Bit(s)	Mnemonic	Name	Description
			3: Reserved 4: Reserved 5: Reserved 6: Reserved 7: Reserved

A0020C24 GPIO_MODE2_SET GPIO Mode Control 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name		**GPIO23**				**GPIO22**				**GPIO21**				**GPIO20**		
Type		WO				WO				WO				WO		
Reset		0	0	0		0	0	0		0	0	0		0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name		**GPIO19**				**GPIO18**				**GPIO17**				**GPIO16**		
Type		WO				WO				WO				WO		
Reset		0	0	0		0	0	0		0	0	0		0	0	0

Overview: For bitwise access of GPIO_MODE2

Bit(s)	Mnemonic	Name	Description
30:28		GPIO23	**Bitwise SET operation for Aux. mode of BPI_BUS1** 0: Keep 1: SET bits
26:24		GPIO22	**Bitwise SET operation for Aux. mode of BPI_BUS2** 0: Keep 1: SET bits
22:20		GPIO21	**Bitwise SET operation for Aux. mode of KROW0** 0: Keep 1: SET bits
18:16		GPIO20	**Bitwise SET operation for Aux. mode of KROW1** 0: Keep 1: SET bits
14:12		GPIO19	**Bitwise SET operation for Aux. mode of KROW2** 0: Keep 1: SET bits
10:8		GPIO18	**Bitwise SET operation for Aux. mode of KROW3** 0: Keep 1: SET bits
6:4		GPIO17	**Bitwise SET operation for Aux. mode of KROW4** 0: Keep 1: SET bits
2:0		GPIO16	**Bitwise SET operation for Aux. mode of KCOL0** 0: Keep 1: SET bits

A0020C28 GPIO_MODE2_CLR GPIO Mode Control 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name		**GPIO23**				**GPIO22**				**GPIO21**				**GPIO20**		

Type		WO				WO				WO				WO		
Reset		0	0	0		0	0	0		0	0	0		0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name		GPIO19				GPIO18				GPIO17				GPIO16		
Type		WO				WO				WO				WO		
Reset		0	0	0		0	0	0		0	0	0		0	0	0

Overview: For bitwise access of GPIO_MODE2

Bit(s)	Mnemonic	Name	Description
30:28		GPIO23	**Bitwise CLR operation for Aux. mode of BPI_BUS1** 0: Keep 1: CLR bits
26:24		GPIO22	**Bitwise CLR operation for Aux. mode of BPI_BUS2** 0: Keep 1: CLR bits
22:20		GPIO21	**Bitwise CLR operation for Aux. mode of KROW0** 0: Keep 1: CLR bits
18:16		GPIO20	**Bitwise CLR operation for Aux. mode of KROW1** 0: Keep 1: CLR bits
14:12		GPIO19	**Bitwise CLR operation for Aux. mode of KROW2** 0: Keep 1: CLR bits
10:8		GPIO18	**Bitwise CLR operation for Aux. mode of KROW3** 0: Keep 1: CLR bits
6:4		GPIO17	**Bitwise CLR operation for Aux. mode of KROW4** 0: Keep 1: CLR bits
2:0		GPIO16	**Bitwise CLR operation for Aux. mode of KCOL0** 0: Keep 1: CLR bits

A0020C30 GPIO_MODE3 GPIO Mode Control 00000001

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name		GPIO31				GPIO30			GPIO29				GPIO28			
Type		RW				RW			RW				RW			
Reset		0	0	0		0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO27					GPIO26				GPIO25				GPIO24		
Type	RW					RW				RW				RW		
Reset	0	0	0	0		0	0	0		0	0	0		0	0	1

Overview: Configures GPIO aux. mode

Bit(s)	Mnemonic	Name	Description
30:28		GPIO31	**Aux. mode of GPIO_31** 0: GPIO31 (IO) 1: MCCK (O) 2: Reserved

Bit(s)	Mnemonic	Name	Description
			3: Reserved 4: U2RXD (I) 5: Reserved 6: Reserved 7: Reserved
26:24		GPIO30	**Aux. mode of GPIO_30** 0: GPIO30 (IO) 1: CMCSK (I) 2: LPTE (I) 3: CMCSD2 (I) 4: EINT16 (I) 5: Reserved 6: JTRCK (O) 7: Reserved
23:20		GPIO29	**Aux. mode of GPIO_29** 0: GPIO29 (IO) 1: CMMCLK (O) 2: LSA0DA1 (O) 3: DAISYNC (O) 4: SPIMISO (IO) 5: FMJTDO (O) 6: JTDO (O) 7: Reserved 8: MC2DA0 (IO)
19:16		GPIO28	**Aux. mode of GPIO_28** 0: GPIO28 (IO) 1: CMCSD1 (I) 2: LSDA1 (IO) 3: DAIPCMOUT (O) 4: SPIMOSI (IO) 5: FMJTRSTB (I) 6: JTRST_B (I) 7: Reserved 8: MC2CK (O)
15:12		GPIO27	**Aux. mode of GPIO_27** 0: GPIO27 (IO) 1: CMCSD0 (I) 2: LSCE_B1 (O) 3: DAIPCMIN (I) 4: SPISCK (IO) 5: FMJTCK (I) 6: JTCK (I) 7: Reserved 8: MC2CM0 (O)
10:8		GPIO26	**Aux. mode of GPIO_26** 0: GPIO26 (IO) 1: CMPDN (O) 2: LSCK1 (O) 3: DAICLK (O) 4: SPICS (IO) 5: FMJTMS (I) 6: JTMS (I) 7: Reserved)
6:4		GPIO25	**Aux. mode of GPIO_25** 0: GPIO25 (IO) 1: CMRST (O) 2: TESTMODE_D (O) 3: CLKO1 (O) 4: EINT15 (I)

Bit(s)	Mnemonic	Name	Description
			5: FMJTDI (I) 6: JTDI (I) 7: Reserved
2:0		GPIO24	**Aux. mode of GPIO_24** 0: GPIO24 (IO) 1: BPI_BUS0 (O) 2: Reserved 3: Reserved 4: Reserved 5: Reserved 6: Reserved 7: Reserved

A0020C34 **GPIO_MODE3_SET** **GPIO Mode Control** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name		**GPIO31**				**GPIO30**			**GPIO29**				**GPIO28**			
Type		WO				WO			WO				WO			
Reset		0	0	0		0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**GPIO27**					**GPIO26**				**GPIO25**				**GPIO24**		
Type	WO					WO				WO				WO		
Reset	0	0	0	0		0	0	0		0	0	0		0	0	0

Overview: For bitwise access of GPIO_MODE3

Bit(s)	Mnemonic	Name	Description
30:28		GPIO31	**Bitwise SET operation for Aux. mode of MCCK** 0: Keep 1: SET bits
26:24		GPIO30	**Bitwise SET operation for Aux. mode of CMCSK** 0: Keep 1: SET bits
23:20		GPIO29	**Bitwise SET operation for Aux. mode of CMMCLK** 0: Keep 1: SET bits
19:16		GPIO28	**Bitwise SET operation for Aux. mode of CMCSD1** 0: Keep 1: SET bits
15:12		GPIO27	**Bitwise SET operation for Aux. mode of CMCSD0** 0: Keep 1: SET bits
10:8		GPIO26	**Bitwise SET operation for Aux. mode of CMPDN** 0: Keep 1: SET bits
6:4		GPIO25	**Bitwise SET operation for Aux. mode of CMRST** 0: Keep 1: SET bits
2:0		GPIO24	**Bitwise SET operation for Aux. mode of BPI_BUS0** 0: Keep 1: SET bits

A0020C38 GPIO_MODE3_CLR GPIO Mode Control 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name		GPIO31				GPIO30			GPIO29				GPIO28			
Type		WO				WO			WO				WO			
Reset		0	0	0		0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO27					GPIO26				GPIO25				GPIO24		
Type	WO					WO				WO				WO		
Reset	0	0	0	0		0	0	0		0	0	0		0	0	0

Overview: For bitwise access of GPIO_MODE3

Bit(s)	Mnemonic	Name	Description
30:28		GPIO31	**Bitwise CLR operation for Aux. mode of MCCK** 0: Keep 1: CLR bits
26:24		GPIO30	**Bitwise CLR operation for Aux. mode of CMCSK** 0: Keep 1: CLR bits
23:20		GPIO29	**Bitwise CLR operation for Aux. mode of CMMCLK** 0: Keep 1: CLR bits
19:16		GPIO28	**Bitwise CLR operation for Aux. mode of CMCSD1** 0: Keep 1: CLR bits
15:12		GPIO27	**Bitwise CLR operation for Aux. mode of CMCSD0** 0: Keep 1: CLR bits
10:8		GPIO26	**Bitwise CLR operation for Aux. mode of CMPDN** 0: Keep 1: CLR bits
6:4		GPIO25	**Bitwise CLR operation for Aux. mode of CMRST** 0: Keep 1: CLR bits
2:0		GPIO24	**Bitwise CLR operation for Aux. mode of BPI_BUS0** 0: Keep 1: CLR bits

A0020C40 GPIO_MODE4 GPIO Mode Control 11100000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name		GPIO39				GPIO38				GPIO37				GPIO36		
Type		RW				RW				RW				RW		
Reset		0	0	1		0	0	1		0	0	1		0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name		GPIO35				GPIO34				GPIO33				GPIO32		
Type		RW				RW				RW				RW		
Reset		0	0	0		0	0	0		0	0	0		0	0	0

Overview: Configures GPIO aux. mode

Bit(s)	Mnemonic	Name	Description
30:28		GPIO39	**Aux. mode of GPIO_39** 0: GPIO39 (IO) 1: SIM1_SCLK (IO) 2: Reserved 3: Reserved 4: Reserved 5: Reserved 6: Reserved 7: Reserved
26:24		GPIO38	**Aux. mode of GPIO_38** 0: GPIO38 (IO) 1: SIM1_SRST (IO) 2: Reserved 3: Reserved 4: Reserved 5: Reserved 6: Reserved 7: Reserved
22:20		GPIO37	**Aux. mode of GPIO_37** 0: GPIO37 (IO) 1: SIM1_SIO (IO) 2: Reserved 3: Reserved 4: Reserved 5: Reserved 6: Reserved 7: Reserved
18:16		GPIO36	**Aux. mode of GPIO_36** 0: GPIO36 (IO) 1: MCDA3 (IO) 2: EINT19 (I) 3: CLKO2 (O) 4: DAIPCMOUT (O) 5: Reserved 6: Reserved 7: Reserved
14:12		GPIO35	**Aux. mode of GPIO_35** 0: GPIO35 (IO) 1: MCDA2 (IO) 2: EINT18 (I) 3: Reserved 4: DAICLK (O) 5: Reserved 6: Reserved 7: Reserved
10:8		GPIO34	**Aux. mode of GPIO_34** 0: GPIO34 (IO) 1: MCDA1 (IO) 2: EINT17 (I) 3: Reserved 4: DAIPCMIN (I) 5: Reserved 6: Reserved 7: Reserved
6:4		GPIO33	**Aux. mode of GPIO_33** 0: GPIO33 (IO)

Bit(s)	Mnemonic	Name	Description
			1: MCDA0 (IO) 2: Reserved 3: Reserved 4: DAISYNC (O) 5: Reserved 6: Reserved 7: Reserved
2:0		GPIO32	**Aux. mode of GPIO_32** 0: GPIO32 (IO) 1: MCCM0 (O) 2: Reserved 3: Reserved 4: U2TXD (O) 5: Reserved 6: Reserved 7: Reserved

A0020C44 **GPIO_MODE4_SET** **GPIO Mode Control** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name		**GPIO39**				**GPIO38**				**GPIO37**				**GPIO36**		
Type		WO				WO				WO				WO		
Reset		0	0	0		0	0	0		0	0	0		0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name		**GPIO35**				**GPIO34**				**GPIO33**				**GPIO32**		
Type		WO				WO				WO				WO		
Reset		0	0	0		0	0	0		0	0	0		0	0	0

Overview: For bitwise access of GPIO_MODE4

Bit(s)	Mnemonic	Name	Description
30:28		GPIO39	**Bitwise SET operation for Aux. mode of SIM1_SCLK** 0: Keep 1: SET bits
26:24		GPIO38	**Bitwise SET operation for Aux. mode of SIM1_SRST** 0: Keep 1: SET bits
22:20		GPIO37	**Bitwise SET operation for Aux. mode of SIM1_SIO** 0: Keep 1: SET bits
18:16		GPIO36	**Bitwise SET operation for Aux. mode of MCDA3** 0: Keep 1: SET bits
14:12		GPIO35	**Bitwise SET operation for Aux. mode of MCDA2** 0: Keep 1: SET bits
10:8		GPIO34	**Bitwise SET operation for Aux. mode of MCDA1** 0: Keep 1: SET bits
6:4		GPIO33	**Bitwise SET operation for Aux. mode of MCDA0** 0: Keep 1: SET bits

Bit(s)	Mnemonic	Name	Description
2:0		GPIO32	**Bitwise SET operation for Aux. mode of MCCM0** 0: Keep 1: SET bits

A0020C48 GPIO_MODE4_CLR GPIO Mode Control 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name		**GPIO39**				**GPIO38**				**GPIO37**				**GPIO36**		
Type		WO				WO				WO				WO		
Reset		0	0	0		0	0	0		0	0	0		0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name		**GPIO35**				**GPIO34**				**GPIO33**				**GPIO32**		
Type		WO				WO				WO				WO		
Reset		0	0	0		0	0	0		0	0	0		0	0	0

Overview: For bitwise access of GPIO_MODE4

Bit(s)	Mnemonic	Name	Description
30:28		GPIO39	**Bitwise CLR operation for Aux. mode of SIM1_SCLK** 0: Keep 1: CLR bits
26:24		GPIO38	**Bitwise CLR operation for Aux. mode of SIM1_SRST** 0: Keep 1: CLR bits
22:20		GPIO37	**Bitwise CLR operation for Aux. mode of SIM1_SIO** 0: Keep 1: CLR bits
18:16		GPIO36	**Bitwise CLR operation for Aux. mode of MCDA3** 0: Keep 1: CLR bits
14:12		GPIO35	**Bitwise CLR operation for Aux. mode of MCDA2** 0: Keep 1: CLR bits
10:8		GPIO34	**Bitwise CLR operation for Aux. mode of MCDA1** 0: Keep 1: CLR bits
6:4		GPIO33	**Bitwise CLR operation for Aux. mode of MCDA0** 0: Keep 1: CLR bits
2:0		GPIO32	**Bitwise CLR operation for Aux. mode of MCCM0** 0: Keep 1: CLR bits

A0020C50 GPIO_MODE5 GPIO Mode Control 01000111

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name		**GPIO47**				**GPIO46**				**GPIO45**				**GPIO44**		
Type		RW				RW				RW				RW		
Reset		0	0	0		0	0	1		0	0	0		0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Name		GPIO43				GPIO42				GPIO41				GPIO40		
Type		RW				RW				RW				RW		
Reset		0	0	0		0	0	1		0	0	1		0	0	1

Overview: Configures GPIO aux. mode

Bit(s)	Mnemonic	Name	Description
30:28		GPIO47	**Aux. mode of GPIO_47** 0: GPIO47 (IO) 1: LSCK0 (O) 2: Reserved 3: CMPDN (O) 4: Reserved 5: Reserved 6: Reserved 7: Reserved
26:24		GPIO46	**Aux. mode of GPIO_46** 0: GPIO46 (IO) 1: LSCE_B0 (O) 2: EINT20 (I) 3: CMCSD0 (I) 4: CLKO4 (O) 5: Reserved 6: Reserved 7: Reserved
22:20		GPIO45	**Aux. mode of GPIO_45** 0: GPIO45 (IO) 1: TESTMODE_D (O) 2: Reserved 3: CMRST (O) 4: Reserved 5: Reserved 6: Reserved 7: Reserved
18:16		GPIO44	**Aux. mode of GPIO_44** 0: GPIO44 (IO) 1: SDA (IO) 2: Reserved 3: Reserved 4: Reserved 5: Reserved 6: Reserved 7: Reserved
14:12		GPIO43	**Aux. mode of GPIO_43** 0: GPIO43 (IO) 1: SCL (IO) 2: Reserved 3: Reserved 4: Reserved 5: Reserved 6: Reserved 7: Reserved
10:8		GPIO42	**Aux. mode of GPIO_42** 0: GPIO42 (IO) 1: SIM2_SCLK (IO) 2: LSCE1_B1 (O) 3: Reserved 4: Reserved 5: Reserved

Bit(s)	Mnemonic	Name	Description
			6: Reserved 7: Reserved
6:4		GPIO41	**Aux. mode of GPIO_41** 0: GPIO41 (IO) 1: SIM2_SRST (IO) 2: CLKO3 (O) 3: U2CTS (I) 4: Reserved 5: Reserved 6: Reserved 7: Reserved
2:0		GPIO40	**Aux. mode of GPIO_40** 0: GPIO40 (IO) 1: SIM2_SIO (IO) 2: Reserved 3: U2RTS (O) 4: Reserved 5: Reserved 6: Reserved 7: Reserved

