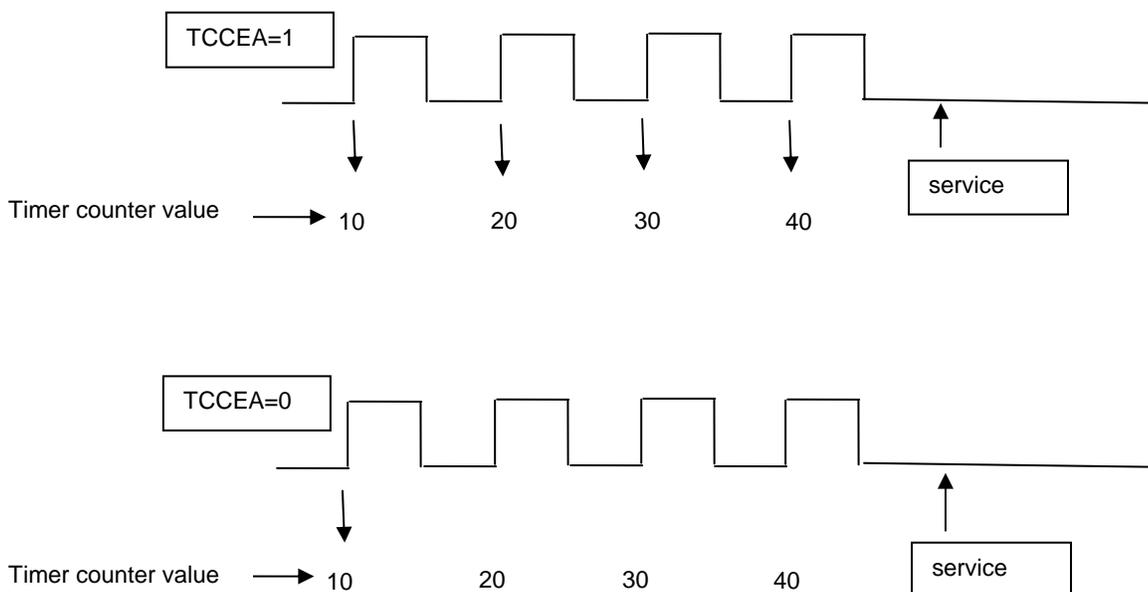


Figure 11. TCCEA behaviour

In the case where TCCEA = 1, every transition will cause the capture register to be updated with the selected timer counter's value when an edge occurs. When the channel is serviced the capture register will contain a value of 40.

In the case where TCCEA = 0, only the first transition will cause the capture register to be updated with the selected timer counter's value when the edge occurred. When the channel is serviced the capture register will contain a value of 10.

2.7 Independent MRLE

On eTPU a single operation negated both MRLE_A and MRLE_B. An eTPU2 enhancement allows MRLE_A and MRLE_B to be negated independently. This enhancement allows more flexible handling of double match events.

Microcode field MRLE (one bit) allows disabling matches on channel selected by CHAN register, for both MatchA and MatchB registers, by clearing their respective MRLE bits. The meaning of MRLE (one bit) is shown in Table 16. Matches can be enabled for each Match register using ERWA and ERWB fields

Table 16. Disable Matches — MRLE

MRLE	Meaning
0	Disable matches for Match 1 and Match 2.
1	Don't change match enabling.

On eTPU2 some instruction formats have a two-bit MRLE field (see Microinstruction Formats) which allows independent disabling of Matches 1 and 2, as shown in Table 17.

Table 17. Two-bit MRLE

MRLE	Meaning
0 0	Disable Match 1 (clear MRLEA)

MRLE	Meaning
0 1	Disable both Matches (clear MRLEA and MRLEB)
1 0	Disable Match 2 (clear MRLEB)
1 1	nop

3 Features where no programming is required

3.1 Prescaler reset on gtbe_in transition

On eTPU it was possible that the eTPUA, eTPUB, and eMIOS TCR values could have a phase error between them. This could happen if any of the timer sub-systems were disabled (for example, using the MDIS bit in the ETPUECR; see

ETPUECR — eTPU Engine Configuration Register) and subsequently re-enabled.

This has been corrected on eTPU2. Timebase prescalers are now reset when the GTBE input is negated. Synchronization with eMIOS prescalers is guaranteed on eTPU2.

Appendix A ETPUECR — eTPU Engine Configuration Register

NOTE

Only the new bits (or bits which have altered meaning) for eTPU2 are described here. Refer to the eTPU Reference Manual for description of the other bits.

