27.4 Registers

This section describes the registers of the STM.

Table 1 Register Overview - STM (ascending Offset Address)

Short Name	Long Name	Offset Address	Access Mode		Reset	Page Number
			Read	Write		
CLC	Clock Control Register	0000_H	U,SV	SV,E,P	Application Reset	7
ID	Module Identification Register	0008_H	U,SV	BE	Application Reset	8
TIM0	Timer Register 0	0010_H	U,SV	BE	Application Reset	9
TIM1	Timer Register 1	0014_H	U,SV	BE	Application Reset	10
TIM2	Timer Register 2	0018_H	U,SV	BE	Application Reset	10
TIM3	Timer Register 3	$001C_H$	U,SV	BE	Application Reset	10
TIM4	Timer Register 4	0020_H	U,SV	BE	Application Reset	11
TIM5	Timer Register 5	0024_H	U,SV	BE	Application Reset	11
TIM6	Timer Register 6	0028_H	U,SV	BE	Application Reset	12
CAP	Timer Capture Register	$002C_H$	U,SV	BE	Application Reset	12
CMPx	Compare Register x	0030_H+x *4	U,SV	U,SV,P	Application Reset	13
CMCON	Compare Match Control Register	0038_H	U,SV	U,SV,P	Application Reset	14
ICR	Interrupt Control Register	$003C_H$	U,SV	U,SV,P	Application Reset	16
ISCR	Interrupt Set/Clear Register	0040_H	U,SV	U,SV,P	Application Reset	17
TIM0SV	Timer Register 0 Second View	0050_H	U,SV	BE	Application Reset	9
CAPSV	Timer Capture Register Second View	0054_H	U,SV	BE	Application Reset	13
OCS	OCDS Control and Status Register	$00E8_H$	U,SV	SV,P,OEN	Debug Reset	19
KRSTCLR	Kernel Reset Status Clear Register	$00EC_H$	U,SV	SV,E,P	Application Reset	22
KRST1	Kernel Reset Register 1	$00F0_H$	U,SV	SV,E,P	Application Reset	21

Table 1 Register Overview - STM (ascending Offset Address) (cont'd)

Short Name	Long Name	Offset Address	Access Mode		Reset	Page Number
			Read	Write		
KRST0	Kernel Reset Register 0	$00F4_H$	U,SV	SV,E,P	Application Reset	21
ACCEN1	Access Enable Register 1	$00F8_H$	U,SV	SV,SE	Application Reset	20
ACCEN0	Access Enable Register 0	$00FC_H$	U,SV	SV,SE	Application Reset	20

27.4.1 Clock Control Register

Clock Control Register

The STM clock control register is used to switch the STM on or off and to control its input clock rate. After reset, the STM is always enabled and starts counting. The STM can be disabled by setting bit DISR to 1.

CLC

Clock Control Register (0000_H) Application Reset Value: $0000\ 0000_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0												EDIS	0	DISS	DISR
r												rw	rw	r	rw

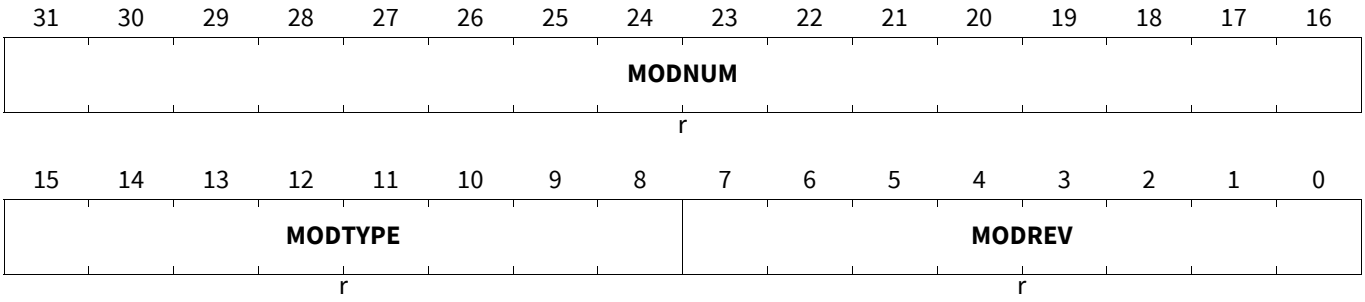
Field	Bits	Type	Description
DISR	0	rw	**Module Disable Request Bit** Used for enable/disable control of the STM module. *Note: f_{STM} is generated by the CCU.* 0_B No disable requested 1_B Disable requested
DISS	1	r	**Module Disable Status Bit** Bit indicates the current status of the STM module. 0_B STM module is enabled 1_B STM module is disabled
EDIS	3	rw	**Sleep Mode Enable Control** Used for module sleep mode control.
0	2	rw	**Reserved** Read as 0; shall be written with 0.
0	31:4	r	**Reserved** Read as 0; should be written with 0.

Module Identification Register

The STM Module Identification Register ID contains read-only information about the module version.

