Document No.: D-SLD-M40-0006-01

Internal Specification

F1K/SARAD113x model for M40PF

(V1.2)

Summary :

This document describes the Detail Specification of F1K/SARAD113x model for M40PF.

Reference Manuals				
No.	***Title name***	***Document Number***	**Description**	***Path***
1	REQ-SLD-15007_M40PF_Maintenance 2015Q2	REQ-SLD-15007	Required specifications for M40PF Maintenance 2015/Q2	Documents/1. General Documents/010_ENG/140_FrontEnd/Project/01_SLD/2_SLD_Project/Model_Documents/01_Project_Document_Management/REQ/2015
2	M40PF uciapadc113x and uciapadc071x outside specifications	-	SARAD113x HWM	uciapadc1130_IPspec_v0.04J.doc
3	Reset Modeling Guide	-	SH72AZ group MG microcomputer virtual PF Reset modeling guide ver0.6.	Documents/010_ENG/140_FrontEnd/Project/01_SLD/2_SLD_Project/RVC_documents/201109_M40PF/SH2A-MG_VP_RstGuide_Eng.pdf.
4	M40 Common Requirement	REQ-SLD-12010_M40PF_Common	The M40 platform common requirement	Documents/010_ENG/140_FrontEnd/Project/01_SLD/2_SLD_Project/Model_Documents/01_Project_Document_Management/REQ/2012/REQ-SLD-12010_M40PF_Common

Index of Figure 5
Index of Tables 6
1. Model summary 7
2. Supported features 7
3. Block diagram 9
4. List of implemented registers 12
5. List of implemented ports 21
6. SARAD113x triggers behavior 28
6.1. Behavior of HW trigger of SG1~SG3 and Temperature sensor 28
6.2. Behavior of HW trigger of PWM-Diag 28
6.3. Behavior of SW trigger 29
6.4. The trigger process of SARAD113x 29
6.5. Interrupt 30
6.6. ADOP interface output 31
6.7. Scanning timing chart 31
6.8. SARAD113x Reset behavior 33
7. Direction for users 35
7.1. File structures 35
7.2. Input/Output file 37
7.3. How to connect Verification Environment 37
7.4. handleCommand 37
7.5. Register RW messages style 39
7.6. Interrupt/error condition messages style 40
7.7. DumpStatInfo messages style 40
7.8. Help messages 41
7.9. Error and debugging messages 42
7.9.1. Error and debugging messages style 42
7.9.2. Error and debugging messages 43
7.10. Defined macro and template 49
8. Flow diagram 49
8.1. Sequence flow 52
8.2. State diagrams 53
8.3. preset_n input port flow 57
8.4. AssertReset function flow 58

8.5. CheckTrigger function processing flow 60
8.6. Start scanning processing flow 62
8.7. Check scanning group setting processing flow 64
8.8. Start virtual channel scanning processing flow....66
8.9. Virtual channel sampling processing flow 67
8.10. Virtual channel conversion processing flow....69
8.11. GetANIPortVal function processing flow....70
8.12. Finish virtual channel conversion processing flow 72
8.13. ADConvert function processing flow....73
8.14. Update conversion data processing flow 75
8.15. IsLastVC function processing flow....77
8.16. StopOperation function processing flow 78
8.17. Registers access condition....78
8.18. handleCommand operation processing flow....80
9. Class explanation 82
9.1. Class relationships....82
9.2. Class Csarad113x_fsm....82
9.3. Class Csarad113x_tha_conv_fsm 82
9.4. Class Csarad113x_thb_conv_fsm 82
9.5. Class Csarad113x 82
10. Restriction....91

Index of Figure

Figure 3.1: Block diagram of F1K/SARAD113x 9
Figure 6.1: Hardware Trigger behavior of Scanning group 1~3 and Temperature sensor 28
Figure 6.2: Hardware Trigger behavior of Scanning group 4 (PWM-Diag) 28
Figure 6.3: Software Trigger behavior of Scanning group n (n=0~4) 29
Figure 6.4: Trigger process of SARAD113x model 29
Figure 6.5: Interrupt output behavior of SARAD113x 30
Figure 6.6: ADOP interface output behavior of SARAD113x 31
Figure 6.7: Scanning timing chart of SARAD113x model 31
Figure 6.8: SARAD113x Reset behavior 33
Figure 7.1: File structures 35
Figure 8.1: Sequence diagram of SARAD113x model 52
Figure 8.2: State diagram of SARAD113x model 53
Figure 8.3: THz_CONV State diagram of SARAD113x model 55
Figure 8.4: preset_n input port flow 57
Figure 8.5: AssertReset function flow 58
Figure 8.6: CheckTrigger function processing flow 60
Figure 8.7: Start scanning processing flow 62
Figure 8.8: Check scanning group setting processing flow 64
Figure 8.9: Start virtual channel scanning processing flow 66
Figure 8.10: Virtual channel sampling processing flow 67
Figure 8.11: Virtual channel conversion processing flow 69
Figure 8.12: GetANIPortVal function processing flow 70
Figure 8.13: Finish virtual channel conversion processing flow 72
Figure 8.14: ADConvert function processing flow 73
Figure 8.15: Update conversion data processing flow 75
Figure 8.16: IsLastVC function processing flow 77
Figure 8.17: StopOperation function processing flow 78
Figure 8.18: handleCommand operation processing flow 80

Index of Tables

Table 2-1: Operation features of SARAD113x model 7
Table 4-1: List of implemented registers of SARAD113x model 12
Table 5-1: List of implemented ports of SARAD113x model 21
Table 6-1: Corresponding tasks at time marks 32
Table 7-1: File description 35
Table 7-2: List of parameters of handleCommand API 37
Table 7-3: List of commands of handleCommand API 39
Table 7-4: Dump Register RW message description 39
Table 7-5: Dump Interrupt/error condition message description 40
Table 7-6: State information message description 40
Table 7-7: Dump help command message description 41
Table 7-8: Dump tgt help command message description 42
Table 7-9: Dump reg help command message description 42
Table 7-10: Dump error message description 42
Table 7-11: Error message 43
Table 7-12: Error and debugging message of handleCommand 48
Table 7-13: Message level description 48
Table 8-1: Features and diagram reference table 49
Table 8-2: State transition table of SARAD113x model. 53
Table 8-3: THz_CONV State transition table of SARAD113x model. 55
Table 8-4: Self-diagnosis voltage level selection 71
Table 8-5: Register access condition 78
Table 9-1: List of implemented functions in Csarad113x class 82
Table 10-1: List of un-supported features of the SARAD113x model 91

1. Model summary

(1) SARAD113x is a module of the M40PF platform and SARAD113x stands for “Successive Approximation A/D converter”.

(2) SARAD113x consists of a module **ADC** which is 12-bit/10-bit successive approximation A/D converter and has up to 36 physical channels.

(3) SARAD113x is used to convert analog signals which are received from physical channels to digital signals.

(4) SARAD113x has following structures and functions:

(4.1) There are 5 scanning groups which can be set for scanning settings individually.

(4.2) There are up to 50 virtual channels (**VC**), each VC has a VC information register (DIRn).

(4.3) The converted data of 2 VC are stored in 1 data register (DRi). (*)

(4.4) SARAD113x can convert a maximum of 6 channels at the same time because it has “track-and-hold” circuits.

(5) SARAD113x is connected to APB (Advanced Peripheral Bus) to write/read the registers.

(6) In this design, the following features are supported :

(6.1) Support both the AT(Approximately time) and LT(Loosely time) coding style.

(6.2) **handleCommand** function with parameters such as **DumpRegisterRW**, **MessageLevel**, **DumpInterrupt, EnableConvertInfo,** etc ... to control how SARAD113x dumps the message during the operation of the SARAD113x model.

(6.3) User can specify the number of channels (physical and virtual), and the channel for external analog multiplexer will be used before the start of simulation.

(6.4) Support 32 bit bus-width socket.

Note: (*) Because converted data of 2 VCs are stored in 1 data register so that data of 2 VCs must be read out at the same time.

2. Supported features

Table 2-1: Operation features of SARAD113x model

Features	Support
1.16. Operation explanation	Yes
1.16.1. Initialization and fundamental motion	Yes
1.16.1.1. Initialization	Yes
1.16.1.2. A/D conversion beginning trigger	Yes
1.16.1.3. A/D conversion operation	Yes
1.16.2. Scanning group operation	Yes
1.16.2.1. Scanning mode of multi cycle	Yes

1.16.2.2. Continuous scanning mode	Yes
1.16.2.3. Channel repetition mode	Yes
1.16.3. Track & holding operation simultaneously	Yes
1.16.3.1. Simultaneous T&H usage restrictions	Yes
1.16.4. A/D conversion operation using external multiplexer	Yes
1.16.4.1. PWM-Diag function	Yes
1.16.4.2. A/D conversion in external multiplexer using VCRj(j=0~49)	Yes
1.16.5. Temperature sensor operation	No
1.16.5.1. Temperature sensor operation	No
1.16.5.2. Temperature sensor self-diagnosis function	No
1.16.6. Example of operating synchronous suspension & resume	Yes
1.16.7. Example of operating asynchronization suspension & resume	Yes
1.16.8. Error detecting function	Yes
1.16.8.1. Upper bound/lower bound error detecting function	Yes
1.16.8.2. Overwrite error detecting function	Yes
1.16.9. Scanning end flag	Yes
1.16.10. SVSTOP operation	No
1.16.11. Switch control signal	Yes
1.16.11.1. The switch control signal	Yes
1.16.11.2. Terminal state at the time of each mode	Yes
1.16.11.3. Control timing	Yes
1.16.12. Trigger input selection of the scan group	Yes
1.16.13. External trigger selection method	Yes
1.16.14. The scan group stop by ADHALT	Yes
1.16.15. The monitor function of the AD conversion monitor terminal	Yes
1.16.16. Scan end interrupt request	Yes
1.16.17. AD error interrupt request	Yes
1.16.18. Self-diagnosis function	Yes
1.16.18.1. Terminal level self-diagnosis function	Yes
1.16.18.2. T & H self-diagnosis function	Yes
1.16.18.3. AD conversion circuit self-diagnosis function	Yes
1.16.18.4. Disconnection detection self-diagnosis function	Yes
1.16.19. Intermittent operation function	No
1.16.20. Synchronization configuration	No
1.17. HM trim adjustment	No
1.18. The definition of the A/D conversion accuracy is shown as follows.	No

3. Block diagram

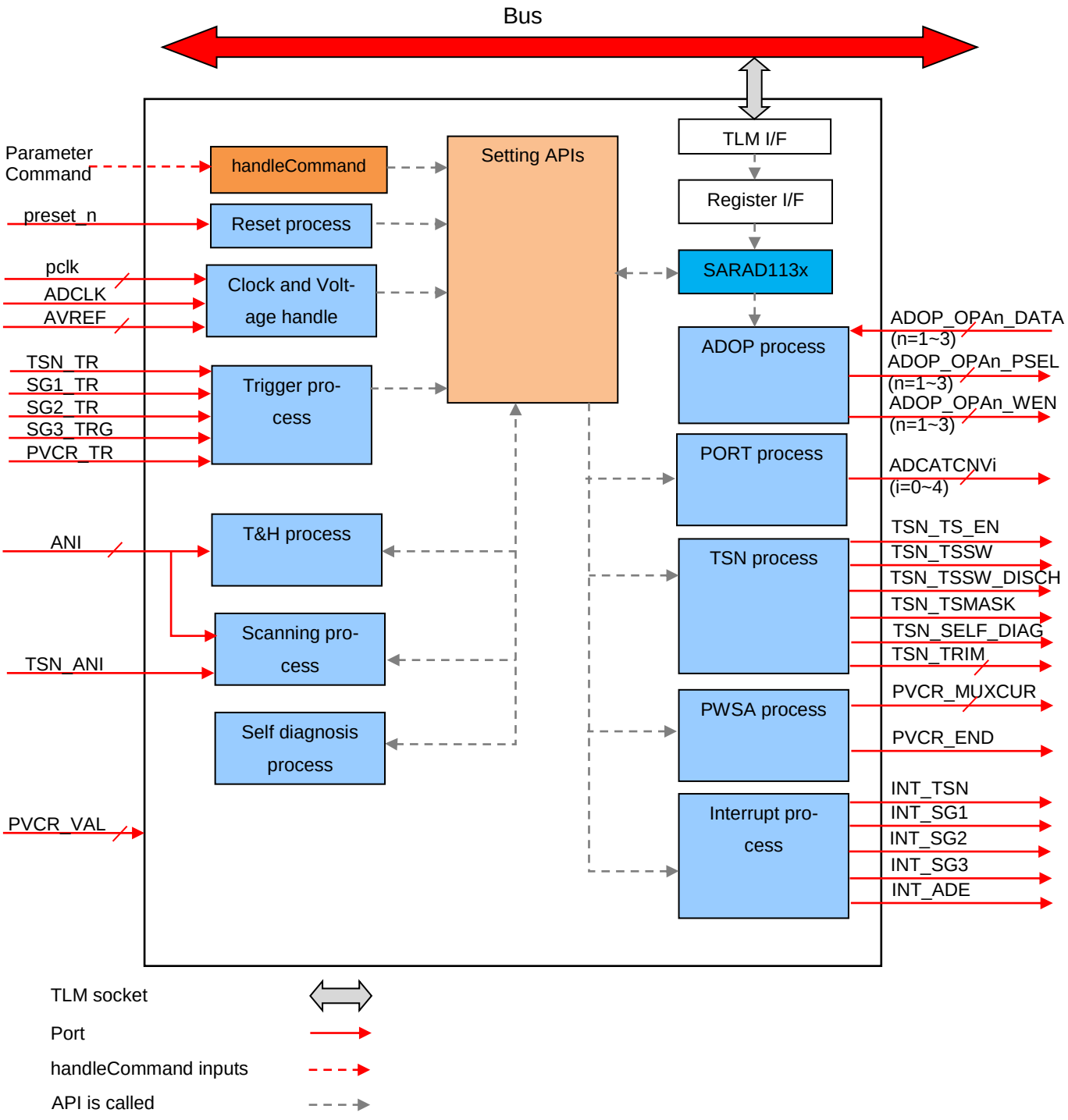

Figure 3.1: Block diagram of F1K/SARAD113x

Explanation:

(1) The Figure 3.1 shows the block diagram of SARAD113x model.

(2) Registers of SARAD113x model can be accessed through "*Register IF*" block. The register value can be written to or read from by TLM transaction.

(3) When the parameters or commands are set by users, the "*handleCommand*" block will transfer them into SARAD113x model. "*handleCommand*" is used for following functions :

(3.1) Setting output messages to dump the statistical information (**DumpRegisterRW, MessageLevel**, **Dumpinterrupt**).

(3.2) Setting enable/disable to dump AD convert activity (**EnableConvertInfo**).

(3.3) Setting frequencies of APB bus (pclk) and internal clock (ADCLK).

(3.4) Setting the reference voltages (**Avrefh, AVcc**).

(3.5) Setting the delay time from starting to hold an analog input until starting AD conversion (**EX_HLD_CDT**) in T&H circuit.

(3.6) Setting the delay time for successive approximation AD conversion (**EX_CNVT**).

(3.7) Setting the starting scanning group (**tD, tPWDD**) and completing scanning group (**tED**).

(3.8) Setting enable/disable to use parameter as delay timing for start or end scanning group (**EnableTimeCalculation**).

(3.9) Assert and de-assert reset function.

(4) The "*Trigger process"* will handle the input triggers. The input triggers will be synchronize with rising edge of ADCLK clock. The first rising edge of ADCLK clock is assumed at the time simulation starts.

(5) The *"T&H process"* will handle the Track and Hold process of SARAD113x. It consists of sampling, holding, holding complete process. There is 5 T&H physical channel (ANI00~ANI05).

(6) The *"Scanning process"* will handle the normal conversion of 5 scanning group (TSN, SG1, SG2, SG3 and PWD).

(7) The *"Self-diagnosis process"* will handle the self-diagnosis process of SARAD113x. There are 4 self-diagnosis function:

(7.1) Channel multiplexer self-diagnosis

(7.2) Channel Sample/Hold self-diagnosis

(7.3) Conversion circuit self-diagnosis

(7.4) ANI pin detection

(8) The *"ADOP process"* is used to control the output value of input ports for ADOP block which used for hardware trigger selection.

(9) The *"Port process"* is used to control the conversion monitor signals ADCATCNVi of each scanning group.

(10) The *"TSN process"* is used to control the input signal for Temperature sensor module when

scanning group for temperature is operating. In this model, the TSN feature is not supported. TSN feature is not supported in this version.

(11) The *"PWSA process"* is used to control the input signal for PWSA module when scanning group for PWD-Diag is operating.

(12) The *"Interrupt process"* is used to control the interrupt signal of SARAD113x after finish A/D conversion process.

Note:

The following symbols are used through this document:

x: 1 ~ 3 : The number of normal scanning group (SGx).
m: 0 ~ 35 : The number of physical channel.
i: 0 ~ 4 : The number of scanning group.
j: 0 ~ 49 : The number for virtual channel (VC).
y: 0 ~ 2 : The number of Upper Limit/Lower Limit Table Register (ULLMTBR)
z: A or B : The group name in T&H circuit.
k: 0 ~ 5 : The sampling channel number of T&H circuit.

4. List of implemented registers

Table 4-1: List of implemented registers of SARAD113x model

Register	Default value	Bit	Bit name	Description	Support Yes/No
VCRn (n=0~49)	0x00000000	[15]	MPXE	MPX is enable	Yes
		[14:12]	MPXV	MPX value	Yes
		[9]	CNVCLS	A/D conversion type selection bit at self-diagnosis	Yes
		[7:6]	ULS	Upper bound value/lower bound value table selection	Yes
		[8]	ADIE	The A/D conversion end interrupt is enabled.	Yes
		[5:0]	GCTRL	Physical channel selection	Yes
TSNVCR	0x00000025	[7:6]	ULS	Upper bound value/lower bound value table selection	No
		[5:0]	TSNGCTRL	Temperature sensor physical channel selection	No
PWDVCR	0x00000000	[15]	MPXE	MPX (PVCR_VALUE 11)	Yes
		[14:12]	MPXV	MPX value (PVCR_VALUE 10:8)	Yes
		[7:6]	ULS	Upper bound value/lower bound value table selection (PVCR_VALUE 7:6)	Yes
		[5:0]	GCTRL	Physical channel selection (PVCR_VALUE 5:0)	Yes
DRn (n=0~24)	0x00000000	[31:16]	DR(j+1)	A/D conversion result data register	Yes
		[15:0]	DR(j)	A/D conversion result data register	Yes
PWDTSNDR	0x00000000	[31:16]	PWDDR	PWM-Diag A/D conversion result data register	Yes
		[15:0]	TSNDR	A/D conversion result data register for temperature sensor	No
DIRn (n=0~49)	0x00000000	[31]	MPXE	MPX enable flag	Yes

Register	Default value	Bit	Bit name	Description	Support Yes/No
		[30:28]	MPX	MPX value	Yes
		[25]	WFLG	Write flag	Yes
		[21:16]	ID	Physical channel data	Yes
		[15:0]	DR(j)	A/D conversion result data	Yes
TSNDIR	0x00250000	[25]	WFLG	Write flag	No
		[21:16]	ID	Physical channel data	No
		[15:0]	TSNDR	A/D conversion result data	No
PWDDIR	0x00000000	[31]	MPXE	MPX enable flag	Yes
		[30:28]	MPX	MPX value	Yes
		[25]	WFLG	Write flag	Yes
		[21:16]	ID	Physical channel data	Yes
		[15:0]	PWDDR	A/D conversion result data	Yes
ADHALTR	0x00000000	[0]	HALT	Forced ending selection bit	Yes
ADCR	0x00000000	[7]	DGON	Self-diagnosis voltage standby control bit	Yes
		[6]	TSN-SELFDIAG	Temperature sensor (HM) self-diagnosis control bit	No
		[5]	CRAC	Alignment control bit	Yes
		[4]	CTYP	12/10bit mode selection bit	Yes
		[1:0]	SUSMTD	Suspension mode selection bit	Yes
SGSTR	0x00000000	[15:14]	SHACT	T&H status flag	Yes
		[13:8]	SGACT	Bit13 SGACT 5: SVSTOP status flag Bit12 SGACT 4: Scanning group (SG4) status flag for PWM-Diag Bit11 SGACT 3: Scanning group 3(SG3) status flag Bit10 SGACT 2: Scanning group 2(SG2) status flag Bit9 SGACT 1: Scanning group 1(SG1) status flag Bit8 SGACT 0: Scanning group (SG0) sta-	Yes

Register	Default value	Bit	Bit name	Description	Support Yes/No
				tus flag for temperature sensor	
		[4:0]	SEF	Bit3 SEF 3: SG3 scanning end flag Bit2 SEF 2: SG2 scanning end flag Bit1 SEF 1: SG1 scanning end flag Bit0 SEF 0: Scanning end flag for tempera-ture sensor	Yes
MPXCURR	0x00000000	[2:0]	MPXCUR	Current MPX value	Yes
TSNCR	0x00000000	[0]	TSNEN	Temperature sensor enable bit	No
THSMPSTCR	0x00000000	[0]	SMPST	T&H sampling beginning control bit	Yes
THCR	0x00000000	[0]	ASMPMSK		Yes
THAHLDSTCR	0x00000000	[0]	HLDST	T&H group A holding beginning control bit	Yes
THBHLDSTCR	0x00000000	[0]	HLDST	T&H group B holding beginning control bit	Yes
THACR	0x00000000	[5]	HLDCTE	T&H group B holding completion cockle navel bit	Yes
		[4]	HLDTE	T&H group B holding cockle navel bit	Yes
		[1:0]	SGS	T&H group B scanning group selection bit	Yes
THBCR	0x00000000	[5]	HLDCTE	T&H group B holding completion cockle navel bit	Yes
		[4]	HLDTE	T&H group B holding cockle navel bit	Yes
		[1:0]	SGS	T&H group B scanning group selection bit	Yes
THER	0x00000000	[5]	TH5E	T&H5 enable bit	Yes
		[4]	TH4E	T&H4 enable bit	Yes
		[3]	TH3E	T&H3 enable bit	Yes
		[2]	TH2E	T&H2 enable bit	Yes
		[1]	TH1E	T&H1 enable bit	Yes
		[0]	TH0E	T&H0 enable bit	Yes
THGSR	0x00000000	[5]	TH5GS	T&H5 group selection bit	Yes
		[4]	TH4GS	T&H4 group selection bit	Yes
		[3]	TH3GS	T&H3 group selection bit	Yes

Register	Default value	Bit	Bit name	Description	Support Yes/No
		[2]	TH2GS	T&H2 group selection bit	Yes
		[1]	TH1GS	T&H1 group selection bit	Yes
		[0]	TH0GS	T&H0 group selection bit	Yes
SFTCR	0x00000000	[4]	RDCLRE	Read clear & enable bit	Yes
		[3]	ULEIE	Upper bound lower bound error interrupt enable bit	Yes
		[2]	OWEIE	Overwrite error interrupt enable bit	Yes
ULLMTBRn (n=0~2)	0xFFF00000	[31:16]	ULMTB	Upper bound value table	Yes
		[15:0]	LLMTB	Lower bound value table	Yes
ECR	0x00000000	[3]	ULEC	Upper bound lower bound error flag enable bit	Yes
		[2]	OWEC	Overwrite error flag enable bit	Yes
		[15]	UE	Upper bound error flag	Yes
ULER	0x00000000	[14]	LE	Lower bound error flag	Yes
		[13:12]	ULSG	Scanning group when upper bound lower bound error occurs	Yes
		[11]	MPXE	MPX use existence	Yes
		[10:8]	MPXV	MPX value when upper bound lower bound error occurs	Yes
		[7]	ULE	Upper bound lower bound error flag	Yes
		[5:0]	ULECAP	Upper bound lower bound error capture bit	Yes
OWER	0x00000000	[7]	OWE	Overwrite error flag	Yes
		[5:0]	OWECAP	Overwrite error capture bit	Yes
DGCTL0	0x00000000	[2]	PSEL2	Self-diagnosis voltage level selection bit	Yes
		[1]	PSEL1	Self-diagnosis voltage level selection bit	Yes
		[0]	PSEL0	Self-diagnosis voltage level selection bit	Yes
DGCTL1	0x00000000	[15]	CDG15	Self-diagnosis channel selection bit	Yes
		[14]	CDG14	Self-diagnosis channel selection bit	Yes

Register	Default value	Bit	Bit name	Description	Support Yes/No
		[13]	CDG13	Self-diagnosis channel selection bit	Yes
		[12]	CDG12	Self-diagnosis channel selection bit	Yes
		[11]	CDG11	Self-diagnosis channel selection bit	Yes
		[10]	CDG10	Self-diagnosis channel selection bit	Yes
		[9]	CDG09	Self-diagnosis channel selection bit	Yes
		[8]	CDG08	Self-diagnosis channel selection bit	Yes
		[7]	CDG07	Self-diagnosis channel selection bit	Yes
		[6]	CDG06	Self-diagnosis channel selection bit	Yes
		[5]	CDG05	Self-diagnosis channel selection bit	Yes
		[4]	CDG04	Self-diagnosis channel selection bit	Yes
		[3]	CDG03	Self-diagnosis channel selection bit	Yes
		[2]	CDG02	Self-diagnosis channel selection bit	Yes
		[1]	CDG01	Self-diagnosis channel selection bit	Yes
		[0]	CDG00	Self-diagnosis channel selection bit	Yes
PDCTL1	0x00000000	[15]	PDNA15	Pull Down enable control bit	Yes
		[14]	PDNA14	Pull Down enable control bit	Yes
		[13]	PDNA13	Pull Down enable control bit	Yes
		[12]	PDNA12	Pull Down enable control bit	Yes
		[11]	PDNA11	Pull Down enable control bit	Yes
		[10]	PDNA10	Pull Down enable control bit	Yes
		[9]	PDNA09	Pull Down enable control bit	Yes
		[8]	PDNA08	Pull Down enable control bit	Yes
		[7]	PDNA07	Pull Down enable control bit	Yes
		[6]	PDNA06	Pull Down enable control bit	Yes
		[5]	PDNA05	Pull Down enable control bit	Yes
		[4]	PDNA04	Pull Down enable control bit	Yes
		[3]	PDNA03	Pull Down enable control bit	Yes