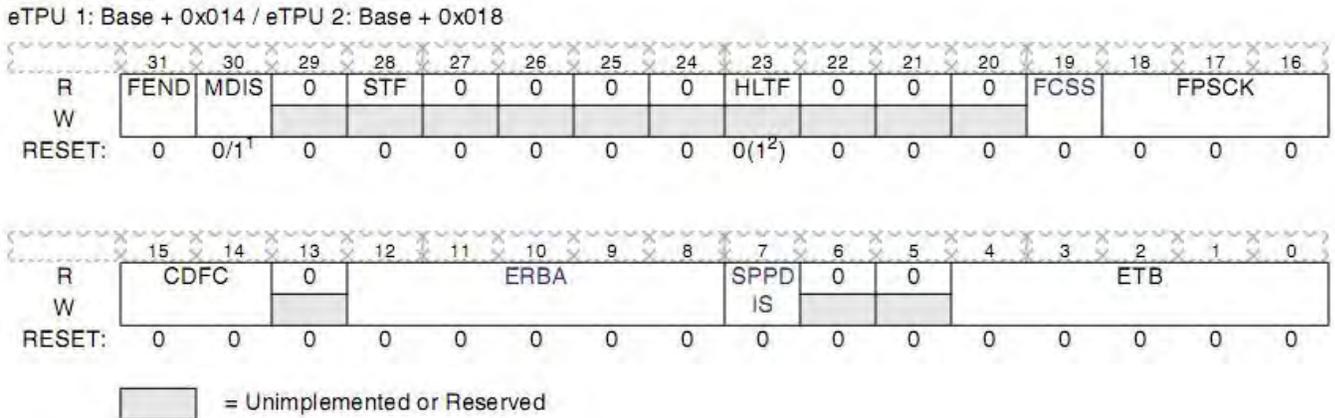


Figure 12. ETPUECR Register

FCSS — Filter Clock Source Selection

Speeds up the filter clock source before the prescaler, allowing more input capture resolution at minimum prescaling.

1 = use eTPU clock as EDF clock source before prescaler

0 = use eTPU clock / 2 as EDF clock source before prescaler

CDFC[1:0] — Channel Digital Filter Control

These bits select a digital filtering mode for the channels when configured as inputs for improved noise immunity (refer to Table 18). The eTPU has three digital filtering modes for the channels which provide programmable trade-off between signal latency and noise immunity. Changing CDFC during eTPU normal input channel operation is not recommended because it changes the behavior of the transition detection logic while executing its operation.

Table 18. Channel Digital Filter Control

CDFC	Selected Digital Filter
00	TPU2/3 Two Sample Mode Using the filter clock which is the eTPU clock divided by (2, 4, 8,..., 256) as a sampling clock (selected by FPSCK field in ETPUECR), comparing two consecutive samples which agree with each other sets the input signal state. This is the default reset state.
01	eTPU bypass mode The input signal is taken unfiltered.
10	eTPU Three Sample Mode

CDFC	Selected Digital Filter
	Similar to the TPU2/3 two sample mode, but comparing three consecutive samples which agree with each other will set the input signal state.
11	eTPU Continuous Mode Signal needs to be stable for the whole filter clock period. This mode compares all the values at the rate of eTPU clock (FCSS = 1) or eTPU clock divided by two (FCSS = 0), between two consecutive filter clock pulses. Signal needs to be continuously stable for the entire period. If all the values agree with each other, input signal is updated.

ERBA — Engine Relative Base Address

This field value is concatenated with the AID instruction field in engine relative address mode to form the SPRAM address (see Engine Relative Addressing Mode).

SPPDIS — Schedule Priority Passing Disable

SPPDIS is used to disable the priority passing mechanism of the microengine scheduler.

1 = Scheduler priority passing mechanism disabled.

0 = Scheduler priority passing mechanism enabled.

NOTE

SPPDIS bit must not be changed while any channel is enabled.

Appendix B ETPUMCR — eTPU Module Configuration Register

NOTE

Only the new bits (or bits which have altered meaning) for eTPU2 are described here. Refer to the eTPU Reference Manual for description of the other bits.