ID

Module Identification Register **(0008_H)** **Application Reset Value: $0000\ C007_H$**



Field	Bits	Type	Description
MODREV	7:0	r	**Module Revision Number** MODREV defines the module revision number. The value of a module revision starts with 01_H (first revision). Current revision is 0x7.
MODTYPE	15:8	r	**Module Type** This bit field defines the module as a 32-bit module: $C0_H$
MODNUM	31:16	r	**Module Number Value** This bit field defines the module identification number for the STM: 0068_H

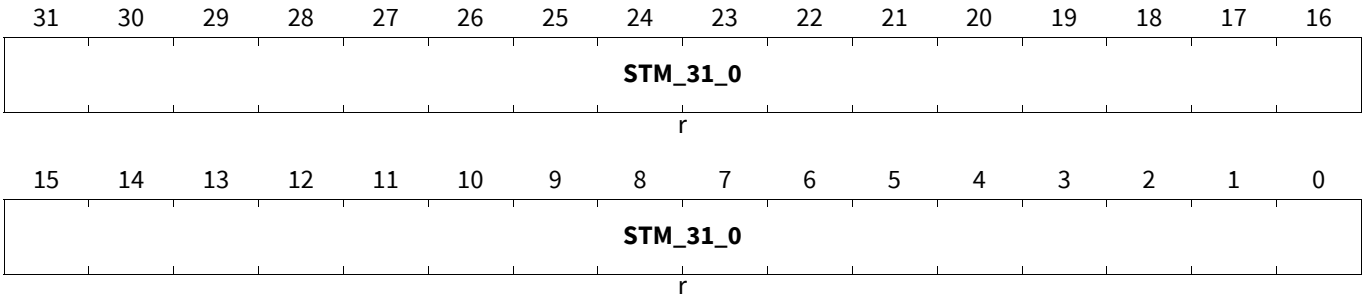
27.4.2 Timer/Capture Registers

Registers TIM0 to TIM6 provide 32-bit views at varying resolutions of the underlying STM counter.

Timer Register 0

Register TIM0SV address the STM[31:0] bits at a second optional address as TIM0.

TIM0
Timer Register 0 **(0010_H)** **Application Reset Value: $0000\ 0000_H$**

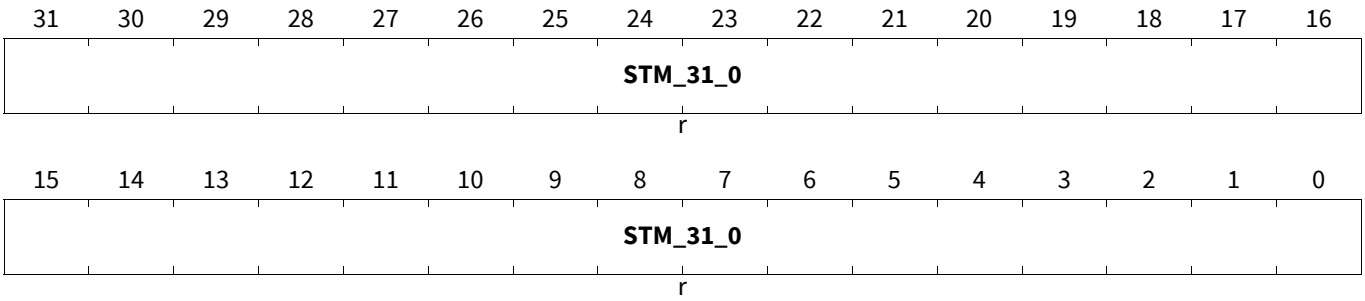


Field	Bits	Type	Description
STM_31_0	31:0	r	**System Timer Bits [31:0] - STM[31:0]** This bit field contains bits [31:0] of the 64-bit STM.

Timer Register 0 Second View

Register TIM0SV address the STM[31:0] bits at a second optional address as TIM0.

TIM0SV
Timer Register 0 Second View **(0050_H)** **Application Reset Value: $0000\ 0000_H$**

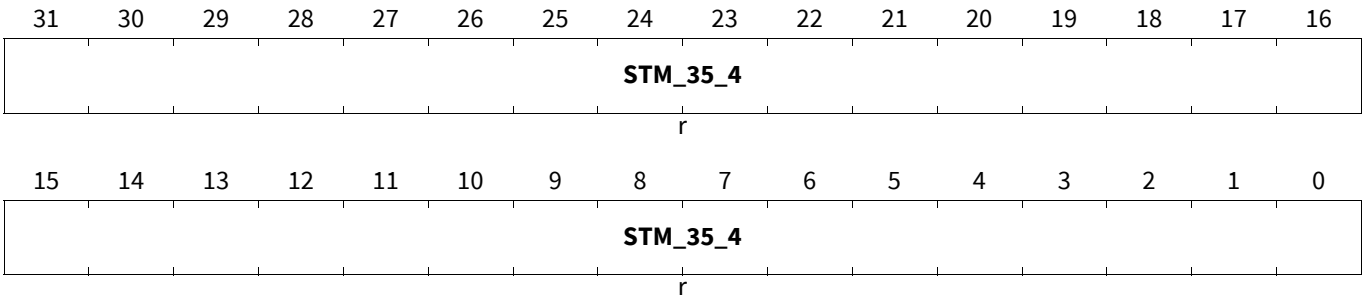


Field	Bits	Type	Description
STM_31_0	31:0	r	**System Timer Bits [31:0] - STM[31:0]** This bit field contains bits [31:0] of the 64-bit STM.