Register	Default value	Bit	Bit name	Description	Support Yes/No
		[2]	PDNA02	Pull Down enable control bit	Yes
		[1]	PDNA01	Pull Down enable control bit	Yes
		[0]	PDNA00	Pull Down enable control bit	Yes
PDCTL2	0x00000000	[19]	PDNA19	Pull Down enable control bit	Yes
		[18]	PDNA18	Pull Down enable control bit	Yes
		[17]	PDNA17	Pull Down enable control bit	Yes
		[16]	PDNA16	Pull Down enable control bit	Yes
		[15]	PDNA15	Pull Down enable control bit	Yes
		[14]	PDNA14	Pull Down enable control bit	Yes
		[13]	PDNA13	Pull Down enable control bit	Yes
		[12]	PDNA12	Pull Down enable control bit	Yes
		[11]	PDNA11	Pull Down enable control bit	Yes
		[10]	PDNA10	Pull Down enable control bit	Yes
		[9]	PDNA09	Pull Down enable control bit	Yes
		[8]	PDNA08	Pull Down enable control bit	Yes
		[7]	PDNA07	Pull Down enable control bit	Yes
		[6]	PDNA06	Pull Down enable control bit	Yes
		[5]	PDNA05	Pull Down enable control bit	Yes
		[4]	PDNA04	Pull Down enable control bit	Yes
		[3]	PDNA03	Pull Down enable control bit	Yes
		[2]	PDNA02	Pull Down enable control bit	Yes
		[1]	PDNA01	Pull Down enable control bit	Yes
		[0]	PDNA00	Pull Down enable control bit	Yes
SMPCR	0x00000018	[7:0]	SMPT	Sampling duration setting bit	Yes
TSNSMPCR	0x000000f0	[7:0]	TSNSMPT	Temperature sensor sampling duration setting bit	No
EMUCR	0x00000000	[7]	SVSDIS	SVSTOP disable bit	No
SGPRCR	0x00043210	[18:16]	SGPR4	SG priority level specification bit priority	Yes

Register	Default value	Bit	Bit name	Description	Support Yes/No
		[14:12]	SGPR3	SG priority level specification bit priority	Yes
		[10:8]	SGPR2	SG priority level specification bit priority	Yes
		[6:4]	SGPR1	SG priority level specification bit priority	Yes
		[2:0]	SGPR0	SG priority level specification bit priority	Yes
TRMCR	0x00000000	[31]	TRMS	Trim selection	No
		[27]	TRMDGSTBY	Trimming test control bit	No
		[15: 14]	TRMTSN-TUNE	Trim tuning for temperature sensor	No
		[13:12]	TRMTTUNE	Trim tuning for T&H	No
		[11:10]	TRMBTUNE	Trim tuning for buffer amplifier	No
		[9:8]	TRMATUNE	Trim tuning for SAR-ADC	No
		[7:6]	TRMTSN	Trim value for temperature sensor	No
		[5:4]	TRMT	Trim value for T&H	No
		[3:2]	TRMB	Trim value for buffer amplifier	No
		[1:0]	TRMA	Trim value for SAR-ADC	No
ADTSTRA	0x00000000	[7:0]	ADTST	Test mode selection bit for SAR-ADC	No
ADTSTRB	0x00000000	[13:0]	ADVAL	Arbitrary A/D conversion value specifica-tion bit	No
ADTSTRC	0x00000000	[13]	CKSTP	CLKAD clock stop control bit	No
		[12]	SYNCERR	Synchronous error	No
		[8]	ADMD8	Shunt width selection bit	No
		[7]	ADMD7	Comparator kickback selection bit	No
		[6]	ADMD6	Option of initial code stage none selection bit	No
		[5]	ADMD5	Subrange method invalid option selection bit	No
		[4]	ADMD4	C short SW	No
		[3]	ADMD3	Single-engined conversion selection bit	No

Register	Default value	Bit	Bit name	Description	Support Yes/No
		[2:0]	ADMD	Comparison method selection bit one by one	No
SGSTCRn (n=1~3)	0x00000000	[0]	SGSTn	SGn beginning trigger	Yes
TSNSGSTCR	0x00000000	[0]	TSNSGST	Beginning trigger for temperature sensor	No
PWDSGSTCR	0x00000000	[0]	PWDSGST	Beginning trigger for PWDSGST:PWM-Diag	Yes
SGCRn (n=1~3)	0x00000000	[5]	SCANMD	Scanning mode selection bit	Yes
		[4]	ADIE	Scanning group n interrupt is enabled.	Yes
		[3:2]	SCT	Channel repetition frequency selection bit	Yes
		[0]	TRGMD	Trigger mode selection bit	Yes
TSNSGCR	0x00000000	[0]	TSNTRGMD	Temperature sensor trigger mode selection bit	No
PWDSGCR	0x00000000	[0]	PWD-TRGMD	Trigger mode selection bit	Yes
SGSEF-CRn(n=1~3)	0x00000000	[0]	SEFCn	Scanning end flag clear trigger	Yes
TSNSGSEFCR	0x00000000	[0]	TSNSEFC	Scanning end flag clear trigger for tem-perature sensor	Yes
PWDSGSEFCR	0x00000000	[0]	PWDSEFC	Scanning end flag clear trigger for PWDSEFC:PWM-Diag	Yes
SGVCSPn (n=1~3)	0x00000000	[5:0]	VCSP	Scanning group n beginning channel point-er	Yes
SGVCEPn (n=1~3)	0x00000000	[5:0]	VCEP	Scanning group n end channel pointer	Yes
SGMCYCRn (n=1~3)	0x00000000	[7:0]	MCYC	The number of specification bit of multi cy-cle	Yes
SGTSEL (n=1~3)	0x00000000	[15]	TxSEL15	A/D conversion trigger (HW trigger) selec-tion	Yes

Register	Default value	Bit	Bit name	Description	Support Yes/No
		[14]	TxSEL14	A/D conversion trigger (HW trigger) selection	Yes
		[13]	TxSEL13	A/D conversion trigger (HW trigger) selection	Yes
		[12]	TxSEL12	A/D conversion trigger (HW trigger) selection	Yes
		[11]	TxSEL11	A/D conversion trigger (HW trigger) selection	Yes
		[10]	TxSEL10	A/D conversion trigger (HW trigger) selection	Yes
		[9]	TxSEL09	A/D conversion trigger (HW trigger) selection	Yes
		[8]	TxSEL08	A/D conversion trigger (HW trigger) selection	Yes
		[7]	TxSEL07	A/D conversion trigger (HW trigger) selection	Yes
		[6]	TxSEL06	A/D conversion trigger (HW trigger) selection	Yes
		[5]	TxSEL05	A/D conversion trigger (HW trigger) selection	Yes
		[4]	TxSEL04	A/D conversion trigger (HW trigger) selection	Yes
		[3]	TxSEL03	A/D conversion trigger (HW trigger) selection	Yes
		[2]	TxSEL02	A/D conversion trigger (HW trigger) selection	Yes
		[1]	TxSEL01	A/D conversion trigger (HW trigger) selection	Yes
		[0]	TxSEL00	A/D conversion trigger (HW trigger) selection	Yes

Note : “No” - Not supported but can be accessed via register I/F.

5. List of implemented ports

Table 5-1: List of implemented ports of SARAD113x model

HW port name	Model Port name	I/O	Type	Initial	Active	Description	Support
PCLK	pclk	IN	sc_dt::uint64	-	-	- Specify APB clock frequency (Hz) (Internal clock, Clean clock)	Yes
CLKAD	ADCLK	IN	sc_dt::uint64	-	-	- Specify AD clock frequency (Hz) (Internal clock, Clean clock)	Yes
PRESETn	preset_n	IN	bool	-	low	Asynchronous hardware reset (active low).	Yes
PSEL, PADDR, PWRITE, PENABLE, PSTRB, PWDATA, PRDATA, PREADY	m_tgt_sockets[0]	I/O	tlm:tlm_target_socket	-	-	TLM target socket (defined in tlm_tgt_if.h).	Yes
INT_TSN	INT_TSN	OUT	bool	false	high	- A/D conversion end interrupt output of Temperature Sensor - Interrupt protocol is 1 pulse of PCLK at rising edge of PCLK.	No*
INT_SG1	INT_SG1	OUT	bool	false	high	- A/D conversion end interrupt output of scanning group 1 (SG1) - Interrupt protocol is 1 pulse of PCLK at rising edge of PCLK.	Yes
INT_SG2	INT_SG2	OUT	bool	false	high	- A/D conversion end interrupt output of scanning group 2 (SG2) - Interrupt protocol is 1 pulse of PCLK at rising edge of PCLK.	Yes
INT_SG3	INT_SG3	OUT	bool	false	high	- A/D conversion end interrupt output of scanning group 3 (SG3) - Interrupt protocol is 1 pulse of PCLK at rising edge of PCLK.	Yes
PVCR_END	PVCR_END	OUT	bool	false	high	- A/D conversion end interrupt output of PWM-Diag - Interrupt protocol is 1 pulse of PCLK at rising edge of PCLK.	Yes
INT_ADE	INT_ADE	OUT	bool	false	high	- Analog to digital interrupt output. - Interrupt protocol is 1 pulse of PCLK at rising edge of PCLK.	Yes

TSN_TRG	TSN_TRG	IN	bool	-	high	- Trigger signal for temperature sensor	No*
SG1_TRG	SG1_TRG	IN	bool	-	high	- Trigger for scanning group 1 (SG1)	Yes
SG2_TRG	SG2_TRG	IN	bool	-	high	- Trigger for scanning group 2 (SG2)	Yes
SG3_TRG	SG3_TRG	IN	bool	-	high	- Trigger for scanning group 3 (SG3)	Yes
PVCR_TRG	PVCR_TRG	IN	bool	-	high	- Trigger Signal for PWM-Diag	Yes
AVREFH	AVREFH	IN	double	-	-	- Reference voltage of analog input port	Yes
ADCATCNV0	ADCATCNV0	OUT	bool	false	high	- Conversion Monitor Signal of Temperature Sensor	No*
ADCATCNV1	ADCATCNV1	OUT	bool	false	high	- Conversion Monitor Signal of Scanning group 1 (SG1)	Yes
ADCATCNV2	ADCATCNV2	OUT	bool	false	high	- Conversion Monitor Signal of Scanning group 2 (SG2)	Yes
ADCATCNV3	ADCATCNV3	OUT	bool	false	high	- Conversion Monitor Signal of Scanning group 3 (SG3)	Yes
ADCATCNV4	ADCATCNV4	OUT	bool	false	high	- Conversion Monitor Signal of PWM-Diag	Yes
ANI00	ANI00	IN	double	-	-	- Analog physical channel 0	Yes
ANI01	ANI01	IN	double	-	-	- Analog physical channel 1	Yes
ANI02	ANI02	IN	double	-	-	- Analog physical channel 2	Yes
ANI03	ANI03	IN	double	-	-	- Analog physical channel 3	Yes
ANI04	ANI04	IN	double	-	-	- Analog physical channel 4	Yes
ANI05	ANI05	IN	double	-	-	- Analog physical channel 5	Yes
ANI06	ANI06	IN	double	-	-	- Analog physical channel 6	Yes
ANI07	ANI07	IN	double	-	-	- Analog physical channel 7	Yes
ANI08	ANI08	IN	double	-	-	- Analog physical channel 8	Yes
ANI09	ANI09	IN	double	-	-	- Analog physical channel 9	Yes
ANI10	ANI10	IN	double	-	-	- Analog physical channel 10	Yes
ANI11	ANI11	IN	double	-	-	- Analog physical channel 11	Yes
ANI12	ANI12	IN	double	-	-	- Analog physical channel 12	Yes
ANI13	ANI13	IN	double	-	-	- Analog physical channel 13	Yes
ANI14	ANI14	IN	double	-	-	- Analog physical channel 14	Yes

ANI15	ANI15	IN	double	-	-	- Analog physical channel 15	Yes
ANI16	ANI16	IN	double	-	-	- Analog physical channel 16	Yes
ANI17	ANI17	IN	double	-	-	- Analog physical channel 17	Yes
ANI18	ANI18	IN	double	-	-	- Analog physical channel 18	Yes
ANI19	ANI19	IN	double	-	-	- Analog physical channel 19	Yes
ANI20	ANI20	IN	double	-	-	- Analog physical channel 20	Yes
ANI21	ANI21	IN	double	-	-	- Analog physical channel 21	Yes
ANI22	ANI22	IN	double	-	-	- Analog physical channel 22	Yes
ANI23	ANI23	IN	double	-	-	- Analog physical channel 23	Yes
ANI24	ANI24	IN	double	-	-	- Analog physical channel 24	Yes
ANI25	ANI25	IN	double	-	-	- Analog physical channel 25	Yes
ANI26	ANI26	IN	double	-	-	- Analog physical channel 26	Yes
ANI27	ANI27	IN	double	-	-	- Analog physical channel 27	Yes
ANI28	ANI28	IN	double	-	-	- Analog physical channel 28	Yes
ANI29	ANI29	IN	double	-	-	- Analog physical channel 29	Yes
ANI30	ANI30	IN	double	-	-	- Analog physical channel 30	Yes
ANI31	ANI31	IN	double	-	-	- Analog physical channel 31	Yes
ANI32	ANI32	IN	double	-	-	- Analog physical channel 32	Yes
ANI33	ANI33	IN	double	-	-	- Analog physical channel 33	Yes
ANI34	ANI34	IN	double	-	-	- Analog physical channel 34	Yes
ANI35	ANI35	IN	double	-	-	- Analog physical channel 35	Yes
-	TSN_ANI	IN	double	-	-	- Analog physical channel 37 (Dedicated port for Temperature sensor)	No*
ULE	ULE	OUT	bool	false	high	- Upper/lower bound error interrupt - Interrupt protocol is level	Yes
PVCR_VALUE	PVCR_VALUE	IN	sc_uint<12>	-	-	- Virtual Channel Data Signal for PWM-Diag	Yes
PVCR_MUXCUR	PVCR_MUXCUR	OUT	sc_uint<3>	0	-	- MPX Current Value for PWM-Diag(true)	Yes
TSN_TS_EN	TSN_TS_EN	OUT	bool	false	high	- Enable Signal of Temperature Sensor	No*
TSN_TSSW	TSN_TSSW	OUT	bool	false	high	- Channel Select Signal for Temperature Sensor(true)	No*

TSN_TSSW_DISCH	TSN_TSSW_DISCH	OUT	bool	false	high	- Enable signal for discharge of temperature sensor's SW	No*
TSN_TSMASK	TSN_TSMASK	OUT	bool	false	high	- Mask Control of Temperature Sensor	No*
TSN_SELF_DIAG	TSN_SELF_DIAG	OUT	bool	false	high	- Self-Diag Control Signal of Temperature Sensor	No*
TSN_TRIM	TSN_TRIM	OUT	sc_uint<2>	0	-	- Bias Trimming of Temperature Sensor	No*
ADOP_OPA1_DATA	ADOP_OPA1_DATA	IN	sc_uint<16>	-	-	- ADOP1 Data Signal	Yes
ADOP_OPA2_DATA	ADOP_OPA2_DATA	IN	sc_uint<16>	-	-	- ADOP2 Data Signal	Yes
ADOP_OPA3_DATA	ADOP_OPA3_DATA	IN	sc_uint<16>	-	-	- ADOP3 Data Signal	Yes
ADOP_OPA1_PSEL	ADOP_OPA1_PSEL	OUT	bool	false	high	- Trigger Priority Select Signal for ADOP1	Yes
ADOP_OPA1_WEN	ADOP_OPA1_WEN	OUT	bool	false	high	- Write Access Signal for ADOP1	Yes
ADOP_OPA2_PSEL	ADOP_OPA2_PSEL	OUT	bool	false	high	- Trigger Priority Select Signal for ADOP2	Yes
ADOP_OPA2_WEN	ADOP_OPA2_WEN	OUT	bool	false	high	- Write Access Signal for ADOP2	Yes
ADOP_OPA3_PSEL	ADOP_OPA3_PSEL	OUT	bool	false	high	- Trigger Priority Select Signal for ADOP3	Yes
ADOP_OPA3_WEN	ADOP_OPA3_WEN	OUT	bool	false	high	- Write Access Signal for ADOP3	Yes
scan_mode	-	IN	bool	-	-	SCAN Mode Signal	No
scan_enable	-	IN	bool	-	-	SCAN Enable Signal	No
test_mode	-	IN	bool	-	-	Test Mode Signal	No
MSTPP_N	-	IN	bool	-	-	Module stop signal for PCLK synchronization	No
MSTPAD_N	-	IN	bool	-	-	Module stop signal for CLKAD synchronization	No
iddq_mode	-	IN	bool	-	-	IDDQ Test Mode Signal	No
sv_mode	-	IN	bool	-	-	Super Visor Mode Signal	No
dt_mode	-	IN	bool	-	-	Direct test mode signal for AD(HM)	No
dt_trg	-	IN	bool	-	-	Direct test conversion trigger	No

RESETAD_N	-	IN	bool	-	-	RESET for ADCLK	No
EX_CNVT	-	IN	sc_uint<6>	-	-	Conversion Count Signal	No
EX_BUFAMP_OFFT	-	IN	sc_uint<8>	-	-	Buffer Amp Count Signal	No
EX_HLD_CDT	-	IN	sc_uint<5>	-	-	Holding t trigger control signal when T&H is executed	No
ADCTRM	-	IN	sc_uint<8>	-	-	AD(HM)(ADcore,RRAMP,T&H,TSN) Trimming Signal	No
ADOUT	-	IN	sc_uint<12>	-	-	AD Core Conversion result	No
ADFLAG3	-	IN	bool	-	-	AD Core Status(Flag3) Signal	No
ADCLK	-	OUT	bool	-	-	AD Core Clock	No
ADRST_N	-	OUT	bool	-	-	AD Core Reset	No
ADSLP_N	-	OUT	bool	-	-	AD Core Sleep Mode Signal	No
dt_adout	-	OUT	sc_uint<12>	-	-	AD core conversion result at direct test	No
dt_adflag3	-	OUT	bool	-	-	AD core status(Flag3) signal at direct test	No
ADMODE	-	OUT	sc_uint<9>	-	-	AD Core Conversion Mode Signal	No
ADST	-	OUT	bool	-	-	AD Core Conversion Start Signal	No
ADNSMP	-	OUT	sc_uint<8>	-	-	AD Core Conversion Number of Sampling	No
ADTEST	-	OUT	sc_uint<4>	-	-	AD Core Test Mode Signal	No
ADVAL	-	OUT	sc_uint<14>	-	-	AD Core Value Data Signal for Test	No
TRMA	-	OUT	sc_uint<2>	-	-	AD Core Bias Trimming for TEST	No
OPON_T	-	OUT	bool	-	-	RRAMP ON Signal(true)	No
OPON_B	-	OUT	bool	-	-	RRAMP ON Signal(bar)	No
TRMB_T	-	OUT	sc_uint<2>	-	-	RRAMP Bias Trimming(true)	No
TRMB_B	-	OUT	sc_uint<2>	-	-	RRAMP Bias Trimming(bar)	No
SHONSL_T	-	OUT	sc_uint<6>	-	-	T&H Standby Control Signal(true)	No
SHONSL_B	-	OUT	sc_uint<6>	-	-	T&H Standby Control Signal(bar)	No
SHSPON1_T	-	OUT	sc_uint<6>	-	-	TH SHAMP sampling control OPamp I/O short and standard voltage output (true)	No

SHSPON1_B	-	OUT	sc_uint<6>	-	-	TH SHAMP sampling control OPamp I/O short and standard voltage output (bar)	No
SHSPON2_T	-	OUT	sc_uint<6>	-	-	TH SHAMP sampling control analog input and standard voltage generation (true)	No
SHSPON2_B	-	OUT	sc_uint<6>	-	-	TH SHAMP sampling control analog input and standard voltage generation (bar)	No
SHHDON_T	-	OUT	sc_uint<6>	-	-	T&H Sampling Hold Amp Hold Control Signal (true)	No
SHHDON_B	-	OUT	sc_uint<6>	-	-	T&H Sampling Hold Amp Hold Control Signal (bar)	No
SHCHSL_T	-	OUT	sc_uint<6>	-	-	T&H Sampling Hold Amp Output Control Signal (true)	No
SHCHSL_B	-	OUT	sc_uint<6>	-	-	T&H Sampling Hold Amp Output Control Signal (bar)	No
TRMT_T	-	OUT	sc_uint<2>	-	-	T&H Bias Trimming(true)	No
TRMT_B	-	OUT	sc_uint<2>	-	-	T&H Baias Triming(bar)	No
DGON_T	-	OUT	bool	-	-	Self-Diag level ON Switch for Self-Diag(true)	No
DGON_B	-	OUT	bool	-	-	Self-Diag level ON Switch for Self-Diag(bar)	No
DGSBY_T	-	OUT	bool	-	-	Standby Switch for Self-Diag(true)	No
DGSBY_B	-	OUT	bool	-	-	Standby Switch for Self-Diag(bar)	No
DGSELAD_T	-	OUT	sc_uint<3>	-	-	Self-Diag Level Select Signal for Self-Diag(true)	No
DGSELAD_B	-	OUT	sc_uint<3>	-	-	Self-Diag level Select Signal for Self-Diag(bar)	No
DGSELSH_T	-	OUT	sc_uint<4>	-	-	Self-Diag level Select Signal of SH ON for Self-Diag(true)	No
DGSELSH_B	-	OUT	sc_uint<4>	-	-	Self-Diag level Select Signal of SH ON for Self-Diag(bar)	No
CHSLC_T	-	OUT	sc_uint<36>	-	-	Channel Select Signal for ANI(true)	No
CHSLC_B	-	OUT	sc_uint<36>	-	-	Channel Select Signal for ANI(bar)	No
CHSLC_DIAG_T	-	OUT	bool	-	-	Channel Select Signal for ANI_SW(DIAG) (true)	No

CHSLC_DIAG_B	-	OUT	bool	-	-	Channel Select Signal for ANI_SW(DIAG) (bar)	No
CHSLC_DIGI_T	-	OUT	bool	-	-	Channel Select Signal for ANI_SW(DIGI) (true)	No
CHSLC_DIGI_B	-	OUT	bool	-	-	Channel Select Signal for ANI_SW(DIGI) (bar)	No
TRON_T	-	OUT	bool	-	-	Transistor Switch On Signal(true)	No
TRON_B	-	OUT	bool	-	-	Transistor Switch On Signal(bar)	No
TRON_OP_T	-	OUT	bool	-	-	Transistor Switch On Signal for Buffer Amp(true)	No
TRON_OP_B	-	OUT	bool	-	-	Transistor Switch On Signal for Buffer Amp(bar)	No
TRON_TH_T	-	OUT	sc_uint<6>	-	-	Transistor Switch On Signal for T&H(true)	No
TRON_TH_B	-	OUT	sc_uint<6>	-	-	Transistor Switch On Signal for T&H(bar)	No
CHDIAG_T	-	OUT	sc_uint<16>	-	-	Self-Diag Channel Select Signal(true)	No
CHDIAG_B	-	OUT	sc_uint<16>	-	-	Self-Diag Channel Select Signal(bar)	No
ANPDON1_T	-	OUT	sc_uint<16>	-	-	ANI Pull Down Signal(true)	No
ANPDON1_B	-	OUT	sc_uint<16>	-	-	ANI Pull Down Signal(bar)	No
ANPDON2_T	-	OUT	sc_uint<20>	-	-	ANI Pull Down Signal(true)	No
ANPDON2_B	-	OUT	sc_uint<20>	-	-	ANI Pull Down Signal(bar)	No

Note: (*) These ports are existed but the TSN feature is not supported so that they have no effect on operation.

6. SARAD113x triggers behavior

6.1. Behavior of HW trigger of SG1~SG3 and Temperature sensor

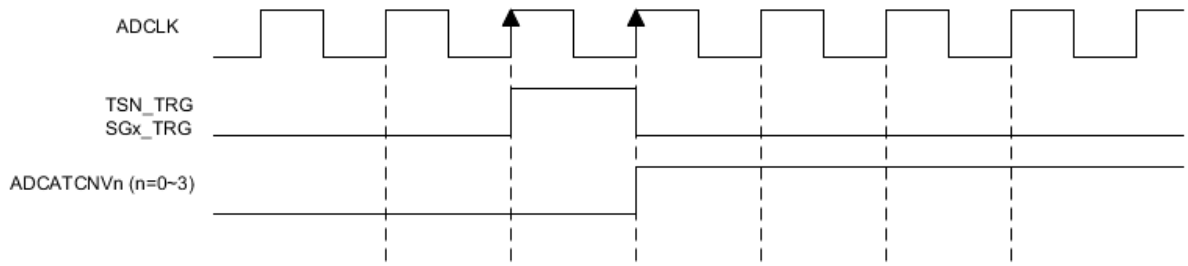

Figure 6.1: Hardware Trigger behavior of Scanning group 1~3 and Temperature sensor

Explanation:

(1) Refer to HWM[2] chapter: 1.16.1.2, 1.16.1.3, 1.16.15.
(2) The Figure 6.1 shows the behavior of TSN_TRG and SGx_TRG. When HW trigger of Temperature sensor (Scanning group 0) or Scanning group 1~3 is asserted, after 1 ADCLK period, the corresponding ADCATCNVn is asserted and the scanning process is started.