A0020C54 **GPIO_MODE5_SET** **GPIO Mode Control** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name		GPIO47				GPIO46				GPIO45				GPIO44		
Type		WO				WO				WO				WO		
Reset		0	0	0		0	0	0		0	0	0		0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name		GPIO43				GPIO42				GPIO41				GPIO40		
Type		WO				WO				WO				WO		
Reset		0	0	0		0	0	0		0	0	0		0	0	0

Overview: For bitwise access of GPIO_MODE5

Bit(s)	Mnemonic	Name	Description
30:28		GPIO47	**Bitwise SET operation for Aux. mode of LSCK** 0: Keep 1: SET bits
26:24		GPIO46	**Bitwise SET operation for Aux. mode of LSCE_B** 0: Keep 1: SET bits
22:20		GPIO45	**Bitwise SET operation for Aux. mode of TESTMODE_D** 0: Keep 1: SET bits
18:16		GPIO44	**Bitwise SET operation for Aux. mode of SDA28** 0: Keep 1: SET bits
14:12		GPIO43	**Bitwise SET operation for Aux. mode of SCL28** 0: Keep 1: SET bits
10:8		GPIO42	**Bitwise SET operation for Aux. mode of SIM2_SCLK** 0: Keep

Bit(s)	Mnemonic	Name	Description
			1: SET bits
6:4		GPIO41	**Bitwise SET operation for Aux. mode of SIM2_SRST** 0: Keep 1: SET bits
2:0		GPIO40	**Bitwise SET operation for Aux. mode of SIM2_SIO** 0: Keep 1: SET bits

A0020C58 GPIO_MODE5_CLR GPIO Mode Control 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name		**GPIO47**				**GPIO46**				**GPIO45**				**GPIO44**		
Type		WO				WO				WO				WO		
Reset		0	0	0		0	0	0		0	0	0		0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name		**GPIO43**				**GPIO42**				**GPIO41**				**GPIO40**		
Type		WO				WO				WO				WO		
Reset		0	0	0		0	0	0		0	0	0		0	0	0

Overview: For bitwise access of GPIO_MODE5

Bit(s)	Mnemonic	Name	Description
30:28		GPIO47	**Bitwise CLR operation for Aux. mode of LSCK** 0: Keep 1: CLR bits
26:24		GPIO46	**Bitwise CLR operation for Aux. mode of LSCE_B** 0: Keep 1: CLR bits
22:20		GPIO45	**Bitwise CLR operation for Aux. mode of TESTMODE_D** 0: Keep 1: CLR bits
18:16		GPIO44	**Bitwise CLR operation for Aux. mode of SDA28** 0: Keep 1: CLR bits
14:12		GPIO43	**Bitwise CLR operation for Aux. mode of SCL28** 0: Keep 1: CLR bits
10:8		GPIO42	**Bitwise CLR operation for Aux. mode of SIM2_SCLK** 0: Keep 1: CLR bits
6:4		GPIO41	**Bitwise CLR operation for Aux. mode of SIM2_SRST** 0: Keep 1: CLR bits
2:0		GPIO40	**Bitwise CLR operation for Aux. mode of SIM2_SIO** 0: Keep 1: CLR bits

A0020C60 GPIO_MODE6 GPIO Mode Control 00001000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name		**GPIO55**				**GPIO54**				**GPIO53**				**GPIO52**		
Type		RW				RW				RW				RW		
Reset		0	0	0		0	0	0		0	0	0		0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name		**GPIO51**			**GPIO50**					**GPIO49**				**GPIO48**		
Type		RW			RW					RW				RW		
Reset		0	0	1	0	0	0	0		0	0	0		0	0	0

Overview: Configures GPIO aux. mode

Bit(s)	Mnemonic	Name	Description
30:28		GPIO55	**Aux. mode of GPIO_55** 0: AGPIO55 (AGIO) 1: Reserved 2: Reserved 3: Reserved 4: Reserved 5: Reserved 6: Reserved 7: Reserved
26:24		GPIO54	**Aux. mode of GPIO_54** 0: AGPIO54 (AGIO) 1: Reserved 2: Reserved 3: Reserved 4: Reserved 5: Reserved 6: Reserved 7: Reserved
22:20		GPIO53	**Aux. mode of GPIO_53** 0: AGPIO53 (AGI) 1: SRCLKENAI (I) 2: EINT24 (I) 3: Reserved 4: Reserved 5: Reserved 6: Reserved 7: Reserved
18:16		GPIO52	**Aux. mode of GPIO_52** 0: AGPI52 (AGI) 1: Reserved 2: EINT23 (I) 3: Reserved 4: Reserved 5: Reserved 6: Reserved 7: Reserved
14:12		GPIO51	**Aux. mode of GPIO_51** 0: GPIO51 (IO) 1: RESETB (IO) 2: Reserved 3: Reserved 4: Reserved 5: Reserved 6: Reserved 7: Reserved
11:8		GPIO50	**Aux. mode of GPIO_50** 0: GPIO50 (IO)

Bit(s)	Mnemonic	Name	Description
			1: LPTE (I) 2: EINT22 (I) 3: CMCSK (I) 4: CMCSD2 (I) 5: Reserved 6: MCINS (I) 7: Reserved 8: Reserved 9: CLKO5 (O)
6:4		GPIO49	**Aux. mode of GPIO_49** 0: GPIO49 (IO) 1: LSA0DA0 (O) 2: LSCE1_B0 (O) 3: CMMCLK (O) 4: Reserved 5: Reserved 6: Reserved 7: Reserved
2:0		GPIO48	**Aux. mode of GPIO_48** 0: GPIO48 (IO) 1: LSDA0 (IO) 2: EINT21 (I) 3: CMCSD1 (I) 4: WIFITOBT (I) 5: Reserved 6: Reserved 7: Reserved

A0020C64 **GPIO_MODE6_SET** **GPIO Mode Control** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name		**GPIO55**				**GPIO54**				**GPIO53**				**GPIO52**		
Type		WO				WO				WO				WO		
Reset		0	0	0		0	0	0		0	0	0		0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name		**GPIO51**			**GPIO50**					**GPIO49**				**GPIO48**		
Type		WO			WO					WO				WO		
Reset		0	0	0	0	0	0	0		0	0	0		0	0	0

Overview: For bitwise access of GPIO_MODE6

Bit(s)	Mnemonic	Name	Description
30:28		GPIO55	**Bitwise SET operation for Aux. mode of SRCLKENAI** 0: Keep 1: SET bits
26:24		GPIO54	**Bitwise SET operation for Aux. mode of EINT** 0: Keep 1: SET bits
22:20		GPIO53	**Bitwise SET operation for Aux. mode of TP4** 0: Keep 1: SET bits
18:16		GPIO52	**Bitwise SET operation for Aux. mode of TP3** 0: Keep

Bit(s)	Mnemonic	Name	Description
			1: SET bits
14:12		GPIO51	**Bitwise SET operation for Aux. mode of RESETB** 0: Keep 1: SET bits
11:8		GPIO50	**Bitwise SET operation for Aux. mode of LPTE** 0: Keep 1: SET bits
6:4		GPIO49	**Bitwise SET operation for Aux. mode of LSA0** 0: Keep 1: SET bits
2:0		GPIO48	**Bitwise SET operation for Aux. mode of LSDA** 0: Keep 1: SET bits

A0020C68 GPIO_MODE6_CLR GPIO Mode Control 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name		GPIO55				GPIO54				GPIO53				GPIO52		
Type		WO				WO				WO				WO		
Reset		0	0	0		0	0	0		0	0	0		0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name		GPIO51			GPIO50					GPIO49				GPIO48		
Type		WO			WO					WO				WO		
Reset		0	0	0	0	0	0	0		0	0	0		0	0	0

Overview: For bitwise access of GPIO_MODE6

Bit(s)	Mnemonic	Name	Description
30:28		GPIO55	**Bitwise CLR operation for Aux. mode of SRCLKENAI** 0: Keep 1: CLR bits
26:24		GPIO54	**Bitwise CLR operation for Aux. mode of EINT** 0: Keep 1: CLR bits
22:20		GPIO53	**Bitwise CLR operation for Aux. mode of TP4** 0: Keep 1: CLR bits
18:16		GPIO52	**Bitwise CLR operation for Aux. mode of TP3** 0: Keep 1: CLR bits
14:12		GPIO51	**Bitwise CLR operation for Aux. mode of RECLRB** 0: Keep 1: CLR bits
11:8		GPIO50	**Bitwise CLR operation for Aux. mode of LPTE** 0: Keep 1: CLR bits
6:4		GPIO49	**Bitwise CLR operation for Aux. mode of LSA0** 0: Keep 1: CLR bits
2:0		GPIO48	**Bitwise CLR operation for Aux. mode of LSDA**

Bit(s)	Mnemonic	Name	Description
			0: Keep 1: CLR bits

A0020D10 GPIO_TDSEL GPIO TDSEL Control 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name															**GPIO50**	
Type															RW	
Reset															0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**GPIO46**		**GPIO42**		**GPIO41**		**GPIO36**		**GPIO25**		**GPIO18**		**GPIO9**		**GPIO8**	
Type	RW		RW		RW		RW		RW		RW		RW		RW	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: GPIO TX duty control register

Bit(s)	Mnemonic	Name	Description
17:16	**GPIO50**	GPIO50_TDSEL	GPIO50 Tx duty control
15:14	**GPIO46**	GPIO46_TDSEL	GPIO46 Tx duty control
13:12	**GPIO42**	GPIO42_TDSEL	GPIO42 Tx duty control
11:10	**GPIO41**	GPIO41_TDSEL	GPIO41 Tx duty control
9:8	**GPIO36**	GPIO36_TDSEL	GPIO36 Tx duty control
7:6	**GPIO25**	GPIO25_TDSEL	GPIO25 Tx duty control
5:4	**GPIO18**	GPIO18_TDSEL	GPIO18 Tx duty control
3:2	**GPIO9**	GPIO9_TDSEL	GPIO9 Tx duty control
1:0	**GPIO8**	GPIO8_TDSEL	GPIO8 Tx duty control

A0020D14 GPIO_TDSEL_SET GPIO TDSEL Control 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name															**GPIO50**	
Type															WO	
Reset															0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**GPIO46**		**GPIO42**		**GPIO41**		**GPIO36**		**GPIO25**		**GPIO18**		**GPIO9**		**GPIO8**	
Type	WO		WO		WO		WO		WO		WO		WO		WO	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: For bitwise access of GPIO_TDSEL

Bit(s)	Mnemonic	Name	Description
17:16	**GPIO50**	GPIO50_TDSEL	**Bitwise SET operation of GPIO50_TDSEL Tx duty control** 0: Keep 1: SET bits
15:14	**GPIO46**	GPIO46_TDSEL	**Bitwise SET operation of GPIO46_TDSEL Tx duty control** 0: Keep 1: SET bits
13:12	**GPIO42**	GPIO42_TDSEL	**Bitwise SET operation of GPIO42_TDSEL Tx duty control**

Bit(s)	Mnemonic	Name	Description
			0: Keep 1: SET bits
11:10	**GPIO41**	GPIO41_TDSEL	**Bitwise SET operation of GPIO41_TDSEL Tx duty control** 0: Keep 1: SET bits
9:8	**GPIO36**	GPIO36_TDSEL	**Bitwise SET operation of GPIO36_TDSEL Tx duty control** 0: Keep 1: SET bits
7:6	**GPIO25**	GPIO25_TDSEL	**Bitwise SET operation of GPIO25_TDSEL Tx duty control** 0: Keep 1: SET bits
5:4	**GPIO18**	GPIO18_TDSEL	**Bitwise SET operation of GPIO18_TDSEL Tx duty control** 0: Keep 1: SET bits
3:2	**GPIO9**	GPIO9_TDSEL	**Bitwise SET operation of GPIO9_TDSEL Tx duty control** 0: Keep 1: SET bits
1:0	**GPIO8**	GPIO8_TDSEL	**Bitwise SET operation of GPIO8_TDSEL Tx duty control** 0: Keep 1: SET bits

A0020D18 **GPIO_TDSEL_CLR** **GPIO TDSEL Control** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name															GPIO50	
Type															WO	
Reset															0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GPIO46		GPIO42		GPIO41		GPIO36		GPIO25		GPIO18		GPIO9		GPIO8	
Type	WO		WO		WO		WO		WO		WO		WO		WO	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: For bitwise access of GPIO_TDSEL

Bit(s)	Mnemonic	Name	Description
17:16	**GPIO50**	GPIO50_TDSEL	**Bitwise CLR operation of GPIO50_TDSEL Tx duty control** 0: Keep 1: CLR bits
15:14	**GPIO46**	GPIO46_TDSEL	**Bitwise CLR operation of GPIO46_TDSEL Tx duty control** 0: Keep 1: CLR bits
13:12	**GPIO42**	GPIO42_TDSEL	**Bitwise CLR operation of GPIO42_TDSEL Tx duty control** 0: Keep 1: CLR bits
11:10	**GPIO41**	GPIO41_TDSEL	**Bitwise CLR operation of GPIO41_TDSEL Tx duty control** 0: Keep 1: CLR bits
9:8	**GPIO36**	GPIO36_TDSEL	**Bitwise CLR operation of GPIO36_TDSEL Tx duty control** 0: Keep 1: CLR bits

Bit(s)	Mnemonic	Name	Description
7:6	**GPIO25**	GPIO25_TDSEL	**Bitwise CLR operation of GPIO25_TDSEL Tx duty control** 0: Keep 1: CLR bits
5:4	**GPIO18**	GPIO18_TDSEL	**Bitwise CLR operation of GPIO18_TDSEL Tx duty control** 0: Keep 1: CLR bits
3:2	**GPIO9**	GPIO9_TDSEL	**Bitwise CLR operation of GPIO9_TDSEL Tx duty control** 0: Keep 1: CLR bits
1:0	**GPIO8**	GPIO8_TDSEL	**Bitwise CLR operation of GPIO8_TDSEL Tx duty control** 0: Keep 1: CLR bits

A0020E00 CLK_OUT0 CLK Out Selection Control 00000004

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													**CLK_OUT0**			
Type													RW			
Reset													0	1	0	0

Overview: CLK OUT0 Setting

Bit(s)	Mnemonic	Name	Description
3:0	**CLK_OUT0**	CFG0	**Selects clock output for CLKO_0** [1]: 26Mhz clock [4]: 32Khz_clock others: debug clock

A0020E10 CLK_OUT1 CLK Out Selection Control 00000004

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													**CLK_OUT1**			
Type													RW			
Reset													0	1	0	0

Overview: CLK OUT1 setting

Bit(s)	Mnemonic	Name	Description
3:0	**CLK_OUT1**	CFG1	**Selects clock output for CLKO_1** [1]: 26Mhz clock [4]: 32Khz_clock others: debug clock

A0020E20 CLK_OUT2 CLK Out Selection Control 00000004

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													CLK_OUT2			
Type													RW			
Reset													0	1	0	0

Overview: CLK OUT2 setting

Bit(s)	Mnemonic	Name	Description
3:0	**CLK_OUT2**	CFG2	**Selects clock output for CLKO_2** [1]: 26Mhz clock [4]: 32Khz_clock others: debug clock

A0020E30 CLK_OUT3 CLK Out Selection Control 00000004

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													CLK_OUT3			
Type													RW			
Reset													0	1	0	0

Overview: CLK OUT3 setting

Bit(s)	Mnemonic	Name	Description
3:0	**CLK_OUT3**	CFG3	**Selects clock output for CLKO_3** [1]: 26Mhz clock [4]: 32Khz_clock others: debug clock

A0020E40 CLK_OUT4 CLK Out Selection Control 00000004

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													CLK_OUT4			
Type													RW			
Reset													0	1	0	0

Overview: CLK OUT4 setting

Bit(s)	Mnemonic	Name	Description
3:0	**CLK_OUT4**	CFG4	**Selects clock output for CLKO_4** [1]: 26Mhz clock [4]: 32Khz_clock others: debug clock

A0020E50 CLK_OUT5 CLK Out Selection Control 00000004

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													**CLK_OUT5**			
Type													RW			
Reset													0	1	0	0

Overview: CLK OUT5 setting

Bit(s)	Mnemonic	Name	Description
3:0	**CLK_OUT5**	CFG5	**Selects clock output for CLKO_5** [1]: 26Mhz clock [4]: 32Khz_clock others: debug clock

3.5 General-purpose Timer

3.5.1 General Descriptions

Three general-purpose timers are provided. Two timers are 16 bits long and one timer is 32 bits long. Each runs independently. GPT1 ~ 2 use 32k clock source to count, whereas GPT4 uses 26M clock source. The 26M clock source can be gated when the system enters the sleep mode, and this will cause GPT4 to stop counting, whereas 32k clock source is always toggling. GPT1 and GPT2 can operate in one of the two modes: one-shot mode and auto-repeat mode. GPT4 are free running timer. In the one-shot mode, when the timer counts down and reaches 0, it will be halted. In the auto-repeat mode, when the timer reaches 0, it will simply be reset to counting down the initial value and repeating the count-down to 0. This loop keeps repeating until the disabling signal is set to 1. Regardless of the timer's mode, if the countdown initial value (i.e. GPTIMER1_DAT for GPT1 or GPTIMER_DAT2 for GPT2) is written when the timer is running, the new initial value will not take effect until the next time the timer is restarted. In the auto-repeat mode, the new countdown start value is used on the next countdown iteration. Therefore, before enabling the timer, the desired values for GPTIMER_DAT and the GPTIMER_PRESCALER registers must first be set.