Timer Register 1

TIM1

Timer Register 1 **(0014_H)** **Application Reset Value: $0000\ 0000_H$**

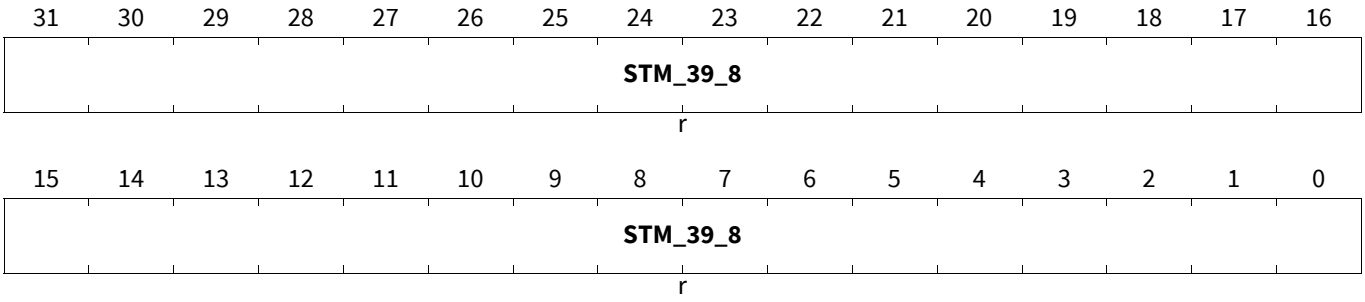


Field	Bits	Type	Description
STM_35_4	31:0	r	**System Timer Bits [35:4] - STM[35:4]** This bit field contains bits [35:4] of the 64-bit STM.

Timer Register 2

TIM2

Timer Register 2 **(0018_H)** **Application Reset Value: $0000\ 0000_H$**

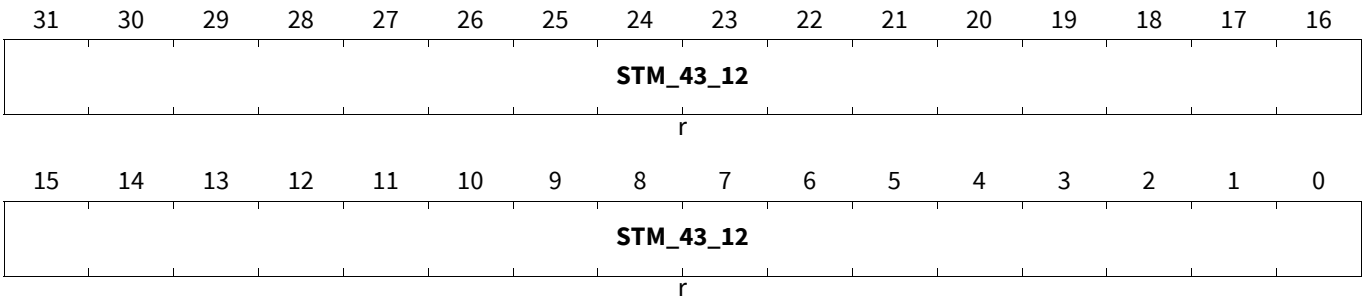


Field	Bits	Type	Description
STM_39_8	31:0	r	**System Timer Bits [39:8] - STM[39:8]** This bit field contains bits [39:8] of the 64-bit STM.

Timer Register 3

TIM3

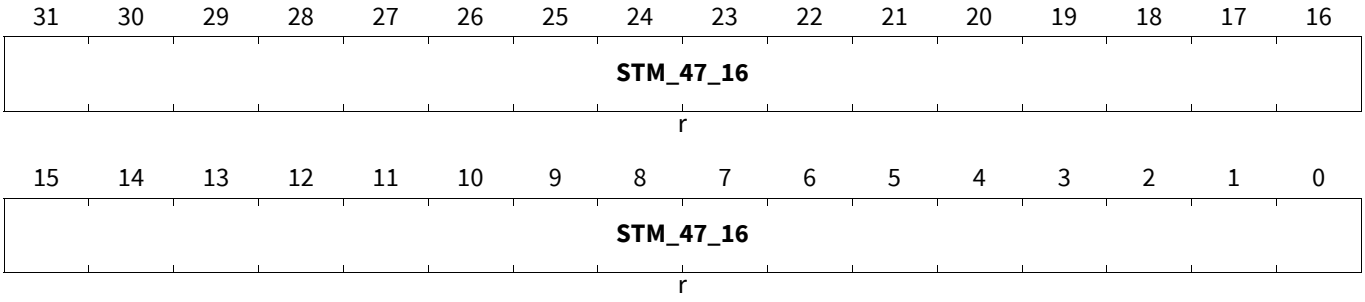
Timer Register 3 **($001C_H$)** **Application Reset Value: $0000\ 0000_H$**



Field	Bits	Type	Description
STM_43_12	31:0	r	**System Timer Bits [43:12] - STM[43:12]** This bit field contains bits [43:12] of the 64-bit STM.

