Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Obsolete
Core Processor	V850ES
Core Size	32-Bit Single-Core
Speed	32MHz
Connectivity	CANbus, CSI, EBI/EMI, I²C, LINbus, UART/USART
Peripherals	DMA, LVD, POR, PWM, WDT
Number of I/O	128
Program Memory Size	256KB (256K x 8)
Program Memory Type	FLASH
EEPROM Size	32K x 8
RAM Size	16K x 8
Voltage - Supply (Vcc/Vdd)	3.3V ~ 5.5V
Data Converters	A/D 24x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	144-LQFP
Supplier Device Package	144-LQFP (20x20)
Purchase URL	https://www.e-xfl.com/product-detail/renesas-electronics-america/upd70f3378m1gja2-gae-ax

2.2.2 Pin function configuration

The registers for pin function configuration define the general function of a pin:

- port mode or alternative mode
- in port mode: input mode or output mode
- in alternative mode: selection of one of the alternative functions in alternative mode
- normal mode or on-chip debug mode (N-Wire interface)

An overview of the register settings is given in the table below.

Table 2-3 Pin function configuration (overview)

Function	Registers					I/O
	OCDM	PMC	PM	PFCE	PFC	
Port mode (output)	0	0	0	X	X	O
Port mode (input)			1	X	X	I
Alternative mode (alternative function 1)		1	X	0	0	I/O[a]
Alternative mode (alternative function 2)					1	
Alternative mode (alternative function 3)				1	0	
Alternative mode (alternative function 4)					1	
On-chip debug mode[b]	1	X	X	X	X	I/O

a) In alternative mode, the corresponding port type defines whether a pin is in input mode or output mode.

b) In on-chip debug mode, the corresponding pins are automatically set as input or output pins to provide the N-Wire interface. In this mode, the configuration of these pins can not be changed by the pin configuration registers. Refer to chapter *"On-Chip Debug Unit" on page 930* for details.