Base + 0x000

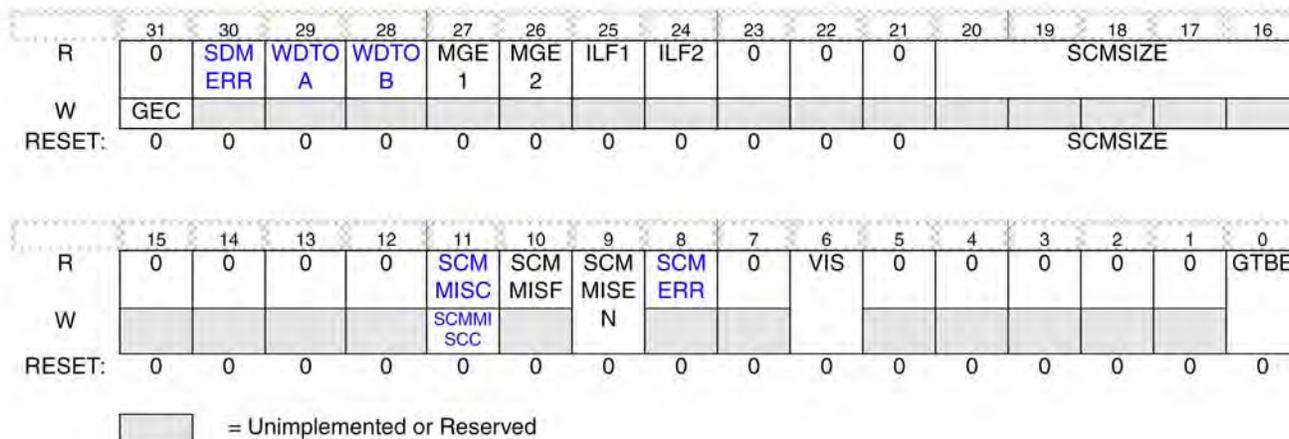


Figure 13. ETPUMCR Register

SDMERR — SDM Read Error

This flag indicates that an SDM non-correctable read error occurred on a microengine read, generating a Global Exception. Errors from Host reads neither set this flag nor generate Global Exceptions. This bit is cleared by writing 1 to GEC.

1 = Global Exception requested by SDM read error is pending.

0 = No Global Exception pending because of SDM read error.

WDTOA/WDTOB — Watchdog Timeout

Flags WDTOA and WDTOB indicate that a Watchdog Timeout occurred in the respective engine, generating a Global Exception. These bits are cleared by writing 1 to GEC.

1 = Global Exception requested by Watchdog timeout.

0 = No Global Exception pending because of Watchdog timeout.

SCMERR — SCM Read Error

This flag indicates that an SCM non-correctable read error occurred on a microengine read, generating a Global Exception. Errors from Host reads neither set this flag nor generate Global Exceptions. This bit is cleared by writing 1 to GEC.

1 = Global Exception requested by SCM read error is pending.

0 = No Global Exception pending because of SCM read error.

SCMMISC, SCMMISCC — SCM MISC Complete, SCM MISC Complete Clear

Flag SCMMISC indicates that MISC has completed the evaluation of the SCM signature since reset or the since the last time it was cleared. SCMMISC is cleared by writing 1 to SCMMISCC (at same bit position), and is not cleared when MISC is disabled (SCMMISEN = 0). SCMMISC asserts at the end of the SCM memory scan, whether the signature matches or not.

1 = MISC completed at least one SCM signature calculation and compare since the last time SCMMISC was cleared.

0 = MISC has not yet completed an SCM signature calculation and compare since the last time SCMMISC was cleared.

Appendix C ETPUTBCR — eTPU TimeBase Configuration Register

NOTE

Only the new bits (or bits which have altered meaning) for eTPU2 are described here. Refer to the eTPU Reference Manual for description of the other bits.

eTPU 1: Base + 0x020 / eTPU 2: Base + 0x040

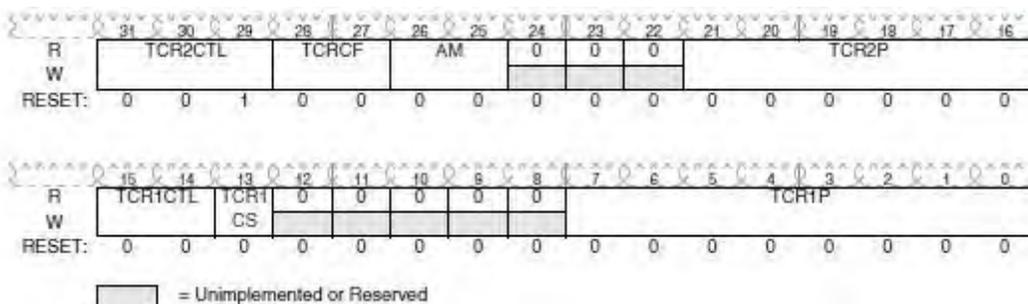


Figure 14. ETPUTBCR Register

AM — Angle Mode Selection

This field enables the Enhanced Angle Counter logic to generate angle information, and also select the tooth signal input and the channel used to process it, as shown in Table 19. When EAC is not disabled by AM and neither TCR1 nor TCR2 are STAC clients, the EAC (eTPU Angle Clock) hardware provides angle information to the channels using the TCR2 bus. When AM is reset (non-angle mode), the EAC operation is disabled, and its internal registers can be used as general purpose.

Table 19. AM — Angle Mode Selection

Value	TCR2 Value	Tooth signal	Tooth processing channel
0 0	Timebase (EAC operation disabled)	not applicable	
0 1	Angle Ticks	TCRCLK input	0
1 0		channel 1 input	1
1 1		channel 2 input	2

If TCR1 or TCR2 is a STAC bus client, the EAC operation is forbidden, and if AM is set the Angle Logic does not work properly.