Timer Register 4

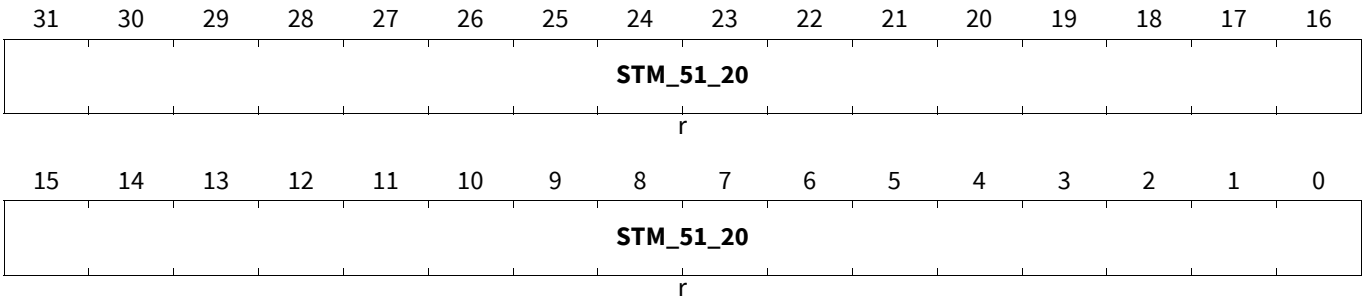
TIM4
Timer Register 4 **(0020_H)** **Application Reset Value: $0000\ 0000_H$**



Field	Bits	Type	Description
STM_47_16	31:0	r	**System Timer Bits [47:16] - STM[47:16]** This bit field contains bits [47:16] of the 64-bit STM.

Timer Register 5

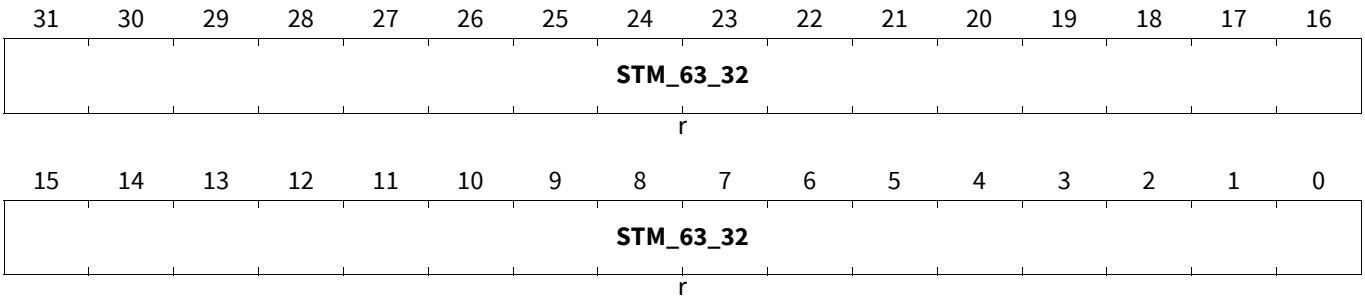
TIM5
Timer Register 5 **(0024_H)** **Application Reset Value: $0000\ 0000_H$**



Field	Bits	Type	Description
STM_51_20	31:0	r	**System Timer Bits [51:20] - STM[51:20]** This bit field contains bits [51:20] of the 64-bit STM.

Timer Register 6

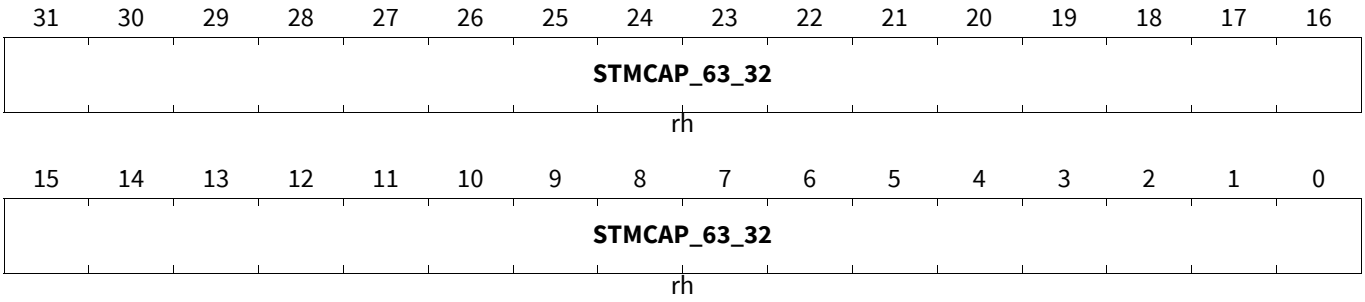
TIM6
Timer Register 6 **(0028_H)** **Application Reset Value: $0000\ 0000_H$**



Field	Bits	Type	Description
STM_63_32	31:0	r	**System Timer Bits [63:32] - STM[63:32]** This bit field contains bits [63:32] of the 64-bit STM.