2.4.25 Port type F1010-U

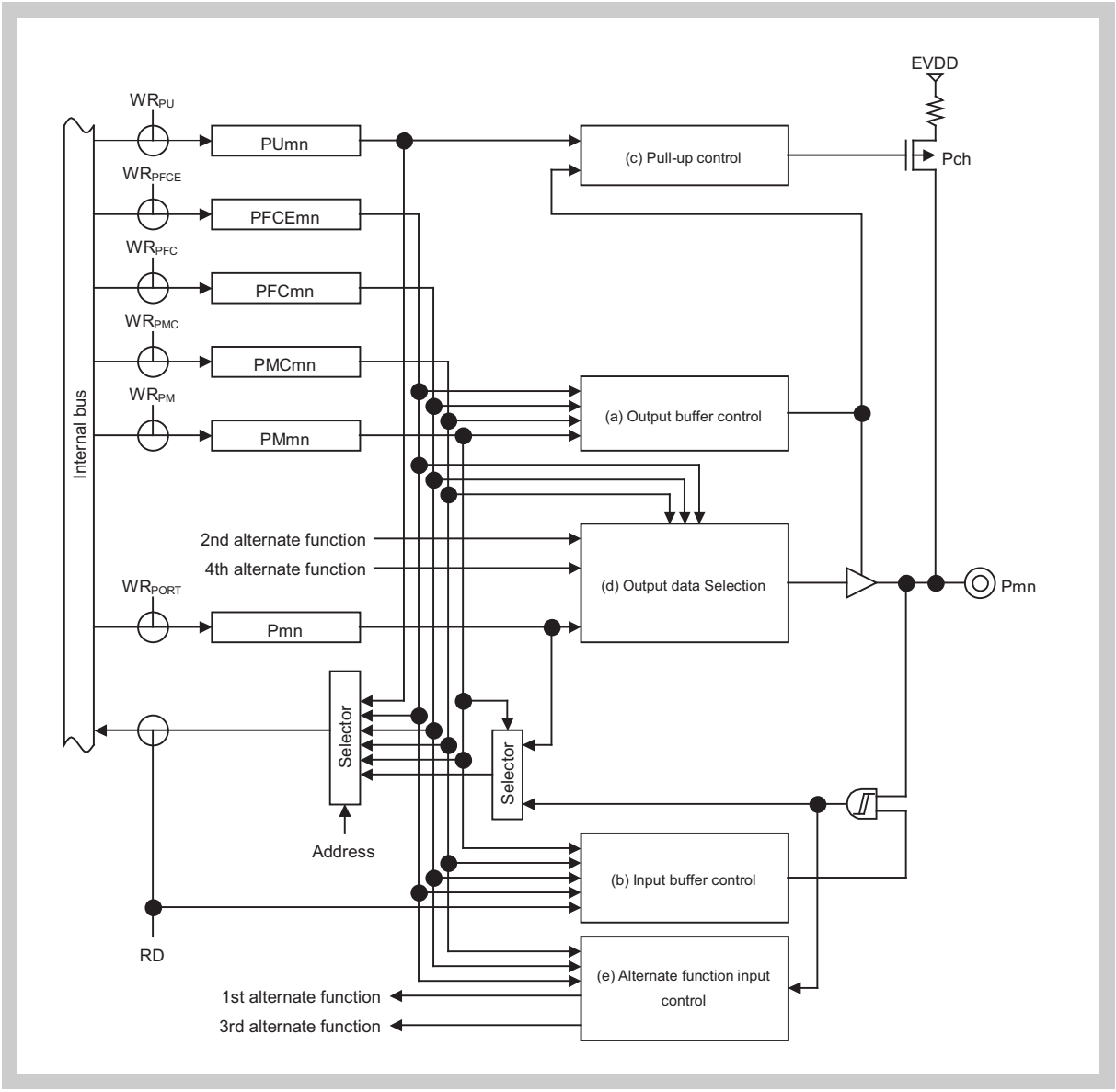

Figure 2-34 Port type F1010-U block diagram

2.4.37 Port type Fx103-UI

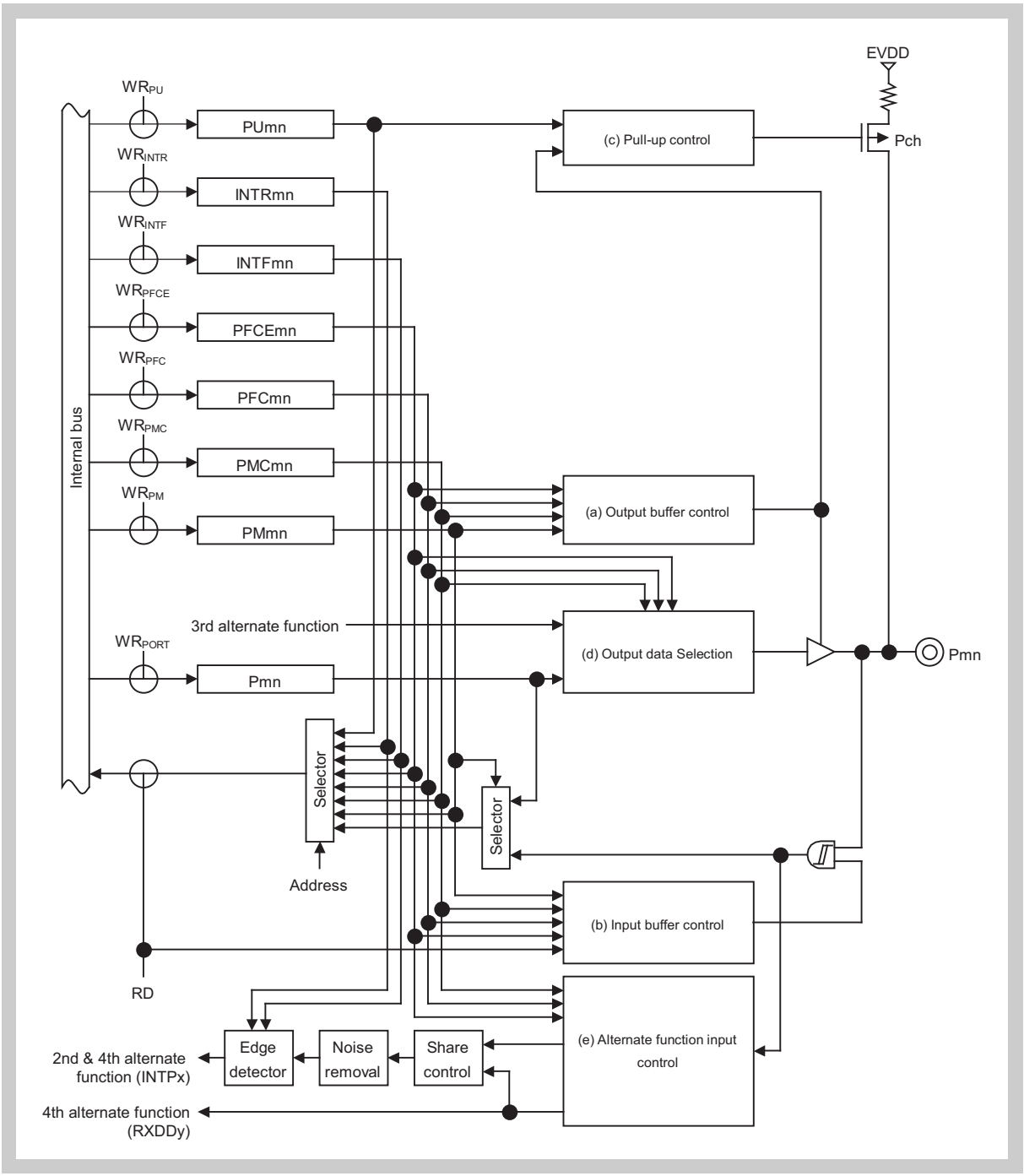

Figure 2-46 Port type Fx103-UI block diagram

2.5.14 Port group 15 (V850ES/FK3)

Note Port group 15 is available only for V850ES/FK3.

Port group 15 is an 8-bit port group. It includes pins for the following functions:

- Timer TAA5 channels (TIAA50, TIAA51 and TOAA50, TOAA51)
- Timer TAA6 channels (TIAA60, TIAA61 and TOAA60, TOAA61)
- Timer TAA7 channels (TIAA70, TIAA71 and TOAA70, TOAA71)
- CAN4 transmit/receive data (CTXD4, CRXD4)

Port group 15 includes the following pins:

Table 2-42 Port group 15: pin functions and port types

Pin functions in different modes			Pin function after reset	Port type	Noise filter[a]	Input charact.[b]
Port mode (PMCnm = 0)	Alternative mode (PMCnm = 1)					
	Function 1 PFC = 0	Function 2 PFC = 1				
P150	TIAA50 (I)	TOAA50 (I)	P150 (I)	E10-U	–	S2
P151	TIAA51 (I)	TOAA51 (I)	P151 (I)	E10-U	A	S2
P152	TIAA60 (I)	TOAA60 (I)	P152 (I)	E10-U	A	S2
P153	TIAA61 (I)	TOAA61 (I)	P153 (I)	E10-U	A	S2
P154	TIAA70 (I)	TOAA70 (I)	P154 (I)	E10-U	A	S2
P155	TIAA71 (I)	TOAA71 (I)	P155 (I)	E10-U	A	S2
P156	CTXD4 (O)	–	P156 (I)	D0-U	–	S1
P157	CRXD4 (I)	–	P157 (I)	D1-U	–	S1

a) A: analog noise filter only for TIAAnm inputs
B: analog and digital noise filter
–: no noise filter

b) S1: Schmitt trigger (30/70%); S2: Schmitt trigger (40/80%); x: CMOS

Table 2-43 Port group 15: configuration registers

Register	Address	Initial value	Used bits							
PMC15	FFFF F45E$_H$	00$_H$	PMC157	PMC156	PMC155	PMC154	PMC153	PMC152	PMC151	PMC150
PM15	FFFF F43E$_H$	FF$_H$	PM157	PM156	PM155	PM154	PM153	PM152	PM151	PM150
PFC15	FFFF F47E$_H$	00$_H$	X	X	PFC155	PFC154	PFC153	PFC152	PFC151	PFC150
P15	FFFF F41E$_H$	undefined	P157	P156	P155	P154	P153	P152	P151	P150
PU15	FFFF FC5E$_H$	00$_H$	PU157	PU156	PU155	PU154	PU153	PU152	PU151	PU150

Access All 8-bit registers can be accessed in 8-bit or 1-bit units.