6.2. Behavior of HW trigger of PWM-Diag

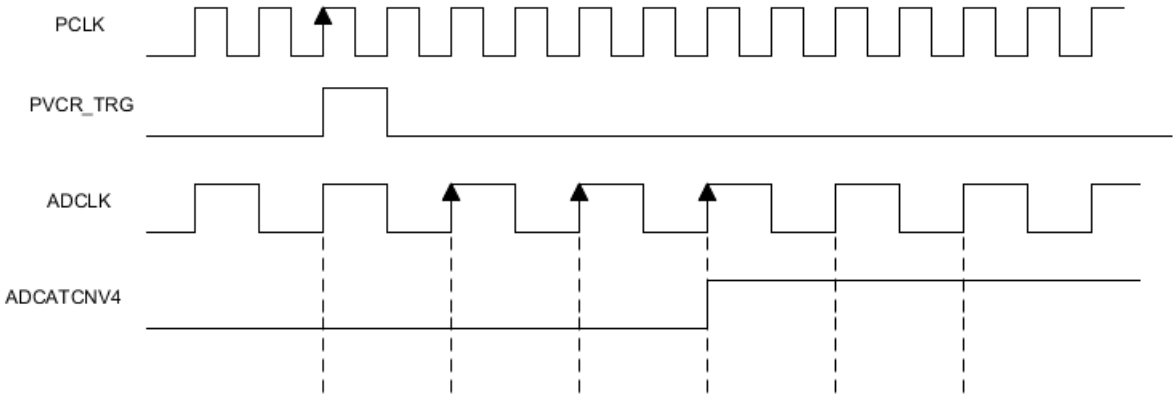

Figure 6.2: Hardware Trigger behavior of Scanning group 4 (PWM-Diag)

Explanation:

(1) Refer to HWM[2] chapter: 1.16.1.2, 1.16.15.
(2) The Figure 6.2 shows the behavior of PVCR_TRG. When HW trigger of PWD-Diag is asserted, after 1 PCLK period and 2 ADCLK (from next rising edge of ADCLK), the corresponding ADCATCNV4 is asserted and the scanning process is started.

6.3. Behavior of SW trigger

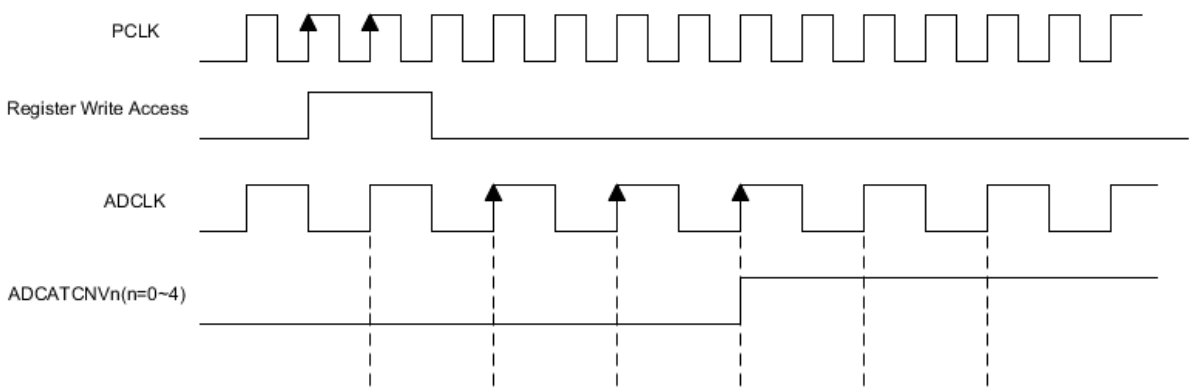

Figure 6.3: Software Trigger behavior of Scanning group n (n=0~4)

Explanation:

(1) Refer to HWM[2] chapter: 1.16.1.2, 1.16.15.
(2) The Figure 6.3 shows the behavior of SW trigger. When SW trigger of is asserted by writing to register, after 1 PCLK period and 2 ADCLK (from next rising edge of ADCLK), the corresponding ADCATCNVn (n=0~4) is asserted and the scanning process is started.

6.4. The trigger process of SARAD113x

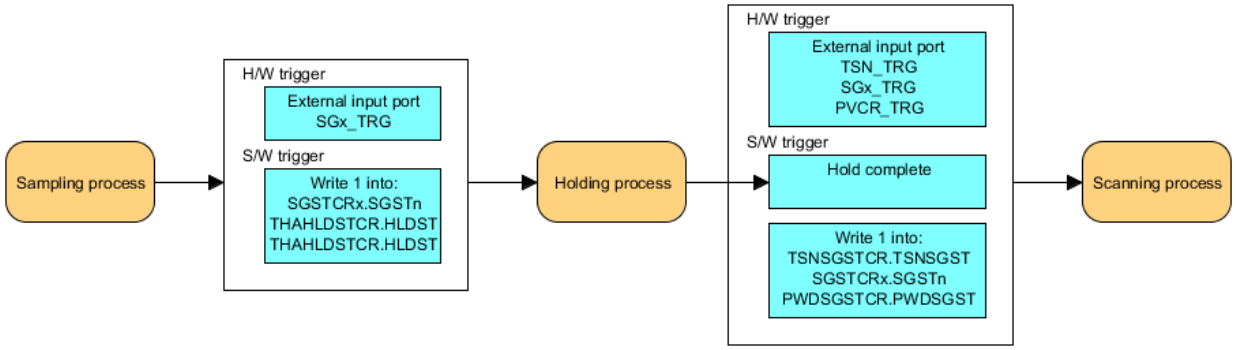

Figure 6.4: Trigger process of SARAD113x model

Explanation:

(1) Refer to HWM[2] chapter: 1.16.1.3.
(2) The Figure 6.4 shows the process of trigger into SARAD113x model.
(3) If SARAD113x model is in Sampling state, users can select suitable trigger based on the setting of register to change to Holding state. After a holding enactment time, the holding process will automatically to notify the hold complete trigger to start the Scanning process if hold complete trigger is enabled.

(4) Users can select several of trigger sources to start the corresponding scanning group. TSN is used as Scanning group 0 and PWD-Diag is used as Scanning group 4.

6.5. Interrupt

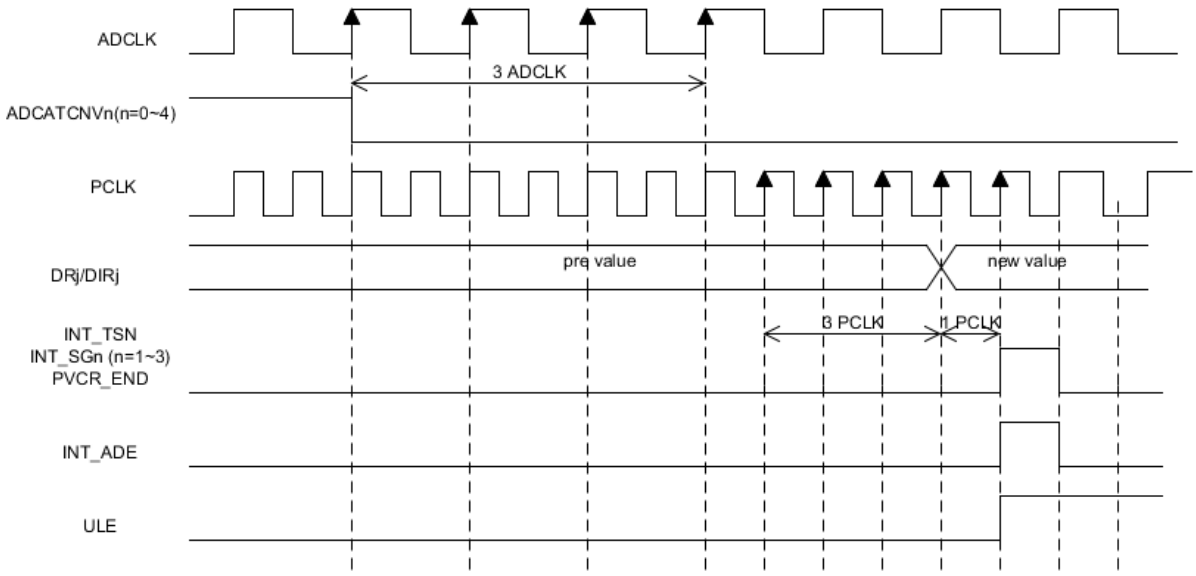

Figure 6.5: Interrupt output behavior of SARAD113x

Explanation:

(1) Refer to HWM[2] chapter: 1.16.4, 1.16.8, 1.16.9, 1.16.16, 1.16.17.

(2) The Figure 6.5 shows the interrupt behavior of SARAD113x model. In SARAD113x model, there are Scan end interrupt, Error interrupt and Upper/Lower bound error interrupt. These interrupts are asserted after 1PCLK period from the time A/D conversion data is updated to DRj register.

(3) Each group has corresponding scan end interrupt. The scan end interrupt indicate the virtual channel of a scanning group finished A/D conversion process if VCRj.ADIE of this channel is set to 1. The scan end interrupt indicate finish a scanning group finish A/D conversion of all virtual channel if SGCRx.ADIE is set to 1. For TSN and PWD-DIag, the scan end interrupt is always asserted. The interrupts is deasserted after 1 PCLK period.

(4) In case there is overwrite error or upper/lower bound error and enable bit is set to 1, the INT_ADE interrupt is asserted for 1 PCLK period.

(5) The ULE interrupt error is level interrupt, it is asserted when there is upper/lower bound error occurred. The ULE interrupt is deasserted when the error flag is cleared or enable bit is cleared to 0.

6.6. ADOP interface output

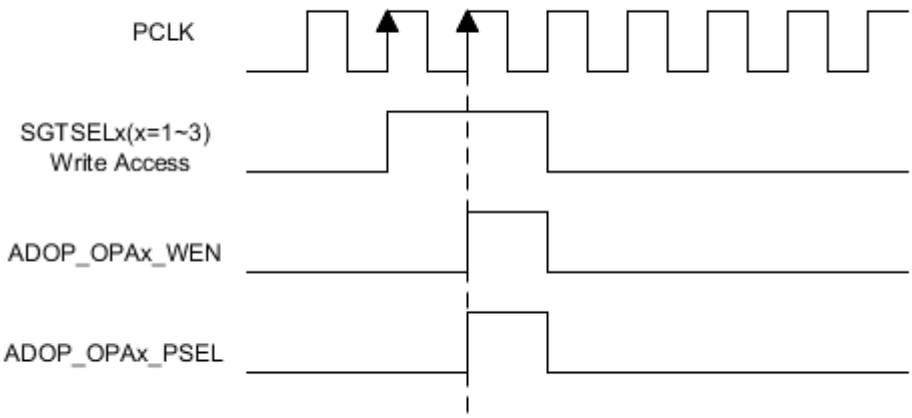

Figure 6.6: ADOP interface output behavior of SARAD113x

Explanation:

(1) Refer to HWM[2] chapter: 1.13.1, 1.16.13
(2) The Figure 6.6 shows the behavior of ADOP interface. When register SGTSELx is written, the corresponding ADOP_OPAx_WEN and ADOP_OPAx_PSEL will be asserted in 1 PCLK period.

6.7. Scanning timing chart

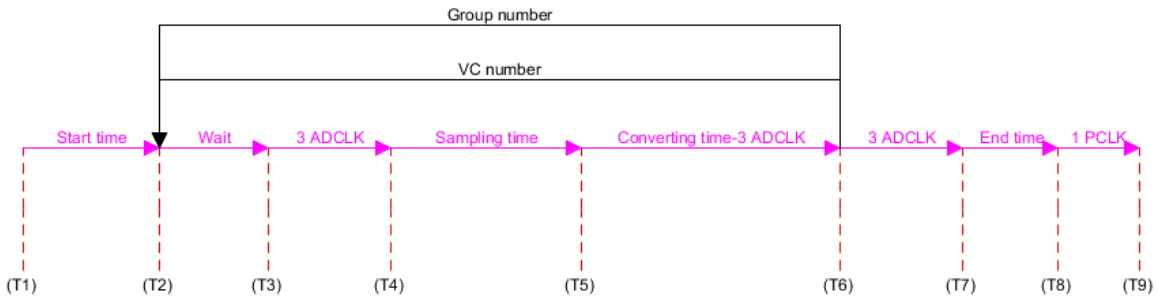

Figure 6.7: Scanning timing chart of SARAD113x model

Explanation:

(1) Refer to HWM[2] chapter: 1.16.1.3, 1.16.3.1, 1.16.8, 1.16.9, 1.16.15, 1.16.16, 1.16.17
(2) The Figure 6.7 shows the scanning timing chart for SARAD113x model. In this timing chart, the operations of SARAD113x model at the exact time marks are showed.
(3) The detail for SARAD113x operations which are corresponding to time marks are listed in the Table 6.9 and Table 6.10.

(4) Base on VC and group setting, the timing map for one group will be different.
(5) The **VC number** shows the number of VC which is calculated by the start VC (SGVCSPx) and the end VC (SGVCEPx).
(6) The **Group number** shows the number which one group is chosen to convert. The group number is based on the mode of group (multiple mode and continuous mode). So, one group can convert many times.
(7) The **Start time** is the time from trigger is asserted to start A/D conversion process. This time is based on each type of trigger. When EnableTimeCalculation is set to false, the Start time is set equal to tD parameter (ns) for Scanning group 0~3, and to tPWDD parameter (ns) for Scanning group 4.
(8) After the start VC of one group finishes VC repetition conversion, the next VC is continued to convert. When all VC is finished conversing, the start VC of this group is converted again. After finishing group number conversions, the group is finished scanning process.
(9) If the group is set up continuous mode (SGCRx.SCANMD = 1), the scanning process of group is only stop by reset function, stop function or suspend function. This group is not finished automatically.
(10) If the group is set up multi cycle mode (SGCRx.SCANMD = 0), the scanning process will finish after multiple number conversions which is configured by SGMCYCRx register.
(11) In the second time of conversion for any VC, if the next VC and the last VC is same, the “wait” time is zero.
(12) The **Sampling time** is defined by number which is value of SMPCR register (SG1~SG4) or TSNSMPCR (SG0) multiply with of ADCLK period.
(13) The **Converting time** is the value of EX_CNV parameter multiply with ADCLK period.
(14) When group is finished, it takes **End time** to finish current A/D conversion process. This End time is equal to 3 pclk periods if EnableTimeCalculation is set to true. When EnableTimeCalculation is set to false, it is equal to tED parameter (ns).
(15) If the other group is ready, the time map is continued. And the current group still finished its A/D conversion process.

Table 6-1: Corresponding tasks at time marks

Time Mark	Operation
(T1)	None
(T2)	- Assert corresponding ADCATCNVn - Set 1 into SGSRx.SGACT. - Write value to PVCR_MUXCUR output port if MPX is enabled. - Update MPXCURR register if MPX is enabled.
(T3)	None.
(T4)	None.
(T5)	None.
(T6)	- Deassert corresponding ADCATCNVn - Set 0 into SGSRx.SGACT.
(T7)	None

(T8)	- Update A/D conversion data to DRj, DIRj register
(T9)	- Assert corresponding scan end interrupt if enabled. - Assert error interrupt (overwrite error or upper/lower bound error) if enabled.

6.8. SARAD113x Reset behavior

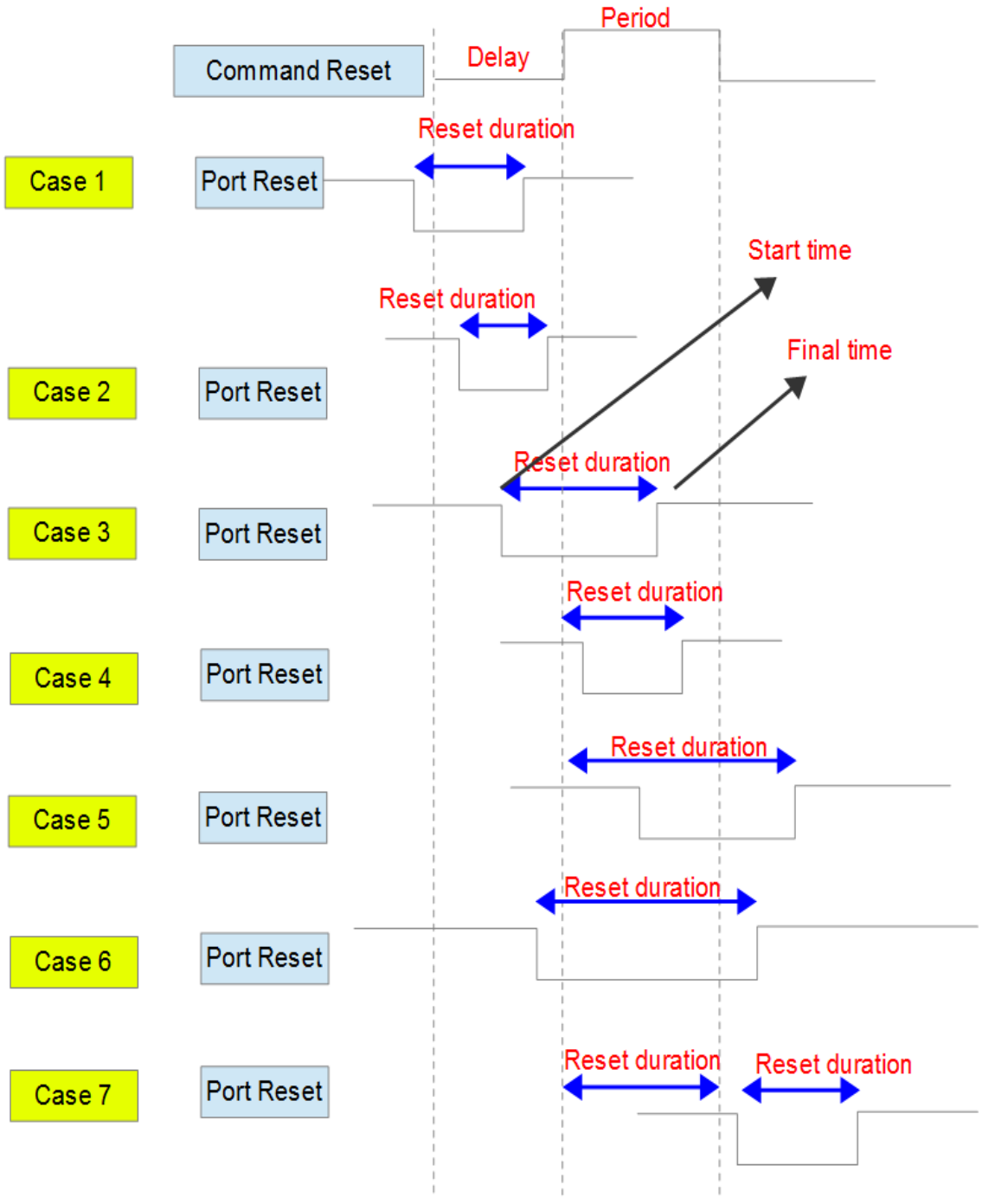

Figure 6.8: SARAD113x Reset behavior

Explanation:

(1) According to the Figure 6.8, there are seven cases for port reset (preset_n) and AssertReset command function. In every case, the “reset duration” for SARAD113x model is showed.

(2) At the “Start time” of “Reset duration”, all features of SARAD113x model is reset and remained during the “Reset duration”.

(3) At the “Final time” of “Reset duration”, the registers of SARAD113x model can be set up again.

7. Direction for users

7.1. File structures

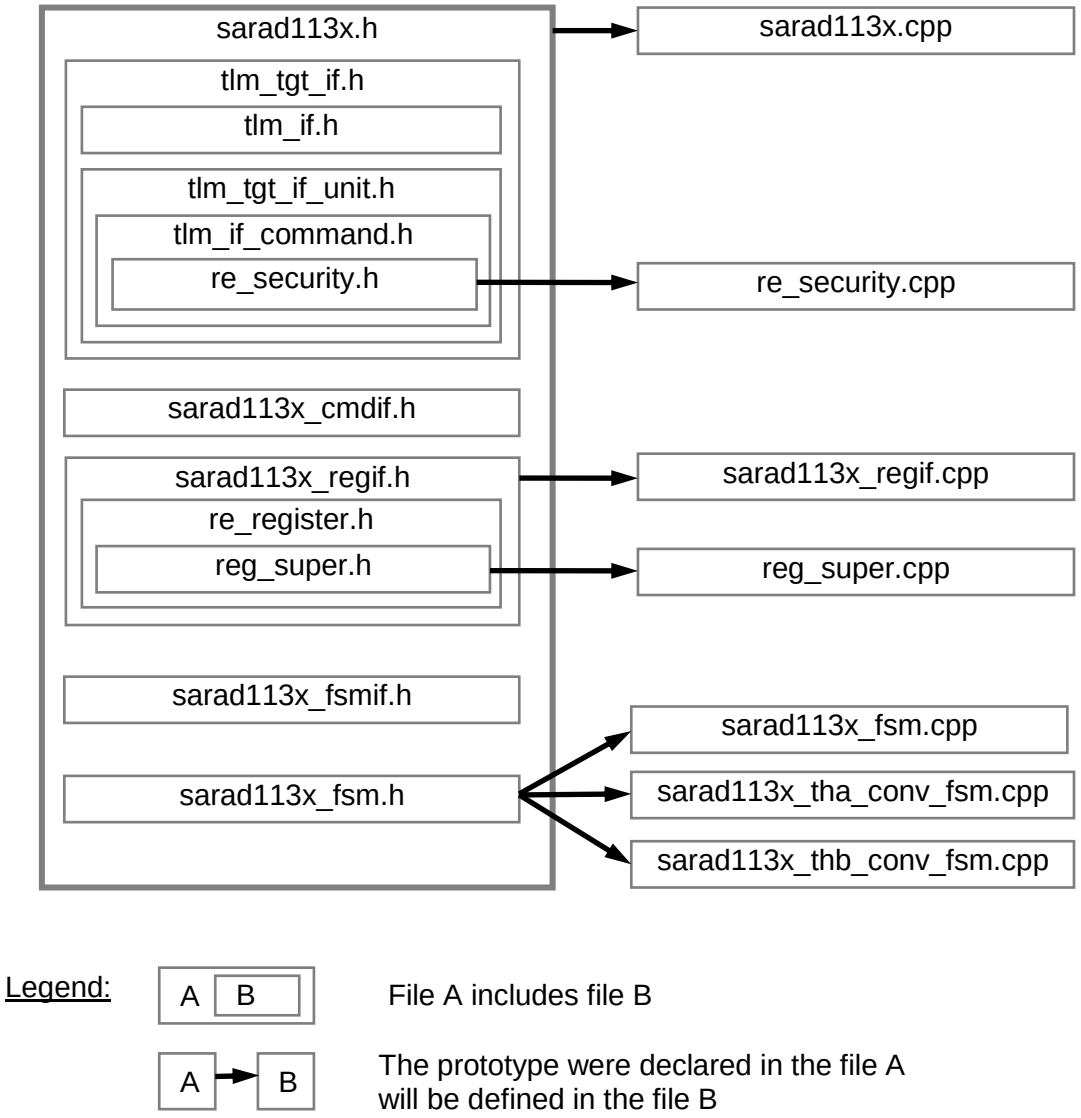

Figure 7.1: File structures

Table 7-1: File description

No.	File name	CVS tag	Developed/ reused	Description
1	re_register.h	v2013_05_28	Reused	Header file of the re_register class.
2	re_register.cpp		Reused	Implement the attributes and the operations of common register class.

No.	File name	CVS tag	Developed/ reused	Description
3	reg_super.h		Reused	General class for models to access to the memory array.
4	tlm_tgt_if.h	v2014_04_02	Reused	Header file of the tlm_tgt_if class.
5	tlm_if.h		Reused	Header file of the tlm_if class.
6	tlm_tgt_if_unit.h		Reused	Header file of the tlm_tgt_if_unit class.
7	tlm_if_command.h		Reused	Header file of the tlm_if_command class.
8	PY_SARAD113x.cpp	v2015_10_28	Generated (1)	Header file of Python IF of SARAD113x model.
9	PY_SARAD113x.h		Generated (1)	Implementation file of Python IF of SARAD113x model.
10	sarad113x_cmdif.h		Generated (1)	Command interface of SARAD113x model.
11	sarad113x_cmdif.txt		Developed	Input file of Command IF generator for SARAD113x model.
12	sarad113x.cpp		Developed	Implementation file of SARAD113x model.
13	sarad113x.h		Developed	Header file of SARAD113x model.
14	sarad113x_fsm.csv		Developed	Input file of FSM generator for SARAD113x model
15	sarad113x_tha_fsm.csv		Developed	Input file of FSM generator for SARAD113x model
16	sarad113x_thb_fsm.csv		Developed	Input file of FSM generator for SARAD113x model
17	sarad113x_fsm.cpp		Generated (2)	Implementation file of FSM control for SARAD113x model
18	sarad113x_fsm.h		Generated (2)	Header file of FSM control for SARAD113x model
19	sarad113x_fsmif.h		Generated (2)	FSM interface of SARAD113x model.
20	sarad113x_tha_conv_fsm.cpp		Generated (2)	Implementation file of FSM control for T&H group A of SARAD113x model
21	sarad113x_thb_conv_fsm.cpp		Generated (2)	Implementation file of FSM control for T&H group B of SARAD113x model
22	sarad113x_regif.cpp (4)		Generated (3)	Implementation file of Register IF of SARAD113x model.

No.	File name	CVS tag	Developed/ reused	Description
23	sarad113x_regif.h		Generated (3)	Header file of Register IF of SARAD113x model.
24	sarad113x_regif.txt		Developed	Input file of register IF generator for SARAD113x model.
25	re_security.h	v100419	Reused	Additional file of tlm_ini_if class and tlm_tgt_if class.
26	re_security.cpp		Reused	Additional file of tlm_ini_if class and tlm_tgt_if class.