Timer Capture Register

CAP
Timer Capture Register **($002C_H$)** **Application Reset Value: $0000\ 0000_H$**

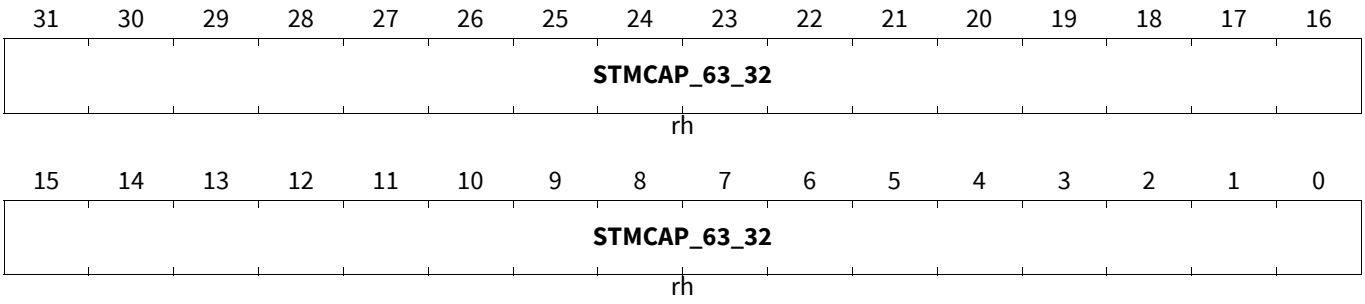


Field	Bits	Type	Description
STMCAP_63_32	31:0	rh	**Captured System Timer Bits [63:32] - STMCAP[63:32]** The capture register STMCAP always captures the STM bits [63:32] when one of the registers TIM0 to TIM6 or TIM0SV is read. This capture operation is performed in order to enable software to operate with a coherent value of all the 64 STM bits at one time stamp.This bit field contains bits [63:32] of the 64-bit STM. *Note: Reading register TIM0SV captures also the read value for register TIM6. In this way reading TIM0SV followed by CAPSV delivers the timer values for the first read request.*

Timer Capture Register Second View

CAPSV

Timer Capture Register Second View **(0054_H)** **Application Reset Value: $0000\ 0000_H$**



Field	Bits	Type	Description
STMCAP_63_32	31:0	rh	**Captured System Timer Bits [63:32] - STMCAP[63:32]** The capture register STMCAP always captures the STM bits [63:32] when one of the registers TIM0 to TIM6 or TIM0SV is read. This capture operation is performed in order to enable software to operate with a coherent value of all the 64 STM bits at one time stamp.This bit field contains bits [63:32] of the 64-bit STM. *Note: Reading register TIM0SV captures also the read value for register TIM6. In this way reading TIM0SV followed by CAPSV delivers the timer values for the first read request.*

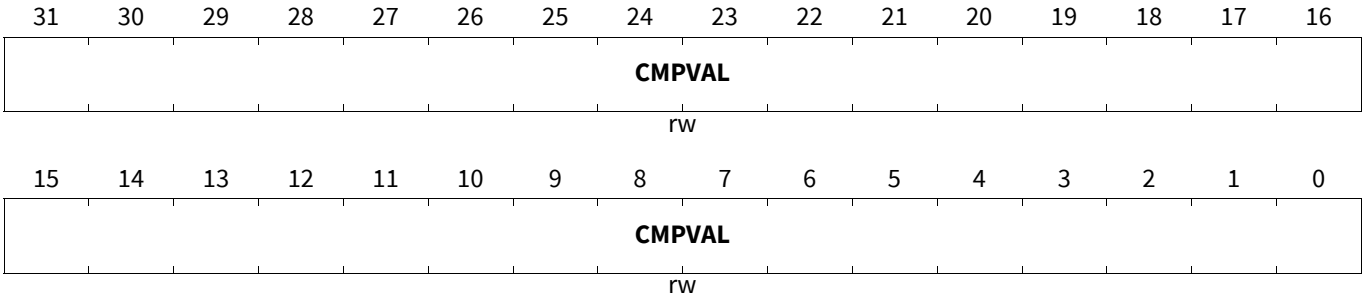
27.4.3 Compare Registers

The compare register CMPx holds up to 32-bits; its value is compared to the value of the STM.

Compare Register x

CMPx (x=0-1)

Compare Register x **(0030_H+x*4)** **Application Reset Value: $0000\ 0000_H$**



Field	Bits	Type	Description
CMPVAL	31:0	rw	**Compare Value of Compare Register x** This bit field holds up to 32 bits of the compare value (right-adjusted).

Compare Match Control Register

The STM Compare Match Control Register controls the parameters of the compare logic.