Note that GPTimer's design is inherited form MT6260, GPT3 is removed for achieving lower cost.

For other timers, register addresses are the same with those of MT6260.

3.5.2 Register Definition

Module name: GPTimer base address: (+A00C0000h)

Address	Name	Width	Register function
A00C0000	GPTIMER1_CON	32	GPT1 control register
A00C0004	GPTIMER1_DAT	32	GPT1 time-out interval register
A00C0008	GPTIMER2_CON	32	GPT2 control register
A00C000C	GPTIMER2_DAT	32	GPT2 time-out Interval register
A00C0010	GPTIMER_STA	32	GPT status register
A00C0014	GPTIMER1_PRESCALER	32	GPT1 prescaler register
A00C0018	GPTIMER2_PRESCALER	32	GPT2 prescaler register
A00C0028	GPTIMER4_CON	32	GPT4 control register
A00C002C	GPTIMER4_DAT	32	GPT4 data register

A00C0000 GPTIMER1_CON GPT1 Control Register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	EN	MODE														
Type	RW	RW														
Reset	0	0														

Bit(s)	Mnemonic	Name	Description
15	EN	EN	**Controls GPT1 to start counting or to stop** 0: Disable GPT1 1: Enable GPT1
14	MODE	MODE	**Controls GPT1 to count repeatedly (in a loop) or just one-shot** 0: One-shot mode is selected. 1: Auto-repeat mode is selected.

A00C0004 GPTIMER1_DAT GPT1 Time-out Interval Register 0000FFFF

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	CNT															
Type	RW															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit(s)	Mnemonic	Name	Description
15:0	**CNT**	CNT	**Initial counting value** GPT1 counts down from GPTIMER1_DAT. When GPT1 counts down to 0, a GPT1 interrupt will be generated.

A00C0008 GPTIMER2_CON GPT2 Control Register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**EN**	**MODE**														
Type	RW	RW														
Reset	0	0														

Bit(s)	Mnemonic	Name	Description
15	**EN**	EN	**Controls GPT2 to start counting or to stop** 0: Disable GPT2 1: Enable GPT2
14	**MODE**	MODE	**Controls GPT2 to count repeatedly (in a loop) or just one-shot** 0: One-shot mode is selected. 1: Auto-repeat mode is selected.

A00C000C GPTIMER2_DAT GPT2 Time-out Interval Register 0000FFFF

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**CNT**															
Type	RW															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit(s)	Mnemonic	Name	Description
15:0	**CNT**	CNT	**Initial counting value** GPT2 counts down from GPTIMER2_DAT. When GPT2 counts down to 0, a GPT2 interrupt will be generated.

A00C0010 GPTIMER_STA GPT Status Register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name															**GPT2**	**GPT1**
Type															RC	RC
Reset															0	0

Bit(s)	Mnemonic	Name	Description
1	**GPT2**	GPT2	**GP timer time-out status** Each flag is set when the corresponding timer count-down is completed. Can be cleared when the CPU reads the status register.
0	**GPT1**	GPT1	**GP timer time-out status** Each flag is set when the corresponding timer count-down is completed. Can be cleared when the CPU reads the status register.

A00C0014 GPTIMER1_PRESCALER GPT1 Prescaler Register 00000004

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														PRESCALER		
Type														RW		
Reset														1	0	0

Bit(s)	Mnemonic	Name	Description
2:0	**PRESCALER**	PRESCALER	**Controls the counting clock for GP timer 1** 0: 16,384Hz 1: 8,192Hz 2: 4,096Hz 3: 2,048Hz 4: 1,024Hz 5: 512Hz 6: 256Hz 7: 128Hz

A00C0018 GPTIMER2_PRESCALER GPT2 Prescaler Register 00000004

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														PRESCALER		
Type														RW		
Reset														1	0	0

Bit(s)	Mnemonic	Name	Description
2:0	**PRESCALER**	PRESCALER	**Controls the counting clock for GP timer 2** 0: 16,384Hz 1: 8,192Hz 2: 4,096Hz 3: 2,048Hz 4: 1,024Hz 5: 512Hz 6: 256Hz 7: 128Hz

A00C0028 GPTIMER4_CON GPT4 Control Register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name															LOCK	EN
Type															RW	RW
Reset															0	0

Bit(s)	Mnemonic	Name	Description
1	**LOCK**	LOCK	**Controls GPT4 EN bit can be modified or not** If LOCK = 0, EN can be modified. If LOCK = 1, the EN value will be fixed, and the LOCK bit will always be 1 and cannot be modified until hardware reset. 0: Unlock 1: Lock
0	**EN**	EN	**Controls GPT4 to start counting or to stop** 0: Disable GPT4 1: Enable GPT4

A00C002C GPTIMER4_DAT GPT4 Data Register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	CNT[31:16]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	CNT[15:0]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
31:0	**CNT**	CNT	If EN = 1, GPT4 will be a free running timer. This register records the GPT4 value. If EN = 0, this register will be cleared to 0. This register does not allow continuous read. It requires at least 1 26M clock cycle between 2 APB reads.

3.5.3 Application Note

When the GPT is in running status, GPTIMER_DAT cannot be configured. To start GPT1 or GPT2, SW should make sure the timer has finished counting for at least 3 cycles of the 32kHz clock.

3.6 MCU OSTIMER

3.6.1 Overview

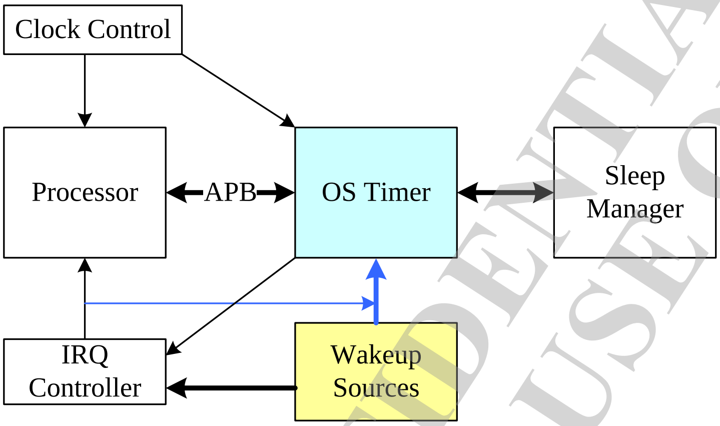


Figure 31. OS timer system view

The OS timer is a hardware timer which specifies the OS time frame duration and generates time-out interrupts by programming the frame counter number. The OS timer has the pause mode. The user can specify pause duration period before the pause mode, and the timer will resume from the pause mode by the external wakeup sources or when the pause duration is timed out.

3.6.2 Terminology

Table 46. Abbreviations

Abbreviation	
R/W	Read/Write
RO	Read only
WO	Write only
W1C	Write 1 to clear
R/W1C	Read/Write 1 to clear
FRC	Free running counter in the system
OST	OS timer

3.6.3 Introduction to Wakeup Source

The OS timer only accepts level trigger wakeup source. The wakeup sources are all treated as asynchronous input (will be changed) and will be synchronized by OST clock in OST wakeup source controller.

The possible wakeup sources are listed in the table below.

Table 47. Wakeup sources

No	Wakeup source
0	GPT
1	EINT
2	Timer trigger
3	KP
4	MSDC
5	ANALOG
6	DSP
7	MSDC2
8	SPISLV
9	Reserved
10	Reserved
11	DSP_ASYNC
12	Reserved
13	Reserved
14	Reserved
15	Reserved
16	Reserved

Table 48. Characteristics of wakeup sources

No	Wakeup source	Edge/Level	SW clear	Clock domain
0	GPT	Edge		F32k_CK
1	EINT	Edge/Level		F32k_CK
2	Timer trigger	Level		F32k_CK
3	KP	Edge		
4	MSDC	Level		
5	ANALOG	Edge/Level		
6	DSP	Level		DSP_CK
7	MSDC2	Level		
8	SPISLV	Level		
9	Reserved			
10	Reserved			
11	DSP_ASYNC	Level		DSP_CK
12	Reserved			
13	Reserved			
14	Reserved			
15	Reserved			
16	Reserved			

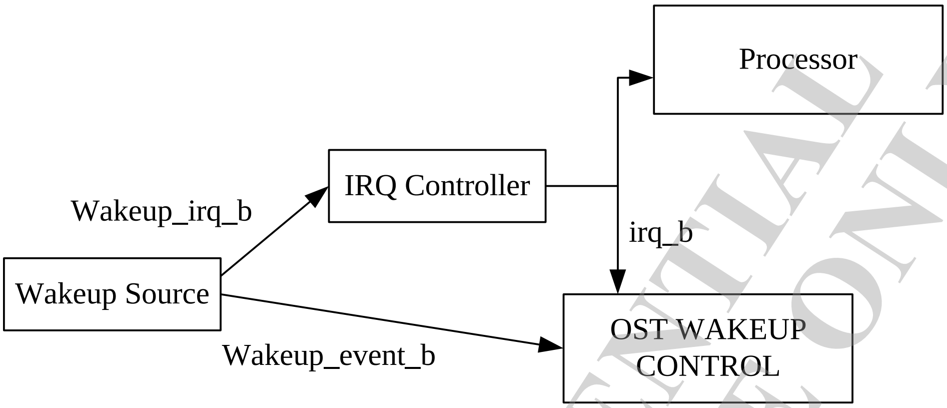


Figure 32. Wakeup event and Irq_b integration diagram

wakeup_irq_b may be asserted before *wakeup_event_b* is asserted from the wakeup source peripheral to ensure the software will enter wakeup ISR after the pause command is set.

Recommended software programming sequence:

1. I-BIT is set when a pause command is executed.
2. Set up IRQ mask registers to select IRQ wakeup sources
3. Pause criteria are met.
4. Set up pause command.
5. Confirm the pause command is executed at OST, and check if the pause command is pending or not.
6. Set up processor in request for interrupt state.
7. The processor clock will be off if there is no interrupt.
8. Check if the pause request command is completed (after the processor clock is active or resumes).
9. Clear I-BIT.

3.6.4 Register Definition

Address	Name	Width	Register function
0XA01F0000	OST_CON	32	**OS timer control register** Only updated when OST_CMD.OST_WR is set and OST_STA.WR_RDY is high.
0XA01F0004	OST_CMD	32	**OS timer command** Only valid when the write data BIT31 to BIT16 is 0x1153.
0XA01F0008	OST_STA	32	**OS timer status**
0XA01F000C	OST_FRM	32	**OS timer frame duration** Specifies OS timer frame duration by micron second resolution. The maximum duration is 8ms and minimum duration is 1ms. This register should be set before OS timer is enabled. The hardware will set up OST_ISR[0] periodically at frame duration period when the OS timer is enabled.
0XA01F0010	OST_FRM_F32K	32	**OS timer frame duration by 32K clock**

Address	Name	Width	Register function
			Specifies OS timer frame duration with 30.5176 micron second (32kHz) resolution. The maximum duration is 8ms and minimum duration is 1ms. This register should be set before the OS timer is enabled. OST_FRAM_F32K*30.5176us should be less than OST_FRM_NUM*OST_FRM*1us - 30.5176us. Set up OST_FRM_NUM if the frame duration is shorter than system settling time.
0XA01F0014	**OST_UFN**	32	**OS timer un-alignment frame number** Specifies OS timer un-alignment event frame number count. This register value is updated to OS timer only when OST_CMD.OST_WR is set. The read value of this register may not be the current OS time Un-alignment frame number. The hardware will count down the OST_UFN counter at frame time-out and stop the count-down when the current value is 0. When the OS timer is in the pause mode, this counter is still active and wakes up the OS timer when OST_UFN becomes 0 at frame time-out boundary. The software should enable the OS timer pause mode when OST_UFN is larger than 1, or the pause command will be ignored by the hardware.
0XA01F0018	**OST_AFN**	32	**OS timer alignment frame number** Specifies OS timer alignment event frame number count. This register value is updated to the OS timer only when OST_CMD.OST_WR is set. The read value of this register may not be the current OS time alignment frame number. The hardware will count down the OST_AFN counter at frame time-out and stop the count-down when the current value is 0. The hardware will set OST_ISR[1] when OST_AFN is 1 or 0 at frame time-out, and the OS timer is in normal mode. OST_AFN is current hardware AFN down counter value, and the software cannot read this register directly. The software can only read the current frame number through the OST_AFN_R register, and there is synchronization latency from OST_AFN to OST_AFN_R.
0XA01F001C	**OST_AFN_DLY**	32	**OS timer alignment frame delay number count** Specifies the OS timer alignment event frame delay count due to OS timer pause mode. The software has to use the OST_CMD.OST_RD command to update this register value, or the value may be out of date.
0XA01F0020	**OST_UFN_R**	32	**Current OS timer un-alignment frame number** Specifies the OS timer current un-alignment frame number. The software has to use the OST_CMD.OST_RD command to update this register value, or the value may be out of date.
0XA01F0024	**OST_AFN_R**	32	**Current OS timer alignment frame number** Specifies the OS timer current alignment frame number. The software has to use the OST_CMD.OST_RD command to update this register value, or the value may be out of date.
0XA01F0030	**OST_INT_MASK**	32	**OS timer interrupt mask** Specifies the OS timer interrupt mask control.
0XA01F0040	**OST_ISR**	32	**OS timer interrupt status** Specifies the OS timer interrupt status. The software has to write 1 at the corresponding bit to clear the interrupt status bit. OSR_ISR[2-0] are also cleared when the OSR_WR command is executed and AFN, UFN are updated.

Address	Name	Width	Register function
0XA01F0050	OST_EVENT_MASK	32	**OS timer event mask** Specifies which wakeup event will be masked to wake up the OS timer in the OS timer pause mode.
0XA01F0054	OST_WAKEUP_STA	32	**OS timer event wakeup status**
0XA01F0060	OST_DBG_WAKEUP	32	**OS timer debug wakeup**

0XA01F0000 OS Timer Control Register OST_CON

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														OST_DBG	UFN_DOWN	EN
Type														RW	RW	RW
Reset														0	1	0

Overview: This registers is only updated when OST_CMD.OST_WR is set and OST_STA.WR_RDY is high.

Bit(s)	Name	Description
2	OST_DBG	**Enables OST wakeup debugging function** 0: Disable 1: Enable
1	UFN_DOWN	**Enables OST_UFN count-down** 0: Disable OST_UFN count-down 1: Enable OST_UFN count-down
0	EN	**Enables OS timer** 0: OS timer is disabled. Then all internal timers are stopped. 1: OS Timer is enabled. The software has to ensure OST_AFN, OST_UFN, OST_FRAM are configured before enabling the OS timer.

0XA01F0004 OS Timer Command OST_CMD

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	OST_KEY															
Type	WO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	OST_CON_WR	OST_AFN_WR	OST_UFN_WR											OST_WR	OST_RD	PAUSE_STR
Type	RW	RW	RW											WO	WO	WO
Reset	0	0	0											0	0	0

Overview: The command is only valid when the write data BIT31 to BIT16 is 0x1153.

Bit(s)	Name	Description
31:16	OST_KEY	
15	OST_CON_WR	Updates OST_CON when the OST_WR command is active.
14	OST_AFN_WR	Updates OST_AFN when the OST_WR command is active.
13	OST_UFN_WR	Updates OST_UFN when the OST_WR command is active.
2	OST_WR	Write 1 to this bit to update the bus clock domain OS timer configuration into the OST SYSCLK domain
1	OST_RD	Write 1 to this bit to update the current OS timer status to the bus clock domain
0	PAUSE_STR	Write 1 to this bit to enable the OS timer pause function. The command will be ignored if the current OST UFN is less than 2. The software has to ensure OST_CMD.OST_WR is completed before the next software pause sequence.