2.8 Recommended Connection of unused Pins

If a pin is not used, it is recommended to connect it as follows:

Table 2-56 Recommended connection of unused pins

Pin	Recommended connection
Port pins	
pins of port groups 0, 1, 3 to 6, 8, 9, 15 (except P05 of Port 0)	• output pins: leave open • input pins: connect to EVDD or EVSS via a resistor, or use the internal pull-up resistor
P05 of port group 0	• output pins: leave open • input pins: connect to EVSS via a resistor
pins of port groups 7, 12	• output pins: leave open • input pins: connect to AVREF0 or AVSS via a resistor
pins of port groups CD, CM, CS, CT, DL	• output pins: leave open • input pins: connect to BVDD or BVSS via a resistor
Non-port pins	
AVREF0, AVREF1	connect to VDD
FLMD0	connect to VSS
REGC, REGC2	connect to regulator output stability capacity
XT1	connect to VSS via a resistor
XT2	leave open
Internally connected pins	
IC	connect to VSS via a resistor

Note
1. When connecting the unused pins with a power supply or ground, it is recommended to connect the pins through a resistance of 1 to 10 KΩ.
2. If the overall maximum output current exceeds its maximum value the output buffer can be damaged. We recommend the placement of a series resistor to prevent damage in case of accidentally enabled outputs. Refer to the absolute maximum rating parameter in the Datasheet.

Note

1. When IDLE2 mode is released, set the stabilization time to the following requirements:
   - In case of PLL mode: PLL lockup time requirements
   - In case of clock-through mode: flash set up time requirement

   For the exact timing values, refer to the Datasheet.
2. When STOP mode is released, set the stabilization time to the following requirements:
   - In case of PLL mode: PLL lockup time requirement
   - In case of clock-through mode:flash set up time requirement

   For the exact timing values, refer to the Datasheet.
3. If the required oscillation stabilization time of the MainOSC exceeds the above times, set the value to the required oscillation stabilization time of the MainOSC.

(2) IMR1 - Interrupt mask register 1

Address FFFF F102$_H$

Initial Value FFFF$_H$. The register is initialized by any reset

IMR1

15	14	13	12	11	10	9	8
CB0RMK	TM0EQMK0	TAA4CCMK1	TAA4CCMK0	TAA4OVMK	TAA3CCMK1	TAA3CCMK0	TAA3OVMK
7	6	5	4	3	2	1	0
TAA2CCMK1	TAA2CCMK0	TAA2OVMK	TAA1CCMK1	TAA1CCMK0	TAA1OVMK	TAA0CCMK1	TAA0CCMK0

(3) IMR2 - Interrupt mask register 2

Address FFFF F104$_H$

Initial Value FFFF$_H$. The register is initialized by any reset

- V850ES/FE3
- V850ES/FF3
- µPD70F3374, µPD70F3375 of V850ES/FG3
- µPD70F3378 of V850ES/FJ3

IMR2

15	14	13	12	11	10	9	8
DMAMK0	C0TRXMK	C0RECMK	C0WUPMK	C0ERRMK	ADMK	IIC0MK	UD1TMK
7	6	5	4	3	2	1	0
UD1RMK	UD1SMK	UD0TMK	UD0RMK	UD0SMK	CB1TMK	CB1RMK	CB0TMK

- µPD70F3376A, µPD70F3377A of V850ES/FG3
- µPD70F3379, µPD70F3380, µPD70F3381, µPD70F3382 of V850ES/FJ3
- V850ES/FK3

IMR2

15	14	13	12	11	10	9	8
DMAMK0	C0TRXMK	C0RECMK	C0WUPMK	C0ERRMK	ADMK	IIC0MK UD4SMK	UD1TMK
7	6	5	4	3	2	1	0
UD1RMK	UD1SMK	UD0TMK	UD0RMK	UD0SMK	CB1TMK	CB1RMK	CB0TMK

(4) IMR3 - Interrupt mask register 3

Address FFFF F106$_H$

Initial Value FFFF$_H$. The register is initialized by any reset

- V850ES/FE3
- V850ES/FF3

IMR3

15	14	13	12	11	10	9	8
1	1	1	1	1	1	1	1
7	6	5	4	3	2	1	0
FLMK	1	WTMK	WTIMK	KRMK	DMAMK3	DMAMK2	DMAMK1

7.2.5 Data flash writing

The data flash can be written by using the data flash library or serial programming with an external flash programmer tool.

Programming during normal operation is achieved by using the data flash access layer software library. The data flash access layer is described in a separate User's Manual.

Note The chip erase command of an external programmer erases also the data flash.

7.3.4 Flash memory programming control

The procedure to program the flash memory is illustrated below.

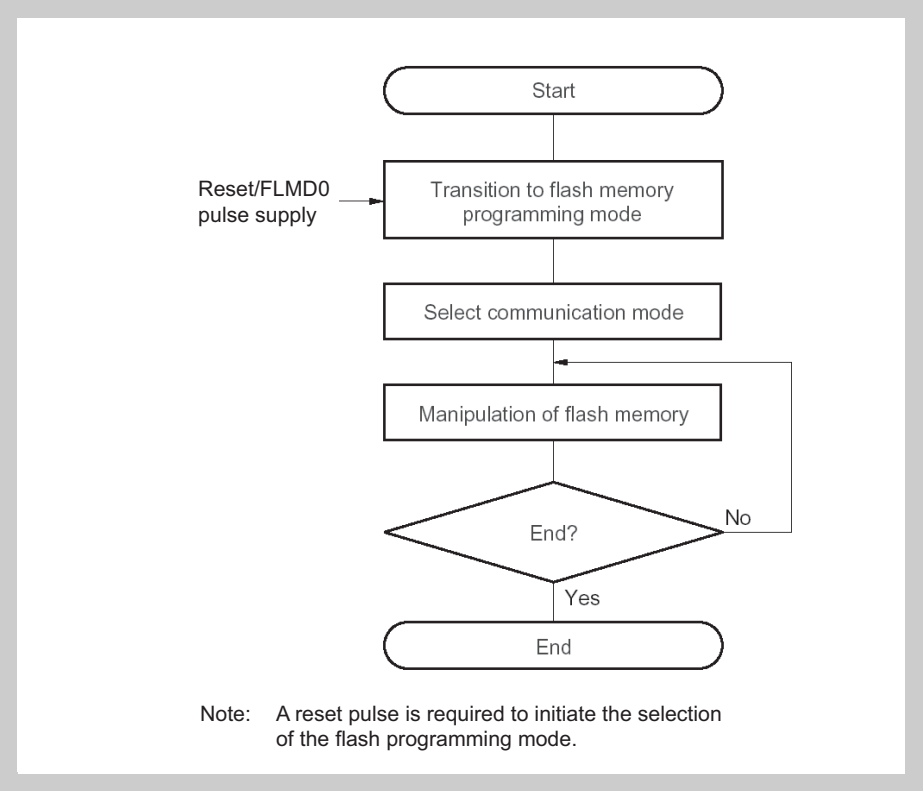

Figure 7-8 Flash memory programming procedure

(1) Operation mode control

To rewrite the contents of the flash memory by using the flash programmer, set the microcontroller in the flash memory programming mode.