Note:
(1) sarad113x_cmdif.h,“PY_SARAD113x.h” and “PY_SARAD113x.cpp” are generated from Command IF Generator v2015_02_12.
(2) sarad113x_fsm.cpp, sarad113x_fsm.h, sarad113x_fsmif.h, sarad113x_tha_conv_fsm.cpp, sarad113x_thb_conv_fsm.cpp files are generated from FSM Generator v2014_07_21.
(3) sarad113x_regif.h, sarad113x_regif.cpp files are generated from Register IF Generator v2014_12_01. **(4)** sarad_regif.cpp is modified to map the register name and bit name to HWM.

7.2. Input/Output file

(1) SARAD113x model does not have output file or input file.

7.3. How to connect Verification Environment

(1) There are 4 basic steps to connect a SARAD113x model to a verification environment

(1.1) Step 1: Declare an instance of the SARAD113x class in the top model.

(1.2) Step 2: Connect SARAD113x's target sockets named m_tgt_sockets with initiator sockets of the bus model.

(1.3) Step 3: Connect SARAD113x's interrupt output signals and the data outputs to suitable models.

(1.4) Step 4: Connect the reset signal preset_n, pclk signal, ADCLK signal and other inputs to SARAD113x model.

7.4. handleCommand

Table 7-2: List of parameters of handleCommand API

No.	Parameters	Type	Default	Description
1	DumpRegisterRW	bool	false	- Dump register access information when register gets accessed. + false ... Not dump register access information.

No.	Parameters	Type	Default	Description
				+ true ... Dump register access information.
2	MessageLevel	string	fatal|error	- Select debug message level from “fatal”, “error”, “warning”, “info” (for register access only). One or more than levels can be connected by vertical bar. Example “fatal|error”.
3	DumpInterrupt	bool	false	Dump interrupt information when interrupt is asserted. This message is info level. + false ... Not dump interrupt information. + true ... Dump interrupt information.
4	EnableConvertInfo	bool	false	- Enable/disable to dump AD convert activity. + true: Enable. + false: Disable.
5	EX_HLD_CDT	double	18	- Number of ADCLK cycle from starting to hold an analog input until starting AD conversion
6	EX_CNVT	double	22	- Number of ADCLK cycle for successive approximation AD conversion
7	tD	double	0	- Delay time until starting scan group for SW trigger (ns) This parameter is only effected when parameter EnableTimeCalculation is set to false.
8	tPWDD	double	0	- Delay time until starting scan group for PVCR_TRG (ns) This parameter is only effected when parameter EnableTimeCalculation is set to false.
9	tED	double	0	- Delay time until competing scanning group. (ns). This parameter is only effected when parameter EnableTimeCalculation is set to false.
10	Avrefh	double	3.3	- Voltage value of AVREFH0. Before setting this parameter, the AVREFH0 port is used. This parameter can be set only at simulation time 0. (The default value is not used)
11	AVcc	double	3.3	-The voltage value of AVCC. This parameter can be set only at simulation time 0.
12	AVss	double	0	-The voltage value of AVSS. This parameter can be set only at simulation time 0.
13	EnableTimeCalculation	bool	true	Enable/disable to calculate the processing time. **(*)** + false : use parameters tD, tPWDD, tED + true : calculate the processing time.

(*) Note: The delay timing processing is specified by setting the parameter EnableTimeCalculation
- When EnableTimeCalculation is set to "false", SARAD113x model uses the parameter tD, tPWDD and tED for corresponding delay time.
- When EnableTimeCalculation is set to "true", the value of delay time is calculated as below:
+ delay time corresponding to tD = wait time to next rising edge of ADCLK + 2 * ADCLK clock period (SW trigger)
+ delay time corresponding to tPWDD = 1 * PCLK clock period + wait time to next ADCLK + 2 * ADCLK clock period (HW trigger for PWD-Diag)
+ delay time corresponding to tED = 3 * PCLK clock period

Table 7-3: List of commands of handleCommand API

No.	Parameters	Type	Argument	Description
1	DumpStatInfo	void	-	Dump the statistical information about activity. When this command is called, SARAD113x model dumps the following information: - Number of each interrupt activation (These values are reset when reset function is active).
2	AssertReset	void	<start-time, period>	Assert and de-assert reset signal to the SARAD113x model. + <start-time>: the time until asserting reset signal from current time. The unit is "ns". + <period>: the time from asserting reset signal to de-asserting it. The unit is "ns".
3	SetCLKfreq	void	<clk_name, clk_frequency>	Change the value of clock frequencies (pclk or ADCLK) + <clk_name>: The clock name is changed the frequency. + <clk_frequency>: The new frequency is updated to the clock.
4	help	void	-	Dump the direction how to use handleCommand parameters and commands.

7.5. Register RW messages style

Table 7-4: Dump Register RW message description

Condition	This message is dumped out when SARAD113x registers are accessed.
Output	This message is printed to standard output (console).
Format: Info: (<hier_instance_name>): [<time>ns] REG [<reg_name>] <operation> Size = <size> Addr = <reg_address> Data = <reg_value> **Example:** Info: reslx.sarad113x: [75 ns] REG [SGCR0] R Size= 1 Addr= 0xFFF92490 Data= 0x0	
Tag name	**Description**
time	Simulation time.
hier_instance_name	Hierarchy instance name of SARAD113x model is being used.
reg_name	Name of accessed register.
operation	R or W (read or write).
size	Accessed register size.

reg_address	Address of accessed register.
reg_value	Register value.

7.6. Interrupt/error condition messages style

Table 7-5: Dump Interrupt/error condition message description

Condition	This message is dumped out when SARAD113x interrupt/error is asserted.
Output	This message is printed to standard output (console).
Format: Info: (hier_instance_name): [<time>ns] INT [SARAD113x: interrupt_name] Assert **Example:** Info: reslx.sarad113x: [622 ns] INT [SARAD113x: INT_SG1] Assert.	
Tag name	**Description**
time	Simulation time.
hier_instance_name	Hierarchy instance name of SARAD113x model is being used.
interrupt/error_name	Interrupt/error factor: INT_SGx, PVCR_END, INT_TSN, INT_ADE, ULE.

7.7. DumpStatInfo messages style

Table 7-6: State information message description

Condition	This message is dumped out when “DumpStatInfo” is transferred to handleCommand.
Output	This message is printed to standard output (console).

Format: PROFILE(StatInfo):F1K/SARAD113x: Info [<time>ns] (hier_instance_name) :
PROFILE(StatInfo): F1K/SARAD113x: INT_TSN active number: <value>
PROFILE(StatInfo): F1K/SARAD113x: INT_SG1 active number: <value>
PROFILE(StatInfo): F1K/SARAD113x: INT_SG2 active number: <value>
PROFILE(StatInfo): F1K/SARAD113x: INT_SG3 active number: <value>
PROFILE(StatInfo): F1K/SARAD113x: PVCR_END active number: <value>
PROFILE(StatInfo): F1K/SARAD113x: INT_ADE active number: <value>
PROFILE(StatInfo): F1K/SARAD113x: ULE active number: <value>
PROFILE(StatInfo): F1K/SARAD113x: EndInfo

Example:
PROFILE(StatInfo): F1K/SARAD113x: Info[208755 ns]
PROFILE(StatInfo): F1K/SARAD113x: INT_SG1 active number : 5
PROFILE(StatInfo): F1K/SARAD113x: INT_SG2 active number : 10
PROFILE(StatInfo): F1K/SARAD113x: INT_SG3 active number : 4
PROFILE(StatInfo): F1K/SARAD113x: PVCR_END active number : 2
PROFILE(StatInfo): F1K/SARAD113x: INT_ADE active number : 0

```
PROFILE(StatInfo): F1K/SARAD113x:  ULE      active number : 0
PROFILE(StatInfo): F1K/SARAD113x: EndInfo.
```

Tag name	Description
time	Simulation time.
hier_instance_name	Hierarchy instance name of SARAD113x model is being used.
value	Register's value or number of each interrupts activation.

7.8. Help messages

Table 7-7: Dump help command message description

Condition	This message is dumped out when “help” is transferred to handleCommand.
Output	The help message is used for handleCommand.

```
--- command ---
help                                Show direction
MessageLevel   <fatal|error|warning|info> Select debug message level (Default: fatal,error)
AssertReset <start_time> <period>        Assert and deassert reset signal to a target model
DumpInterrupt <enable>                Display interrupt information when interrupt get set[ns] ( Default:false  )
EnableConvertInfo <enable>          Display AD convert activity ( Default:false  )
Avrefh <Avrefh>                     Reference voltage of AVREFH ( Default:3.3  )
AVcc <AVcc>                         Reference voltage of AVCC ( Default:3.3  )
AVss <AVss>                         Reference voltage of AVSS ( Default:0  )
EX_HLD_CDT  <EX_HLD_CDT> Number of ADCLK cycle from starting to hold an analog input until starting AD conversion ( Default:18  )
EX_CNVT <EX_CNVT>               Number of ADCLK cycle for successive approximation AD conversion ( Default:22  )
tD <tD>                             Delay time until starting scan group for SW trigger (ns) ( Default:0  )
tPWDD <tPWDD>                       Delay time until starting scan group for PVCR_TRG (ns) ( Default:0  )
tED <tED>                           Delay time until ending scan group (ns) ( Default:0  )
EnableTimeCalculation <enable>    Enable using formula to calculate the delay times ( Default:true  )
DumpStatInfo                        Dump the statistical information about SAR model activity
SetCLKfreq <clk_name> <clk_freq>         Setup the new clock frequency
```

Table 7-8: Dump tgt help command message description

Condition	This message is dumped out when “tgt help” is transferred to handleCommand.
Output	The help message is used for handleCommand.

```
Command                          Description

-------------------------------------------------------------------------------
set_param <term> <value> : Set simulation information about access to target.
            <term>  : m_bus_clk | m_bus_gnt | m_bus_rgnt | m_buf_size
                    | m_wr_latency | m_rd_latency | m_phase_mode
                    | m_p_log_file | m_wr_log | m_rd_log | m_msg_out_lvl
            <value> : Please see tlm_common_class spec sheet.
get_param <term>   : Get simulation information about access to target.
init_param         : Initialize simulation information.
```

Table 7-9: Dump reg help command message description

Condition	This message is dumped out when “tgt help” is transferred to handleCommand.
Output	The help message is used for handleCommand.

```
--- reg ---
reg MessageLevel <fatal|error|warning|info> Select debug message level (Default: fatal,error)
reg DumpRegisterRW <true/false>              Select dump register access information  (Default: false)
reg DumpFileNameLineNum <true/false>     Select dump information about file name and line number (Default: false)
reg <register_name> MessageLevel <fatal|error|warning|info> Select debug message level for register (Default:
fatal,error)
reg <register_name> force <value>            Force register with setting value
reg <register_name> release                  Release register from force value
reg <register_name> <value>                  Write a value into register
reg <register_name>                          Read value of register
reg help                                     Show a direction
```

7.9. Error and debugging messages

7.9.1. Error and debugging messages style

Table 7-10: Dump error message description

Condition	This kind of message is dumped out when error occurs or some important events occur. Detailed conditions are described in the “Description” column of table 7.11.
Output	This kind of message is printed to standard output (console).

Format: <Severity>: (<hier_instance_name>): [<time>ns] [Message content]
Example: Error: reslx.sarad.mADC0: [1135 ns] The condition of tED, tSPL and tSAR is not correct.

Tag name	Description
severity	Kind of severity of the message.
time	Simulation time.
hier_instance_name	Hierarchy instance name of SARAD113x model is being used.

7.9.2. Error and debugging messages

Table 7-11: Error message

No.	Type	Severity	Message	Description
1	User	Error	Clock name is invalid.	Dump this message when clock name is invalid
2	User	Info	%s for PWM-Diag.	Dump this message when SARAD113x models changing processing state for PWM-Diag channel. The processing state are: + Start A/D conversion process + Finish A/D conversion process + Start sampling + Start A/D converting
3	User	Info	%s for Temperature sensor.	Dump this message when SARAD113x models changing processing state for TSN channel. The processing state are: + Start A/D conversion process + Finish A/D conversion process + Start sampling + Start A/D converting
4	User	Info	%s for Virtual channel %d.	Dump this message when SARAD113x models changing processing state for virtual channel 0~49. The processing state are: + Start A/D conversion process + Finish A/D conversion process + Start sampling + Start A/D converting
5	User	Info	A/D conversion data of scanning group %d is stored to register	Dump this message when SARAD113x stored converted value into DRj register
6	User	Info	Hardware trigger group %d is asserted.	Dump this message when hardware trigger of a scanning group is asserted
7	User	Info	Hold process for T&H group A completed.	Dump this message when holding process of group A is completed
8	User	Info	Hold process for T&H group B completed.	Dump this message when holding process of group B is completed
9	User	Info	Reset signal is asserted.	Dump this message when reset signal is asserted
10	User	Info	Reset signal is negated.	Dump this message when reset signal is deasserted
11	User	Info	SARAD operation is stopped.	Dump this message when SARAD113x operation is stopped

No.	Type	Severity	Message	Description
12	User	Info	SARAD113x moves from RESET state to IDLE state.	Dump this message when SARAD113x move from RESET state to IDLE state after reset signal is deasserted or reset time is ended
13	User	Info	SARAD113x moves to RESET state.	Dump this message when SARAD113x move to RESET state from other states after reset signal is asserted or reset time is started
14	User	Info	SARAD113x will reset for %f ns after %f ns.	Dump this message when SARAD113x start to reset by handleCommand
15	User	Info	Scanning group %d is suspended.	Dump this message when scanning group is suspend by other higher priority group
16	User	Info	Scanning group %d starts for A/D conversion.	Dump this message when scanning group starts the A/D conversion process
17	User	Info	Software trigger group %d is asserted.	Dump this message when software trigger of a scanning group is asserted
18	User	Info	Start hold process for T&H group A.	Dump this message when holding process of group A is started
19	User	Info	Start hold process for T&H group B.	Dump this message when holding process of group B is started
20	User	Info	Start sampling process for all T&H channels.	Dump this message when start sampling process for all T&H channels
21	User	Info	The ADCLK is set with a frequency as %f	Dump this message when users set value to ADCLK clock
22	User	Info	The pclk is set with a frequency as %f	Dump this message when users set value to pclk clock
23	User	Info	INT [SARAD113x: %s] Assert.	Dump this message when interrupt (INT_TSN, INT_SGx, PVCR_END, INT_ADE, ULE) is asserted and DumpInterrupt parameter is set to true
24	User	Info	INT [SARAD113x: %s] Deassert.	Dump this message when interrupt (INT_TSN, INT_SGx, PVCR_END, INT_ADE, ULE) is deasserted and DumpInterrupt parameter is set to true
25	User	Info	Start simulation time : %f End simulation time : %f The scanning group : %d The scanning mode : Multi cycle mode The scanning mode : Continuous mode The frequency count : %d/%d The VC number : %d The VCR value : 0x%X The repetition count : %d/%d The A/D conversion type : %s The A/D conversion value : 0x%X	Dump this message when SARAD113x finished an A/D conversion and EnableConvertInfo parameter is set to true
26	User	Warning	Changing AVcc is invalid when simulation time is greater than zero.	Dump this message when users change AVcc after simulation started
27	User	Warning	Changing AVss is invalid when simulation time is greater than zero.	Dump this message when users change AVss after simulation started

No.	Type	Severity	Message	Description
28	User	Warning	Changing delay times calculation is invalid when simulation time is greater than zero.	Dump this message when users change time calculation method after simulation started
29	User	Warning	Changing reference voltage AVREFH0 is invalid when simulation time is greater than zero.	Dump this message when value of port AVREFH0 is changed after simulation started
30	User	Warning	Changing reference voltage Avrefh is invalid when simulation time is greater than zero.	Dump this message when users change Avrefh parameter after simulation started
31	User	Warning	Changing the EX_CNVT parameter is not affected when SARAD113x model is operating.	Dump this message when users change EX_CNVT parameter while model is operating
32	User	Warning	Changing the EX_HLD_CDT parameter is not affected when SARAD113x model is operating.	Dump this message when users change EX_HLD_CDT parameter while model is operating
33	User	Warning	Changing the tD parameter is not affected when SARAD113x model is operating.	Dump this message when users change tD parameter while model is operating
34	User	Warning	Changing the tED parameter is not affected when SARAD113x model is operating.	Dump this message when users change tED parameter while model is operating
35	User	Warning	Changing the tPWDD parameter is not affected when SARAD113x model is operating.	Dump this message when users change tPWDD parameter while model is operating
36	User	Warning	Continuous scanning mode is not supported in T&H operation.	Dump this message when users set continuous mode for scanning group used for T&H
37	User	Warning	EX_CNVT must be equal or greater than 3 cycles (AD clock).	Dump this message when users set the EX_CNVT parameter less than 3
38	User	Warning	Holding process can not start because all of T&H channel is disabled.	Dump this message when users start holding process while all of T&H channels are disabled (THER = 0)
39	User	Warning	Input analog value of physical channel %d (%f) is greater than AVREFH value (%f).	Dump this message when input value of analog input port greater than AVREFH value
40	User	Warning	MPX function should not be enabled for T&H operation.	Dump this message when users use MPX function for scanning group used for T&H
41	User	Warning	Physical channel %d should not be used in both T&H group A and B.	Dump this message when users use the same physical channel in both group A and B
42	User	Warning	Register %s can be accessed only when SGACT and TRGMD of all scanning groups are 0.	Dump this message when users access to register while SGACT and TRGMD of all scanning group are not 0
43	User	Warning	Register %s can be accessed only when SGACT and TRGMD of all scanning groups n (n=1~3) are 0.	Dump this message when users access to register while SGACT and TRGMD of all scanning group SGx (x=1~3) are not 0
44	User	Warning	Register %s can be accessed only when SGACT and TRGMD of scanning group %d are 0.	Dump this message when users access to register while SGACT and TRGMD of scanning group SGn (n=0~4) are not 0

No.	Type	Severity	Message	Description
45	User	Warning	Register %s can not be accessed when A/D conversion is operating with HW trigger.	Dump this message when users access to register SGVCSP of which scanning group is operating with H/W trigger
46	User	Warning	Reset is in progress.	Dump this message when users assert reset by handleCommand while SARAD113x model is in RESET state
47	User	Warning	SGPRCR can not be changed when SARAD113x is operating.	Dump this message when users written to SGPRCR register while SARAD113x is operating
48	User	Warning	Sampling process can not start because HLDCTE is not set 1 or HLDTE is set to 0 when T&H self-diagnosis is used.	Dump this message when users start sampling process in T&H self-diagnosis mode without setting HLDCTE to 1 and HLDTE to 0
49	User	Warning	Sampling process can not start because T&H group A and B are disabled.	Dump this message when users start sampling process without enable T&H group A or B
50	User	Warning	Sampling process can not start because all of T&H channel is disabled.	Dump this message when users start sampling process without enable any T&H channel (THER = 0)
51	User	Warning	Sampling process can not start because operation of Scanning group A or B is not ended.	Dump this message when users start sampling process while both scanning group A and B are not ended
52	User	Warning	Sampling process can not start because the same scanning group is set for both T&H group A and B.	Dump this message when users start sampling process set same scanning group for both T&H group A and B
53	User	Warning	Sampling time for T&H channel of %s must be equal or greater than %d ns.	Dump this message when users start hold process while sampling time less than 450ns
54	User	Warning	Scanning group %d can not start due to error or conflict setting.	Dump this message when scanning group can not start A/D conversion process due to error or conflict setting
55	User	Warning	Scanning group %d is stopped.	Dump this message when scanning group stop the A/D conversion process
56	User	Warning	Self-diagnosis channel should not be used in Virtual channel %d when Self-diagnosis function is disabled (ADCR.DGON = 0).	Dump this message when users use self-diagnosis channel when self-diagnosis function is disabled
57	User	Warning	Setting value to %s.GCTRL must be equal or less than %d.	Dump this message when users set the value to VCRj.GCTRL greater than 36
58	User	Warning	Setting value to %s.MCYC must be equal or less than %d.	Dump this message when users set the value to SGMCYCRx.MCYC greater than 0x3
59	User	Warning	Setting value to %s.VCEP must be equal or greater than value of %s.VCSP.	Dump this message when users set the value of SGCVEPx.VCEP less than SGVCSPx.VCSP
60	User	Warning	Setting value to %s.VCEP must be equal or less than %d.	Dump this message when users set the value of SGCVEPx.VCEP greater than 49
61	User	Warning	Setting value to %s.VCSP must be equal or less than %d.	Dump this message when users set the value of SGCVSPx.VCSP greater than 49
62	User	Warning	Setting value to %s.VCSP must be equal or less than value of %s.VCEP.	Dump this message when users set the value of SGCVSPx.VCSP greater than SGVCEPx.VCEP
63	User	Warning	Setting value to SMPCR.SMPT must be equal or greater than %d.	Dump this message when users set the value of SMPCR.SMPT less than 0x12
64	User	Warning	Setting value to TSNSMPCR.TSNSMPT must be equal or greater than %d.	Dump this message when users set the value of TSNSMPCR.TSNSMPT less than 0x12

No.	Type	Severity	Message	Description
65	User	Warning	T&H channel %d of T&H group A is used for T&H group B.	Dump this message when T&H channel of T&H group A is used for T&H group B
66	User	Warning	T&H channel %d of T&H group B is used for T&H group A.	Dump this message when T&H channel of T&H group B is used for T&H group A
67	User	Warning	The ADCLK period is less than 1 unit time of system.	Dump this message when ADCLK period is set less than 1 unit time of system
68	User	Warning	The T&H channels 0-2 should not set to same group with channels 3-5.	Dump this message when users set the T&H channels 0-2 to same group of T&H channels 3-5
69	User	Warning	The frequency of ADCLK clock is changed while SARAD113x model is operating.	Dump this message when the frequency of ADCLK clock is changed during SARAD113x is operating
70	User	Warning	The frequency of pclk clock is changed while SARAD113x model is operating.	Dump this message when the frequency of pclk clock is changed during SARAD113x is operating
71	User	Warning	The hold process of group A can not start because T&H group A is disabled.	Dump this message when users start hold process for T&H group A but it is disabled
72	User	Warning	The hold process of group A can not start because all of T&H channel for group A is disabled.	Dump this message when users start hold process for T&H group A but all of T&H channel of group A are disabled
73	User	Warning	The hold process of group B can not start because T&H group B is disabled.	Dump this message when users start hold process for T&H group B but it is disabled
74	User	Warning	The hold process of group B can not start because all of T&H channel for group B is disabled.	Dump this message when users start hold process for T&H group B but all of T&H channel of group B are disabled
75	User	Warning	The pclk period is less than 1 unit time of system.	Dump this message when pclk period is set less than 1 unit time of system
76	User	Warning	The physical channels setting should not be changed when T&H is operating.	Dump this message when users update the physical channel setting of group A or B during operating
77	User	Warning	The scanning frequency should be set to 1 in multi cycle scanning mode of T&H operation (%s.MCYC = 0).	Dump this message when users set the SGMCYCRx.MCYC greater than 0 for T&H group in multi cycle scanning mode
78	User	Warning	The value of %s.ULMTB should be equal or greater than %s.LLMTB.	Dump this message when users set the value of ULLMTBRx.ULMTB less than ULLMTBR.LLMTB
79	User	Warning	The value of register %s should be equal or less than value of register %s.	Dump this message when users set the value of SGCVEPx.VCEP less than SGVCSPx.VCSP
80	User	Warning	The value of register OWER is not updated because it is not cleared from previous error.	Dump this message when overwrite error occurs but the register OWER is not cleared from previous error
81	User	Warning	The value of register ULER is not updated because it is not cleared from previous error.	Dump this message when upper/lower bound error occurs but the register ULER is not cleared from previous error
82	User	Warning	The virtual channels setting should not be changed when T&H is operating.	Dump this message when users update the virtual channel setting of group A or B during operating
83	User	Warning	When A/D conversion 10bit is selected (ADCR.CTYP = 1), %s.ULMTB[1:0] should be set to 2'b11 and %s.LLMTB[1:0] should be set to 2'b00.	Dump this message when users set ULLMTBRx.ULMTB[1:0] different from 2'b11 and ULLMTBRx.LLMTB[1:0] different from 2'b00 while 10bit conversion is selected (ADCR.CTYP = 1)
84	User	Warning	When HLDCTE = 1 and HLDTE = 1, HW trigger must be asserted after HLDST is set.	Dump this message when users assert H/W trigger for T&H group while HLDCTE = 1 and HLDTE = 1 before set 1 to HLDST

Table 7-12: Error and debugging message of handleCommand

No.	Type	Severity	Message	Description
1	User	Error	wrong number of arguments (<command>) : Type reslx.sarad113x help	Dump this message when a command is called with wrong number of arguments.
2	User	Error	wrong argument: <argument> (<command>) : Type reslx.sarad113x help	Dump this message when a command is called with illegal argument.
3	Internal	Error	command name “<command name>” is invalid.	This message is returned when an invalid command is called. (Command handler model dumps “Unknown command name” instead if this message is returned).
4	User	Error	<command name> has too much arguments.	Dump this message when a command of Register IF is called with wrong number of arguments.
5	User	Error	<command name> command needs an argument [<valid value>]	Dump this message when a command of Register IF is called with illegal argument.
6	User	Error	[<register name>] Invalid force value	Dump this message when users force an invalid value to a register.
7	User	Error	Wrong command : <reg ...>	Dump this message when an invalid command of Register IF is called.
8	User	Error	[<register name>] Invalid write value	Dump this message when users write an invalid value through Register IF command to a register.
9	Internal	Error	Register name is invalid	This message is returned when an unknown register is access through Register IF command. (Command handler model dumps “Unknown command name” instead if this message is returned).

Table 7-13: Message level description

No.	Message level	Description
1	Fatal	Illegal operation, which causes simulation to stop.
2	Error	Illegal operation, which cannot be processed but possible to recover (continue simulation).
3	Warning	Illegal operation, but it is ignorable.
4	Info	Information of internal operation.