0XA01F0008 **OS Timer Status** **OST_STA**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	CPU_SLEEP									AFN_DLY_OVER		PAUSE_REQ			CMD_CPL	READY
Type	RO									RO		RO			RO	RO
Reset	0									0		0	0		1	0

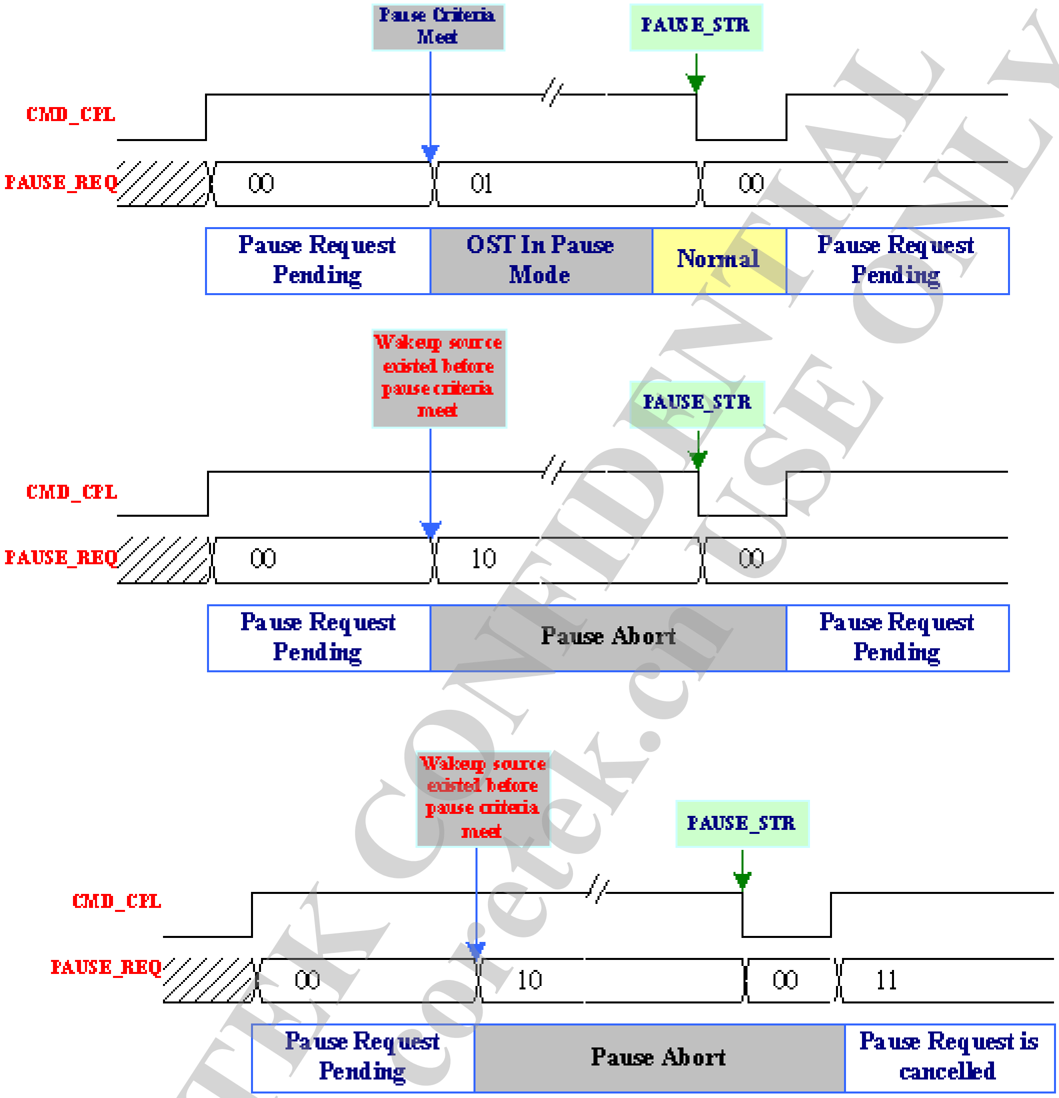


Figure 33. Pause command complete and pause request state

Bit(s)	Name	Description
15	CPU_SLEEP	**The processor is in the sleep mode. (for debugging)** 0: Active 1: Sleep
6	AFN_DLY_OVER	**AFN_DLY counter overflows. This bit is cleared when AFN is updated.** 0: Does not overflow 1: Overflow
4:3	PAUSE_REQ	**An OS timer pause request is pending. A pause command will be completed when the pause command is set.** 1. Processor is in the sleep mode (processor clock is off), UFN ≥ 2 and no wakeup sources 2. Any wakeup sources are sensed after the pause command is set. 3. UFN < 2 00: The last pause command request is not completed yet (CP15 is not enable

Bit(s)	Name	Description
		and no wakeup source). 01: The last pause command request is completed with OST pause mode being active. 10: The last pause command request is completed with wakeup sources. 11: The last pause command request is completed with UFN < 2.
1	CMD_CPL	**OST command is completed. It takes several clocks from OST_CMD being updated to command being active.** 0: OST command is not completed. 1: OST command is completed.
0	READY	**OS timer status** 0: OST is in pause mode. 1: OST is in normal mode.

0XA01F000C OS Timer Frame Duration OST_FRM

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name				**OST_FRM**												
Type				RW												
Reset				1	0	0	0	0	0	1	0	0	1	0	0	1

Overview: This register specifies the OS timer frame duration by micron second resolution. The maximum duration is 8ms and minimum duration is 1ms. This register should be set before the OS timer is enabled. The hardware will set up OST_ISR[0] periodically at frame duration period when the OS timer is enabled.

Bit(s)	Name	Description
12:0	OST_FRM	

0XA01F0010 OS Timer Frame Duration by 32K Clock OST_FRM_F32K

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**OST_UFN**								**OST_FRM_F32K**							
Type	RW								RW							
Reset	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: This register specifies the OS timer frame duration with 30.5176 micron second (32kHz) resolution. The maximum duration is 8ms and minimum duration is 1ms. This register should be set up before the OS timer is enabled. OST_FRAM_F32K*30.5176us should be less than OST_FRM_NUM*OST_FRM*1us - 30.5176us. Set up OST_FRM_NUM if the frame duration is shorter than the system settling time.

Bit(s)	Name	Description

Bit(s)	Name	Description
15:12	OST_UFN	
11:0	OST_FRM_F32K	

0XA01F0014 OS Timer Un-alignment Frame Number OST_UFN

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	OST_UFN[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	OST_UFN[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: This register specifies the OS timer un-alignment event frame number count. This register value is updated to the OS timer only when OST_CMD.OST_WR is set. The read value of this register may not be the current OS time un-alignment frame number. The hardware will count down the OST_UFN counter at frame time-out and stop the count-down when the current value is 0. When the OS timer is in the pause mode, this counter is still active and wakes up the OS timer when OST_UFN becomes 0 at frame time-out boundary. The software should enable the OS timer pause mode when OST_UFN is larger than 1, or the pause command will be ignored by hardware.

Bit(s)	Name	Description
31:0	OST_UFN	

0XA01F0018 OS Timer Alignment Frame Number OST_AFN

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	OST_AFN[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	OST_AFN[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: This register specifies the OS timer alignment event frame number count. This register value is updated to the OS timer only when OST_CMD.OST_WR is set. The read value of this register may not be the current OS time alignment frame number. The hardware will count down the OST_AFN counter at frame time-out and stop the count-down when the current value is 0. The hardware will set up OST_ISR[1] when OST_AFN is 1 or 0 at frame time-out and the OS timer is in the normal mode. OST_AFN is the current hardware AFN down counter value, and the software cannot read this register directly. The software can only read the current frame number through the OST_AFN_R register, and there is synchronization latency from OST_AFN to OST_AFN_R.

Bit(s)	Name	Description
31:0	OST_AFN	

0XA01F001C OS Timer Alignment Frame Delay Number Count OST_AFN_DLY

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	OST_AFN_DLY[31:16]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	OST_AFN_DLY[15:0]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: This register specifies the OS Timer Alignment event frame delay count due to OS timer pause mode. Software has to use OST_CMD.OST_RD command to update this register value, or the value maybe out of date.

Bit(s)	Name	Description
31:0	OST_AFN_DLY	

0XA01F0020 Current OS Timer Un-alignment Frame Number OST_UFN_R

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	OST_UFN_R[31:16]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	OST_UFN_R[15:0]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: This register specifies the OS timer current un-alignment frame number. The software has to use the OST_CMD.OST_RD command to update this register value, or the value may be out of date.

Bit(s)	Name	Description
31:0	OST_UFN_R	

0XA01F0024 Current OS Timer Alignment Frame Number OST_AFN_R

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	OST_AFN_R[31:16]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	OST_AFN_R[15:0]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: This register specifies the OS timer current alignment frame number. The software has to use the OST_CMD.OST_RD command to update this register value, or the value may be out of date.

Bit(s)	Name	Description
31:0	OST_AFN_R	

0XA01F0030 **OS Timer Interrupt Mask** **OST_INT_MASK**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name												**OST_INT_MASK**				
Type												RW				
Reset												1	1	1	1	1

Overview: This register specifies the OS timer interrupt mask control.

Bit(s)	Name	Description
4:0	OST_INT_MASK	0: Mask OS timer frame time-out interrupt 1: Mask OS timer alignment frame time-out interrupt 2: Mask OS timer un-alignment frame time-out interrupt 3: Mask OS timer pause abort interrupt 4: Mask OS timer pause interrupt

0XA01F0040 **OS Timer Interrupt Status** **OST_ISR**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name												**OST_ISR**				
Type												W1C				
Reset												0	0	0	0	0

Overview: This register specifies the OS timer interrupt status. The software has to write 1 to the corresponding bit to clear the interrupt status bit. OSR_ISR[2-0] are also cleared when the OSR_WR command is executed and AFN, UFN are updated.

Bit(s)	Name	Description
4:0	OST_ISR	0: OS timer frame time-out interrupt status 1: OS timer alignment frame time-out interrupt status 2: OS timer un-alignment frame time-out interrupt status 3: OS timer pause abort interrupt status 4: OS timer pause interrupt status

0XA01F0050 **OS Timer Event Mask** **OST_EVENT_MASK**

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

Name														OST_EVENT_MASK [18:16]		
Type														RW		
Reset														0	1	1
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	OST_EVENT_MASK[15:0]															
Type	RW															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Overview: This register specifies which wakeup event will be masked to wake up the OS timer in the OS timer pause mode.

Bit(s)	Name	Description
18:0	OST_EVENT_MASK	

0XA01F0054 OS Timer Event Wakeup Status OST_WAKEUP_STA

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name														OST_WAKEUP_STA [18:16]		
Type														RO		
Reset														0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	OST_WAKEUP_STA[15:0]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
18:0	OST_WAKEUP_STA	

0XA01F0060 OS Timer Debug Wakeup OST_DBG_WAKEOSUP

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	CIRQ_MASK_EN													OST_DBG_WAKEUP[18:16]		
Type	RW															
Reset	0													0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	OST_DBG_WAKEUP[15:0]															
Type																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Name	Description
31	CIRQ_MASK_EN	0: Disable cirq mask function 1: Enable cirq mask function
18:0	OST_DBG_WAKEUP	Controls wakeup status in debug timing event1

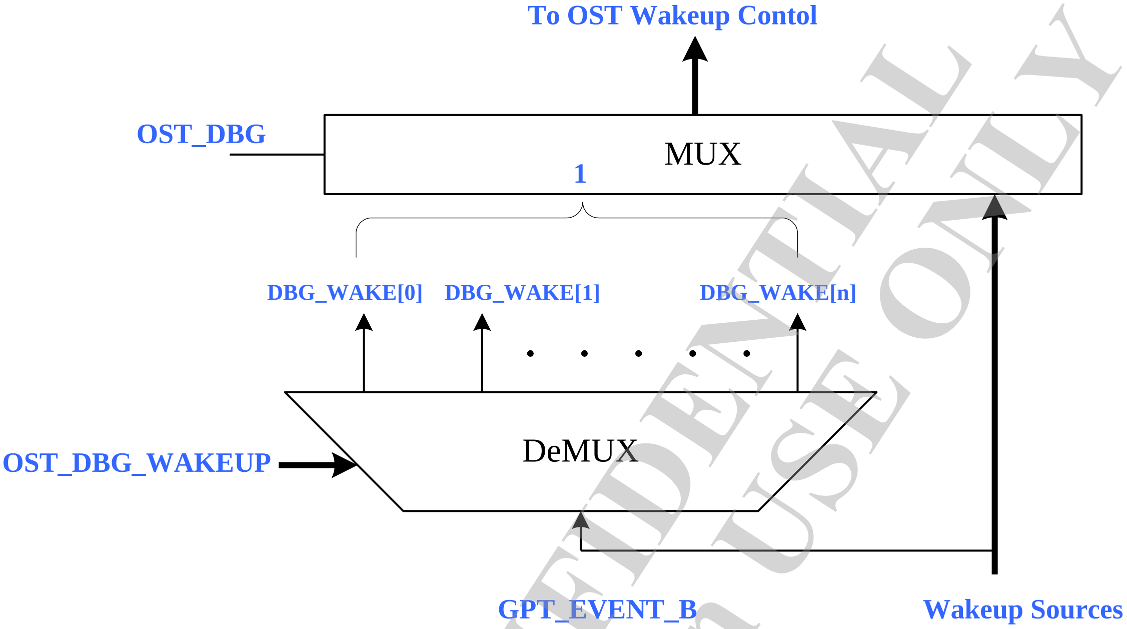


Figure 34. Debug wakeup events

3.7 UART1

3.7.1 General Description

The baseband chipset houses two UARTs. The UARTs provide full duplex serial communication channels between baseband chipset and external dev ices.

The UART has M16C450 and M16550A modes of operation, which are compatible with a range of standard software drivers. The extensions have been designed to be broadly software compatible with 16550A variants, but certain areas offer no consensus.

In common with the M16550A, the UART supports word lengths **from 5 to 8 bits, an optional parity bit** and one or two stop bits, and is fully programmable by an 8-bit CPU interface. A 16-bit programmable baud rate generator and an 8-bit scratch register are included, together with separate transmit and receive FIFOs. 8 modem control lines and a diagnostic loop-back mode are provided. UART also includes two DMA handshake lines, indicating when the FIFOs are ready to transfer data to the CPU. Interrupts can be generated from any of the 10 sources.

Note that UART is designed so that all internal operation is synchronized by the CLK signal. This synchronization results in minor timing differences between the UART and industry standard 16550A device, which means that the core is not clock for clock identical to the original device.

After a hardware reset, UART will be in M16C450 mode. Its FIFOs can then be enabled and UART can enter M16550A mode. UART has further functions beyond the M16550A mode. Each of the extended functions can be selected individually under software control.

UART provides more powerful enhancements than the industry-standard 16550:

- Hardware flow control. This feature is very useful when the ISR latency is hard to predict and control in the embedded applications. The MCU is relieved of having to fetch the received data within a fixed amount of time.

Note that in order to enable any of the enhancements, the Enhanced Mode bit, EFR[4], must be set. If EFR[4] is not set, IER[7:5], FCR[5:4], IIR[5:4] and MCR[7:6] cannot be written. The enhanced mode bit ensures that UART is backward compatible with the software that has been written for 16C450 and 16550A devices. **Figure 35** is the block diagram of the UART1 device.

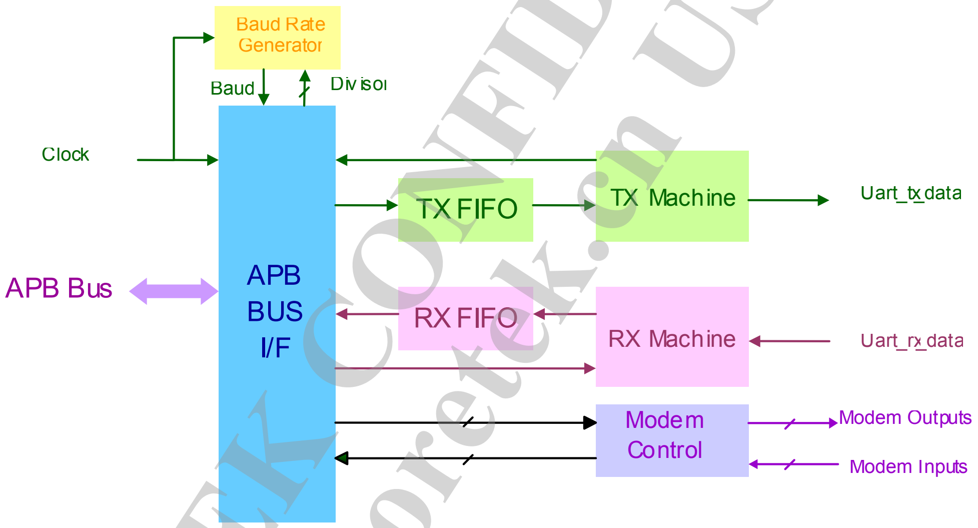


Figure 35. Block Diagram of UART1

3.7.2 Register Definition

Module name: UART1 Base address: (+A0080000h)

Address	Name	Width	Register Function
A0080000	**RBR**	8	**RX Buffer Register** Note: Only when LCR[7] = 0.
A0080000	**THR**	8	**TX Holding Register** Note: Only when LCR[7] = 0
A0080000	**DLL**	8	**Divisor Latch (LS)** Divides the bclk frequency Note: Nodified when LCR[7]!=0
A0080004	**IER**	8	**Interrupt Enable Register** Note: Only when LCR[7] = 0. By storing 1 to a specific bit position, the interrupt associated with that bit is enabled. Otherwise, the interrupt will be disabled.