To set this mode, set the FLMD0 and FLMD1 pins as shown in *Table 7-8 on page 316* and release $\overline{\text{RESET}}$.

In the normal operation mode, VSS is input to the FLMD0 pin. A pull-down resistor at the FLMD0 pin ensures normal operation mode if no flash programmer is connected. In the flash memory programming mode, the V_{DD} write voltage is supplied to the FLMD0 pin. Additionally the FLMD1 pin has to hold VSS level.

Table 7-8 Operation mode setting

Pins		Operation mode
FLMD0	**FLMD1**	
V_{SS}	Don't care	Normal operation mode
V_{DD}	V_{SS}	Flash programming mode
	V_{DD}	Setting prohibited

An example of connection of the FLMD0 and FLMD1 pins is shown below. FLMD1 can be connected to ground via a resistor. Alternatively the FLMD1 pin may also be connected directly to the FLMD1 signal of the flash programmer.

7.4.4 Interrupt handling during flash self-programming

This microcontroller provides functions to maintain interrupt servicing during the self-programming procedure.

Since neither the interrupt vector table nor the interrupt handler routines, which are normally located in the flash memory, are accessible while self-programming is active, interrupt acknowledges have to be re-routed to non-flash memory, i.e. to the internal RAM.

Therefore two prerequisites are necessary to enable interrupt servicing during self-programming:

- The concerned interrupt handler routine needs to be copied to the internal RAM, respectively external memory. The user has to initiate this copy process.
- The concerned interrupt acknowledge has to be re-routed to that handler. Re-routing to the handler is done by the internal firmware. Thus the user doesn't have to care about.

The internal firmware and the self-programming library provide functions to initialize and process such interrupts.

The interrupt handler routines can be copied from flash to the internal RAM by use of self-programming library functions.

The addresses of the interrupt handler routines are set up via the self-programming library as well.

Note
1. Note that this special interrupt handling adds some interrupt latency time.
2. Special interrupt handling is done only during the flash programming environment is activated. If self-programming is deactivated, the normal interrupt vector table in the flash memory is used.

All interrupt vectors are relocated to one entry point in the internal RAM:

- New entry point of *all* maskable interrupts is the 1st address of the internal RAM. A handler routine must check the interrupt source. The interrupt request source can be identified via the interrupt/exception source register ECR.EICC (refer to *"System register set" on page 159*)
- New entry point of *all* non maskable interrupts is the word address following the maskable interrupt entry, i.e. the second address of the internal RAM. The interrupt request source can be identified via the interrupt/exception source register ECR.FECC (refer to *"System register set" on page 159*).

In general a jump to a special handler routine will be placed at the 1st and 2nd internal RAM address, which identifies the interrupt sources and branches to the correct interrupt service routine.

The function serving the interrupt needs to be compiled as an interrupt function (i.e. terminate with a RETI instruction, save/restore all used registers, etc.).

It is recommended to refer to the application note "Self-Programming" (document nr. U16929EE) for comprehensive information concerning flash self-programming. This document explains also the functions of the self-programming library. The latest version of this document can be loaded via the URL

http://www.renesas.eu/updates

(3) Read with bus hold state and idle state insertion (bus size: 16 bits)

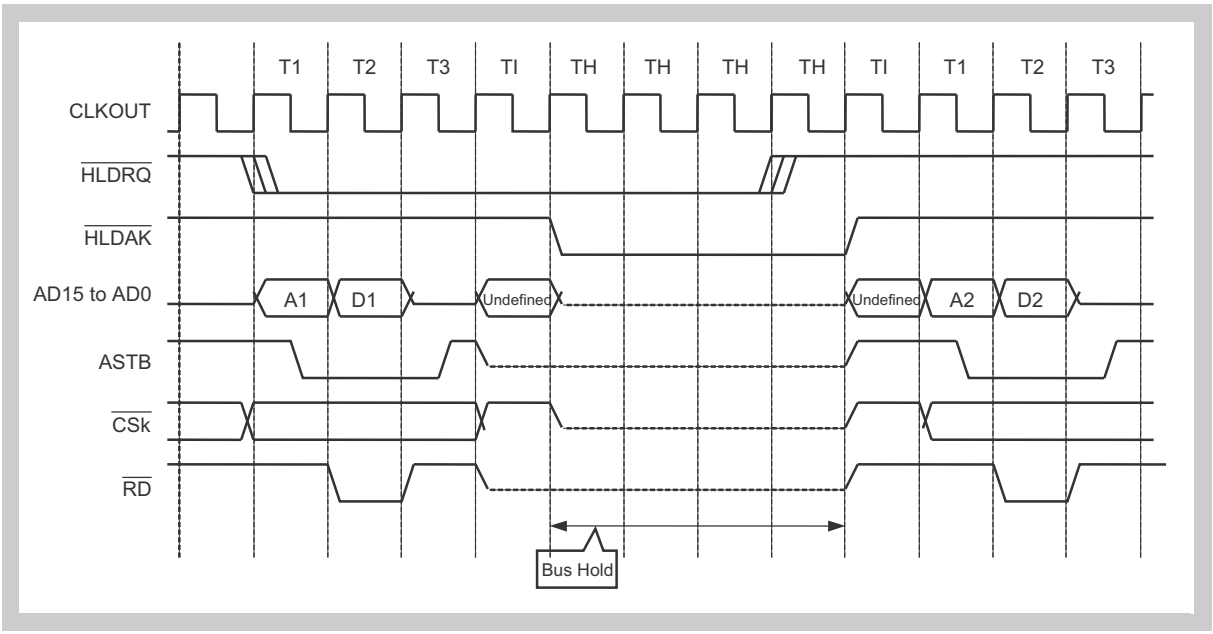

Figure 9-12 Timing: read data with bus hold state and idle state insertion (bus size: 16 bits)

Register settings:

- BSC.BSk0 = 1_B (16 bit data bus size)
- AWC.AHWk = AWC.ASWk = 0 (no address setup/hold wait states inserted)
- DWC0.DWk[2:0] = 001_B (one programmable data wait state inserted)
- BCC.BCk1 = 0_B (no idle state inserted, see also the Note below)

Note

1. The arrows indicate the sampling timing.
2. AD[7:0] holds the address for accessing the odd address byte
   AD[15:8] holds the address for accessing the even address byte
3. The idle state (TI) is independent of the setting of BCC.BCk1.
4. $\overline{CSk}$ with k = 0 to 3.