7.10. Defined macro and template

(1) This model uses “IS_RESET_ACTIVE_LOW” to define the active level (active low) for the reset port.

(2) This model uses "REGIF_SC_REPORT" to dump output message by SC_REPORT.

(3) This model uses "SARAD113x_ENABLE_TSN" to enable the TSN features.

(4) This model supports only 32 bit bus-width socket, so there is no bus-width template.

8. Flow diagram

Summary:

(1) The chapter 8.1 describes the sequence flow of the SARAD113x model.

(2) The chapter 8.2 describes the state of SARAD113x model.

(3) The chapter 8.3 and 8.4 describes the reset process for SARAD113x model.

(4) The chapter 8.18 describes the parameters and handleCommand functions in for SARAD113x model.

(5) The other chapters describe the operations of SARAD113x model.

Table 8-1: Features and diagram reference table

Model Features	HWM Chapter	Diagram	Description	Figure
Sequence flow	-	Sequence flow	Sequence flow of model.	Figure 8.1
State diagram	-	State diagram	State information of model.	Figure 8.2 Figure 8.3
preset_n input port flow	-	preset_n input port flow	preset_n input port flow	Figure 8.4
AssertReset function flow	-	AssertReset function flow	The reset flow of model	Figure 8.5
CheckTrigger function processing flow	1.10.17, 1.10.18, 1.11.14, 1.16.1.2, 1.16.1.3, 1.16.2, 1.16.3, 1.16.12, 1.16.15	CheckTrigger function processing flow	The processing flow of CheckTrigger function	Figure 8.6
Start scanning processing flow	1.16.2, 1.16.3, 1.16.4, 1.16.15	Start scanning processing flow	The processing flow of start scanning process of scanning group	Figure 8.7

Model Features	HWM Chapter	Diagram	Description	Figure
Check scanning group setting processing flow	1.3.1.13, 1.10.1, 1.10.2, 1.10.3, 1.10.4, 1.11.7, 1.11.8, 1.16.3.1	Check scanning group setting processing flow	The processing flow of checking a scanning group	Figure 8.8
Start virtual channel scanning processing flow	1.3.1.13, 1.10.1, 1.10.2, 1.10.3, 1.10.4, 1.11.7, 1.11.8, 1.16.3.1	Start virtual channel scanning processing flow	The processing flow of start A/D conversion process of virtual channel	Figure 8.9
Virtual channel sampling processing flow	1.16.1.3, 1.16.2, 1.16.3, 1.16.4, 1.16.5	Virtual channel sampling processing flow	The processing flow of of sampling process of virtual channel	Figure 8.10
Virtual channel conversion processing flow	1.16.1.3, 1.16.2, 1.16.3, 1.16.4, 1.16.5	Virtual channel conversion processing flow	The processing flow of of conversion process of virtual channel	Figure 8.11
GetANIPortVal function processing flow	1.10.1, 1.10.2, 1.10.3, 1.10.4, 1.10.11, 1.14.1, 1.14.2, 1.14.3, 1.14.4, 1.16.18	GetANIPortVal function processing flow	The processing flow of GetANIPortVal function of model	Figure 8.12
Finish virtual channel conversion processing flow	1.16.1.3, 1.16.2, 1.16.3, 1.16.4, 1.16.5	Finish virtual channel conversion processing flow	The processing flow of finishing A/D conversion process of virtual channel	Figure 8.13
ADConvert function processing flow	1.10.1, 1.10.2, 1.10.3, 1.10.4, 1.10.11, 1.10.22, 1.10.24, 1.16.8	ADConvert function processing flow	The processing flow of ADConvert function of model	Figure 8.14
Update conversion data processing flow	1.10.1~1.10.9, 1.10.21, 1.10.24, 1.10.25, 1.16.8, 1.16.9, 1.16.16, 1.16.17	Update conversion data processing flow	The processing flow of updating conversion data of model	Figure 8.15
IsLastVC function processing flow	1.10.11, 1.11.4, 1.16.2, 1.16.3, 1.16.4.1, 1.16.5, 1.16.6, 1.16.15	IsLastVC function processing flow	The processing flow of IsLastVC function of model	Figure 8.16
StopOperation function processing flow	1.10.10, 1.16.14	StopOperation function processing flow	The processing flow of StopOperation function of model	Figure 8.17

Model Features	HWM Chapter	Diagram	Description	Figure
handleCommand operation processing flow	-	handleCommand operation processing flow	The processing flow of handleCommand function of model	Figure 8.18

8.1. Sequence flow

The SARAD113x model is described via Csarad113x class, Csarad113x_fsm class, Csarad113x_tha_conv_fsm class and Csarad113x_thb_conv_fsm class. The relationships of these classes are showed in Figure 8.1.

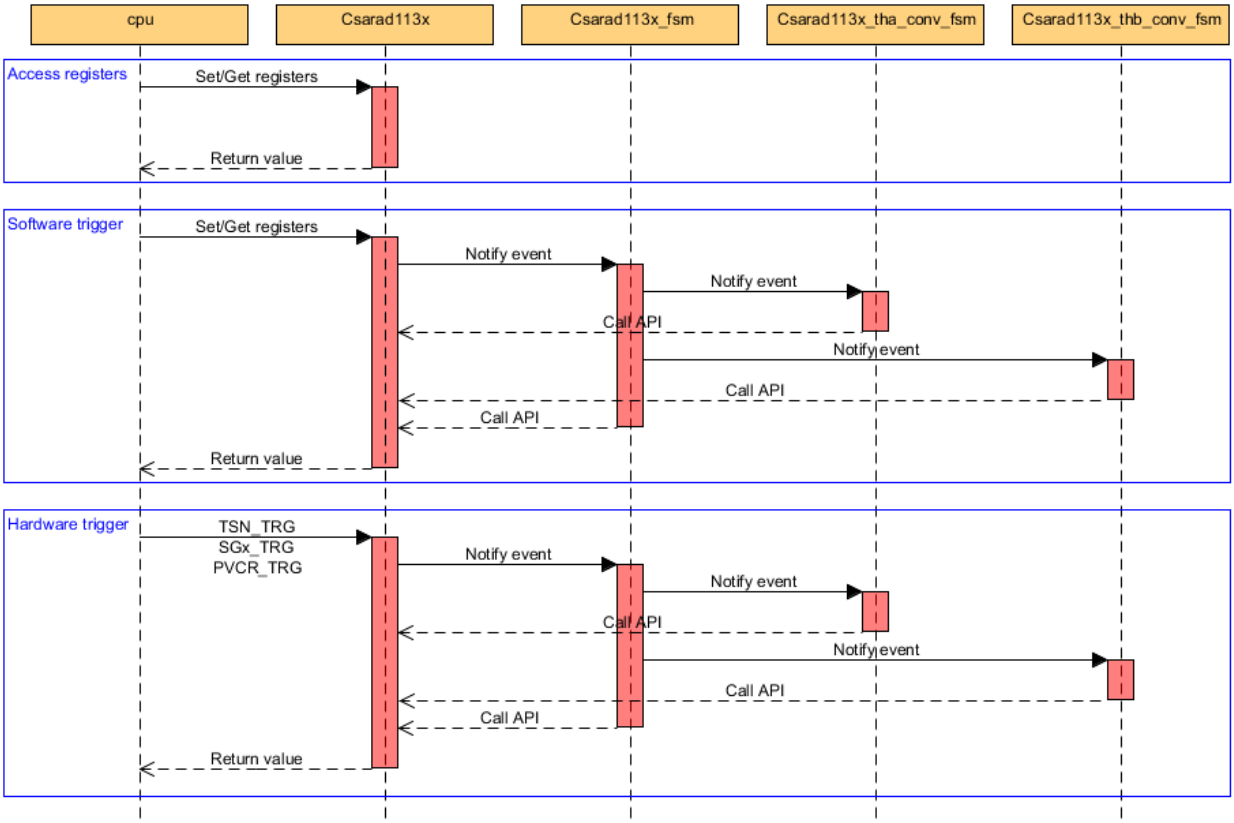

Figure 8.1: Sequence diagram of SARAD113x model

Explanation:

(1) Users set or get the value of registers through tlm_tgt_if to the SARAD113x model.

(2) **Software trigger:** When users write 1 to register TSNSGSTCR.TSNSGST, SGSTCRx.SGST or PWDSGSTCR.PWDSGST, the SARAD113x model will process the corresponding scanning group. The corresponding state of SARAD113x will be controlled by Csarad113x_fsm, Csarad113x_tha_conv_fsm and Csarad113x_thb_conv_fsm classes.

(3) **External trigger:** This trigger feature uses TSN_TRG, SGx_TRG and PVCR_TRG input ports to trigger scanning process in SARAD113x model. . The corresponding state of SARAD113x will be controlled by Csarad113x_fsm, Csarad113x_tha_conv_fsm and Csarad113x_thb_conv_fsm classes same as software trigger.

8.2. State diagrams

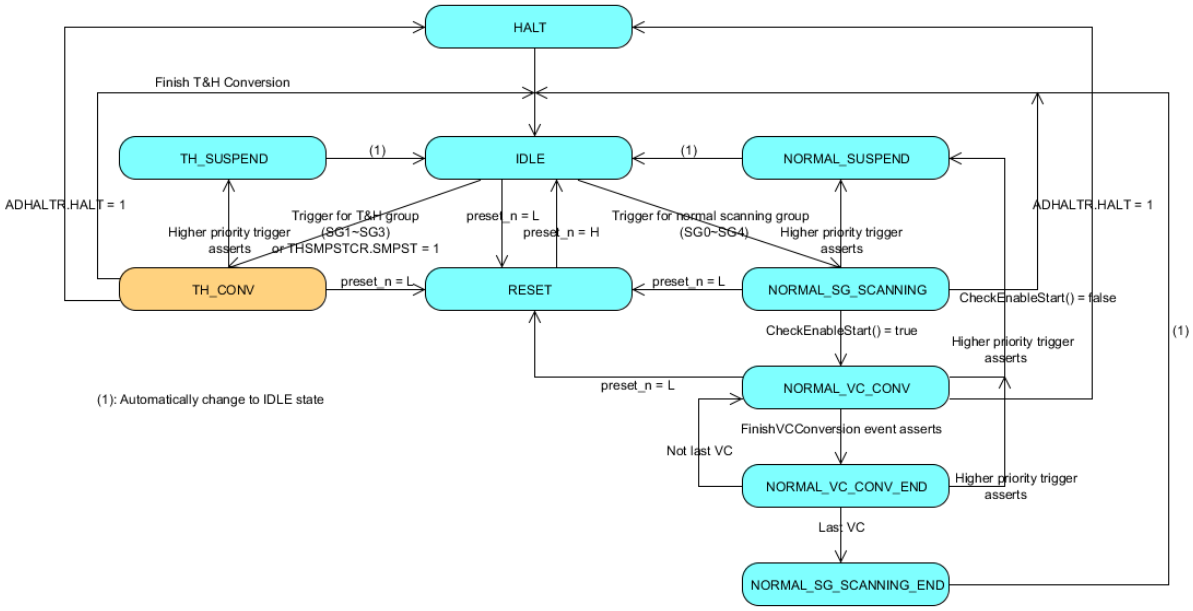

Figure 8.2: State diagram of SARAD113x model

Explanation:

(1) The detail explanation is described as the Table 8-2.

Table 8-2: State transition table of SARAD113x model.

State Name	Transition condition	Next Transition State
RESET	+ Deasserting preset_n or period time that is set by AssertReset command is over.	IDLE
IDLE (Initial state)	+ Asserting preset_n or AssertReset command is called.	RESET
	+ Trigger for T&H scanning group is asserted (set by THACR.SGS and THBCR.SGS) + THSMPSTCR.SMPST is set to 1	TH_CONV
	+ Trigger for normal scanning group is asserted	NORMAL_SG_SCANNING
HALT	+ Automatically change to IDLE state	IDLE
TH_CONV **(*)**	+ Asserting preset_n or AssertReset command is called.	RESET
	+ ADHALTR.HALT is set to 1	HALT
	+ T&H scanning group A and B finish A/D conversion process	IDLE
	+ Higher priority trigger is asserted	TH_SUSPEND
TH_SUSPEND	+ Automatically change to IDLE state	IDLE
NORMAL_SG_SCANNING	+ Automatically change to IDLE state if the setting for current scanning group is error or conflict	IDLE
	+ Automatically change to NORMAL_VC_CONV state if the setting for current scanning group is not error or conflict	NORMAL_VC_CONV

State Name	Transition condition	Next Transition State
NORMAL_SG_SCANNING_END	+ Automatically change to IDLE state	IDLE
NORMAL_VC_CONV	+ Asserting preset_n or AssertReset command is called.	RESET
	+ ADHALTR.HALT is set to 1	HALT
	+ Finish A/D conversion process of a virtual channel	NORMAL_VC_CONV_END
	+ Higher priority trigger is asserted	NORMAL_SUSPEND
NORMAL_VC_CONV_END	+ Automatically change to NORMAL_SUSPEND state if higher priority trigger is asserted and current VC is not last VC of current scanning group	NORMAL_SUSPEND
	+ Automatically change to NORMAL_SG_SCANNING_END state if current VC is last VC of current scanning group	NORMAL_SG_SCANNING_END
	+ Automatically change to NORMAL_VC_CONV state if there is no higher priority trigger and current VC is not last VC of current scanning group	NORMAL_VC_CONV
NORMAL_SUSPEND	+ Automatically change to IDLE state	IDLE

Note: (*) TH_CONV includes 2 sub state diagram THA_CONV and THB_CONV which is described as THz_CONV as below

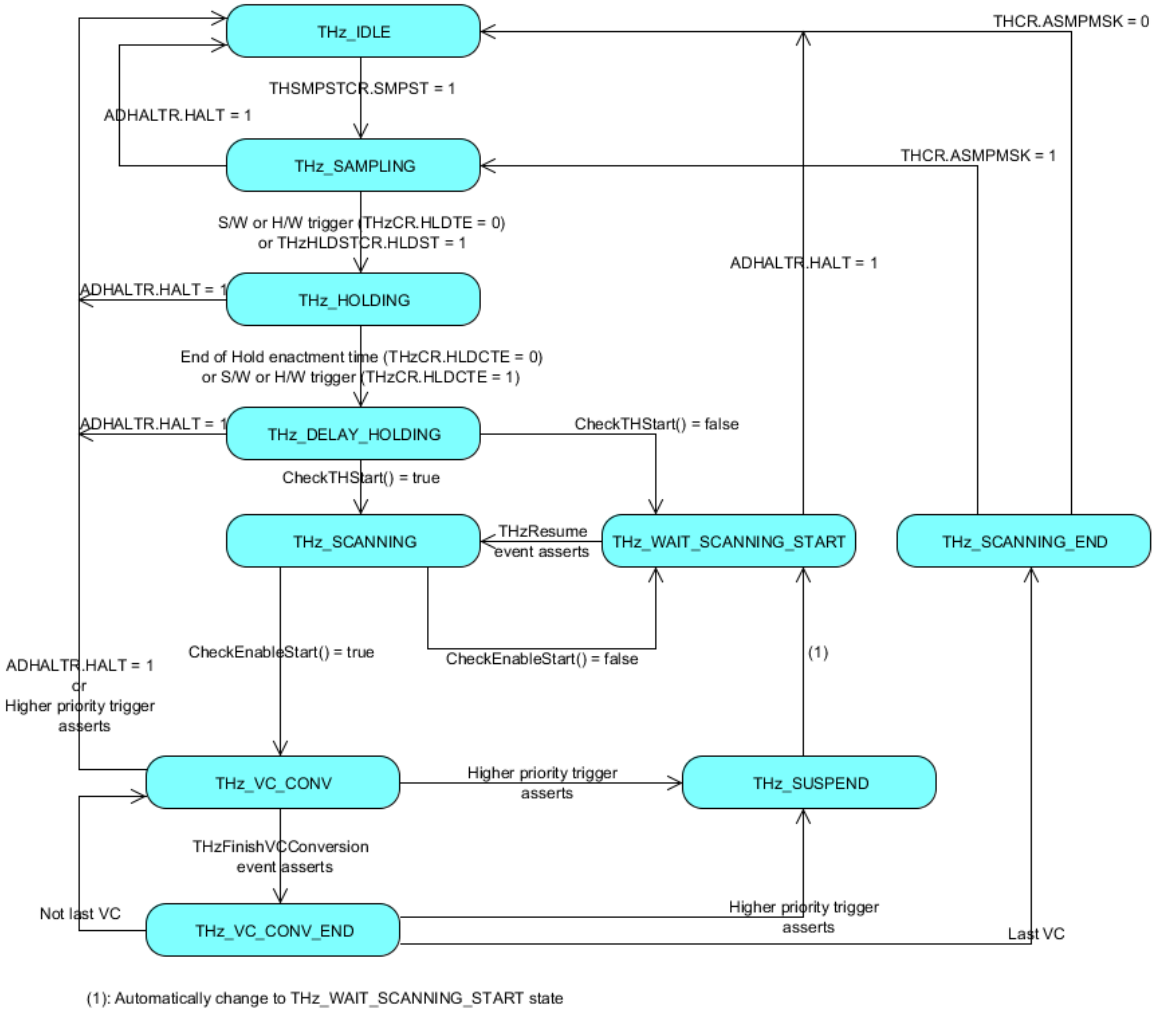

Figure 8.3: THz_CONV State diagram of SARAD113x model

Explanation:

(1) The detail explanation is described as the Table 8-3Table 8-2.

Table 8-3: THz_CONV State transition table of SARAD113x model.

State Name	Transition condition	Next Transition State
THz_IDLE (Initital state)	+ THSMPSTCR.SMPST is set to 1	THz_SAMPLING
THz_SAMPLING	+ ADHALTR.HALT is set to 1	THz_IDLE
	+ H/W or S/W trigger is asserted when THzCR.HLDTE = 0 + THzHLDSTCR.HLDST is set to 1	THz_HOLDING
THz_HOLDING	+ ADHALTR.HALT is set to 1	THz_IDLE
	+ H/W or S/W trigger is asserted when THzCR.HLDCTE = 1 + End of hold enactment time when THzCR.HLDCTE = 0	THz_DELAY_HOLDING

State Name	Transition condition	Next Transition State
	+ In T&H self-diagnosis mode, THzCR.HLDCTE = 1 and THzCR.HLDTE = 0, S/W trigger is asserted to enter THz_HOLDING state	
THz_DELAY_HOLDING	+ At the end of waiting 3ADCLK, if current T&H scanning group can start (suspend other operating scanning group or no scanning group is operating)	THz_SCANNING
	+ At the end of waiting 3ADCLK, if current T&H scanning group can not start (can not suspend other operating scanning group)	THz_WAIT_SCANNING_START
THz_SCANNING	+ Automatically change to THz_IDLE state if the setting for current scanning group is error or conflict	THz_IDLE
	+ Automatically change to THz_VC_CONV state if the setting for current scanning group is not error or conflict	THz_VC_CONV
THz_SCANNING_END	+ Automatically change to THz_IDLE state if THCR.ASMPMSK is equal to 0	THz_IDLE
	+ Automatically change to THz_SAMPLING state if THCR.ASMPMSK is equal to 1	THz_SAMPLING
THz_VC_CONV	+ ADHALTR.HALT is set to 1	THz_IDLE
	+ Higher priority trigger is asserted	THz_SUSPEND
	+ Finish A/D conversion process of a virtual channel	THz_VC_CONV_END
THz_VC_CONV_END	+ Automatically change to THz_SUSPEND state if higher priority trigger is asserted and current VC is not last VC of current scanning group	THz_SUSPEND
	+ Automatically change to THz_SCANNING_END state if current VC is last VC of current scanning group	THz_VC_CONV_END
	+ Automatically change to THz_VC_CONV state if there is no higher priority trigger and current VC is not last VC of current scanning group	THz_VC_CONV
THz_SUSPEND	+ Automatically change to THz_WAIT_SCANNING_START state	THz_WAIT_SCANNING_START
THz_WAIT_SCANNING_START	+ ADHALTR.HALT is set to 1	THz_IDLE
	+ All of other higher priority scanning group finish A/D conversion process	THz_SCANNING

8.3. preset_n input port flow

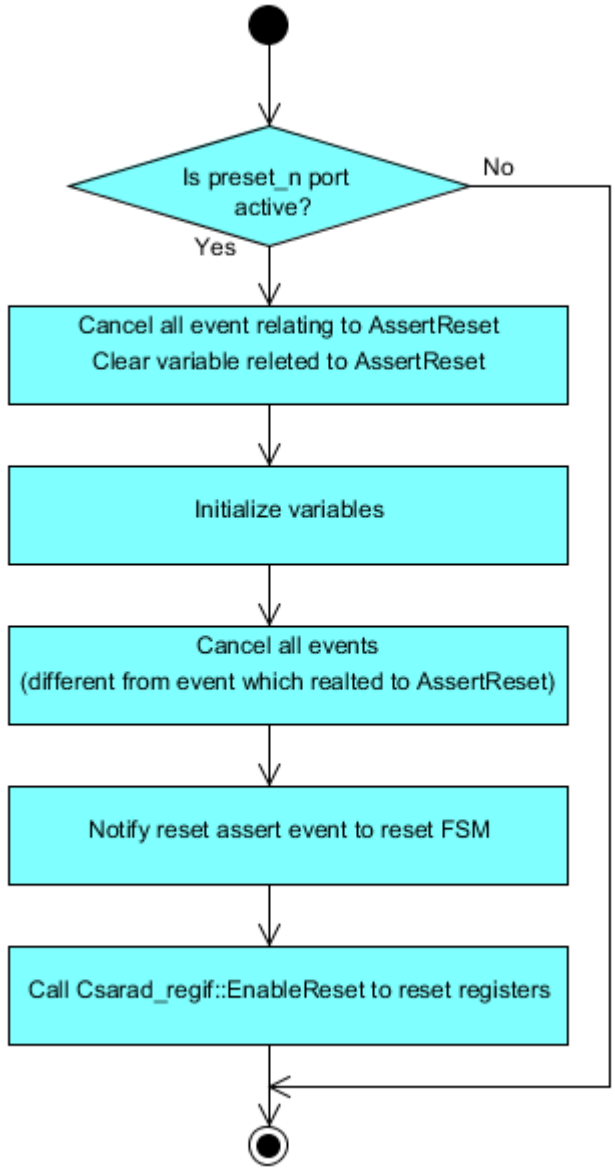

Figure 8.4: preset_n input port flow

Explanation:

(1) The reset function is implemented base on the behavior of following functions and methods:

(1.1) ResetMethod method, AssertReset function, AssertResetMethod method, DeAssertResetMethod method, EnableReset functions of Csarad113x class.

(1.2) EnableReset function of Csarad113x class.

(2) If preset_n port is active low - ResetMethod is notified, all events and variables which are relative to AssertReset are canceled and cleared because the preset_n port has higher

priority.

(3) Next, the internal variables, registers, output ports are initialized by calling Csarad:: EnableReset function. However, the parameters set by commandIF are not cleared.

(4) The ResetAssert event is notified to reset the FSM of SARAD113x model.

8.4. AssertReset function flow

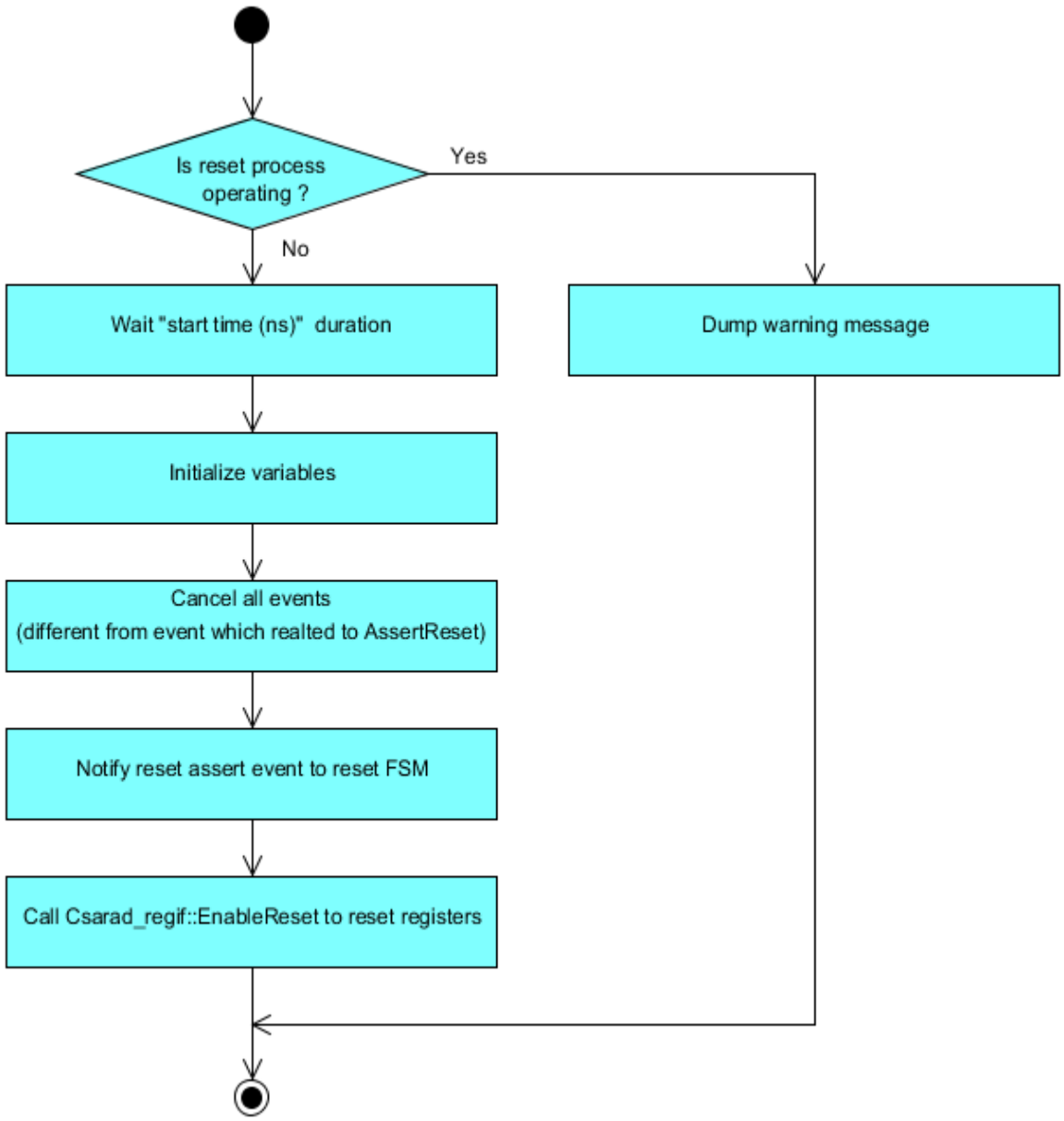

Figure 8.5: AssertReset function flow

Explanation:

(1) The reset function is implemented base on the behavior of following functions and methods:

(1.1) ResetMethod method, AssertReset function, AssertResetMethod method, DeAssertResetMethod method, EnableReset functions of Csarad113x class.