Address	Name	Width	Register Function
			IER[3:0] are modified w hen LCR[7] = 0. IER[7:4] are modified w hen LCR[7] = 0 & EFR[4] = 1.
A0080004	DLM	8	**Divisor Latch (MS)** uesd to divid the bclk frequency .*NOTE: modified w hen LCR[7]!=0
A0080008	IIR	8	**Interrupt Identification Register** Note: Only w hen LCR!=BF'h. Priority is from high to low as the follow ing. IIR[5:0]=0X1: No interrupt pending IIR[5:0]=0X6: Line status interrupt (under IER[2]=1) IIR[5:0]=0Xc: RX data time-out interrupt (under IER[0]=1) IIR[5:0]=0X4: RX data are placed in the RX bBuffer register or the RX trigger level is reached. (under IER[0]=1). IIR[5:0]=0X2: TX holding register is empty or the contents of the TX FIFO have been reduced to its trigger level (under IER[1]=1). IIR[5:0]=0X10: XOFF character received (under IER[5]=1, EFR[4] = 1).
A0080008	FCR	8	**FIFO Control Register** FCR is used to control the trigger levels of the FIFOs or flush the FIFOs. FCR[7:6] is modified w hen LCR != BFh FCR[5:4] is modified w hen LCR != BFh & EFR[4] = 1 FCR[4:0] is modified w hen LCR != BFh
A0080008	EFR	8	**Enhanced Feature Register** Note: Only w hen LCR=BF'h
A008000C	LCR	8	**Line Control Register** Determines characteristics of serial communication signals.
A0080010	MCR	8	**Modem Control Register** Controls interface signals of the UART. MCR[5:0] are modified w hen LCR != 8'hBF, MCR[7] can be read w hen LCR != 8'hBF & EFR[4] = 1.
A0080010	XON1	8	**XON1 Char Register** Note: XON1 is modified only w hen LCR=BF'h.
A0080014	LSR	8	**Line Status Register** Modified w hen LCR != BFh.
A0080018	XOFF1	8	**XOFF1 Char Register** *Note: XOFF1 is modified only w hen LCR=BF'h.
A008001C	SCR	8	**Scratch Register** A general purpose read/w rite register. After reset, its value w ill be un-defined. Modified w hen LCR != BFh.
A0080020	AUTOBAUD_EN	8	**Auto Baud Detect Enable Register**
A0080024	HIGHSPEED	8	**High Speed Mode Register**
A0080028	SAMPLE_COUNT	8	**Sample Counter Register** When HIGHSPEED=3, sample_count w ill be the threshold value for UART sample counter (sample_num). Count from 0 to sample_count.
A008002C	SAMPLE_POINT	8	**Sample Point Register** When HIGHSPEED=3, UART gets the input data w hen sample_count=sample_num, e.g. system clock = 13MHz,

Address	Name	Width	Register Function
			921600 = 13000000/14. Therefore, sample_count = 13, and sample point = 6 (sampling the central point to decrease the inaccuracy) SAMPLE_POINT is usually (SAMPLE_COUNT-1)/2 without decimal.
A0080030	AUTOBAUD_REG	8	**Auto Baud Monitor Register** the autobaud detection state ,it will not change until enable the autobaud_en again.
A0080034	RATEFIX_AD	8	**Clock Rate Fix Register**
A0080038	AUTOBAUDSAMPLE	8	**Auto Baud Sample Register** Since the system clock may change, autobaud sample duration should change as the system clock changes. When system clock = 13MHz, autobaudsample = 6; when system clock = 26MHz, autobaudsample = 13. When system clock = 52MHz, autobaudsample = 27.
A008003C	GUARD	8	**Guard Time Added Register**
A0080040	ESCAPE_DAT	8	**Escape Character Register**
A0080044	ESCAPE_EN	8	**Escape Enable Register**
A0080048	SLEEP_EN	8	**Sleep Enable Register**
A008004C	DMA_EN	8	**DMA Enable Register**
A0080050	RXTRI_AD	8	**Rx Trigger Address**
A0080054	FRACDIV_L	8	**Fractional Divider LSB Address**
A0080058	FRACDIV_M	8	**Fractional Divider MSB Address**
A008005C	FCR_RD	8	**FIFO Control Register**

A0080000 RBR RX Buffer Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									RBR							
Type									RU							
Reset									0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7:0	RBR	**Read-only register** The received data can be read by accessing this register. Only when LCR[7] = 0.

A0080000 THR TX Holding Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									THR							
Type									WO							
Reset									0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7:0	THR	**TX Holding Register** Write-only register. The data to be transmitted are written to this register and sent to the PC via serial communication.

Bit(s)	Name	Description
		Only when LCR[7] = 0.

A0080000 DLL Divisor Latch (LS) 01

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									DLL							
Type									RW							
Reset									0	0	0	0	0	0	0	1

Bit(s)	Name	Description
7:0	DLL	**Divisor latch low 8-bit data** *Note: Modified when LCR[7]!=0.*

A0080004 IER Interrupt Enable Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									CTSI	RTSI	XOFFI		EDSSI	ELSI	ETBEI	ERBFI
Type									RW	RW	RW		RW	RW	RW	RW
Reset									0	0	0		0	0	0	0

Bit(s)	Name	Description
7	CTSI	**Masks an interrupt that is generated when a rising edge is detected on the CTS modem control line.** *Note: This interrupt is only enabled when hardware flow control is enabled.* 0: Mask an interrupt that is generated when a rising edge is detected on the CTS modem control line. 1: Unmask an interrupt that is generated when a rising edge is detected on the CTS modem control line.
6	RTSI	**Masks an interrupt that is generated when a rising edge is detected on the RTS modem control line.** *Note: This interrupt is only enabled when hardware flow control is enabled.* 0: Mask an interrupt that is generated when a rising edge is detected on the RTS modem control line. 1: Unmask an interrupt that is generated when a rising edge is detected on the RTS modem control line.
5	XOFFI	**Masks an interrupt that is generated when an XOFF character is received.** *Note: This interrupt is only enabled when software flow control is enabled.* 0: Mask an interrupt that is generated when an XOFF character is received. 1: Unmask an interrupt that is generated when an XOFF character is received.
3	EDSSI	**When set to 1, an interrupt will be generated if DCTS (MSR[0]) becomes set.** 0: No interrupt is generated if DCTS (MSR[0]) becomes set. 1: An interrupt is generated if DCTS (MSR[0]) becomes set.
2	ELSI	**When set to1, an interrupt will be generated if BI, FE, PE or OE (LSR[4:1]) becomes set.** 0: No interrupt is generated if BI, FE, PE or OE (LSR[4:1]) becomes set. 1: An interrupt is generated if BI, FE, PE or OE (LSR[4:1]) becomes set.
1	ETBEI	**When set to 1, an interrupt will be generated if the TX holding register is empty or the contents of the TX FIFO have been reduced to its trigger level.** 0: No interrupt will be generated if the TX holding register is empty or the contents of the TX FIFO have been reduced to its trigger level. 1: An interrupt will be generated if the TX folding register is empty or the contents of the TX FIFO have been reduced to its trigger level.
0	ERBFI	**When set to 1, an interrupt will be generated if RX data are placed in RX buffer register or the RX trigger level is reached.**

Bit(s)	Name	Description
		0: No interrupt will be generated if RX data are placed in the RX buffer register or the RX trigger level is reached. 1: An interrupt will be generated if RX Data are placed in the RX buffer register or the RX trigger level is reached.

A0080004 DLM Divisor Latch (MS) 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									DLM							
Type									RW							
Reset									0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7:0	DLM	**Divisor latch high 8-bit data** *Note: Modified when LCR[7]!=0. DLL & DLM can only be updated when DLAB(LCR[7]) is set to 1. Division by 1 will generate a BAUD signal that is constantly high. DLL & DLM setting formula is {DLH,DLL}=(system clock frequency/baud_pulse/baud_rate).* When RATE_FIX(RATEFIX_AD[0])=0, system clock frequency = 52MHz. When RATE_FIX(RATEFIX_AD[0])=1 and RATE_FIX(RATEFIX_AD[2])=0, system clock frequency = 26MHz. When RATE_FIX(RATEFIX_AD[0])=1 and RATE_FIX(RATEFIX_AD[2])=1, system clock frequency = 13MHz. For baud_pulse value, refer to HIGH_SPEED(offset=24H) register For example, when at 52MHz, default speed mode and 115200 baud rate, {DLH,DLL}=52MHz/16/115200=28.

A0080008 IIR Interrupt Identification Register 01

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									FIFOE		ID					
Type									RO		RU					
Reset									0	0	0	0	0	0	0	1

Bit(s)	Name	Description
7:6	FIFOE	
5:0	ID	**IIR[5:0] Priority level interrupt source** 000001 - No interrupt pending 000110 1 Line status interrupt: BI, FE, PE or OE set in LSR. (Under IER[2]=1) 001100 2 RX data time-out: Time-out on character in RX FIFO. (Under IER[0]=1) 000100 3 RX data received: RX data received or RX trigger level reached. (Under IER[0]=1) 000010 4 TX holding register empty: 000000 5 Modem status change: DCTS set in MSR. (Under IER[3]=1) TX Holding Register empty or TX FIFO trigger level reached. (Under IER[1]=1) 010000 6 Software flow control: XOFF Character received. (Under IER[5]=1) 100000 7 Hardware flow control: CTS or RTS Rising Edge. (Under IER[7]=1 or IER[6]=1) **Line status interrupt:** A RX line status interrupt (IIR[5:0] = 000110b) will be generated if ELSI (IER[2]) is set and any of BI, FE, PE or OE (LSR[4:1]) becomes set. The interrupt is cleared by reading the line status register. **RX data time-out interrupt:** When the virtual FIFO mode is disabled, RX data time-out

Bit(s)	Name	Description
		interrupt will be generated if all of the following conditions are applied: 1. FIFO contains at least one character. 2. The most recent character is received longer than four character periods ago (including all start, parity and stop bits); 3. The most recent CPU read of the FIFO is longer than four character periods ago. The timeout timer is restarted upon receipt of a new byte from the RX shift register or upon a CPU read from the RX FIFO. The RX data time-out interrupt is enabled by setting EFRBI (IER[0]) to 1 and is cleared by reading RX FIFO. When the virtual FIFO mode is enabled, RX data time-out interrupt will be generated if all of the following conditions are applied: 1. FIFO is empty. 2. The most recent character is received longer than four character periods ago (including all start, parity and stop bits). 3. The most recent CPU read of the FIFO is longer than four character periods ago. The timeout timer is restarted upon receipt of a new byte from the RX shift register or reading DMA_EN register. The RX Data Timeout Interrupt is enabled by setting EFRBI (IER[0]) to 1 and is cleared by reading DMA_EN register. **RX data received interrupt:** A RX received interrupt (IER[5:0] = 000100b) is generated if EFRBI (IER[0]) is set and either RX data are placed in the RX buffer register or the RX trigger level is reached. The interrupt is cleared by reading the RX buffer register or the RX FIFO (if enabled). **TX holding register empty interrupt:** A TX holding register empty interrupt (IIR[5:0] = 000010b) is generated if ETRBI (IER[1]) is set and either the TX holding register is empty or the contents of the TX FIFO are reduced to its trigger level. The interrupt is cleared by writing to the TX holding register or TX FIFO if FIFO is enabled. **Modem status change interrupt:** A modem status change Interrupt (IIR[5:0] = 000000b) will be generated if EDSSI (IER[3]) is set and e DCTS (MSR[3:0]) becomes set. The interrupt is cleared by reading the modem status register. **Software flow control interrupt:** A software flow control interrupt (IIR[5:0] = 010000b) will be generated if the software flow control is enabled and XOFFI (IER[5]) becomes set, indicating that an XOFF character has been received. The interrupt is cleared by reading the interrupt identification register. **Hardware flow control interrupt:** A hardware flow control interrupt (IER[5:0] = 100000b) will be generated if the hardware flow control is enabled and either RTSI (IER[6]) or CTSI (IER[7]) becomes set indicating that a rising edge has been detected on either the RTS/CTS modem control line. The interrupt is cleared by reading the interrupt identification register.

A0080008 FCR FIFO Control Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									RFTL1_RFTL0		TFTL1_TFTL0			CLRT	CLRR	FIFOE
Type									WO		WO			WO	WO	WO
Reset									0	0	0	0		0	0	0

Bit(s)	Name	Description
7:6	RFTL1_RFTL0	**RX FIFO trigger threshold** RX FIFO contains total 32 bytes. 0: 1 1: 6 2: 12

Bit(s)	Name	Description
		3: RXTRIG
5:4	TFTL1_TFTL0	**TX FIFO trigger threshold** TX FIFO contains total 32 bytes. 0: 1 1: 4 2: 8 3: 14
2	CLRT	**Control bit to clear TX FIFO** 0: No effect 1: Clear TX FIFO
1	CLRR	**Control bit to clear RX FIFO** 0: No effect 1: Clear RX FIFO
0	FIFOE	**Enables FIFO** This bit must be set to 1 for any of other bits in the registers to have any effect. 0: Disable both RX and TX FIFOs. 1: Enable both RX and TX FIFOs.

A0080008 EFR Enhanced Feature Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									AUTO_CTS	AUTO_RTS		ENABLE_E	SW_FLOW_CONT			
Type									RW	RW		RW	RW			
Reset									0	0		0	0	0	0	0

Bit(s)	Name	Description
7	AUTO_CTS	**Enables hardware transmission flow control** 0: Disable 1: Enable
6	AUTO_RTS	**Enables hardware reception flow control** 0: Disable 1: Enable
4	ENABLE_E	**Enables enhancement feature** 0: Disable 1: Enable
3:0	SW_FLOW_CONT	**Software flow control bits** 00xx: No TX flow control 01xx: No TX flow control 10xx: Transmit XON1/XOFF1 as flow control bytes xx00: No RX flow control xx01: No RX flow control xx10: Receive XON1/XOFF1 as flow control bytes

A008000C LCR Line Control Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									DLAB	SB	SP	EPS	PEN	STB	WLS1_WLS0	
Type									RW	RW	RW	RW	RW	RW	RW	
Reset									0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7	DLAB	**Divisor latch access bit**

Bit(s)	Name	Description
		0: RX and TX registers are read/written at Address 0 and the IER register is read/written at Address 4. 1: Divisor Latch LS is read/written at Address 0 and the Divisor Latch MS is read/written at Address 4.
6	SB	**Set break** 0: No effect 1: SOUT signal is forced to the 0 state.
5	SP	**Stick parity** 0: No effect. 1: The parity bit is forced to a defined state, depending on the states of EPS and PEN: If EPS=1 & PEN=1, the parity bit will be set and checked = 0. If EPS=0 & PEN=1, the parity bit will be set and checked = 1.
4	EPS	**Selects even parity** 0: When EPS=0, an odd number of ones is sent and checked. 1: When EPS=1, an even number of ones is sent and checked.
3	PEN	**Enables parity** 0: The parity is neither transmitted nor checked. 1: The parity is transmitted and checked.
2	STB	**Number of STOP bits** 0: One STOP bit is always added. 1: Two STOP bits are added after each character is sent; unless the character length is 5 when 1 STOP bit is added.
1:0	WLS1_WLS0	**Selects word length** 0: 5 bits 1: 6 bits 2: 7 bits 3: 8 bits

A0080010 MCR Modem Control Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									**XOFF_STATUS**			**Loop**			**RTS**	
Type									RU			RW			RW	
Reset									0			0			0	

Bit(s)	Name	Description
7	XOFF_STATUS	**Read-only bit** 0: When an XON character is received. 1: When an XOFF character is received.
4	Loop	**Loop-back control bit** 0: No loop-back is enabled. 1: Loop-back mode is enabled.
1	RTS	**Controls the state of the output NRTS, even in loop mode.** 0: RTS will always output 1. 1: RTS's output will be controlled by flow control condition.

A0080010 XON1 XON1 Char Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									**XON1**							
Type									RW							
Reset									0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7:0	XON1	**XON1 character for software flow control** Modified only when LCR=BF'h.