(3) Word access (32 bits)

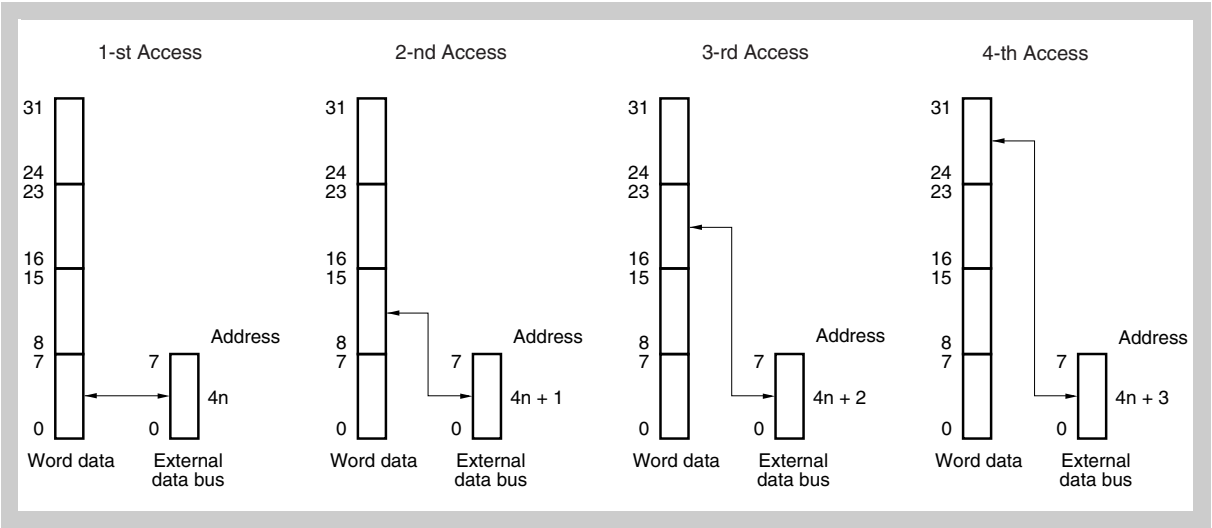

Figure 9-15 Access to address 4n

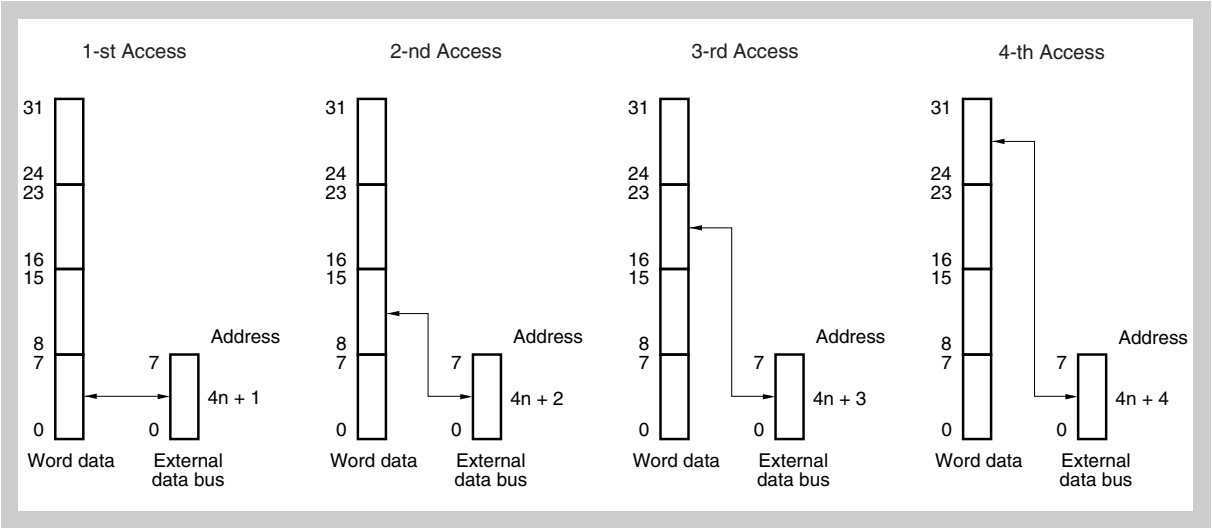

Figure 9-16 Access to address 4n + 1

(2) DDA0 to DDA3 - DMA destination address registers 0 to 3

The DDA0 to DDA3 registers set the DMA destination address (26 bits each) for DMA channel n (n = 0 to 3). These registers are divided into two 16-bit registers, DDAnH and DDAnL.

Access These registers can be read/written in 16-bit units.

Address

DDAL0:	FFFF F084$_H$	DDAH0:	FFFF F086$_H$
DDAL1:	FFFF F08C$_H$	DDAH1:	FFFF F08E$_H$
DDAL2:	FFFF F094$_H$	DDAH2:	FFFF F096$_H$
DDAL3:	FFFF F09C$_H$	DDAH3:	FFFF F09E$_H$

Initial Value undefined

DDAnH

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IR	0	0	0	0	0	DAn25	DAn24	DAn23	DAn22	DAn21	DAn20	DAn19	DAn18	DAn17	DAn16
R/W															

DDAnL

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DAn15	DAn14	DAn13	DAn12	DAn11	DAn10	DAn9	DAn8	DAn7	DAn6	DAn5	DAn4	DAn3	DAn2	DAn1	DAn0
R/W															

IR	Specification of DMA transfer source
0	External memory, on-chip peripheral or data flash
1	Internal RAM

DAn25 to DAn16	Set the address (A25 to A16) of the DMA transfer destination (default value is undefined). During DMA transfer the next DMA transfer destination address is held. When DMA transfer is completed the DMA address set first is held.