(1.2) EnableReset function of Csarad113x class.

(2) If the reset function is active by calling the AssertReset function, the reset process is check that it is operating or not (the reset process is operating when reset input port is active low or AssertReset function is called before).

(3) If the reset process is operating, the AssertReset function is ignored and warning message is dumped. If the reset process is not operating, the Csarad113x::EnableReset will be called after start_time duration to reset SARAD113x model.

(4) After start_time duration, the internal variables, registers, output ports are initialized. However, the parameters set by commandIF are not cleared.

8.5. CheckTrigger function processing flow

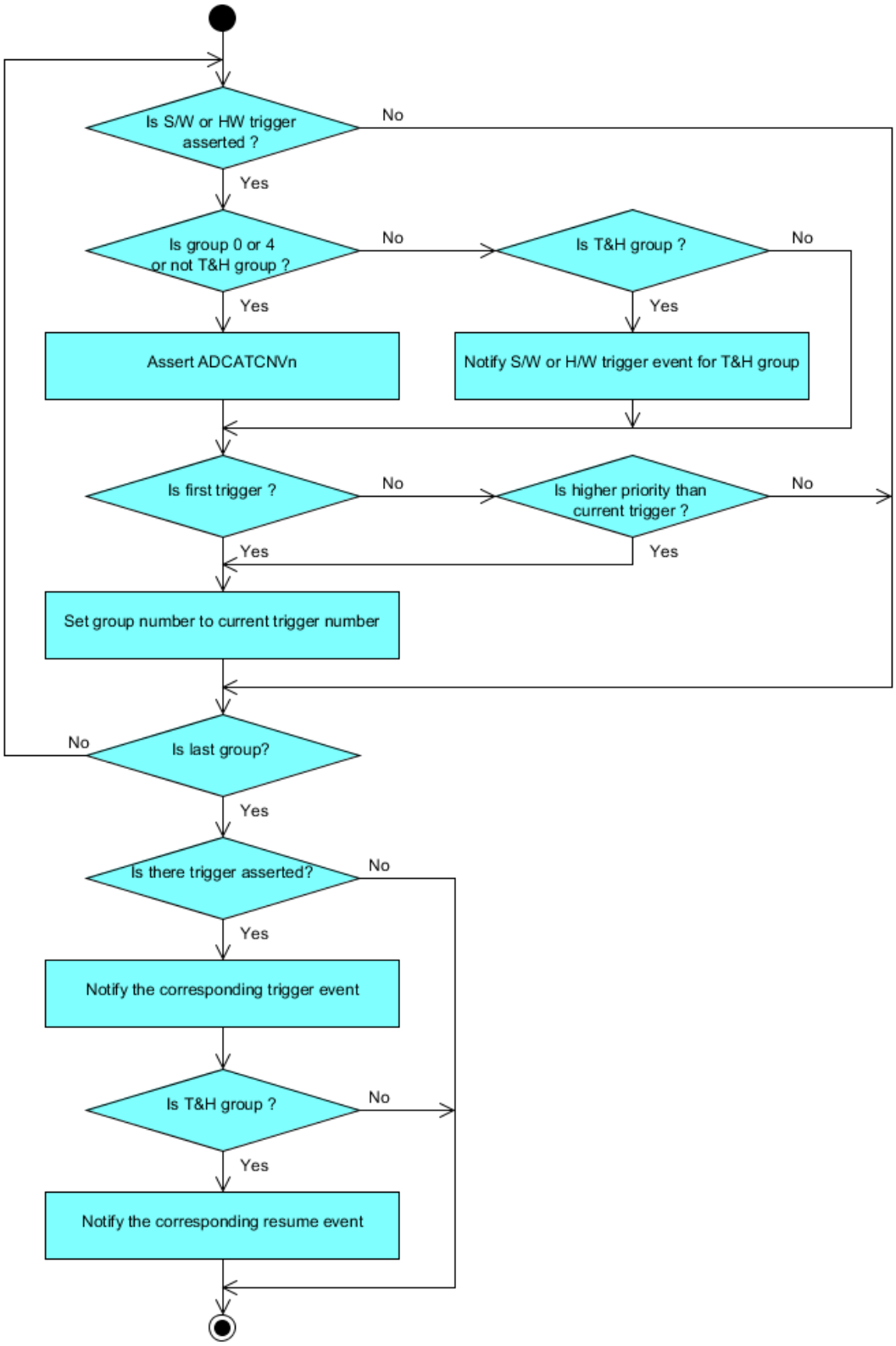

Figure 8.6: CheckTrigger function processing flow

Explanation:

(1) Refer to HWM[2] chapter: 1.10.17, 1.10.18, 1.11.14, 1.16.1.2, 1.16.1.3, 1.16.2, 1.16.3, 1.16.12, 1.16.15

(2) The CheckTrigger function is called when H/W or S/W trigger is asserted or SARAD113x model enters to the IDLE state.

(3) The SARAD113x model will check the H/W and S/W trigger of group 0 to 4. If is there trigger of normal scanning group the corresponding output port ADCATCNVn will be asserted. If the trigger which is relating to T&H scanning group decided by THACR.SGS and THBCR.SGS is asserted, the SARAD113x notify the corresponding T&H trigger event to T&H group.

(4) The highest priority trigger will be selected for A/D conversion process by triggering its trigger event.

(5) If the highest priority trigger is relating to T&H scanning group it will notify the THzResume event if it is in THz_WAIT_SCANNING_START state.

8.6. Start scanning processing flow

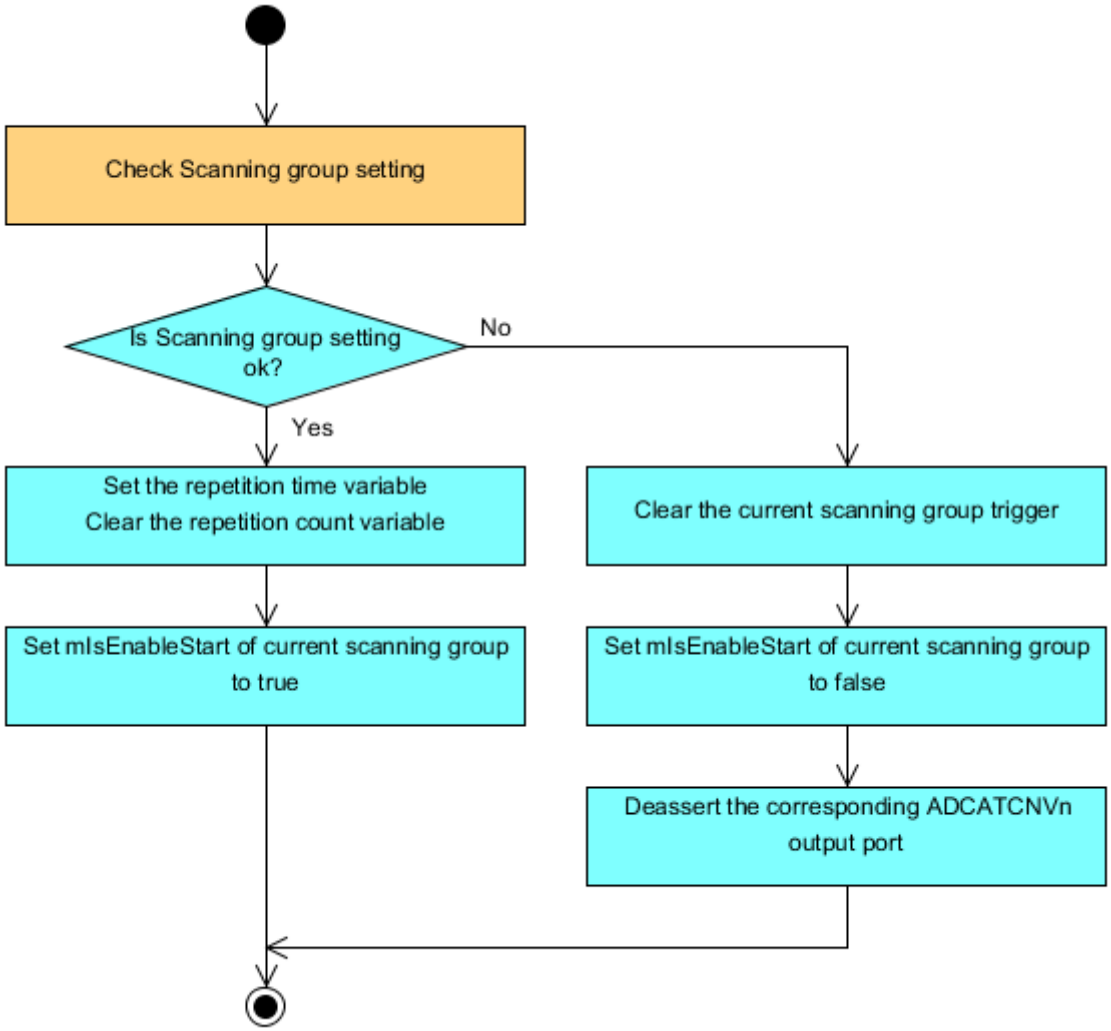

Figure 8.7: Start scanning processing flow

Explanation:

(1) Refer to HWM[2] chapter: 1.16.2, 1.16.3, 1.16.4, 1.16.15

(2) The SARAD113x calls the function CheckSGSetting to check the setting of current scanning group before start the A/D conversion process. If the setting is not error or conflict, the SARAD113x model calculate the repetition time and reset the repetition count variable for current virtual channel. After that, the mIsEnableStart variable of current scanning group is set to true to start the A/D conversion process.

(3) If there is any error or conflict, the SARAD113x model clear H/W and S/W trigger and set mIsEnableStart variable of current scanning group to false to ignore the current trigger. The corresponding output port ADCATCNVn will be deasserted.

8.7. Check scanning group setting processing flow

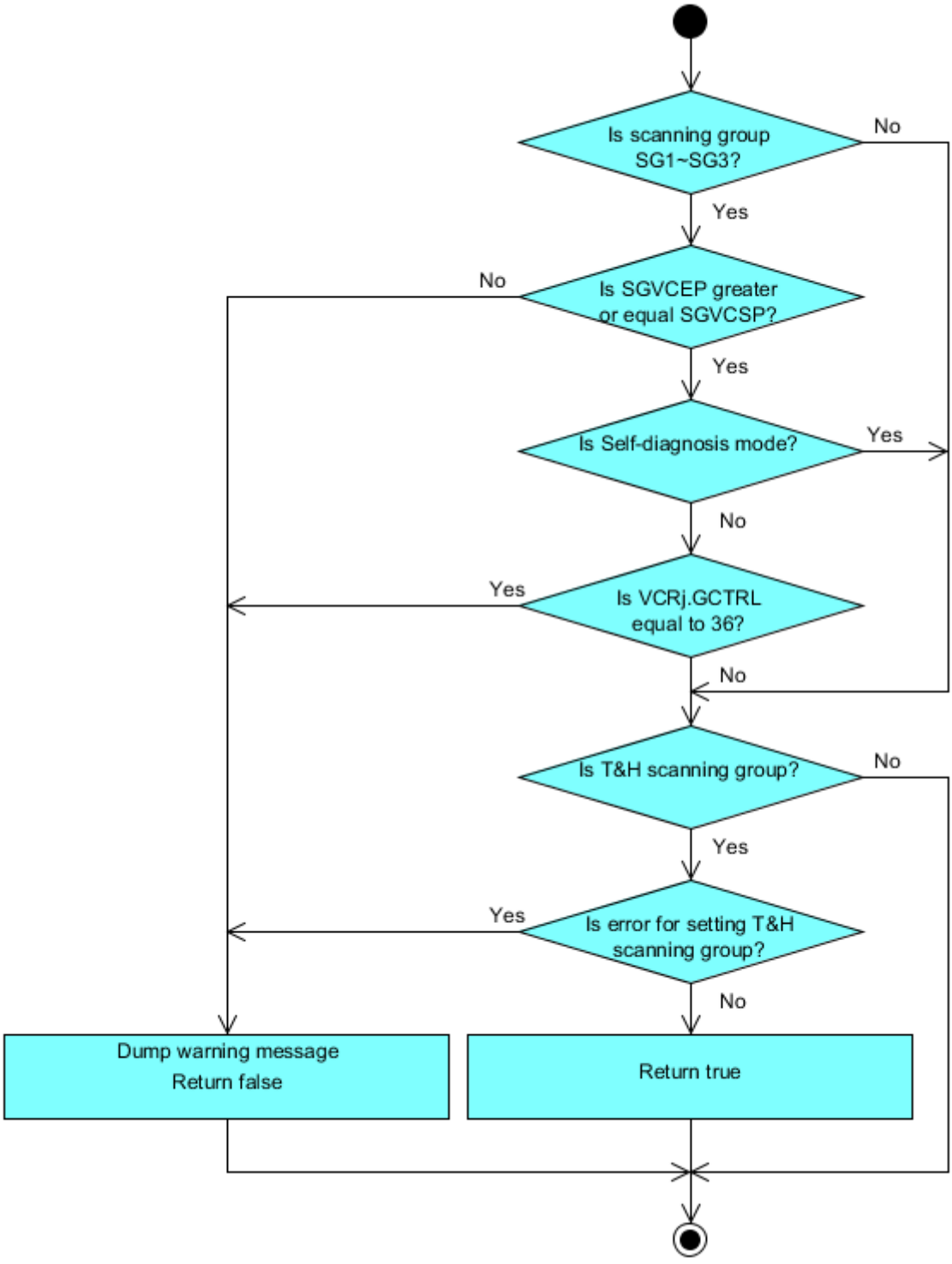

Figure 8.8: Check scanning group setting processing flow

Explanation:

(1) Refer to HWM[2] chapter: 1.3.1.13, 1.10.1, 1.10.2, 1.10.3, 1.10.4, 1.11.7, 1.11.8, 1.16.3.1

(2) If current scanning group is SG1~SG3, SARAD113x check whether the setting of SGVCEPx is equal or greater than SGVCSPx or not. If not, a warning message is dumped and false value is returned. In case self-diagnosis mode is disabled, if the GCTRL value of virtual channel is set to 36 (self-diagnosis channel), warning message is also dumped and false value is returned.

(3) If current scanning group is T&H scanning group, the false value will be returned if the following error or conflict occur:

(3.1) MPX function is enabled for virtual channel

(3.2) T&H channel is used for both T&H group A and B

(3.3) Same physical channel is used for both T&H group A and B

(3.4) T&H channel of T&H group A is used for group B and vice versa

(3.5) T&H channels 0-2 is same group with T&H channels 3-5

(3.6) Continuous scanning mode is used for T&H group A or B

(3.7) Multi cycle scanning mode is used for T&H group A or B and the scanning frequency is greater than 1

(4) If there is no error or conflict, this function returns true value.

8.8. Start virtual channel scanning processing flow

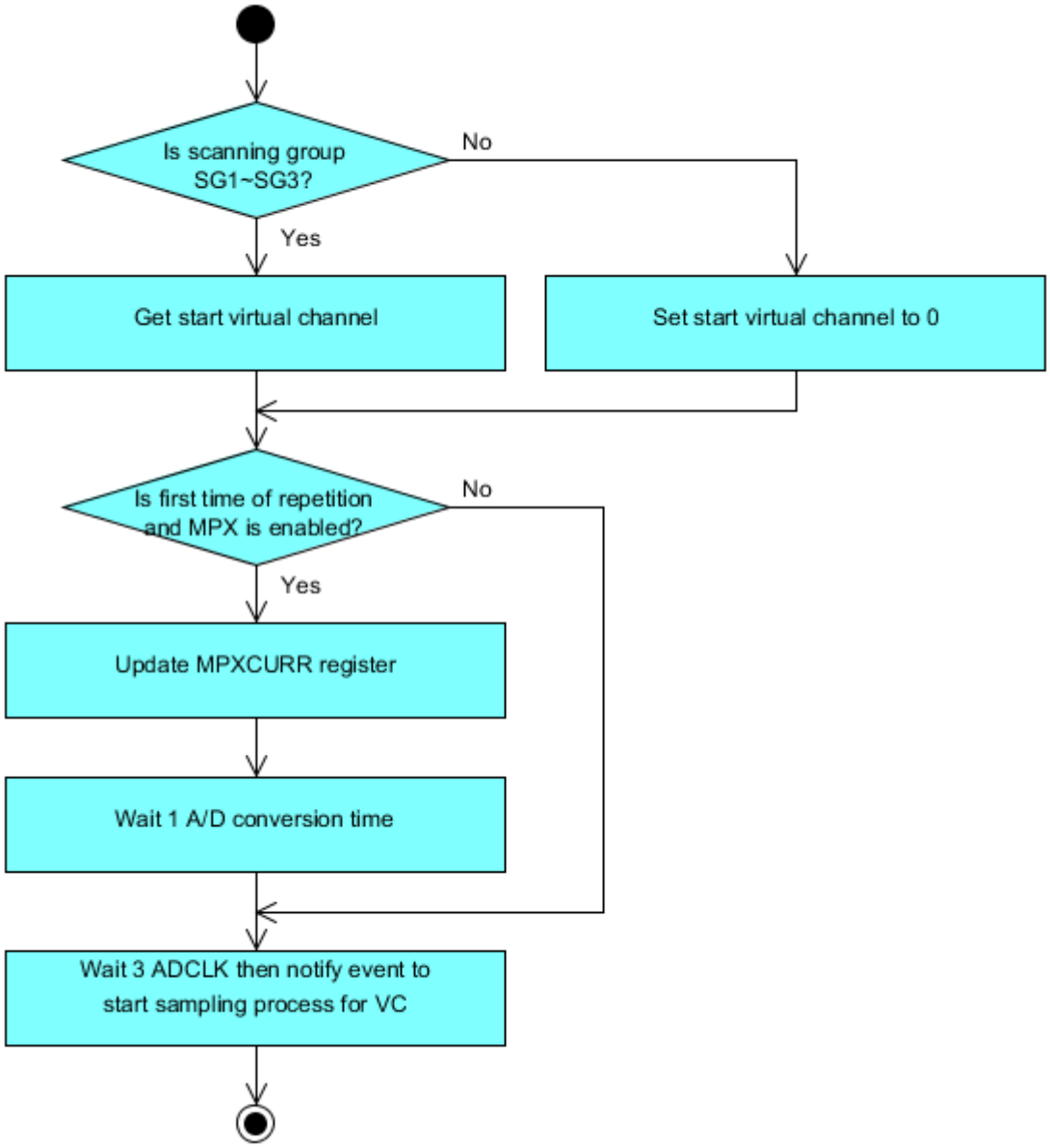

Figure 8.9: Start virtual channel scanning processing flow

Explanation:

(1) Refer to HWM[2] chapter: 1.3.1.13, 1.10.1, 1.10.2, 1.10.3, 1.10.4, 1.11.7, 1.11.8, 1.16.3.1

(2) The SARAD113x model will check the scanning group to get the start virtual channel before start the A/D conversion process. If the scanning group is SG1~SG3, the start virtual channel will be SGVCSPx value if scanning group start A/D conversion, if not the start virtual channel is continued from previous virtual channel. If the scanning group is TSN or PWD, the virtual channel is 0 (don't care because they have special virtual channel).

(3) In case MPX function is enabled for current virtual channel, the register MPXCURR will be updated following the corresponding MPXV value. If this is the first time of repetition time, the SARAD113x model will wait 1 A/D conversion time (sampling time + converting time) before starting the A/D conversion for virtual channel.

(4) After wait time if enabled, the SARAD113x model will notify the event after 3 ADCLK periods to start the sampling for A/D conversion process of current virtual channel.

8.9. Virtual channel sampling processing flow

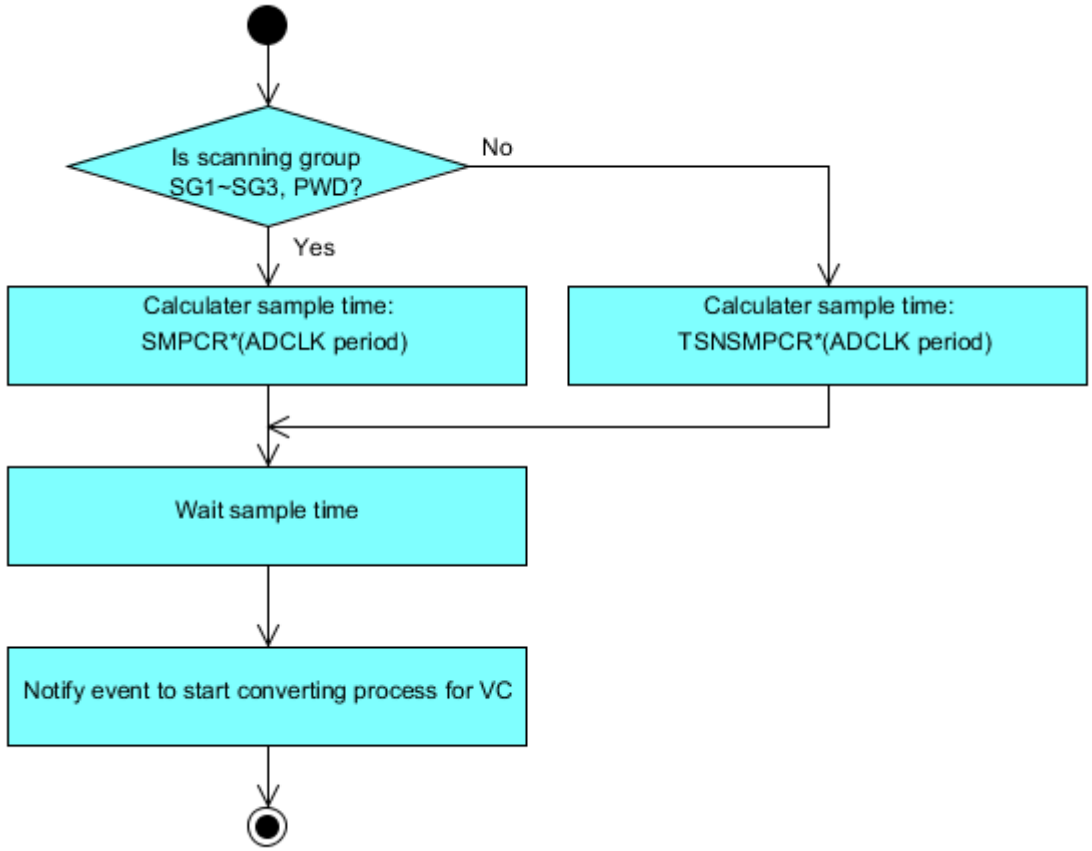

Figure 8.10: Virtual channel sampling processing flow

Explanation:

(1) Refer to HWM[2] chapter: 1.16.1.3, 1.16.2, 1.16.3, 1.16.4, 1.16.5

(2) The SARAD113x model will check the scanning group to calculate the sample time. If the scanning group is SG1~SG3 or PWD, the sample time is equal to the value of SMPCR register multiply with ADCLK period. If the scanning group is TSN, the sample time is equal to the value of TSNSMPCR register multiply with ADCLK period.

(3) The SARAD113x models waits for the sample time then notify the event to start the conversion process.