A0080014 LSR Line Status Register 60

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									**FIFOERR**	**TEMT**	**THRE**	**BI**	**FE**	**PE**	**OE**	**DR**
Type									RU	RU	RU	RU	RU	RU	RU	RU
Reset									0	1	1	0	0	0	0	0

Bit(s)	Name	Description
7	FIFOERR	**RX FIFO error indicator** 0: No PE, FE, BI set in the RX FIFO. 1: Set to 1 when there is at least one PE, FE or BI in the RX FIFO.
6	TEMT	**TX holding register (or TX FIFO) and the TX shift register are empty.** 0: Empty conditions below are not met. 1: If FIFOs are enabled, the bit will be set whenever the TX FIFO and the TX shift register are empty. If FIFOs are disabled, the bit will be set whenever TX holding register and TX shift register are empty.
5	THRE	**Indicates if there is room for TX holding register or TX FIFO is reduced to its trigger level** 0: Reset whenever the contents of the TX FIFO are more than its trigger level (FIFOs are enabled), or whenever TX holding register is not empty (FIFOs are disabled). 1: Set whenever the contents of the TX FIFO are reduced to its trigger level (FIFOs are enabled), or whenever TX holding register is empty and ready to accept new data (FIFOs are disabled).
4	BI	**Break interrupt** 0: Reset by the CPU reading this register 1: If the FIFOs are disabled, this bit will be set whenever the SIN is held in the 0 state for more than one transmission time (START bit + DATA bits + PARITY + STOP bits). If the FIFOs are enabled, this error will be associated with a corresponding character in the FIFO and is flagged when this byte is at the top of the FIFO. When a break occurs, only one zero character is loaded into the FIFO: The next character transfer is enabled when SIN goes into the marking state and receives the next valid start bit.
3	FE	**Framing error** 0: Reset by the CPU reading this register 1: If the FIFOs are disabled, this bit will be set if the received data do not have a valid STOP bit. If the FIFOs are enabled, the state of this bit will be revealed when the byte it refers to is the next to be read.
2	PE	**Parity error** 0: Reset by the CPU reading this register 1: If the FIFOs are disabled, this bit will be set if the received data do not have a valid parity bit. If the FIFOs are enabled, the state of this bit will be revealed when the referred byte is the next to be read.
1	OE	**Overrun error** 0: Reset by the CPU reading this register. 1: If the FIFOs are disabled, this bit will be set if the RX buffer is not read by the CPU before the new data from the RX shift register overwrites the previous contents. If the FIFOs are enabled, an overrun error will occur when the RX FIFO is full and the RX shift register becomes full. OE will be set as soon as this happens. The character in the shift register is then overwritten, but not transferred to the FIFO.
0	DR	**Data ready** 0: Cleared by the CPU reading the RX buffer or by reading all the FIFO bytes. 1: Set by the RX buffer becoming full or by the FIFO becoming no empty.

A0080018 XOFF1 XOFF1 Char Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									XOFF1							
Type									RW							
Reset									0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7:0	XOFF1	**XOFF1 character for software flow control** Modified only when LCR=BF'h.

A008001C SCR Scratch Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									SCR							
Type									RW							
Reset									0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7:0	SCR	**General purpose read/write register** After reset, its value will be undefined. Modified when LCR != BFh.

A0080020 AUTOBAUD_EN Auto Baud Detect Enable Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														SLEEP_ACK_SEL	AUTOBAUD_SEL	AUTOBAUD_EN
Type														RW	RW	RW
Reset														0	0	0

Bit(s)	Name	Description
2	SLEEP_ACK_SEL	**Selects sleep ack when autobaud_en** 0: Support sleep_ack when autobaud_en is opened . 1: Does not support sleep_ack when autobaud_en is opened .
1	AUTOBAUD_SEL	**Selects auto-baud** 0: Support standard baud rate detection 1: Support non_standard baud rate detection (support baud from 110 to 115200; recommended to use 52MHz to auto fix).
0	AUTOBAUD_EN	**Auto-baud enabling signal** 0: Disable auto-baud function 1: Enable auto-baud function (UARTn+0024h SPEED should be set to 0.) *Note: When AUTOBAUD_EN is active, there should not be A*/a* char before the auto baud char AT/at. If A*/a* is Inevitable, autobaud will fail and please disable AUTOBAUD_EN to reset the autobaud feature and autobaud_en again.*

A0080024 HIGHSPEED High Speed Mode Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name															SPEED	
Type															RW	

Reset															0	0

Bit(s)	Name	Description

1:0	SPEED	**UART sample counter base**
		0: Based on 16*baud_pulse, baud_rate = system clock frequency/16/{DLH, DLL}
		1: Based on 8*baud_pulse, baud_rate = system clock frequency/8/{DLH, DLL}
		2: Based on 4*baud_pulse, baud_rate = system clock frequency/4/{DLH, DLL}
		3: Based on sampe_count * baud_pulse, baud_rate = system clock frequency / (sampe_count+1)/{DLM, DLL}

A0080028 SAMPLE_COUNT Sample Counter Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									**SAMPLECOUNT**							
Type									RW							
Reset									0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7:0	SAMPLECOUNT	Only useful when HIGHSPEED mode = 3.

A008002C SAMPLE_POINT Sample Point Register FF

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									**SAMPLEPOINT**							
Type									RW							
Reset									1	1	1	1	1	1	1	1

Bit(s)	Name	Description
7:0	SAMPLEPOINT	SAMPLE_POINT is usually (SAMPLE_COUNT-1)/2 without decimal. Effective only when HIGHSPEED=3.

A0080030 AUTOBAUD_REG Auto Baud Monitor Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									**BAUD_STAT**				**BAUD_RATE**			
Type									RU				RU			
Reset									0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7:4	BAUD_STAT	**Autobaud state (only true value in standard autobaud detection)** 0: Autobaud is detecting. 1: AT_7N1 2: AT_7O1 3: AT_7E1 4: AT_8N1 5: AT_8O1 6: AT_8E1 7: at_7N1 8: at_7E1 9: at_7O1 10: at_8N1

Bit(s)	Name	Description
		11: at_8E1 12: at_8O1 13: Autobaud detection fails
3:0	BAUD_RATE	**Autobaud baud rate (only true v alue in standard autobaud detection)** 0: 115,200 1: 57,600 2: 38,400 3: 19,200 4: 9,600 5: 4,800 6: 2,400 7: 1,200 8: 300 9: 110

A0080034 RATEFIX_AD Clock Rate Fix Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														FREQ_SEL	AUTOBAUD_RATE_FIX	RATE_FIX
Type														RW	RW	RW
Reset														0	0	0

Bit(s)	Name	Description
2	FREQ_SEL	0: Select 26MHz as system clock 1: Select 13MHz as system clock
1	AUTOBAUD_RATE_FIX	0: Use 52MHz as system clock for UART auto baud detection 1: Use 26MHz/13MHz (depending on FREQ_SEL) as system clock for UART auto baud detection
0	RATE_FIX	0: Use 52MHz as system clock for UART TX/RX 1: Use 26MHz/13MHz (depending on FREQ_SEL) as system clock for UART TX/RX

A0080038 AUTOBAUDSAMPLE Auto Baud Sample Register 0D

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name											AUTOBAUDSAMPLE					
Type											RW					
Reset											0	0	1	1	0	1

Bit(s)	Name	Description
5:0	AUTOBAUDSAMPLE	**clk diveision for autobaud rate detection** For standard baud rate detection. System clk 52m: 'd 27 System clk 26m: 'd 13 System clk 13m: 'd 6 For non-standard baud rate detection. :15.

A008003C GUARD Guard Time Added Register 0F

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name												GUARD_EN	GUARD_CNT			
Type												RW	RW			
Reset												0	1	1	1	1

Bit(s)	Name	Description
4	GUARD_EN	**Guard interval add enabling signal** 0: No guard interval added 1: Add guard interval after stop bit.
3:0	GUARD_CNT	**Guard interval count value** Guard interval = [1/(system clock/div_step/div)]*GUARD_CNT.

A0080040 ESCAPE_DAT Escape Character Register FF

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									ESCAPE_DAT							
Type									RW							
Reset									1	1	1	1	1	1	1	1

Bit(s)	Name	Description
7:0	ESCAPE_DAT	**Escape character added before software flow control data and escape character** If TX data are xon (31h), with esc_en =1, UART will transmit data as esc + CEh (~xon).

A0080044 ESCAPE_EN Escape Enable Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																ESC_EN
Type																RW
Reset																0

Bit(s)	Name	Description
0	ESC_EN	**Adds escape character in transmitter and removes escape character in receiver by UART** 0: Does not deal with the escape character 1: Add escape character in transmitter and remove escape character in receiver

A0080048 SLEEP_EN Sleep Enable Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																SLEEP_EN
Type																RW
Reset																0

Bit(s)	Name	Description
0	SLEEP_EN	**For sleep mode issue** 0: Does not deal with sleep mode indicate signal 1: Activate hardware flow control or software control according to software initial setting when the chip enters sleep mode. Release hardware flow when the chip wakes up. However, for software control, UART sends xon when awaken and when FIFO does not

Bit(s)	Name	Description
		reach threshold level.

A008004C DMA_EN DMA Enable Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													FIFO_lsr_sel	TO_CNT_AUTORST	TX_DMA_EN	RX_DMA_EN
Type													RW	RW	RW	RW
Reset													0	0	0	0

Bit(s)	Name	Description
3	FIFO_lsr_sel	**Selects FIFO LSR mode** 0: LSR will hold the first line status error state until you read the LSR register. 1: LSR will update automatically.
2	TO_CNT_AUTORST	**Time-out counter auto reset register** 0: After RX time-out happens, SW shall reset the interrupt by reading UART 0x4C. 1: The time-out counter will be auto reset. Set this register when Rain's new DMA is used.
1	TX_DMA_EN	**TX_DMA mechanism enabling signal** 0: Does not use DMA in TX 1: Use DMA in TX. When this register is enabled, the flow control will be based on the DMA threshold and generate a time-out interrupt for DMA.for DMA.
0	RX_DMA_EN	**RX_DMA mechanism enabling signal** 0: Does not use DMA in RX 1: Use DMA in RX. When this register is enabled, the flow control will be based on the DMA threshold and generate a time-out interrupt

A0080050 RXTRI_AD Rx Trigger Address 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													RXTRIG			
Type													RW			
Reset													0	0	0	0

Bit(s)	Name	Description
3:0	RXTRIG	When {rtm,rtl}=2'b11, the RX FIFO threshold will be Rxtrig. The value is suggested to be smaller than half of RX FIFO size, which is 32 bytes.

A0080054 FRACDIV_L Fractional Divider LSB Address 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									FRACDIV_L							
Type									RW							
Reset									0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7:0	FRACDIV_L	Adds sampling count (+1) from state data7 to data0 to contribute fractional divisor.only when high_speed=3.

A0080058 FRACDIV_M Fractional Divider MSB Address 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name															**FRACDIV_M**	
Type															RW	
Reset															0	0

Bit(s)	Name	Description
1:0	FRACDIV_M	Adds sampling count when in state stop to parity to contribute fractional divisor.only when high_speed=3.

A008005C FCR_RD FIFO Control Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									**RFTL1_RFTL0**		**TFTL1_TFTL0**			**CLRT**	**CLRR**	**FIFOE**
Type									RO		RO			RO	RO	RO
Reset									0	0	0	0		0	0	0

Bit(s)	Name	Description
7:6	RFTL1_RFTL0	**RX FIFO trigger threshold** RX FIFO contains total 32 bytes. 0: 1 1: 6 2: 12 3: RXTRIG
5:4	TFTL1_TFTL0	**TX FIFO trigger threshold** TX FIFO contains total 32 bytes. 0: 1 1: 4 2: 8 3: 14
2	CLRT	0: TX FIFO is not cleared. 1: TX FIFO is cleared.
1	CLRR	0: RX FIFO is not cleared. 1: RX FIFO is cleared.
0	FIFOE	**Enables FIFO** This bit must be set to 1 for any of other bits in the registers to have any effect. 0: RX and TX FIFOs are not enabled. 1: RX and TX FIFOs are enabled.

3.8 UART2

3.8.1 General Description

The baseband chipset houses two UARTs. The UARTs provide full duplex serial communication channels between baseband chipset and external dev ices.

The UART has M16C450 and M16550A modes of operation, which are compatible with a range of standard software drivers. The extensions have been designed to be broadly software compatible with 16550A variants, but certain areas offer no consensus.

In common with the M16550A, the UART supports word lengths **from 5 to 8 bits**, **an optional parity bit** and one or two stop bits, and is fully programmable by an 8-bit CPU interface. A 16-bit programmable baud rate generator and an 8-bit scratch register are included, together with separate transmit and receive FIFOs. 8 modem control lines and a diagnostic loop-back mode are provided. UART also includes two DMA handshake lines, indicating when the FIFOs are ready to transfer data to the CPU. Interrupts can be generated from any of the 10 sources.

Note that UART is designed so that all internal operation is synchronized by the CLK signal. This synchronization results in minor timing differences between the UART and industry standard 16550A device, which means that the core is not clock for clock identical to the original device.

After a hardware reset, UART will be in M16C450 mode. Its FIFOs can then be enabled and UART can enter M16550A mode. UART has further functions beyond the M16550A mode. Each of the extended functions can be selected individually under software control.

UART provides more powerful enhancements than the industry-standard 16550:

- Hardware flow control. This feature is very useful when the ISR latency is hard to predict and control in the embedded applications. The MCU is relieved of having to fetch the received data within a fixed amount of time.

Note that in order to enable any of the enhancements, the Enhanced Mode bit, EFR[4], must be set. If EFR[4] is not set, IER[7:5], FCR[5:4], IIR[5:4] and MCR[7:6] cannot be written. The enhanced mode bit ensures that UART is backward compatible with the software that has been written for 16C450 and 16550A devices. **Figure 35** is the block diagram of the UART2 device.

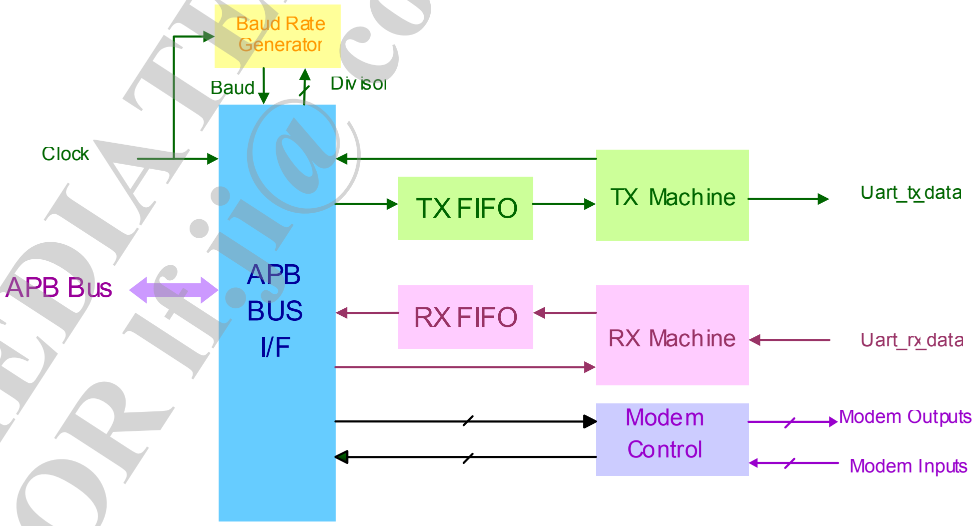


Figure 36. Block Diagram of UART2

3.8.2 Register Definition

Module name: UART2 Base address: (+A0090000h)

Address	Name	Width	Register Function
A0090000	RBR	8	**RX Buffer Register** Note: Only w hen LCR[7] = 0.
A0090000	THR	8	**TX Holding Register** Note: Only w hen LCR[7] = 0
A0090000	DLL	8	**Divisor Latch (LS)** Divides the bclk frequency Note: Nodified w hen LCR[7]!=0
A0090004	IER	8	**Interrupt Enable Register** Note: Only w hen LCR[7] = 0. By storing 1 to a specific bit position, the interrupt associated w ith that bit is enabled. Otherw ise, the interrupt w ill be disabled. IER[3:0] are modified w hen LCR[7] = 0. IER[7:4] are modified w hen LCR[7] = 0 & EFR[4] = 1.
A0090004	DLM	8	**Divisor Latch (MS)** uesd to divid the bclk frequency .*NOTE: modified w hen LCR[7]!=0
A0090008	IIR	8	**Interrupt Identification Register** Note: Only w hen LCR!=BF'h. Priority is from high to low as the follow ing. IIR[5:0]=0X1: No interrupt pending IIR[5:0]=0X6: Line status interrupt (under IER[2]=1) IIR[5:0]=0Xc: RX data time-out interrupt (under IER[0]=1) IIR[5:0]=0X4: RX data are placed in the RX bBuffer register or the RX trigger level is reached. (under IER[0]=1). IIR[5:0]=0X2: TX holding register is empty or the contents of the TX FIFO have been reduced to its trigger level (under IER[1]=1). IIR[5:0]=0X10: XOFF character received (under IER[5]=1, EFR[4] = 1).
A0090008	FCR	8	**FIFO Control Register** FCR is used to control the trigger levels of the FIFOs or flush the FIFOs. FCR[7:6] is modified w hen LCR != BFh FCR[5:4] is modified w hen LCR != BFh & EFR[4] = 1 FCR[4:0] is modified w hen LCR != BFh
A0090008	EFR	8	**Enhanced Feature Register** Note: Only w hen LCR=BF'h
A009000C	LCR	8	**Line Control Register** Determines characteristics of serial communication signals.
A0090010	MCR	8	**Modem Control Register** Controls interface signals of the UART. MCR[5:0] are modified w hen LCR != 8'hBF, MCR[7] can be read w hen LCR != 8'hBF & EFR[4] = 1.
A0090010	XON1	8	**XON1 Char Register** Note: XON1 is modified only w hen LCR=BF'h.
A0090014	LSR	8	**Line Status Register** Modified w hen LCR != BFh.