Dn15 to DAn0	Set the address (A15 to A0) of the DMA transfer destination (default value is undefined). During DMA transfer the next DMA transfer destination address is held. When DMA transfer is completed the DMA address set first is held.

Caution

1. Be sure to clear bits 14 to 10 of the DDAnH register to 0.
2. Set the DDAnH and DDAnL registers at the following timing when DMA transfer is disabled (DCHCn.Enn bit = 0).
   - Period from after reset to start of first DMA transfer
   - Period from after channel initialization by DCHCn.INITn bit to start of DMA transfer
   - Period from after completion of DMA transfer (DCHCn.TCn bit = 1) to start of the next DMA transfer
3. When the value of the DDAn register is read, two 16-bit registers, DDAnH and DDAnL, are read. If reading and updating conflict, a value being updated may be read (see *"Cautions" on page 395*).
4. Following reset, set the DSAnH, DSAnL, DDAnH, DDAnL, and DBCn registers before starting DMA transfer. If these registers are not set, the operation when DMA transfer is started is not guaranteed.

(2) Reload

When data is written to the TAAnCCR0 and TAAnCCR1 registers during timer operation, it is compared with the value of the 16-bit counter via the CCRm buffer register. The values of the TAAnCCR0 and TAAnCCR1 registers can be rewritten when TAAnCE = 1.

So that the set values of the TAAnCCR0 and TAAnCCR1 registers are compared with the value of the 16-bit counter (the set values are reloaded to the CCRm buffer register), the value of the TAAnCCR0 register must be rewritten and then a value must be written to the TAAnCCR1 register before the value of the 16-bit counter matches the value of TAAnCCR0. When the value of the TAAnCCR0 register matches the value of the 16-bit counter, the values of the TAAnCCR0 and TAAnCCR1 registers are reloaded.

Whether the next reload timing is made valid or not is controlled by writing to the TAAnCCR1 register. Therefore, write the same value to the TAAnCCR1 register when it is necessary to rewrite the value of only the TAAnCCR0 register.

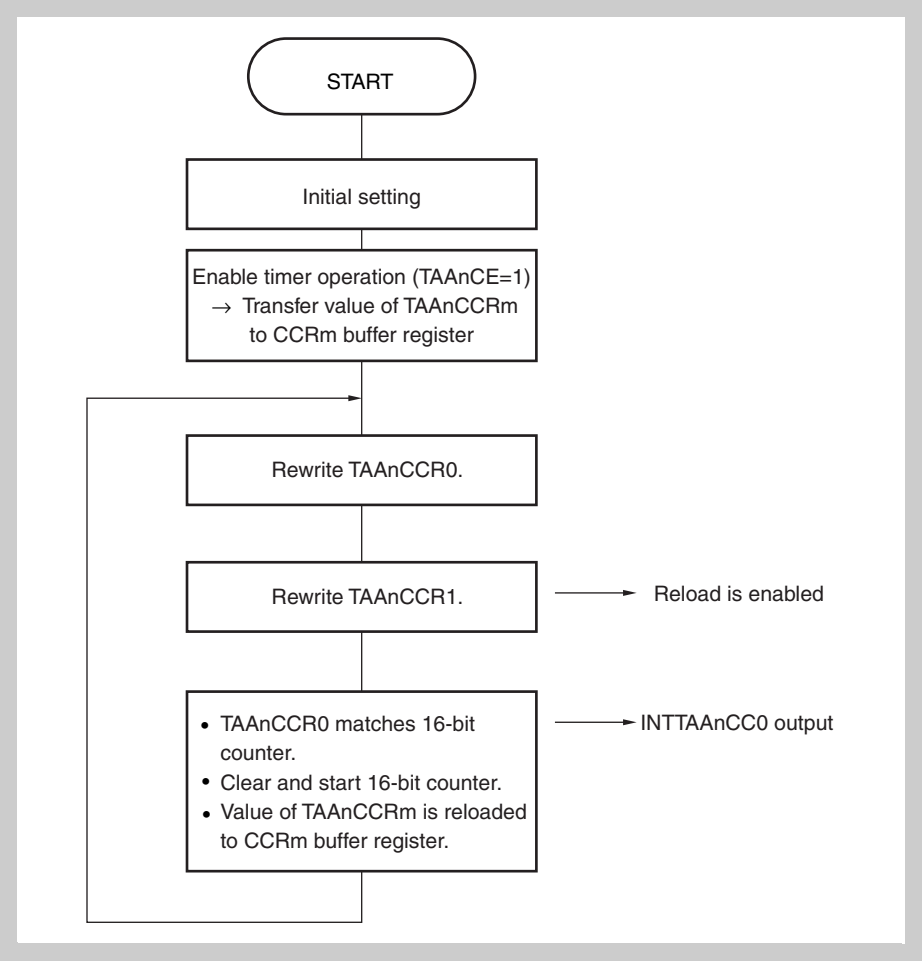

Figure 11-8 Flowchart of basic operation for reload

Caution Writing to the TAAnCCR1 register includes an operation to enable reload. Therefore, rewrite the TAAnCCR1 register after rewriting the TAAnCCR0 register.

Note 1. Above flowchart illustrates an example of the PWM mode operation.
2. m = 0, 1

12.5.6 PWM mode (TABnMD2 to TABnMD0 = 100)

In the PWM mode, TABn capture/compare register k (TABnCCRk) is used to set the duty factor and TABn capture/compare register 0 (TABnCCR0) is used to set the cycle.

By using these four registers and operating the timer, variable-duty PWM is output.

Rewriting the TABnCCRm register is enabled when TABnCE = 1.

So that the set value of the TABnCCRm register is compared with the value of the 16-bit counter (reloaded to the CCRm buffer register), a value must be written to the TABnCCR1 register before the value of the 16-bit counter matches the value of the TABnCCR0 register. The value of the TABnCCRm register is reloaded to the CCRm buffer registers when the value of the TABnCCR0 register later matches the value of the 16-bit counter.

Whether the next reload timing is made valid or not is controlled by writing to the TABnCCR1 register. Therefore, write the same value to the TABnCCR1 register even when only the value of the TABnCCR0 register needs to be rewritten. Reload is invalid when only the value of the TABnCCR0/TABnCCR2/TABnCCR3 register is rewritten.