NOTE

Changing AM may cause spurious transition detections on the channel selected by AM, depending on the channel mode and state

TCR1CS — TCR1 Clock Source

TCR1CS provides the option to double the TCR1 incrementing speed, using eTPU clock as its clock source instead of eTPU clock / 2.

1 = use eTPU clock as TCR1 clock source before the prescaler; can only be set in specific combinations with TCR1CTL (see Table 20).

0 = use eTPU clock / 2 as TCR1 clock source before the prescaler, if that clock source is selected by TCR1CTL.

NOTE

The clock source of the EAC angle tick generator will still be an even division of eTPU clock if TCR1CS=1, obeying to the fields TCR1P as if TCR1CS=0

Table 20. TCR1 Clock Source

TCR1CTL	TCR1CS ¹	TCR1 Clock before prescaler
00	0	Selects TCRCLK as clock source for the TCR1 prescaler ²
01	0	Selects Peripheral Timebase clock as source for the TCR1 prescaler.
10	0	Selects eTPU clock divided by 2 as clock source for the TCR1 prescaler.
10	1	Selects eTPU clock as clock source for the TCR1 prescaler.
11	0	TCR1 frozen, except as a STAC client.

NOTES:

¹ All other combinations of TCR1CTL and TCR1CS are reserved.

² This selection must not be used in Angle Mode.

Appendix D Microinstruction Formats

Table 21. Microinstruction Formats

format	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0										
A1	0 0 0			IMM[15:3]			IMM[7:2]						IMM[23:16]										FTN		IMM[11:9]					T2D		0 0										
A2													T4ABS					T2ABD					IMM[2]		CCS		[0:1]MMI					A B S E		A B D E		IMM[8]		0 1				
A3				ALUOP		CCSV													ALUO		PI[3:2]		AS/CE					ALUO		PI[1:0]		0		1 0								
A4				FLC		SHF															rsv		FLC [1:0]							A B S E		A B D E		1								
B1	1 0		0															SRC		P/D		AID[7:0] (global param)																				
B2			1															SRC		P/D		AID[8:0] (channel param)																				
B3	0 0 0			END														SRC		P/D		CCS		ZRO		STC		A B S E		A B D E		rsv		1 1								
B4	0 0 1			0		CCSV													1		AS/CE																					
B5								CIN						BINV					FL		rsv																					
B6				1															SRC		SEXT		A B S E					A B D E														
B7	0 1 1			END			SHF													SRC		PSC		MFLA		ERMA		MRLB		ERWB		ABSE		ABDE		CCS		MRLA		PSCS		
C1	0 1 0			0		OPACA		OPACB						TBSA					TBSB					LSR		MFLA		ERMA		MRLB		ERWB		PDCM								
C2				1		IPACA		IPACB																																		
D0	1 1 0			0		0		PSC						FLC										0		RSIZ		ZRO		AID[6:0] (engine param)												
D1						MRLE		1						CIRG										1		P/D		AID[7:0] (global param)														
D2																								1		P/D		AID[6:0] (channel param)														
D3	1 1 1			1		MRLE		FLS						PSCS					CIRQ[1:0]					1		P/D		ZRO		STC		1 1		0 0		rsv						
D4																								0		rsv		SMP R														
D5	1 1 0			1		MRLE		rsv						MTD										0		RSIZ		AID[7:0] (global param)														
D6																								1		P/D		AID[6:0] (channel param)														
D7	1 1 1			1		MRLE		OMW						TDL					MRLA					1		P/D		ZRO		STC		1 1		0 1		rsv						
D8																								0		rsv		SMP R														
D9	1 1 0			0		MRLE		0																1		P/D		ZRO		AID[6:0] (engine param)												
E1	1 1 1			BCC[4:0]														BAF[13:0]											00		P/D		STC									
E2				BCC[5]			J/C																						01		AID[2:0]											
E3																													10		rsv		SMP R									
E4																													11		1		rsv									
F1							rsv																						111		rsv											

ALU Operations
RAM Operations

Channel Control/Config Operations
Flow Control Operations

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

For Literature Requests Only:

Freescale Semiconductor Literature Distribution Center
1-800-441-2447 or 303-675-2140
Fax: 303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

Information in this document is provided solely to enable system and software implementers to use Freescale Semiconductor products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.

Freescale Semiconductor reserves the right to make changes without further notice to any products herein. Freescale Semiconductor makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale Semiconductor 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 Semiconductor 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 Semiconductor does not convey any license under its patent rights nor the rights of others. Freescale Semiconductor products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Freescale Semiconductor product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescale Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify and hold Freescale Semiconductor and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Freescale Semiconductor was negligent regarding the design or manufacture of the part.



Freescale™ and the Freescale logo are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners.

© Freescale Semiconductor, Inc. 2010. All rights reserved.