8.10. Virtual channel conversion processing flow

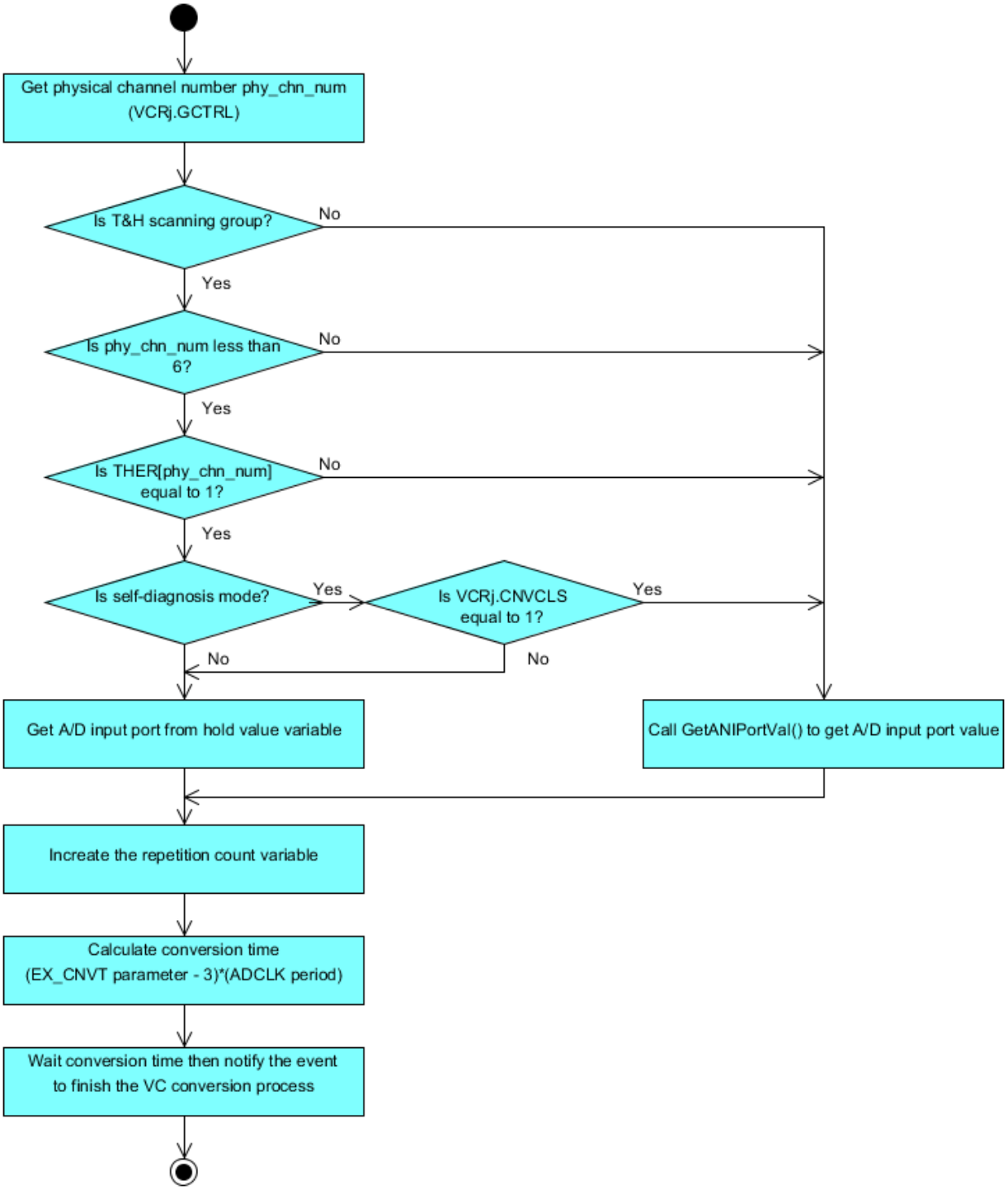

Figure 8.11: Virtual channel conversion processing flow

Explanation:

(1) Refer to HWM[2] chapter: 1.16.1.3, 1.16.2, 1.16.3, 1.16.4, 1.16.5

(2) At first, the SARAD113x model will get the physical channel number of current virtual channel.

(3) If the current scanning group is not T&H scanning group, the SARAD113x model calls the function GetANIPortVal() to get the corresponding input port value.

(4) If the current scanning group is T&H scanning group, the SARAD113x model calls the function GetANIPortVal() to get the corresponding input port value in case the physical channel is not T&H channel or it is disabled by setting corresponding bit of register THER to 0. In self-diagnosis mode, if VCRj.CNVCLS (j = 33~35) equal to 1, the SARAD113x model also calls the function GetANIPortVal() to get the corresponding input port value. Otherwise the value of corresponding hold port value variable will be get for A/D conversion process. The hold port value variable is updated by hold process of T&H channel.

(5) After input value is get, the SARAD113x model calculates the conversion time. The conversion time equal to number of ADCLK period of EX_CNVT parameter (exclude 3 ADCLK periods which is delayed at the beginning of A/D conversion process).

(6) After wait conversion time, the SARAD113x model notifies event to finish current VC conversion process.

8.11. GetANIPortVal function processing flow

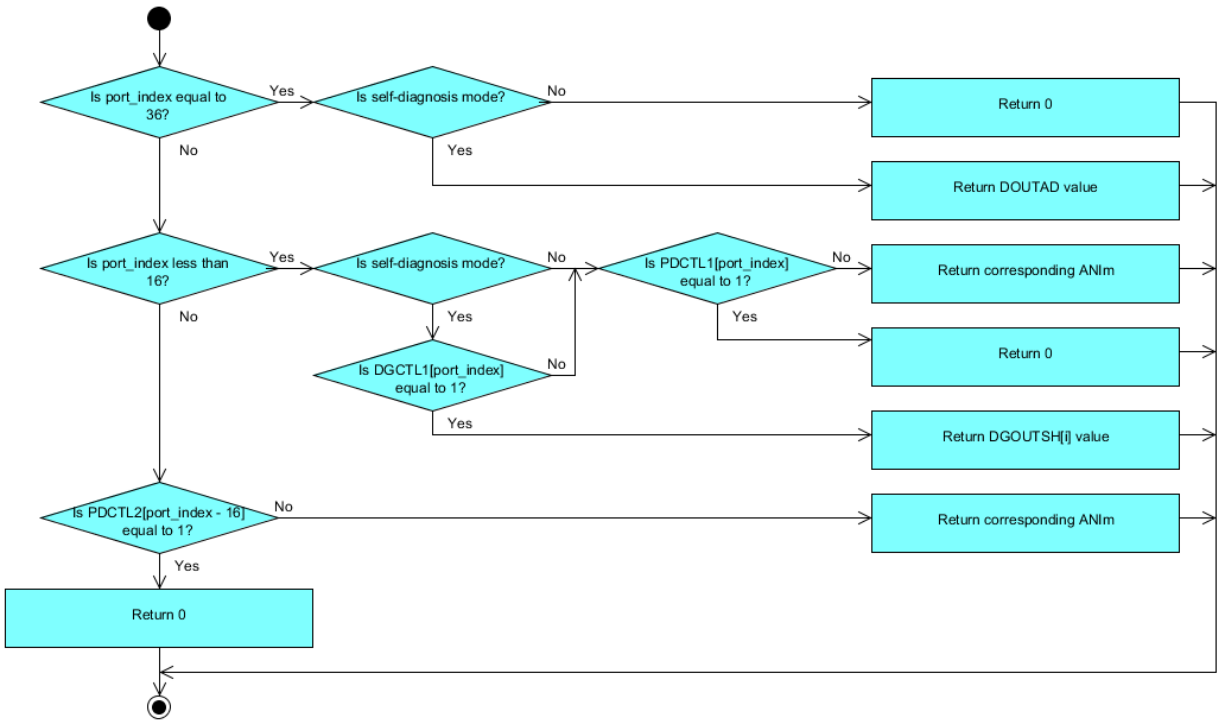

Figure 8.12: GetANIPortVal function processing flow

Explanation:

(1) Refer to HWM[2] chapter: 1.10.1, 1.10.2, 1.10.3, 1.10.4, 1.10.11, 1.14.1, 1.14.2, 1.14.3, 1.14.4, 1.16.18

(2) The port_index variable indicates the physical channel number which is defined by VCRj.GCTRL value.

(3) In case the port_index is equal to 36 (self-diagnosis channel), if self-diagnosis function is disabled, the function GetANIPortVal returns 0. If self-diagnosis is enabled, the value of DOUTAD value is returned.

(4) In case port_index is less than 16 (ANI group 1), if self-diagnosis function is enabled and corresponding bit of DGCTL1 is set to 1, the corresponding value of DGOUTSH is returned (refer to). Otherwise, this function will check whether the Pull-down function is set or not by corresponding bit of PDCTL1 register. If Pull-down function is enabled for physical channel, the value 0 is returned. If not the corresponding ANIm (m = port_index) value is returned.

(5) In case port_index is equal or greater than 16 (ANI group 2), if Pull-down function is enabled for physical channel, the value 0 is returned. If not the corresponding ANIm (m = port_index) value is returned. The Pull-down function for ANI group 2 ports is enabled by register PDCTL2.

(6) The following table is the value of DOUTAD, DGOUTSH corresponding to setting of DGCTL0 register. In case of “Hi-z” value, the value 0 is returned.

Table 8-4: Self-diagnosis voltage level selection

DGCTL0			**Output signal**			
PSEL2	**PSEL1**	**PSEL0**	**DGOUTAD**	**DGOUTSH2**	**DGOUTSH1**	**DGOUTSH0**
0	0	0	Hi-z	Hi-z	Hi-z	Hi-z
0	0	1	AVSS	2/3AVCC	1/2AVCC	1/3AVCC
0	1	0	1/3AVCC	1/3AVCC	2/3AVCC	1/2AVCC
0	1	1	1/2AVCC	1/2AVCC	1/3AVCC	2/3AVCC
1	0	0	2/3AVCC	Hi-z	Hi-z	Hi-z
1	0	1	AVCC	1/3AVCC	1/3AVCC	1/3AVCC
1	1	0	AVCC	1/2AVCC	1/2AVCC	1/2AVCC
1	1	1	AVCC	2/3AVCC	2/3AVCC	2/3AVCC
Corresponding physical channel			-	**2,5,8,11,14**	**1,4,7,10.13**	**0,3,6,9,12,15**

8.12. Finish virtual channel conversion processing flow

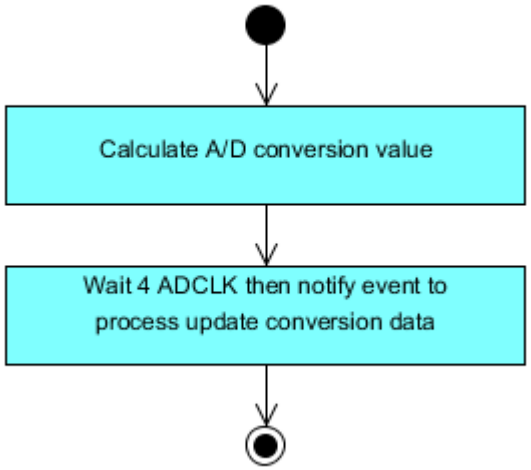

Figure 8.13: Finish virtual channel conversion processing flow

Explanation:

(1) Refer to HWM[2] chapter: 1.16.1.3, 1.16.2, 1.16.3, 1.16.4, 1.16.5

(2) The SARAD113x calls ADConvert() function to calculate the A/D conversion value of current virtual channel. After that it waits 4 ADCLK then notifies the event to start the update conversion data process.

8.13. ADConvert function processing flow

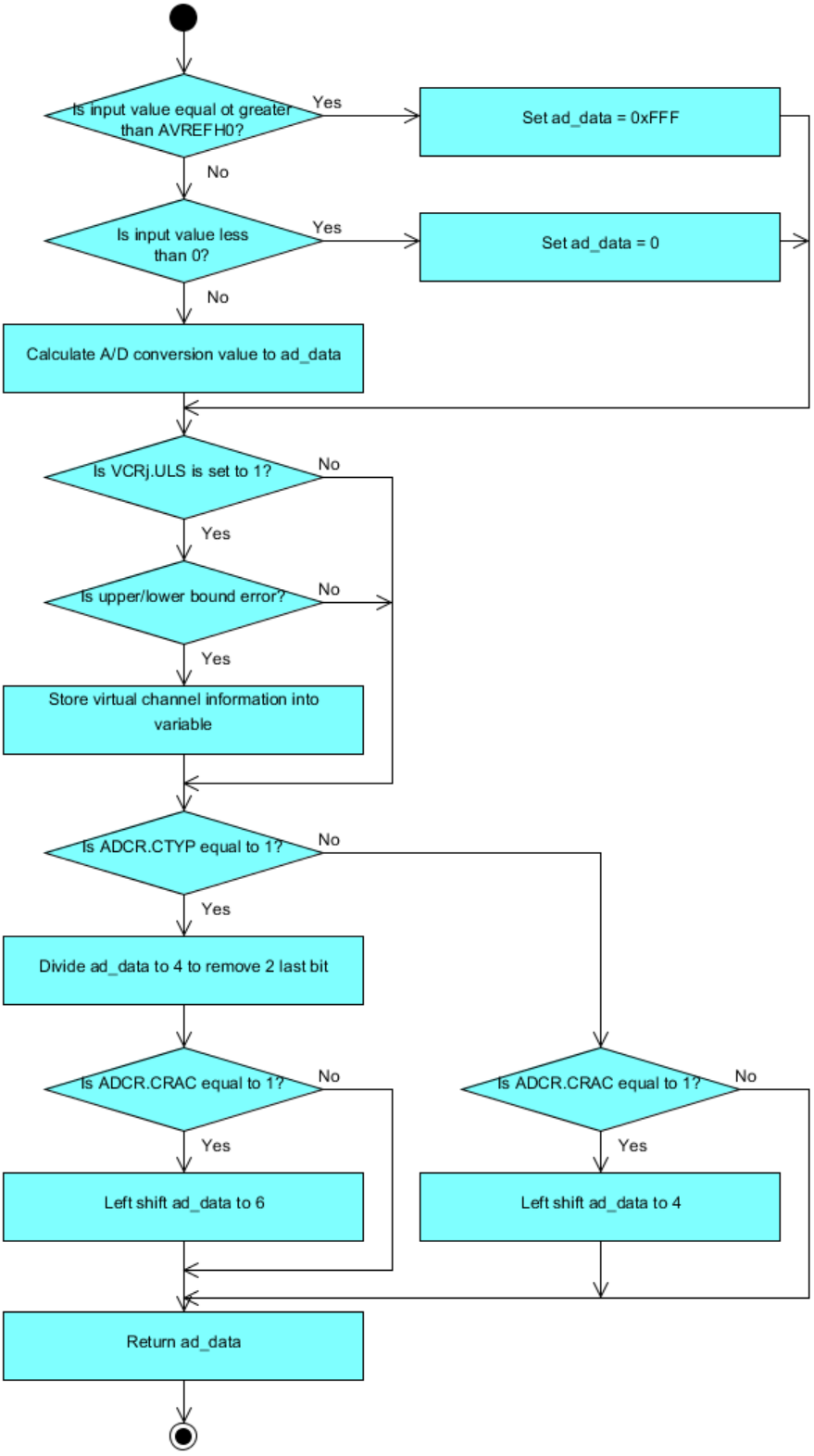

Figure 8.14: ADConvert function processing flow

Explanation:

(1) Refer to HWM[2] chapter: 1.10.1, 1.10.2, 1.10.3, 1.10.4, 1.10.11, 1.10.22, 1.10.24, 1.16.8

(2) The ADConvert function will check the input analog value before start the A/D conversion. If the input value is equal or greater than AVREFH0 value, the ad_data is set to maximum value (0xFFF). If the input value is less than 0, the ad_data is set to 0. Otherwise, the input value will be calculated by formula ad_data = (value*(2^12))/AVREFH0.

(3) ADConvert function will check the converted value for upper/lower bound error if VCRj.ULS is set to 1. The error virtual channel information will be stored into variable.

(4) In case 10bit conversion mode is selected (ADCR.CTYP = 1), the converted data is divided to 4 to remove 2 last bits. If left align mode is selected (ADCR.CRAC = 1), the ad_data is left shifted to 6 for alignment.

(5) In case 12bit conversion mode is selected (ADCR.CTYP = 0) and left align mode is selected (ADCR.CRAC = 1), the ad_data is left shifted to 4 for alignment.

(6) The ad_data is returned at the end of the process.

8.14. Update conversion data processing flow

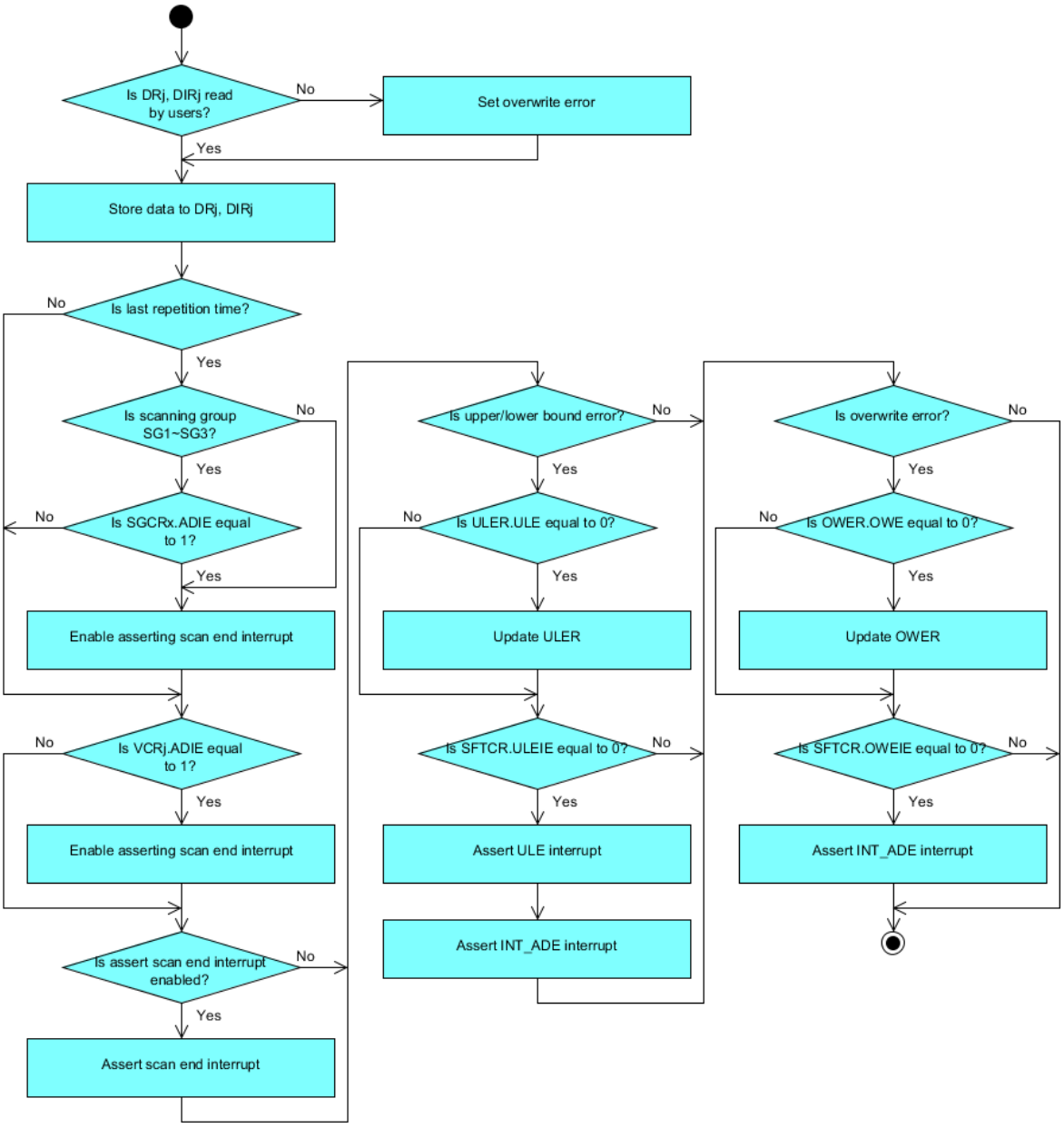

Figure 8.15: Update conversion data processing flow

Explanation:

(1) Refer to HWM[2] chapter: 1.10.1~1.10.9, 1.10.21, 1.10.24, 1.10.25, 1.16.8, 1.16.9, 1.16.16, 1.16.17

(2) The SARAD113x will check whether the DRj, DIRj is read by users or not. If previous value is not read, the overwrite error will occur. The conversion data and virtual channel information is stored into corresponding DRj and DIRj.

(3) At the last time of repetition time, the SARAD113x model will assert corresponding scan end interrupt if scanning group is TSN or PWD. In case scanning group is SG1~SG3, the scan interrupt is asserted if SGCRx.ADIE is set to 1.

(4) When finishing A/D conversion of a virtual channel, if the VCRj.ADIE is set to 1, the scan end interrupt is asserted too.

(5) In case there is an upper/lower bound error, the ULER is updated with information of current virtual channel if ULER.ULE is equal to 0 (previous error is cleared). If the SFTCR.ULEIE is set to 1, the ULE interrupt and INT_ADE interrupt is asserted.

(6) In case there is an overwrite error, the OWER is updated with information of current virtual channel if OWER.OWE is equal to 0 (previous error is cleared). If the SFTCR.OWEIE is set to 1, the INT_ADE interrupt is asserted.

8.15. IsLastVC function processing flow

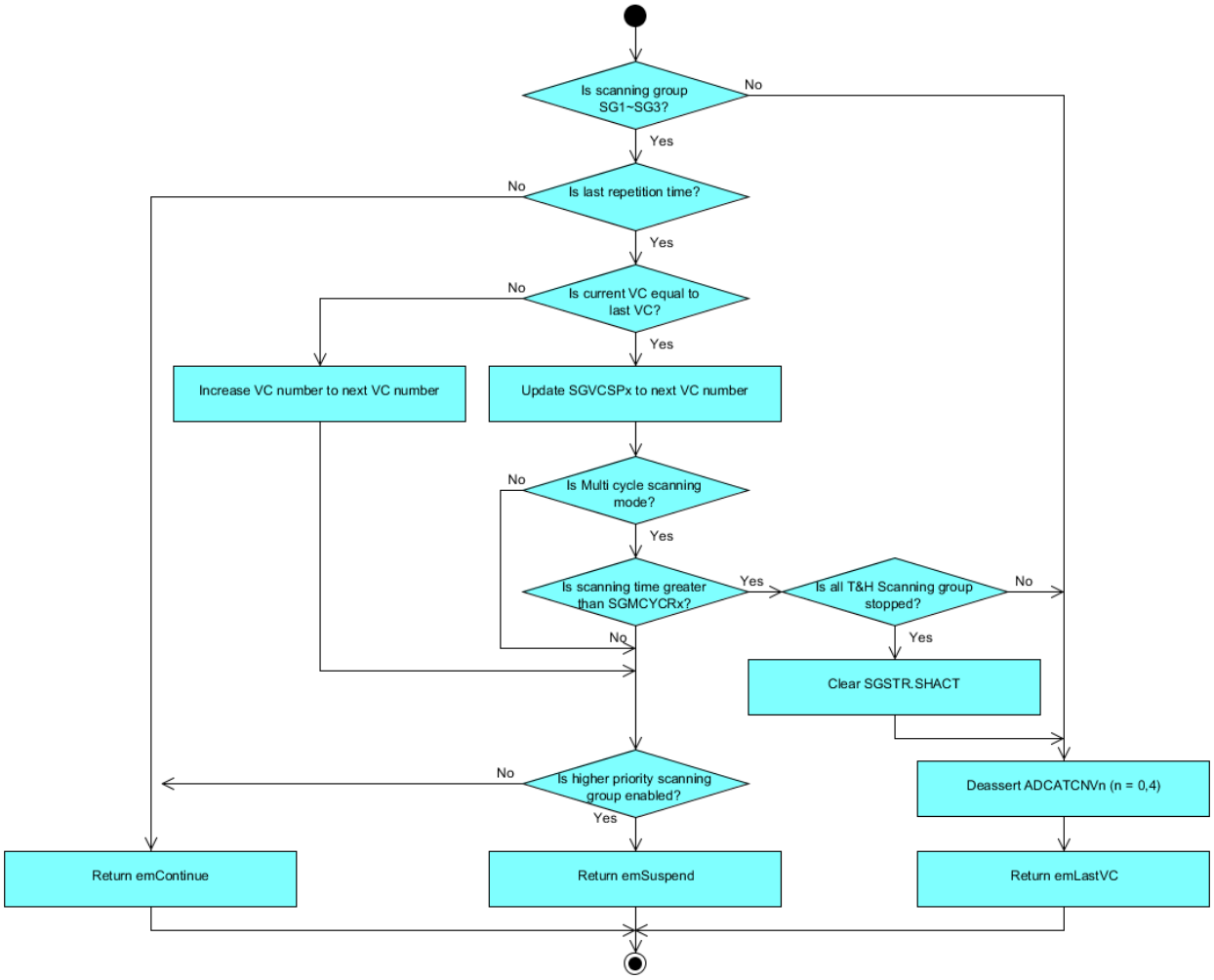

Figure 8.16: IsLastVC function processing flow

Explanation:

(1) Refer to HWM[2] chapter: 1.10.11, 1.11.4, 1.16.2, 1.16.3, 1.16.4.1, 1.16.5, 1.16.6, 1.16.15

(2) The IsLastVC function is used to check whether scanning group is stopped or suspended or continued when A/D conversion process for virtual channel is finished.

(3) If scanning group is TSN or PWD, the emLastVC value is returned to indicate that the scanning group TSN or PWD finish the A/D conversion and it will be stopped. The corresponding ADCATCNVn is deasserted.

(4) In case the scanning group is SG1~SG3, this function will continue if it is not last repetition time of A/D conversion of virtual channel.

(5) If current VC is not last VC, the VC index is increased for next A/D conversion.

(6) If current VC is last VC, the value of SGVCSPx is set for next A/D conversion. If the scanning mode is Multi cycle mode and the scanning time greater than value of SGMCYCRx. The current scanning group will be stopped, emLastVC value will be returned.

The corresponding ADCATCNVn is deasserted. If all of T&H scanning group is stopped, the SGSTR.SHACT bit is cleared to 0.

(7) If there is higher priority scanning group is enabled, the IsLastVC function return the emSuspend value to indicate that current scanning group is suspend.

8.16. StopOperation function processing flow

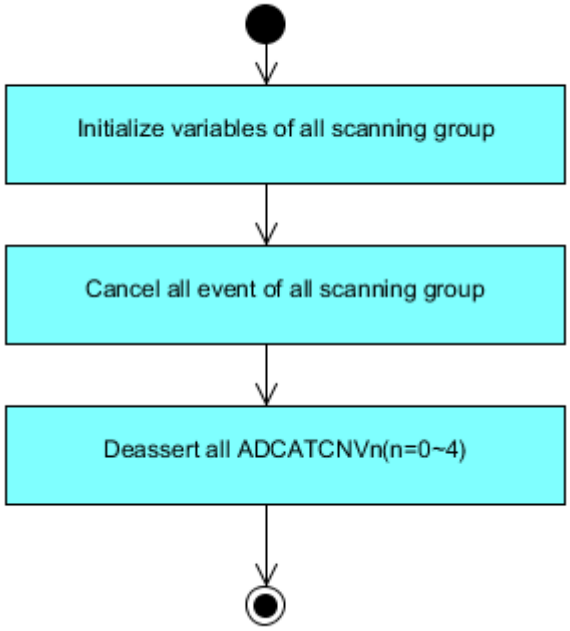

Figure 8.17: StopOperation function processing flow

Explanation:

(1) Refer to HWM[2] chapter: 1.10.10, 1.16.14

(2) This function is called when ADHALT.HALT is set to 1. When this function is called, all variables relating to all scanning groups are initialized. All events of scanning groups are canceled. And SARAD113x model deassertes all ADCATCNVn output port.