To stop timer AB, clear TABnCE to 0.

The waveform of PWM is output from the TOABnk pin. The TOABn0 pin produces a toggle output when the 16-bit counter matches the TABnCCR0 register.

In the PWM mode, the TABnCCRm register is used only as a compare register. It cannot be used as a capture register.

Note m = 0 to 3; k = 1 to 3

(4) UDnOPT0 - UARTDn option control register 0

The UDnOPT0 register is an 8-bit register that controls the serial transfer operation of the UARTDn register.

Access This register can be read/written in 8-bit or 1-bit units.

Address

UD0OPT0: $FFFFFA03_H$	UD1OPT0: $FFFFFA13_H$
UD2OPT0: $FFFFFA23_H$	UD3OPT0: $FFFFFA33_H$
UD4OPT0: $FFFFFA43_H$	UD5OPT0: $FFFFFA53_H$
UD6OPT0: $FFFFFA63_H$	UD7OPT0: $FFFFFA73_H$

Initial Value 14_H. This register is initialized by any reset.

UDnOPT0

7	6	5	4	3	2	1	0
UDnSRF	UDnSRT	UDnSTT	UDnSLS2	UDnSLS1	UDnSLS0	UDnTDL	UDnRDL
R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 17-3 UDnOPT0 register contents (1/3)

Bit position	Bit name	Function
7	UDnSRF	SBF Reception Flag 0: When UDnCTL0.UDnPWR = 0 and UDnCTL0.UDnRXE = 0 are set to 1. Also upon normal end of SBF reception. 1: During SBF reception • SBF (Sync Brake Field) reception is judged during LIN communication. • The UDnSRF bit is held at 1 when an SBF reception error occurs, and then if the SBF reception is started again and ended normally, the UDnSRF bit is cleared to 0. Clearing by the instruction is disabled. • UDnSRF bit is read-only. When the UDnSRF = 1, the judgment process that SBF reception ended normally differs depending on the values of the SBF reception mode selection bit (UDnSRS). If the UDnSRS bit = 0, when any high level inputs including noises are applied to the reception input data even only for a second, the judgment of whether the low level period is more than 11 bits or not is executed. If the UDnSRS bit = 1, the received input data is sampled along with the set baud rate and when the low level period is 11 bits or more, it is judged as the successful SBF reception.

20.9 Message Reception

20.9.1 Message reception

In all the operation modes, the complete message buffer area is analyzed to find a suitable buffer to store a newly received message. All message buffers satisfying the following conditions are included in that evaluation (RX-search process).

- Used as a message buffer
  (MA0 bit of CnMCONFm register set to 1.)
- Set as a receive message buffer
  (MT[2:0] bits of CnMCONFm register are set to 001_B, 010_B, 011_B, 100_B, or 101_B.)
- Ready for reception
  (RDY bit of CnMCTRLm register is set to 1.)

When two or more message buffers of the CAN module receive a message, the message is stored according to the priority explained below. The message is always stored in the message buffer with the highest priority, not in a message buffer with a low priority. For example, when an unmasked receive message buffer and a receive message buffer linked to mask 1 have the same ID, the received message is not stored in the message buffer linked to mask 1, even if that message buffer has not received a message and a message has already been received in the unmasked receive message buffer. In other words, when a condition has been set in two or more message buffers with different priorities, the message buffer with the highest priority always stores the message; the message is not stored in message buffers with a lower priority. This also applies when the message buffer with the highest priority is unable to store a message (i.e., when DN = 1 indicating that a message has already been received, but rewriting is disabled because OWS = 0). In this case, the message is not actually stored in the candidate message buffer with the highest priority, but neither is it stored in a message buffer with a lower priority.

Table 20-32 MBRB priorities

Priority	Storing condition if same ID is set	
1 (high)	Unmasked message buffer	DN bit = 0
		DN bit = 1 and OWS bit = 1
2	Message buffer linked to mask 1	DN bit = 0
		DN bit = 1 and OWS bit = 1
3	Message buffer linked to mask 2	DN bit = 0
		DN bit = 1 and OWS bit = 1
4	Message buffer linked to mask 3	DN bit = 0
		DN bit = 1 and OWS bit = 1
5 (low)	Message buffer linked to mask 4	DN bit = 0
		DN bit = 1 and OWS bit = 1

Table A-6 Other special function registers (5/17)

Address	Register name	Shortcut	1	8	16	32
0xFFFFF1C0	Interrupt control register	C2ERRIC	R/W	R/W	-	-
0xFFFFF1C2	Interrupt control register	C2WUPIC	R/W	R/W	-	-
0xFFFFF1C4	Interrupt control register	C2RECIC	R/W	R/W	-	-
0xFFFFF1C6	Interrupt control register	C2TRXIC	R/W	R/W	-	-
0xFFFFF1C8	Interrupt control register	C3ERRIC	R/W	R/W	-	-
0xFFFFF1CA	Interrupt control register	C3WUPIC	R/W	R/W	-	-
0xFFFFF1CC	Interrupt control register	C3RECIC	R/W	R/W	-	-
0xFFFFF1CE	Interrupt control register	C3TRXIC	R/W	R/W	-	-
0xFFFFF1D0	Interrupt control register	PIC15	R/W	R/W	-	-
0xFFFFF1D2	Interrupt control register	TAA5OVIC	R/W	R/W	-	-
0xFFFFF1D4	Interrupt control register	TAA5CCIC0	R/W	R/W	-	-
0xFFFFF1D6	Interrupt control register	TAA5CCIC1	R/W	R/W	-	-
0xFFFFF1D8	Interrupt control register	TAA6OVIC	R/W	R/W	-	-
0xFFFFF1DA	Interrupt control register	TAA6CCIC0	R/W	R/W	-	-
0xFFFFF1DC	Interrupt control register	TAA6CCIC1	R/W	R/W	-	-
0xFFFFF1DE	Interrupt control register	TAA7OVIC	R/W	R/W	-	-
0xFFFFF1E0	Interrupt control register	TAA7CCIC0	R/W	R/W	-	-
0xFFFFF1E2	Interrupt control register	TAA7CCIC1	R/W	R/W	-	-
0xFFFFF1E4	Interrupt control register	UD6SIC	R/W	R/W	-	-
0xFFFFF1E6	Interrupt control register	CB3RIC	R/W	R/W	-	-
		UD6RIC	R/W	R/W	-	-
0xFFFFF1E8	Interrupt control register	CB3TIC	R/W	R/W	-	-
		UD6TIC	R/W	R/W	-	-
0xFFFFF1EA	Interrupt control register	UD7SIC	R/W	R/W	-	-
0xFFFFF1EC	Interrupt control register	UD7RIC	R/W	R/W	-	-
0xFFFFF1EE	Interrupt control register	UD7TIC	R/W	R/W	-	-
0xFFFFF1F0	Interrupt control register	AD1IC	R/W	R/W	-	-
0xFFFFF1F2	Interrupt control register	C4ERRIC	R/W	R/W	-	-
0xFFFFF1F4	Interrupt control register	C4WUPIC	R/W	R/W	-	-
0xFFFFF1F6	Interrupt control register	C4RECIC	R/W	R/W	-	-
0xFFFFF1F8	Interrupt control register	C4TRXIC	R/W	R/W	-	-
0xFFFFF1FA	In-service priority register	ISPR	R	R	-	-
0xFFFFF1FC	Command register	PRCMD	-	W	-	-
0xFFFFF1FE	Power save control register	PSC	R/W	R/W	-	-
0xFFFFF200	ADC0 mode register 0	ADA0M0	R/W	R/W	-	-
0xFFFFF201	ADC0 mode register 1	ADA0M1	R/W	R/W	-	-
0xFFFFF202	ADC0 channel specification register	ADA0S	R/W	R/W	-	-
0xFFFFF203	ADC0 mode register 2	ADA0M2	R/W	R/W	-	-
0xFFFFF204	ADC0 Power fail comparison mode register	ADA0PFM	R/W	R/W	-	-
0xFFFFF205	ADC0 Power fail comparison threshold value register	ADA0PFT	R/W	R/W	-	-
0xFFFFF20C	ADC0 conversion result register DD	ADA0CRDD	-	-	R	-