CMCON

Compare Match Control Register **(0038_H)** **Application Reset Value: $0000\ 0000_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0			**MSTART1**					**0**			**MSIZE1**				
r			rw					r			rw				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0			**MSTART0**					**0**			**MSIZE0**				
r			rw					r			rw				

Field	Bits	Type	Description
MSIZE0	4:0	rw	**Compare Register Size for CMP0** This bit field determines the number of bits in register CMP0 (starting from bit 0) that are used for the compare operation with the System Timer. ... 00_H CMP0[0] used for compare operation 01_H CMP0[1:0] used for compare operation $1E_H$ CMP0[30:0] used for compare operation $1F_H$ CMP0[31:0] used for compare operation
MSTART0	12:8	rw	**Start Bit Location for CMP0** This bit field determines the lowest bit number of the 64-bit STM that is compared with the content of register CMP0 bit 0. The number of bits to be compared is defined by bit field MSIZE0. ... 00_H STM[0] is the lowest bit number 01_H STM[1] is the lowest bit number $1E_H$ STM[30] is the lowest bit number $1F_H$ STM[31] is the lowest bit number
MSIZE1	20:16	rw	**Compare Register Size for CMP1** This bit field determines the number of bits in register CMP1 (starting from bit 0) that are used for the compare operation with the System Timer. ... 00_H CMP1[0] used for compare operation 01_H CMP1[1:0] used for compare operation $1E_H$ CMP1[30:0] used for compare operation $1F_H$ CMP1[31:0] used for compare operation

Field	Bits	Type	Description
MSTART1	28:24	rw	**Start Bit Location for CMP1** This bit field determines the lowest bit number of the 64-bit STM that is compared with the content of register CMP1 bit 0. The number of bits to be compared is defined by bit field MSIZE1. ... 00_H STM[0] is the lowest bit number 01_H STM[1] is the lowest bit number $1E_H$ STM[30] is the lowest bit number $1F_H$ STM[31] is the lowest bit number
0	7:5, 15:13, 23:21, 31:29	r	**Reserved** Read as 0; should be written with 0.

27.4.4 Interrupt Registers

Interrupt Control Register

The two compare match interrupts of the STM are controlled by the STM Interrupt Control Register.

ICR
Interrupt Control Register **(003C$_H$)** **Application Reset Value: 0000 0000$_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0									CMP1OS	CMP1IR	CMP1EN	0	CMP0OS	CMP0IR	CMP0EN
r									rw	rh	rw	r	rw	rh	rw

Field	Bits	Type	Description
CMP0EN	0	rw	**Compare Register CMP0 Interrupt Enable Control** This bit enables the compare match interrupt with compare register CMP0. 0_B Interrupt on compare match with CMP0 disabled 1_B Interrupt on compare match with CMP0 enabled
CMP0IR	1	rh	**Compare Register CMP0 Interrupt Request Flag** This bit indicates whether or not a compare match event of compare register CMP0 has occured. CMP0IR can be cleared by software and can be set by software, too (see ISCR register). After a STM reset operation, CMP0IR is immediately set as a result of a compare match event with the reset values of the STM and the compare registers CMP0. *Note: This flag does not gate the interrupt generation. i.e., even if this flag is not cleared, compare match event interrupts are forwarded if CMPxEN is set.* 0_B A compare match event has not been detected since the bit was last cleared. 1_B A compare match event has been detected.
CMP0OS	2	rw	**Compare Register CMP0 Interrupt Output Selection** This bit determines the interrupt output that is activated on a compare match event of compare register CMP0. 0_B Interrupt output STMIR0 selected 1_B Interrupt output STMIR1 selected

Field	Bits	Type	Description
CMP1EN	4	rw	**Compare Register CMP1 Interrupt Enable Control** This bit enables the compare match interrupt with compare register CMP1. 0_B Interrupt on compare match with CMP1 disabled 1_B Interrupt on compare match with CMP1 enabled
CMP1IR	5	rh	**Compare Register CMP1 Interrupt Request Flag** This bit indicates whether or not a compare match event of compare register CMP1 has occured. CMP1IR can be cleared by software and can be set by software, too (see ISCR register). After a STM reset operation, CMP1IR is immediately set as a result of a compare match event with the reset values of the STM and the compare register CMP1. *Note: This flag does not gate the interrupt generation. i.e., even if this flag is not cleared, compare match event interrupts are forwarded if CMPxEN is set.* 0_B A compare match event has not been detected since the bit was last cleared. 1_B A compare match event has been detected.
CMP1OS	6	rw	**Compare Register CMP1 Interrupt Output Selection** This bit determines the interrupt output that is activated on a compare match event of compare register CMP1. 0_B Interrupt output STMIR0 selected 1_B Interrupt output STMIR1 selected
0	3, 31:7	r	**Reserved** Read as 0; should be written with 0.

Interrupt Set/Clear Register

The bits in the STM Interrupt Set/Clear Register make it possible to set or cleared the compare match interrupt request status flags of register ICR.

Note: Reading register ISCR always returns 0000 0000$_H$.