Address	Name	Width	Register Function
A0090018	XOFF1	8	**XOFF1 Char Register** *Note: XOFF1 is modified only when LCR=BF'h.
A009001C	SCR	8	**Scratch Register** A general purpose read/write register. After reset, its value will be un-defined. Modified when LCR != BFh.
A0090020	AUTOBAUD_EN	8	**Auto Baud Detect Enable Register**
A0090024	HIGHSPEED	8	**High Speed Mode Register**
A0090028	SAMPLE_COUNT	8	**Sample Counter Register** When HIGHSPEED=3, sample_count will be the threshold value for UART sample counter (sample_num). Count from 0 to sample_count.
A009002C	SAMPLE_POINT	8	**Sample Point Register** When HIGHSPEED=3, UART gets the input data when sample_count=sample_num, e.g. system clock = 13MHz, 921600 = 13000000/14. Therefore, sample_count = 13, and sample point = 6 (sampling the central point to decrease the inaccuracy) SAMPLE_POINT is usually (SAMPLE_COUNT-1)/2 without decimal.
A0090030	AUTOBAUD_REG	8	**Auto Baud Monitor Register** the autobaud detection state ,it will not change until enable the autobaud_en again.
A0090034	RATEFIX_AD	8	**Clock Rate Fix Register**
A0090038	AUTOBAUDSAMPLE	8	**Auto Baud Sample Register** Since the system clock may change, autobaud sample duration should change as the system clock changes. When system clock = 13MHz, autobaudsample = 6; when system clock = 26MHz, autobaudsample = 13. When system clock = 52MHz, autobaudsample = 27.
A009003C	GUARD	8	**Guard Time Added Register**
A0090040	ESCAPE_DAT	8	**Escape Character Register**
A0090044	ESCAPE_EN	8	**Escape Enable Register**
A0090048	SLEEP_EN	8	**Sleep Enable Register**
A009004C	DMA_EN	8	**DMA Enable Register**
A0090050	RXTRI_AD	8	**Rx Trigger Address**
A0090054	FRACDIV_L	8	**Fractional Divider LSB Address**
A0090058	FRACDIV_M	8	**Fractional Divider MSB Address**
A009005C	FCR_RD	8	**FIFO Control Register**

A0090000 RBR RX Buffer Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									RBR							
Type									RU							
Reset									0	0	0	0	0	0	0	0

Bit(s)	Name	Description

Bit(s)	Name	Description
7:0	RBR	**Read-only register** The received data can be read by accessing this register. Only w hen LCR[7] = 0.

A0090000 THR TX Holding Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									THR							
Type									WO							
Reset									0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7:0	THR	**TX Holding Register** Write-only register. The data to be transmitted are w ritten to this register and sent to the PC via serial communication. Only w hen LCR[7] = 0.

A0090000 DLL Divisor Latch (LS) 01

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									DLL							
Type									RW							
Reset									0	0	0	0	0	0	0	1

Bit(s)	Name	Description
7:0	DLL	**Divisor latch low 8-bit data** *Note: Modified when LCR[7]!=0.*

A0090004 IER Interrupt Enable Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									CTSI	RTSI	XOFFI		EDSSI	ELSI	ETBEI	ERBFI
Type									RW	RW	RW		RW	RW	RW	RW
Reset									0	0	0		0	0	0	0

Bit(s)	Name	Description
7	CTSI	**Masks an interrupt that is generated when a rising edge is detected on the CTS modem control line.** *Note: This interrupt is only enabled when hardware flow control is enabled.* 0: Mask an interrupt that is generated w hen a rising edge is detected on the CTS modem control line. 1: Unmask an interrupt that is generated w hen a rising edge is detected on the CTS modem control line.
6	RTSI	**Masks an interrupt that is generated when a rising edge is detected on the RTS modem control line.** *Note: This interrupt is only enabled when hardware flow control is enabled.* 0: Mask an interrupt that is generated w hen a rising edge is detected on the RTS modem control line. 1: Unmask an interrupt that is generated w hen a rising edge is detected on the

Bit(s)	Name	Description
		RTS modem control line.
5	XOFFI	**Masks an interrupt that is generated when an XOFF character is received.** *Note: This interrupt is only enabled when software flow control is enabled.* 0: Mask an interrupt that is generated when an XOFF character is received. 1: Unmask an interrupt that is generated when an XOFF character is received.
3	EDSSI	**When set to 1, an interrupt will be generated if DCTS (MSR[0]) becomes set.** 0: No interrupt is generated if DCTS (MSR[0]) becomes set. 1: An interrupt is generated if DCTS (MSR[0]) becomes set.
2	ELSI	**When set to1, an interrupt will be generated if BI, FE, PE or OE (LSR[4:1]) becomes set.** 0: No interrupt is generated if BI, FE, PE or OE (LSR[4:1]) becomes set. 1: An interrupt is generated if BI, FE, PE or OE (LSR[4:1]) becomes set.
1	ETBEI	**When set to 1, an interrupt will be generated if the TX holding register is empty or the contents of the TX FIFO have been reduced to its trigger level.** 0: No interrupt will be generated if the TX holding register is empty or the contents of the TX FIFO have been reduced to its trigger level. 1: An interrupt will be generated if the TX folding register is empty or the contents of the TX FIFO have been reduced to its trigger level.
0	ERBFI	**When set to 1, an interrupt will be generated if RX data are placed in RX buffer register or the RX trigger level is reached.** 0: No interrupt will be generated if RX data are placed in the RX buffer register or the RX trigger level is reached. 1: An interrupt will be generated if RX Data are placed in the RX buffer register or the RX trigger level is reached.

A0090004 DLM Divisor Latch (MS) 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									**DLM**							
Type									RW							
Reset									0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7:0	DLM	**Divisor latch high 8-bit data** *Note: Modified when LCR[7]!=0. DLL & DLM can only be updated when DLAB(LCR[7]) is set to 1. Division by 1 will generate a BAUD signal that is constantly high. DLL & DLM setting formula is {DLH,DLL}=(system clock frequency/baud_pulse/baud_rate).* When RATE_FIX(RATEFIX_AD[0])=0, system clock frequency = 52MHz. When RATE_FIX(RATEFIX_AD[0])=1 and RATE_FIX(RATEFIX_AD[2])=0, system clock frequency = 26MHz. When RATE_FIX(RATEFIX_AD[0])=1 and RATE_FIX(RATEFIX_AD[2])=1, system clock frequency = 13MHz. For baud_pulse value, refer to HIGH_SPEED(offset=24H) register For example, when at 52MHz, default speed mode and 115200 baud rate, {DLH,DLL}=52MHz/16/115200=28.

A0090008 IIR Interrupt Identification Register 01

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									FIFOE		ID					
Type									RO		RU					
Reset									0	0	0	0	0	0	0	1

Bit(s)	Name	Description
7:6	FIFOE	
5:0	ID	**IIR[5:0] Priority level interrupt source**

000001 - No interrupt pending
000110 1 Line status interrupt:
BI, FE, PE or OE set in LSR. (Under IER[2]=1)
001100 2 RX data time-out:
Time-out on character in RX FIFO. (Under IER[0]=1)
000100 3 RX data received:
RX data received or RX trigger level reached. (Under IER[0]=1)
000010 4 TX holding register empty:
000000 5 Modem status change:
DCTS set in MSR. (Under IER[3]=1)
TX Holding Register empty or TX FIFO trigger level reached. (Under IER[1]=1)
010000 6 Software flow control:
XOFF Character received. (Under IER[5]=1)
100000 7 Hardware flow control:
CTS or RTS Rising Edge. (Under IER[7]=1 or IER[6]=1)

Line status interrupt: A RX line status interrupt (IIR[5:0] = 000110b) will be generated if ELSI (IER[2]) is set and any of BI, FE, PE or OE (LSR[4:1]) becomes set. The interrupt is cleared by reading the line status register.

RX data time-out interrupt: When the virtual FIFO mode is disabled, RX data time-out interrupt will be generated if all of the following conditions are applied:
1. FIFO contains at least one character.

2. The most recent character is received longer than four character periods ago (including all start, parity and stop bits);

3. The most recent CPU read of the FIFO is longer than four character periods ago.

The timeout timer is restarted upon receipt of a new byte from the RX shift register or upon a CPU read from the RX FIFO.
The RX data time-out interrupt is enabled by setting EFRBI (IER[0]) to 1 and is cleared by reading RX FIFO.

When the virtual FIFO mode is enabled, RX data time-out interrupt will be generated if all of the following conditions are applied:
1. FIFO is empty.
2. The most recent character is received longer than four character periods ago (including all start, parity and stop bits).
3. The most recent CPU read of the FIFO is longer than four character periods ago.

The timeout timer is restarted upon receipt of a new byte from the RX shift register or reading DMA_EN register.
The RX Data Timeout Interrupt is enabled by setting EFRBI (IER[0]) to 1 and is cleared by reading DMA_EN register.

RX data received interrupt: A RX received interrupt (IER[5:0] = 000100b) is generated if EFRBI (IER[0]) is set and either RX data are placed in the RX buffer register or the RX trigger level is reached. The interrupt is cleared by

Bit(s)	Name	Description
		reading the RX buffer register or the RX FIFO (if enabled). **TX holding register empty interrupt:** A TX holding register empty interrupt (IIR[5:0] = 000010b) is generated if ETRBI (IER[1]) is set and either the TX holding register is empty or the contents of the TX FIFO are reduced to its trigger level. The interrupt is cleared by writing to the TX holding register or TX FIFO if FIFO is enabled. **Modem status change interrupt:** A modem status change Interrupt (IIR[5:0] = 000000b) will be generated if EDSSI (IER[3]) is set and e DCTS (MSR[3:0]) becomes set. The interrupt is cleared by reading the modem status register. **Software flow control interrupt:** A software flow control interrupt (IIR[5:0] = 010000b) will be generated if the software flow control is enabled and XOFFI (IER[5]) becomes set, indicating that an XOFF character has been received. The interrupt is cleared by reading the interrupt identification register. **Hardware flow control interrupt:** A hardware flow control interrupt (IER[5:0] = 100000b) will be generated if the hardware flow control is enabled and either RTSI (IER[6]) or CTSI (IER[7]) becomes set indicating that a rising edge has been detected on either the RTS/CTS modem control line. The interrupt is cleared by reading the interrupt identification register.

A0090008 FCR FIFO Control Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									RFTL1_RFTL0		TFTL1_TFTL0			CLRT	CLRR	FIFOE
Type									WO		WO			WO	WO	WO
Reset									0	0	0	0		0	0	0

Bit(s)	Name	Description
7:6	RFTL1_RFTL0	**RX FIFO trigger threshold** RX FIFO contains total 32 bytes. 0: 1 1: 6 2: 12 3: RXTRIG
5:4	TFTL1_TFTL0	**TX FIFO trigger threshold** TX FIFO contains total 32 bytes. 0: 1 1: 4 2: 8 3: 14
2	CLRT	**Control bit to clear TX FIFO** 0: No effect 1: Clear TX FIFO
1	CLRR	**Control bit to clear RX FIFO** 0: No effect 1: Clear RX FIFO
0	FIFOE	**Enables FIFO** This bit must be set to 1 for any of other bits in the registers to have any effect. 0: Disable both RX and TX FIFOs. 1: Enable both RX and TX FIFOs.

A0090008 EFR Enhanced Feature Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									AUTO_CTS	AUTO_RTS		ENABLE_E	SW_FLOW_CONT			
Type									RW	RW		RW	RW			
Reset									0	0		0	0	0	0	0

Bit(s)	Name	Description
7	AUTO_CTS	**Enables hardware transmission flow control** 0: Disable 1: Enable
6	AUTO_RTS	**Enables hardware reception flow control** 0: Disable 1: Enable
4	ENABLE_E	**Enables enhancement feature** 0: Disable 1: Enable
3:0	SW_FLOW_CONT	**Software flow control bits** 00xx: No TX flow control 01xx: No TX flow control 10xx: Transmit XON1/XOFF1 as flow control bytes xx00: No RX flow control xx01: No RX flow control xx10: Receive XON1/XOFF1 as flow control bytes

A009000C LCR Line Control Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									DLAB	SB	SP	EPS	PEN	STB	WLS1	WLS0
Type									RW	RW	RW	RW	RW	RW	RW	
Reset									0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7	DLAB	**Divisor latch access bit** 0: RX and TX registers are read/written at Address 0 and the IER register is read/written at Address 4. 1: Divisor Latch LS is read/written at Address 0 and the Divisor Latch MS is read/written at Address 4.
6	SB	**Set break** 0: No effect 1: SOUT signal is forced to the 0 state.
5	SP	**Stick parity** 0: No effect. 1: The parity bit is forced to a defined state, depending on the states of EPS and PEN: If EPS=1 & PEN=1, the parity bit will be set and checked = 0. If EPS=0 & PEN=1, the parity bit will be set and checked = 1.
4	EPS	**Selects even parity** 0: When EPS=0, an odd number of ones is sent and checked. 1: When EPS=1, an even number of ones is sent and checked.

Bit(s)	Name	Description
3	PEN	**Enables parity** 0: The parity is neither transmitted nor checked. 1: The parity is transmitted and checked.
2	STB	**Number of STOP bits** 0: One STOP bit is always added. 1: Two STOP bits are added after each character is sent; unless the character length is 5 when 1 STOP bit is added.
1:0	WLS1_WLS0	**Selects word length** 0: 5 bits 1: 6 bits 2: 7 bits 3: 8 bits

A0090010 MCR Modem Control Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									XOFF_STATUS			Loop			RTS	
Type									RU			RW			RW	
Reset									0			0			0	

Bit(s)	Name	Description
7	XOFF_STATUS	**Read-only bit** 0: When an XON character is received. 1: When an XOFF character is received.
4	Loop	**Loop-back control bit** 0: No loop-back is enabled. 1: Loop-back mode is enabled.
1	RTS	**Controls the state of the output NRTS, even in loop mode.** 0: RTS will always output 1. 1: RTS's output will be controlled by flow control condition.

A0090010 XON1 XON1 Char Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									XON1							
Type									RW							
Reset									0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7:0	XON1	**XON1 character for software flow control** Modified only when LCR=BF'h.

A0090014 LSR Line Status Register 60

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									FIFOERR	TEMT	THRE	BI	FE	PE	OE	DR

Type									RU	RU	RU	RU	RU	RU	RU	RU
Reset									0	1	1	0	0	0	0	0

Bit(s)	Name	Description
7	FIFOERR	**RX FIFO error indicator** 0: No PE, FE, BI set in the RX FIFO. 1: Set to 1 when there is at least one PE, FE or BI in the RX FIFO.
6	TEMT	**TX holding register (or TX FIFO) and the TX shift register are empty.** 0: Empty conditions below are not met. 1: If FIFOs are enabled, the bit will be set whenever the TX FIFO and the TX shift register are empty. If FIFOs are disabled, the bit will be set whenever TX holding register and TX shift register are empty.
5	THRE	**Indicates if there is room for TX holding register or TX FIFO is reduced to its trigger level** 0: Reset whenever the contents of the TX FIFO are more than its trigger level (FIFOs are enabled), or whenever TX holding register is not empty (FIFOs are disabled). 1: Set whenever the contents of the TX FIFO are reduced to its trigger level (FIFOs are enabled), or whenever TX holding register is empty and ready to accept new data (FIFOs are disabled).
4	BI	**Break interrupt** 0: Reset by the CPU reading this register 1: If the FIFOs are disabled, this bit will be set whenever the SIN is held in the 0 state for more than one transmission time (START bit + DATA bits + PARITY + STOP bits). If the FIFOs are enabled, this error will be associated with a corresponding character in the FIFO and is flagged when this byte is at the top of the FIFO. When a break occurs, only one zero character is loaded into the FIFO: The next character transfer is enabled when SIN goes into the marking state and receives the next valid start bit.
3	FE	**Framing error** 0: Reset by the CPU reading this register 1: If the FIFOs are disabled, this bit will be set if the received data do not have a valid STOP bit. If the FIFOs are enabled, the state of this bit will be revealed when the byte it refers to is the next to be read.
2	PE	**Parity error** 0: Reset by the CPU reading this register 1: If the FIFOs are disabled, this bit will be set if the received data do not have a valid parity bit. If the FIFOs are enabled, the state of this bit will be revealed when the referred byte is the next to be read.
1	OE	**Overrun error** 0: Reset by the CPU reading this register. 1: If the FIFOs are disabled, this bit will be set if the RX buffer is not read by the CPU before the new data from the RX shift register overwrites the previous contents. If the FIFOs are enabled, an overrun error will occur when the RX FIFO is full and the RX shift register becomes full. OE will be set as soon as this happens. The character in the shift register is then overwritten, but not transferred to the FIFO.
0	DR	**Data ready** 0: Cleared by the CPU reading the RX buffer or by reading all the FIFO bytes. 1: Set by the RX buffer becoming full or by the FIFO becoming no empty.