8.17. Registers access condition

(1) Refer to HWM[2] chapter: 1.3.1.5, 1.10 to 1.14

(2) The registers access condition is described in detail as following table.

Table 8-5: Register access condition

Register Group	Conditions
VCRj TSNVCR ADCR THCR THGSR SFTCR	+ SGSTR.SGACT = 0 + All SGCRx.TRGMD = 0 + PWDSGCR.PWDTRGMD = 0 + TSNSGCR.TSNTRGMD = 0

ULLMTBRx SMPCR TSNSMPCR EMUCR SGCRx TSNSGCR PWDSGCR	
DGCTL1 PDCTL1 PDCTL2 SGMCYCRx SGTSELx	+ SGSTR.SGACT [3:1] = 0 + All SGCRx.TRGMD = 0
TSNCR	+ SGSTR.SGACT [0] = 0 + TSNSGCR.TSNTRGMD = 0
THACR THBCR THER	+ SGSTR.SGACT = 0 + All SGCRx.TRGMD = 0 + PWDSGCR.PWDTRGMD = 0 + TSNSGCR.TSNTRGMD = 0 + THzCR.HLDTE, THzCR.HLDCTE and THER can be cleared during operation
Others	No condition

8.18. handleCommand operation processing flow

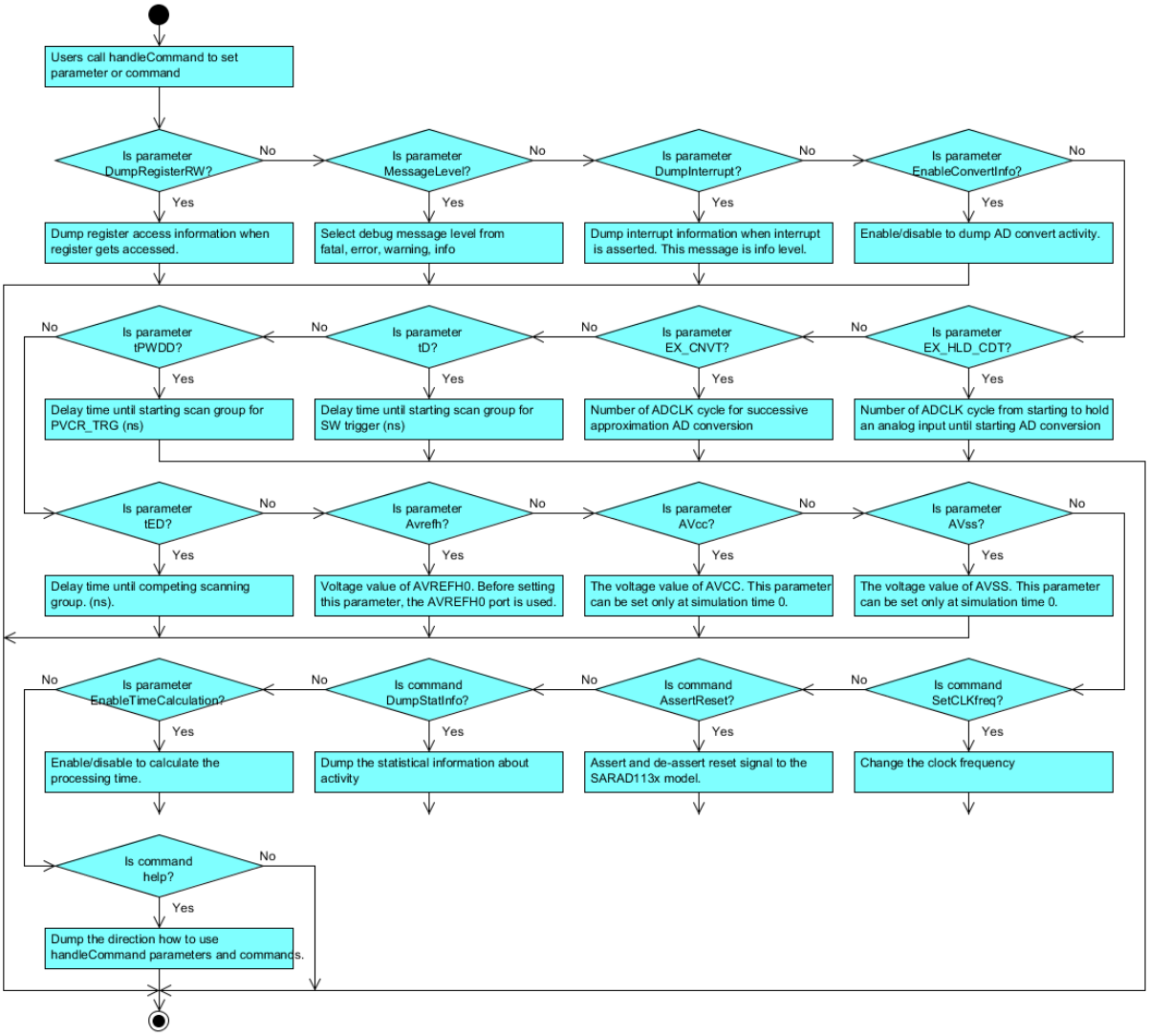

Figure 8.18: handleCommand operation processing flow

Explanation:

(1) Users call handleCommand to set parameter or command. The parameters are referred to Table 7-2 and the commands are referred to Table 7-3.

(2) If parameter or command is:

(2.1) DumpRegisterRW, it enables the SARAD113x model to dump register access information when register gets accessed.

(2.2) MessageLevel, it selects debug message level of the SARAD113x model.

(2.3) DumpInterrupt, it enables dumping interrupt information when interrupt is asserted.

(2.4) Avrefh, it sets the reference voltage of AVREFH0 for SARAD113x model.

(2.5) AVcc, it sets the reference voltage of AVCC for SARAD113x model.

(2.6) AVss, it sets the reference voltage of AVSS for SARAD113x model.

(2.7) EnableConvertInfo, it enables to dump AD convert activity.

(2.8) EX_HLD_CDT, it sets number of ADCLK cycle from starting to hold an analog input until starting AD conversion.

(2.9) EX_CNVT, it sets the number of ADCLK cycle for successive approximation AD conversion.

(2.10)tD, it sets delay time until starting scan group for SW trigger (ns)

(2.11)tPWDD, it sets delay time until starting scan group for PVCR_TRG (ns)

(2.12)tED, it sets delay time until competing scanning group (ns).

(2.13)EnableTimeCalculation, it enables using formula to calculate the delay time.

(2.14)DumpStatInfo, it dumps information about the SARAD113x model. When this command is called, the statistical information about transfer activity is dumped.

(2.15)AssertReset, it resets the SARAD113x model.

(2.16)SetCLKfreq, it changes the clock frequency.

(2.17)help, it dumps the direction how to use handleCommand parameters and commands.

9. Class explanation

9.1. Class relationships

(1) Please refer to “INT-SLD-15007_SARAD113x_refman.pdf” file for more detail.

9.2. Class Csarad113x_fsm

(1) Please refer to “INT-SLD-15007_SARAD113x_refman.pdf” file for more detail.

(2) This class is generated by FSM generator (v2014_07_21).

9.3. Class Csarad113x_tha_conv_fsm

(1) Please refer to “INT-SLD-15007_SARAD113x_refman.pdf” file for more detail.

(2) This class is generated by FSM generator (v2014_07_21).

9.4. Class Csarad113x_thb_conv_fsm

(1) Please refer to “INT-SLD-15007_SARAD113x_refman.pdf” file for more detail.

(2) This class is generated by FSM generator (v2014_07_21).

9.5. Class Csarad113x

(1) Please refer to “INT-SLD-15007_SARAD113x_refman.pdf” file for more detail.

Table 9-1: List of implemented functions in Csarad113x class

No.	Function	Level	Description
1	Csarad113x (sc_module_name name);	Public	Constructor of Csarad113x class
2	~Csarad113x (void);	Public	Destructor of Csarad113x class
3	std::string handleCommand (const std::vector<std::string>& args);	Public	Receive parameters, commands, and their arguments to support users in debugging.
4	unsigned int GetWrittenData(void);	Public	Return the last written data to SARAD113x model
5	void tgt_acc (tlm::tlm_generic_payload &trans, sc_time &t);	Private	Issue a transport request to SARAD113x model.
6	unsigned int tgt_acc_dbg (tlm::tlm_generic_payload &trans);	Private	Issue a transport request to SARAD113x model for debugging.
7	double GetTimeResolution(void)	Private	Function is used to get the time resolution
8	void SetLatency_TLM(const double pclk_period, const bool is_constructor)	Private	Function is used to set the latency of tlm_tgt_if
9	void SetCLKfreq(std::string clk_name, double clk_freq)	Private	Function is used to set clock frequency

10	std::string CommandCB(const std::vector<std::string>& args)	Private	Command callback function
11	void DumpStatInfo(void)	Private	DumpStatInfo command of handleCommand
12	void AssertReset(const double start_time, const double period)	Private	AssertReset command of handleCommand
13	void EnableReset(const bool is_active)	Private	Function is used to reset the SARAD113x model
14	unsigned int GetTRGMD(const unsigned int sg)	Private	Function is used to get the TRGMD of scanning group
15	void Initialize(void)	Private	Function is used to initialize SARAD113x model
16	void InitOperation(void)	Private	Function is used to initialize operation of SARAD113x model
17	void ClearScanningEndFlag(const unsigned int sg)	Private	Function is used to clear the scanning end flag of scanning group
18	void WriteADOPControl(const unsigned int sg)	Private	Function is used to control the ADOP output ports
19	double NextPCLKPosedge(double offset)	Private	Function is used to calculate wait time to next rising edge of PCLK clock (first rising edge at the time simulation started)
20	double NextADCLKPosedge(double offset)	Private	Function is used to calculate wait time to next rising edge of ADCLK clock (first rising edge at the time simulation started)
21	double GetANIPortVal(unsigned int port_index)	Private	Function is used to get the A/D input port value
22	void UpdateSelfDiag(void)	Private	Function is used to update self-diagnosis DGOUTAD and DGOUTSH when ADCR.DGON is set to 1
23	void SetCurrentSG(unsigned int group,bool is_th)	Private	Function is used to set the current scanning group number to variable
24	void StartScanning(unsigned int sg, bool is_th)	Private	Function is used to start the scanning process of scanning group
25	void FinishScanning(unsigned int sg)	Private	Function is used to finish the scanning pro-

			cess of scanning group
26	bool CheckSGSetting(unsigned int sg, bool is_th)	Private	Function is used to check the setting of scanning group
27	void SuspendScanning(unsigned int group, bool is_th)	Private	Function is used to suspend the current scanning group
28	void StartVCConv(unsigned int sg)	Private	Function is used to start the virtual channel A/D conversion process
29	void FinishVCConv(unsigned int sg)	Private	Function is used to finish the virtual channel A/D conversion process
30	void UpdateInternalCount(unsigned int sg)	Private	Function is used to update the internal variable which is used for counting during operation
31	bool CheckSuspend(unsigned int trg_num, unsigned int current_sg)	Private	Function is used to check the suspend ability of a scanning group to current group
32	unsigned int IsLastVC(unsigned int sg)	Private	Function is used to check whether current VC is last VC of scanning group or not
33	void CheckTrigger(void)	Private	Function is used to check and notify corresponding trigger of scanning group
34	bool ComparePriority(unsigned int check_sg, unsigned int current_sg)	Private	Function is used to compare the priority of 2 scanning group
35	void SetStartSmpTime(unsigned int channel)	Private	Function is used to set the start sampling time of T&H channel to variable
36	bool CheckSmpTime(unsigned int group, bool hw_trg)	Private	Function is used to check the sampling time of T&H channel
37	bool IsReset(void)	Private	Function is used to check whether SARAD113x model is resetting or not.
38	bool CheckTH(unsigned int sg)	Private	Function is used to check scanning group is T&H group or not
39	bool CheckEnableTH(unsigned int group)	Private	Function is used to check T&H scanning group is enabled or not
40	bool CheckTHStart(unsigned int group)	Private	Function is used to check whether the T&H scanning group can start A/D conversion process or not

41	void StartHoldProcess(unsigned int group)	Private	Function is used to start the hold process for T&H scanning group
42	void HoldPortVal(unsigned int group)	Private	Function is used to hold the A/D input port value of T&H channel
43	void DelayEndHolding(unsigned int group)	Private	Function is used to process the delay when holding ended
44	void ResumeTH(void)	Private	Function is used to notify event to resume T&H scanning group
45	void StartTrigger(unsigned int sg, double delay_time)	Private	Function is used to notify event of corresponding trigger
46	bool IsAutoStartSampling(void)	Private	Function is used to check whether SARAD113x is operated in auto sampling mode or not
47	bool CheckEnableStart(unsigned int sg)	Private	Function is used to check whether scanning group can be started or not
48	bool CheckHoldStart(unsigned int group, bool is_trg, eTriggerType trg_type)	Private	Function is used to check the hold start condition for T&H scanning group
49	bool CheckHoldComplete(unsigned int group, eTriggerType trg_type)	Private	Function is used to check hold complete condition for T&H scanning group
50	void EndHolding(unsigned int group)	Private	Function is used to process end holding process
51	void HWTrigger(unsigned int sg)	Private	Function is used to control the hardware trigger
52	void SWTrigger(unsigned int sg)	Private	Function is used to control the software trigger
53	bool IsContinuousMode(unsigned int sg)	Private	Function is used to check whether scanning group is operated in continuous scanning mode or not
54	void FinishTHConversion(void)	Private	Function is used to notify event for finishing T&H conversion
55	void StopOperation(void)	Private	Function is used to stop the operation of all scanning groups of SARAD113x model
56	void AssertADCATCNVTH(unsigned int sg)	Private	Function is used to control asserting the AD-

			CATCNVn of T&H scanning group
57	double GetSampleTime(unsigned int sg)	Private	Function is used to calculate the sample time of A/D conversion process
58	double GetConversionTime(void)	Private	Function is used to calculate the conversion time of A/D conversion process
59	unsigned int ADConvert(double value, unsigned int sg, unsigned int vc_num)	Private	Function is used to convert the A/D input port value
60	bool StoreADData(unsigned int data, unsigned int sg, unsigned int vc_num)	Private	Function is used to store the A/D converted value into DRj and DIRj
61	unsigned int GetRepetitionTime(unsigned int sg)	Private	Function is used to calculate the repetition time of a virtual channel
62	unsigned int GetMPXE(unsigned int sg, unsigned int vc_num)	Private	Function is used to get the value of MPXE bit of a virtual channel
63	unsigned int GetMPXV(unsigned int sg, unsigned int vc_num)	Private	Function is used to get the value of MPXV bit of a virtual channel
64	unsigned int GetADIE(unsigned int sg, unsigned int vc_num)	Private	Function is used to get the value of MPXE bit of a virtual channel
65	unsigned int GetULS(unsigned int sg, unsigned int vc_num)	Private	Function is used to get the value of MPXE bit of a virtual channel
66	unsigned int GetGCTRL(unsigned int sg, unsigned int vc_num)	Private	Function is used to get the value of MPXE bit of a virtual channel
67	unsigned int GetCNVCLSSelfDiag(unsigned int vc_num)	Private	Function is used to get the value of MPXE bit of a virtual channel
68	void ClearDRProcess(unsigned int sg, unsigned int channel)	Private	Function is used to control clearing DRj when DRj is read
69	void ClearDIRProcess(unsigned int sg, unsigned int vc_num)	Private	Function is used to control clearing DIRj when DIRj is read
70	void PrintVCmessage(std::string msg, unsigned int sg, unsigned int vc_num)	Private	Function is used to dump debug message of A/D conversion process
71	void DumpInfo (const char *type, const char *message, ...)	Private	Function is used to dump the statistic information
72	void DumpInterruptMessage (const std::string intr_name, const bool is_assert)	Private	Function is used to dump the interrupt info message

73	void DumpActivity(unsigned int sg, unsigned int vc_num, double start_time)	Private	Function is used to dump the A/D conversion information
74	void PCLKMethod(void)	Private	Function is used to set the pclk value
75	void ADCLKMethod(void)	Private	Function is used to set the ADCLK value
76	void ResetMethod(void)	Private	Method is used to detect the preset_n port
77	void AssertResetMethod(void)	Private	Method is used to assert reset of AssertReset command
78	void DeAssertResetMethod(void)	Private	Method is used to deassert reset of Asser-tReset command
79	void SG1TRGMethod(void)	Private	Method is used to control the input SG1_TRG
80	void SG2TRGMethod(void)	Private	Method is used to control the input SG2_TRG
81	void SG3TRGMethod(void)	Private	Method is used to control the input SG3_TRG
82	void PVCRTRGMethod(void)	Private	Method is used to control the input PVCR_TRG
83	void PVCR_VALUEMethod(void)	Private	Method is used to control the input PVCR_VALUE
84	void WriteADOPControlMethod(void)	Private	Method is used to control the ADOP output ports
85	void WritePVCR_MUXCURMethod(void)	Private	Method is used to control the PVCR_MUXCUR output port
86	void WriteADCATCNVControlMethod(unsigned int sg)	Private	Method is used to control the ADCATCNVn output ports
87	void WriteSGEndInterruptMethod(unsigned int sg)	Private	Method is used to control the scanning end interrupt
88	void WriteULEInterruptMethod(void)	Private	Method is used to control the ULE interrupt
89	void WriteADEInterruptMethod(void)	Private	Method is used to control the INT_ADE inter-rupt
90	void AVREFHMethod(void)	Private	Method is used to control the input AVREFH0 port
91	void InitialAVREFHMethod(void)	Private	Method is used to update the AVREFH0 val-ue to variable when simulation starts
92	void HWTriggerProcessMethod(unsigned int	Private	Method is used to control hardware trigger

	sg)		process
93	void SWTriggerProcessMethod(unsigned int sg)	Private	Method is used to control software trigger process
94	void CheckTriggerMethod(void)	Private	Method is used to call CheckTrigger() function
95	void UpdateSGACTMethod(unsigned int sg)	Private	Method is used to control updating SGSTR.SGACT bit
96	void UpdateSHACTMethod(void)	Private	Method is used to control updating SGSTR.SHACT bit
97	void StartTHSamplingMethod(void)	Private	Method is used to start T&H sampling process
98	void VCSamplingMethod(void)	Private	Method is used to control the virtual channel sampling process
99	void VCConversionMethod(void)	Private	Method is used to control the virtual channel conversion process
100	void VCEndConversionMethod(void)	Private	Method is used to control virtual channel conversion ending process
101	void UpdateConversionDataMethod(unsigned int sg)	Private	Method is used to control the updating conversion data process
102	void ClearDRMethod(unsigned int channel)	Private	Method is used to control clearing DRj register process
103	void ClearPWDDRMethod(void)	Private	Method is used to control control clearing PWDDR register process
104	void ClearDIRMethod(unsigned int vc)	Private	Method is used to control clearing DIRj register process
105	void ClearPWDDIRMethod(void)	Private	Method is used to control control clearing PWDDIR register process
106	bool CheckAccess(const unsigned int sg, vpcl::re_register* reg, RegCBstr str)	Private	Function is used to check the write condition of register
107	void cb_VCR_GCTRL(RegCBstr str)	Private	Callback funtion of register VCR
108	void cb_TSNVCR_ULS(RegCBstr str)	Private	Callback funtion of register TSNVCR
109	void cb_DR_DR0(RegCBstr str)	Private	Callback funtion of register DR

110	void cb_PWDTSNDR_TSNDR(RegCBstr str)	Private	Callback funtion of register PWDTSNDR
111	void cb_DIR_DR(RegCBstr str)	Private	Callback funtion of register DIR
112	void cb_TSNDIR_TSNDR(RegCBstr str)	Private	Callback funtion of register TSNDIR
113	void cb_PWDDIR_PWDDR(RegCBstr str)	Private	Callback funtion of register PWDDIR
114	void cb_ADHALTR_HALT(RegCBstr str)	Private	Callback funtion of register ADHALTR
115	void cb_ADCR_SUSMTD(RegCBstr str)	Private	Callback funtion of register ADCR
116	void cb_TSNCR_TSNEN(RegCBstr str)	Private	Callback funtion of register TSNCR
117	void cb_THSMPSTCR_SMPST(RegCBstr str)	Private	Callback funtion of register THSMPSTCR
118	void cb_THCR_ASMPMSK(RegCBstr str)	Private	Callback funtion of register THCR
119	void cb_THAHLDSTCR_HLDST(RegCBstr str)	Private	Callback funtion of register THAHLDSTCR
120	void cb_THBHLDSTCR_HLDST(RegCBstr str)	Private	Callback funtion of register THBHLDSTCR
121	void cb_THACR_SGS(RegCBstr str)	Private	Callback funtion of register THACR
122	void cb_THBCR_SGS(RegCBstr str)	Private	Callback funtion of register THBCR
123	void cb_THER_TH0E(RegCBstr str)	Private	Callback funtion of register THER
124	void cb_THGSR_TH0GS(RegCBstr str)	Private	Callback funtion of register THGSR
125	void cb_SFTCR_OWEIE(RegCBstr str)	Private	Callback funtion of register SFTCR
126	void cb_ULLMTBR_ULMTB(RegCBstr str)	Private	Callback funtion of register ULLMTBR
127	void cb_ECR_ULEC(RegCBstr str)	Private	Callback funtion of register ECR
128	void cb_DGCTL0_PSEL0(RegCBstr str)	Private	Callback funtion of register DGCTL0
129	void cb_DGCTL1_CDG00(RegCBstr str)	Private	Callback funtion of register DGCTL1
130	void cb_PDCTL1_PDNA00(RegCBstr str)	Private	Callback funtion of register PDCTL1
131	void cb_PDCTL2_PDNB00(RegCBstr str)	Private	Callback funtion of register PDCTL2
132	void cb_SMPCR_SMPT(RegCBstr str)	Private	Callback funtion of register SMPCR
133	void cb_TSNSMPCR_TSNSMPT(RegCBstr str)	Private	Callback funtion of register TSNSMPCR
134	void cb_EMUCR_SVSDIS(RegCBstr str)	Private	Callback funtion of register EMUCR
135	void cb_SGPRCR_SGPR0(RegCBstr str)	Private	Callback funtion of register SGPRCR
136	void cb_SGSTCR_SGSTn(RegCBstr str)	Private	Callback funtion of register SGSTCR
137	void cb_TSNSGSTCR_TSNSGST(RegCBstr	Private	Callback funtion of register TSNSGSTCR

	str)		
138	void cb_PWDSGSTCR_PWDSGST(RegCBstr str)	Private	Callback funtion of register PWDSGSTCR
139	void cb_SGCR_ADIE(RegCBstr str)	Private	Callback funtion of register SGCR
140	void cb_TSNSGCR_TSNTRGMD(RegCBstr str)	Private	Callback funtion of register TSNSGCR
141	void cb_PWDSGCR_PWDTRGMD(RegCBstr str)	Private	Callback funtion of register PWDSGCR
142	void cb_SGSEFCR_SEFCn(RegCBstr str)	Private	Callback funtion of register SGSEFCR
143	void cb_TSNSGSEFCR_TSNSEFC(RegCBstr str)	Private	Callback funtion of register TSNSGSEFCR
144	void cb_PWDSGSEFCR_PWDSEFC(RegCBstr str)	Private	Callback funtion of register PWDSGSEFCR
145	void cb_SGVCSP_VCSP(RegCBstr str)	Private	Callback funtion of register SGVCSP
146	void cb_SGVCEP_VCEP(RegCBstr str)	Private	Callback funtion of register SGVCEP
147	void cb_SGMCYCR_MCYC(RegCBstr str)	Private	Callback funtion of register SGMCYCR
148	void cb_SGTSEL_TxSEL00(RegCBstr str)	Private	Callback funtion of register SGTSEL

10. Restriction

The SARAD113x model does not support the features in Table 10.1.

Table 10-1: List of un-supported features of the SARAD113x model

Features	Support
1.16. Operation explanation	Yes
1.16.5. Temperature sensor operation	No
1.16.5.1. Temperature sensor operation	No
1.16.5.2. Temperature sensor self-diagnosis function	No
1.16.10. SVSTOP operation	No
1.16.19. Intermittent operation function	No
1.16.20. Synchronization configuration	No
1.17. HM trim adjustment	No
1.18. The definition of the A/D conversion accuracy is shown as follows.	No

Revision History					
Rev.	**Modified Contents**	**Agreed by Customer**	**Approved by RVC**	**Reviewed by**	**Created by**
1.0	New creation.	A.Imoto 09/18/2015	Vu Pham 09/11/2015	Uyen Le 09/11/2015	Son Tran 09/10/2015
1.1	- Update Table 7-1: Update revision of source code - Update Table 9-1: Add public API GetWrittenData () to return the last written data to SARAD113x model	A.Imoto 11/13/2015	Vu Pham 10/28/2015	Duc Duong 10/28/2015	Son Tran 10/28/2015
1.2	- Update chapter 9: Update the explaination of referring to “INT-SLD-15007_SARAD113x_refman.pdf” file	A.Imoto 12/29/2015	Vu Pham 12/29/2015	Vu Pham 12/29/2015	Son Tran 12/29/2015