ISCR

Interrupt Set/Clear Register (0040$_H$) Application Reset Value: 0000 0000$_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0												CMP1IRS	CMP1IRR	CMP0IRS	CMP0IRR
r												w	w	w	w

Field	Bits	Type	Description
CMP0IRR	0	w	**Reset Compare Register CMP0 Interrupt Flag** 0_B Bit ICR.CMP0IR is not changed. 1_B Bit ICR.CMP0IR is cleared.
CMP0IRS	1	w	**Set Compare Register CMP0 Interrupt Flag** 0_B Bit ICR.CMP0IR is not changed. 1_B Bit ICR.CMP0IR is set. The state of bit CMP0IRR is “don’t care” in this case.
CMP1IRR	2	w	**Reset Compare Register CMP1 Interrupt Flag** 0_B Bit ICR.CMP1IR is not changed. 1_B Bit ICR.CMP1IR is cleared.
CMP1IRS	3	w	**Set Compare Register CMP1 Interrupt Flag** 0_B Bit ICR.CMP1IR is not changed. 1_B Bit ICR.CMP1IR is set. The state of bit CMP1IRR is “don’t care” in this case.
0	31:4	r	**Reserved** Read as 0; should be written with 0.

27.4.5 Interface Registers

OCDS Control and Status Register

The OCDS Control and Status (OCS) register is cleared by Debug Reset.

The OCS register can only be written when the OCDS is enabled.

If OCDS is being disabled, the OCS register value will not change.When OCDS is disabled the OCS suspend control is ineffective.

OCS

OCDS Control and Status Register (00E8$_H$) Debug Reset Value: 0000 0000$_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0		SUSSTA	SUS_P	SUS				0							
r		rh	w	rw				r							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0														0	
r														rw	

Field	Bits	Type	Description
SUS	27:24	rw	**OCDS Suspend Control** Controls the sensitivity of STMx to the suspend signal coming from CPUx (CPUxSUSOUT) 0_H Will not suspend 1_H Reserved, do not use this combination 2_H 64-bit counter will be stopped **others**, Reserved, do not use this combination
SUS_P	28	w	**SUS Write Protection** SUS is only written when SUS_P is 1, otherwise unchanged. Read as 0.
SUSSTA	29	rh	**Suspend State** 0_B Module is not (yet) suspended 1_B Module is suspended
0	2:0	rw	**Reserved** Read as 0; must be written with 0.
0	23:3, 31:30	r	**Reserved** Read as 0; must be written with 0.

Table 2 Access Mode Restrictions of OCS sorted by descending priority

Mode Name	Access Mode		Description
write 1 to **SUS_P**	rw	SUS	
(default)	r	SUS	

Access Enable Register 0

The Access Enable Register 0 controls write access for transactions with the on chip bus master TAG ID 000000B to 011111B (see On Chip Bus chapter for the products TAG ID <-> master peripheral mapping). The BPI_FPI is prepared for a 6-bit TAG ID. The registers ACCEN0 / ACCEN1 are providing one enable bit for each possible 6-bit TAG ID encoding.

Mapping of TAG IDs to ACCEN0.ENx: EN0 -> TAG ID 000000B, EN1 -> TAG ID 000001B , …,EN31 -> TAG ID 011111B.

ACCEN0

Access Enable Register 0 **($00FC_H$)** **Application Reset Value: $FFFF\ FFFF_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EN31	**EN30**	**EN29**	**EN28**	**EN27**	**EN26**	**EN25**	**EN24**	**EN23**	**EN22**	**EN21**	**EN20**	**EN19**	**EN18**	**EN17**	**EN16**
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EN15	**EN14**	**EN13**	**EN12**	**EN11**	**EN10**	**EN9**	**EN8**	**EN7**	**EN6**	**EN5**	**EN4**	**EN3**	**EN2**	**EN1**	**EN0**
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Field	Bits	Type	Description
ENn (n=0-31)	n	rw	**Access Enable for Master TAG ID n** This bit enables write access to the module kernel addresses for transactions with the Master TAG ID n 0_B Write access will not be executed 1_B Write access will be executed

Access Enable Register 1

The Access Enable Register 1 controls write access for transactions with the on chip bus master TAG ID 100000B to 111111B (see On Chip Bus chapter for the products TAG ID master peripheral mapping).

Mapping of TAG IDs to ACCEN1.ENx: EN0 -> TAG ID 100000B, EN1 -> TAG ID 100001B, …, EN31 -> TAG ID 111111B.

ACCEN1

Access Enable Register 1 **($00F8_H$)** **Application Reset Value: $0000\ 0000_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0															
r															

Field	Bits	Type	Description
0	31:0	r	**Reserved** Read as 0; should be written with 0.

Kernel Reset Register 0

The Kernel Reset Register 0 is used to reset the related module kernel. To reset a module kernel it is necessary to set the RST bits by writing with ´1´ in both Kernel Reset Registers related to the module kernel that should be reset (kernel 0 or kernel 1). In order support modules with two kernel the BPI_FPI provides two set of kernel reset registers. The RST bit will be re-set by the BPI with the end of the BPI kernel reset sequence.

Kernel Reset Register 0 includes a kernel reset status bit that is set to ´1´ by the BPI_FPI in the same clock cycle the RST bit is re-set by the BPI_FPI. This bit can be used to detect that a kernel reset was processed. The bit can be re-set to ´0´ by writing to it with ´1´.