Table A-6 Other special function registers (11/17)

Address	Register name	Shortcut	1	8	16	32
0xFFFFF59A	TAA0 counter read buffer register	TAA0CNT	-	-	R	-
0xFFFFF59C	TAA0 I/O control register 4	TAA0IOC4	R/W	R/W	-	-
0xFFFFF5A0	TAA1 control register 0	TAA1CTL0	R/W	R/W	-	-
0xFFFFF5A1	TAA1 control register 1	TAA1CTL1	R/W	R/W	-	-
0xFFFFF5A2	TAA1 I/O control register 0	TAA1IOC0	R/W	R/W	-	-
0xFFFFF5A3	TAA1 I/O control register 1	TAA1IOC1	R/W	R/W	-	-
0xFFFFF5A4	TAA1 I/O control register 2	TAA1IOC2	R/W	R/W	-	-
0xFFFFF5A5	TAA1 option register 0	TAA1OPT0	R/W	R/W	-	-
0xFFFFF5A6	TAA1 capture/compare register 0	TAA1CCR0	-	-	R/W	-
0xFFFFF5A8	TAA1 capture/compare register 1	TAA1CCR1	-	-	R/W	-
0xFFFFF5AA	TAA1 counter read buffer register	TAA1CNT	-	-	R	-
0xFFFFF5AC	TAA1 I/O control register 4	TAA1IOC4	R/W	R/W	-	-
0xFFFFF5AD	TAA1 option register 1	TAA1OPT1	R/W	R/W	-	-
0xFFFFF5B0	TAA2 control register 0	TAA2CTL0	R/W	R/W	-	-
0xFFFFF5B1	TAA2 control register 1	TAA2CTL1	R/W	R/W	-	-
0xFFFFF5B2	TAA2 I/O control register 0	TAA2IOC0	R/W	R/W	-	-
0xFFFFF5B3	TAA2 I/O control register 1	TAA2IOC1	R/W	R/W	-	-
0xFFFFF5B4	TAA2 I/O control register 2	TAA2IOC2	R/W	R/W	-	-
0xFFFFF5B5	TAA2 option register 0	TAA2OPT0	R/W	R/W	-	-
0xFFFFF5B6	TAA2 capture/compare register 0	TAA2CCR0	-	-	R/W	-
0xFFFFF5B8	TAA2 capture/compare register 1	TAA2CCR1	-	-	R/W	-
0xFFFFF5BA	TAA2 counter read buffer register	TAA2CNT	-	-	R	-
0xFFFFF5BC	TAA2 I/O control register 4	TAA2IOC4	R/W	R/W	-	-
0xFFFFF5C0	TAA3 control register 0	TAA3CTL0	R/W	R/W	-	-
0xFFFFF5C1	TAA3 control register 1	TAA3CTL1	R/W	R/W	-	-
0xFFFFF5C2	TAA3 I/O control register 0	TAA3IOC0	R/W	R/W	-	-
0xFFFFF5C3	TAA3 I/O control register 1	TAA3IOC1	R/W	R/W	-	-
0xFFFFF5C4	TAA3 I/O control register 2	TAA3IOC2	R/W	R/W	-	-
0xFFFFF5C5	TAA3 option register 0	TAA3OPT0	R/W	R/W	-	-
0xFFFFF5C6	TAA3 capture/compare register 0	TAA3CCR0	-	-	R/W	-
0xFFFFF5C8	TAA3 capture/compare register 1	TAA3CCR1	-	-	R/W	-
0xFFFFF5CA	TAA3 counter read buffer register	TAA3CNT	-	-	R	-
0xFFFFF5CC	TAA3 I/O control register 4	TAA3IOC4	R/W	R/W	-	-
0xFFFFF5CD	TAA3 option register 1	TAA3OPT1	R/W	R/W	-	-
0xFFFFF5D0	TAA4 control register 0	TAA4CTL0	R/W	R/W	-	-
0xFFFFF5D1	TAA4 control register 1	TAA4CTL1	R/W	R/W	-	-
0xFFFFF5D2	TAA4 I/O control register 0	TAA4IOC0	R/W	R/W	-	-
0xFFFFF5D3	TAA4 I/O control register 1	TAA4IOC1	R/W	R/W	-	-
0xFFFFF5D4	TAA4 I/O control register 2	TAA4IOC2	R/W	R/W	-	-
0xFFFFF5D5	TAA4 option register 0	TAA4OPT0	R/W	R/W	-	-
0xFFFFF5D6	TAA4 capture/compare register 0	TAA4CCR0	-	-	R/W	-