KRST0

Kernel Reset Register 0 **(00F4$_H$)** **Application Reset Value: 0000 0000$_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0														RSTSTAT	RST
r														rh	rwh

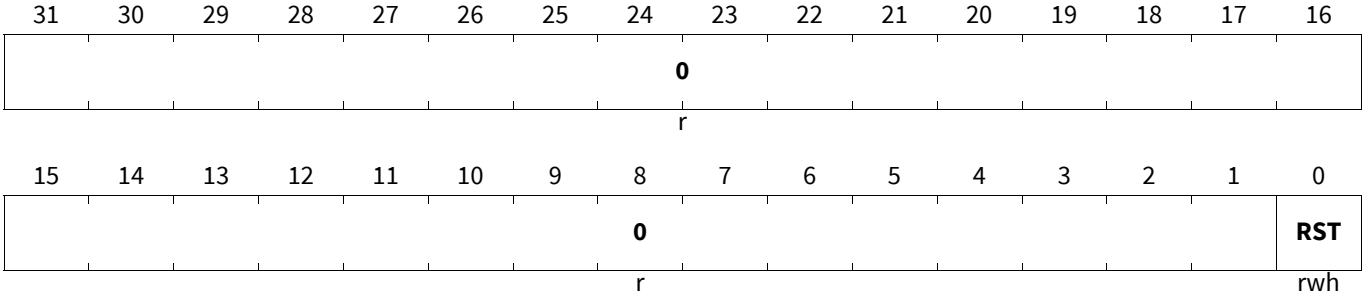
Field	Bits	Type	Description
RST	0	rwh	**Kernel Reset** This reset bit can be used to request for a kernel reset. The kernel reset will be executed if the reset bits of both kernel registers are set. The RST bit will be cleared (re-set to ´0´) by the BPI_FPI after the kernel reset was executed. 0_B No kernel reset was requested 1_B A kernel reset was requested
RSTSTAT	1	rh	**Kernel Reset Status** This bit indicates wether a kernel reset was executed or not. This bit is set by the BPI_FPI after the execution of a kernel reset in the same clock cycle both reset bits. This bit can be cleared by writing with ´1´ to the CLR bit in the related KRSTCLR register. 0_B No kernel reset was executed 1_B Kernel reset was executed
0	31:2	r	**Reserved** Read as 0; should be written with 0.

Kernel Reset Register 1

The Kernel Reset Register 1 is used to reset the STM kernel. STM kernel registers related to the Debug Reset (Class 1) are not influenced. To reset the STM kernel it is necessary to set the RST bits by writing with ´1´ in both Kernel Reset registers. The RST bit will be cleared with the end of the BPI kernel reset sequence.

KRST1

Kernel Reset Register 1 **(00F0$_H$)** **Application Reset Value: 0000 0000$_H$**



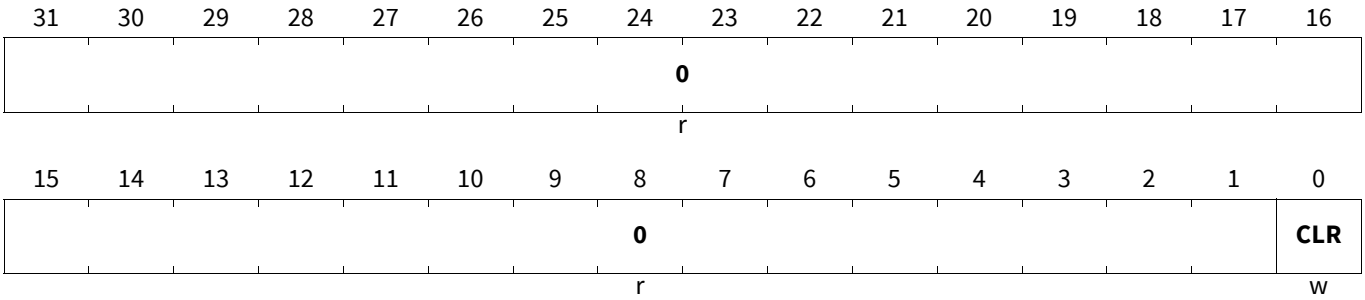
Field	Bits	Type	Description
RST	0	rwh	**Kernel Reset** This reset bit can be used to request for a kernel reset. The kernel reset will be executed if the reset bits of both kernel reset registers is set. The RST bit will be cleared (re-set to ´0´) after the kernel reset was executed. 0_B No kernel reset was requested 1_B A kernel reset was requested
0	31:1	r	**Reserved** Read as 0; should be written with 0.

Kernel Reset Status Clear Register

The Kernel Reset Register Clear register is used to clear the Kernel Reset Status bit (<>_KRST0.RSTSTAT).

KRSTCLR

Kernel Reset Status Clear Register **(00EC$_H$)** **Application Reset Value: 0000 0000$_H$**



Field	Bits	Type	Description
CLR	0	w	**Kernel Reset Status Clear** Read always as 0. 0_B No action 1_B Clear Kernel Reset Status KRST0.RSTSTAT
0	31:1	r	**Reserved** Read as 0; should be written with 0.