A0090018 XOFF1 XOFF1 Char Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									XOFF1							
Type									RW							
Reset									0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7:0	XOFF1	**XOFF1 character for software flow control** Modified only when LCR=BF'h.

A009001C SCR Scratch Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									SCR							
Type									RW							
Reset									0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7:0	SCR	**General purpose read/write register** After reset, its value will be undefined. Modified when LCR != BFh.

A0090020 AUTOBAUD_EN Auto Baud Detect Enable Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														SLEEP_ACK_SEL	AUTOBAUD_SEL	AUTOBAUD_EN
Type														RW	RW	RW
Reset														0	0	0

Bit(s)	Name	Description
2	SLEEP_ACK_SEL	**Selects sleep ack when autobaud_en** 0: Support sleep_ack when autobaud_en is opened . 1: Does not support sleep_ack when autobaud_en is opened .
1	AUTOBAUD_SEL	**Selects auto-baud** 0: Support standard baud rate detection 1: Support non_standard baud rate detection (support baud from 110 to 115200; recommended to use 52MHz to auto fix) .
0	AUTOBAUD_EN	**Auto-baud enabling signal** 0: Disable auto-baud function 1: Enable auto-baud function (UARTn+0024h SPEED should be set to 0.) *Note: When AUTOBAUD_EN is active, there should not be A*/a* char before the auto baud char AT/at. If A*/a* is Inevitable, autobaud will fail and please disable AUTOBAUD_EN to reset the autobaud feature and autobaud_en again.*

A0090024 HIGHSPEED High Speed Mode Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name															SPEED	
Type															RW	
Reset															0	0

Bit(s)	Name	Description

1:0	SPEED	**UART sample counter base**
		0: Based on 16*baud_pulse, baud_rate = system clock frequency/16/{DLH, DLL}
		1: Based on 8*baud_pulse, baud_rate = system clock frequency/8/{DLH, DLL}
		2: Based on 4*baud_pulse, baud_rate = system clock frequency/4/{DLH, DLL}
		3: Based on sampe_count * baud_pulse, baud_rate = system clock frequency / (sampe_count+1)/{DLM, DLL}

A0090028 SAMPLE_COUNT Sample Counter Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									SAMPLECOUNT							
Type									RW							
Reset									0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7:0	SAMPLECOUNT	Only useful when HIGHSPEED mode = 3.

A009002C SAMPLE_POINT Sample Point Register FF

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									SAMPLEPOINT							
Type									RW							
Reset									1	1	1	1	1	1	1	1

Bit(s)	Name	Description
7:0	SAMPLEPOINT	SAMPLE_POINT is usually (SAMPLE_COUNT-1)/2 without decimal. Effective only when HIGHSPEED=3.

A0090030 AUTOBAUD_REG Auto Baud Monitor Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									BAUD_STAT				BAUD_RATE			
Type									RU				RU			
Reset									0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7:4	BAUD_STAT	**Autobaud state (only true value in standard autobaud detection)** 0: Autobaud is detecting. 1: AT_7N1 2: AT_7O1 3: AT_7E1 4: AT_8N1 5: AT_8O1 6: AT_8E1 7: at_7N1

Bit(s)	Name	Description
		8: at_7E1 9: at_7O1 10: at_8N1 11: at_8E1 12: at_8O1 13: Autobaud detection fails
3:0	BAUD_RATE	**Autobaud baud rate (only true value in standard autobaud detection)** 0: 115,200 1: 57,600 2: 38,400 3: 19,200 4: 9,600 5: 4,800 6: 2,400 7: 1,200 8: 300 9: 110

A0090034 RATEFIX_AD Clock Rate Fix Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														**FREQ_SEL**	**AUTOBAUD_RATE_FIX**	**RATE_FIX**
Type														RW	RW	RW
Reset														0	0	0

Bit(s)	Name	Description
2	FREQ_SEL	0: Select 26MHz as system clock 1: Select 13MHz as system clock
1	AUTOBAUD_RATE_FIX	0: Use 52MHz as system clock for UART auto baud detection 1: Use 26MHz/13MHz (depending on FREQ_SEL) as system clock for UART auto baud detection
0	RATE_FIX	0: Use 52MHz as system clock for UART TX/RX 1: Use 26MHz/13MHz (depending on FREQ_SEL) as system clock for UART TX/RX

A0090038 AUTOBAUDSAMPLE Auto Baud Sample Register 0D

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name											**AUTOBAUDSAMPLE**					
Type											RW					
Reset											0	0	1	1	0	1

Bit(s)	Name	Description
5:0	AUTOBAUDSAMPLE	**clk diveision for autobaud rate detection** For standard baud rate detection. System clk 52m: 'd 27 System clk 26m: 'd 13 System clk 13m: 'd 6

Bit(s)	Name	Description
		For non-standard baud rate detection. :15.

A009003C GUARD Guard Time Added Register 0F

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name												GUARD_EN	GUARD_CNT			
Type												RW	RW			
Reset												0	1	1	1	1

Bit(s)	Name	Description
4	GUARD_EN	**Guard interval add enabling signal** 0: No guard interval added 1: Add guard interval after stop bit.
3:0	GUARD_CNT	**Guard interval count value** Guard interval = [1/(system clock/div_step/div)]*GUARD_CNT.

A0090040 ESCAPE_DAT Escape Character Register FF

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									ESCAPE_DAT							
Type									RW							
Reset									1	1	1	1	1	1	1	1

Bit(s)	Name	Description
7:0	ESCAPE_DAT	**Escape character added before software flow control data and escape character** If TX data are xon (31h), with esc_en =1, UART will transmit data as esc + CEh (~xon).

A0090044 ESCAPE_EN Escape Enable Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																ESC_EN
Type																RW
Reset																0

Bit(s)	Name	Description
0	ESC_EN	**Adds escape character in transmitter and removes escape character in receiver by UART** 0: Does not deal with the escape character 1: Add escape character in transmitter and remove escape character in receiver

A0090048 SLEEP_EN Sleep Enable Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																SLEEP_EN
Type																RW
Reset																0

Bit(s)	Name	Description
0	SLEEP_EN	**For sleep mode issue** 0: Does not deal with sleep mode indicate signal 1: Activate hardware flow control or software control according to software initial setting when the chip enters sleep mode. Release hardware flow when the chip wakes up. However, for software control, UART sends xon when awaken and when FIFO does not reach threshold level.

A009004C DMA_EN DMA Enable Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													FIFO_lsr_sel	TO_CNT_AUTORST	TX_DMA_EN	RX_DMA_EN
Type													RW	RW	RW	RW
Reset													0	0	0	0

Bit(s)	Name	Description
3	FIFO_lsr_sel	**Selects FIFO LSR mode** 0: LSR will hold the first line status error state until you read the LSR register. 1: LSR will update automatically.
2	TO_CNT_AUTORST	**Time-out counter auto reset register** 0: After RX time-out happens, SW shall reset the interrupt by reading UART 0x4C. 1: The time-out counter will be auto reset. Set this register when Rain's new DMA is used.
1	TX_DMA_EN	**TX_DMA mechanism enabling signal** 0: Does not use DMA in TX 1: Use DMA in TX. When this register is enabled, the flow control will be based on the DMA threshold and generate a time-out interrupt for DMA.for DMA.
0	RX_DMA_EN	**RX_DMA mechanism enabling signal** 0: Does not use DMA in RX 1: Use DMA in RX. When this register is enabled, the flow control will be based on the DMA threshold and generate a time-out interrupt

A0090050 RXTRI_AD Rx Trigger Address 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													RXTRIG			
Type													RW			
Reset													0	0	0	0

Bit(s)	Name	Description

Bit(s)	Name	Description
3:0	RXTRIG	When {rtm,rtl}=2'b11, the RX FIFO threshold will be Rxtrig. The value is suggested to be smaller than half of RX FIFO size, which is 32 bytes.

A0090054 FRACDIV_L Fractional Divider LSB Address 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									FRACDIV_L							
Type									RW							
Reset									0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7:0	FRACDIV_L	Adds sampling count (+1) from state data7 to data0 to contribute fractional divisor.only when high_speed=3.

A0090058 FRACDIV_M Fractional Divider MSB Address 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name															FRACDIV_M	
Type															RW	
Reset															0	0

Bit(s)	Name	Description
1:0	FRACDIV_M	Adds sampling count when in state stop to parity to contribute fractional divisor.only when high_speed=3.

A009005C FCR_RD FIFO Control Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									RFTL1_RFTL0		TFTL1_TFTL0			CLRT	CLRR	FIFOE
Type									RO		RO			RO	RO	RO
Reset									0	0	0	0		0	0	0

Bit(s)	Name	Description
7:6	RFTL1_RFTL0	**RX FIFO trigger threshold** RX FIFO contains total 32 bytes. 0: 1 1: 6 2: 12 3: RXTRIG
5:4	TFTL1_TFTL0	**TX FIFO trigger threshold** TX FIFO contains total 32 bytes. 0: 1 1: 4 2: 8 3: 14
2	CLRT	0: TX FIFO is not cleared. 1: TX FIFO is cleared.

Bit(s)	Name	Description
1	CLRR	0: RX FIFO is not cleared. 1: RX FIFO is cleared.
0	FIFOE	**Enables FIFO** This bit must be set to 1 for any of other bits in the registers to have any effect. 0: RX and TX FIFOs are not enabled. 1: RX and TX FIFOs are enabled.

3.9 UART3

3.9.1 General Description

The baseband chipset houses two UARTs. The UARTs provide full duplex serial communication channels between baseband chipset and external dev ices.

The UART has M16C450 and M16550A modes of operation, which are compatible with a range of standard software drivers. The extensions have been designed to be broadly software compatible with 16550A variants, but certain areas offer no consensus.

In common with the M16550A, the UART supports word lengths **from 5 to 8 bits**, **an optional parity bit** and one or two stop bits, and is fully programmable by an 8-bit CPU interface. A 16-bit programmable baud rate generator and an 8-bit scratch register are included, together with separate transmit and receive FIFOs. Eight modem control lines and a diagnostic loop-back mode are provided. The UART also includes two DMA handshake lines, used to indicate when the FIFOs are ready to transfer data to the CPU. Interrupts can be generated from any of the 10 sources.

Note: The UART has been designed so that all internal operations are synchronized by the CLK signal. This synchronization results in minor timing differences between the UART and the industry standard 16550A device, which means that the core is not clock for clock identical to the original device.

After a hardware reset, the UART is in M16C450 mode. Its FIFOs can be enabled and the UART can then enter M16550A mode. The UART adds further functionality beyond M16550A mode. Each of the extended functions can be selected individually under software control.

The UART provides more powerful enhancements than the industry-standard 16550:

Note: In order to enable any of the enhancements, the Enhanced Mode bit, EFR[4], must be set. If EFR[4] is not set, IER[7:5], FCR[5:4], IIR[5:4] and MCR[7:6] cannot be written. The Enhanced Mode bit ensures that the UART is backward compatible with software that has been written for 16C450 and 16550A devices.

Figure 35 shows the block diagram of the UART3 device.

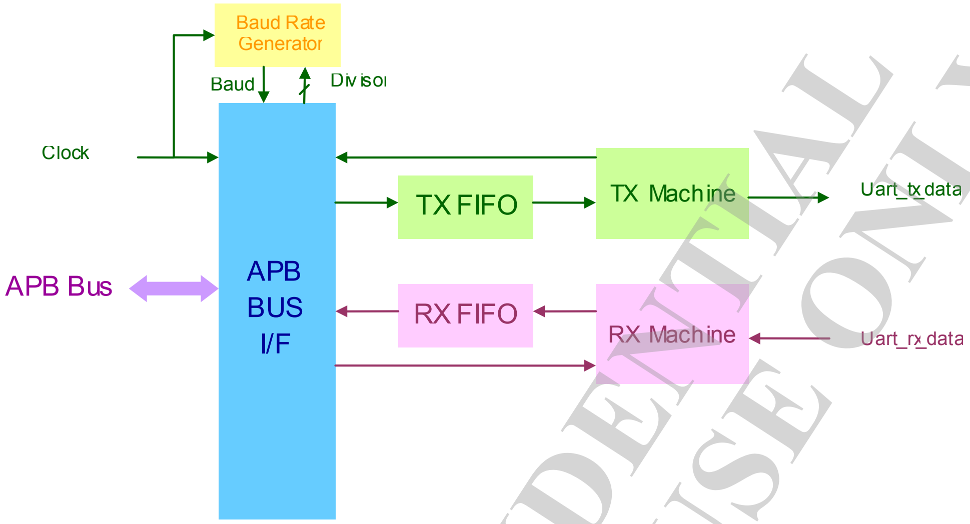


Figure 37. Block Diagram of UART3

3.9.2 Register Definitions

Module name: UART3 Base address: (+A00A0000h)

Address	Name	Width	Register Function
A00A0000	RBR	8	**RX Buffer Register** *NOTE:onnly when LCR[7] = 0.
A00A0000	THR	8	**TX Holding Register** *NOTE:only when LCR[7] = 0.
A00A0000	DLL	8	**Divisor Latch (LS)** uesd to divid the bclk frequency .*NOTE: modified when LCR[7]!=0
A00A0004	IER	8	**Interrupt Enable Register** *NOTE:onnly when LCR[7] = 0. By storing a '1' to a specific bit position, the interrupt associated with that bit is enabled. Otherwise, the interrupt is disabled. IER[3:0] are modified when LCR[7] = 0. IER[7:4] are modified when LCR[7] = 0 & EFR[4] = 1.
A00A0004	DLM	8	**Divisor Latch (MS)** uesd to divid the bclk frequency .*NOTE: modified when LCR[7]!=0
A00A0008	IIR	8	**Interrupt Identification Register** *NOTE:only when LCR!=BF'h. priority is from high to low as following .IIR[5:0]==0X1: no interrupt pending .IIR[5:0]==0X6:line status interrup(Under IER[2]=1). IIR[5:0]==0Xc:rx data timeout interrup(Under IER[0]=1). IIR[5:0]==0X4:RX Data is placed in the RX Buffer Register or the RX Trigger Level is reached.(Under IER[0]=1) . IIR[5:0]==0X2: TX Holding Register is empty or the contents of the TX FIFO have been reduced to its Trigger Level(Under IER[1]=1). IIR[5:0]==0X10: XOFF character recieved (Under IER[5]=1,EFR[4] = 1).
A00A0008	FCR	8	**FIFO Control Register** FCR is used to control the trigger levels of the FIFOs, or flush the FIFOs. FCR[7:6] is modified when LCR != BFh FCR[5:4] is modified when LCR != BFh & EFR[4] = 1 FCR[4:0] is modified when LCR != BFh
A00A0008	EFR	8	**Enhanced Feature Register** *NOTE: Only when LCR=BF'h
A00A000C	LCR	8	**Line Control Register** Line Control Register. Determines characteristics of serial communication signals.

A00A0010	**MCR**	8	**Modem Control Register** Modem Control Register. Control interface signals of the UART. MCR[5:0] are modified when LCR != 8'hBF, MCR[7] can be read when LCR != 8'hBF & EFR[4] = 1.
A00A0010	**XON1**	8	**XON1 Char Register** *Note: XON1modified only when LCR=BF'h.
A00A0014	**LSR**	8	**Line Status Register** Line Status Register. Modified when LCR != BFh.
A00A0018	**XOFF1**	8	**XOFF1 Char Register** *Note: , XOFF1 modified only when LCR=BF'h.
A00A001C	**SCR**	8	**Scratch Register** A general purpose read/write register. After reset, its value is un-defined. Modified when LCR != BFh.
A00A0020	**AUTOBAUD_EN**	8	**Auto Baud Detect Enable Register**
A00A0024	**HIGHSPEED**	8	**High Speed Mode Register**
A00A0028	**SAMPLE_COUNT**	8	**Sample Counter Register** When HIGHSPEED=3, the sample_count is the threshold value for UART sample counter (sample_num). Count from 0 to sample_count.
A00A002C	**SAMPLE_POINT**	8	**Sample Point Register** When HIGHSPEED=3, UART gets the input data when sample_count=sample_num. e.g. system clock = 13MHz, 921600 = 13000000 / 14 sample_count = 13 and sample point = 6 (sample the central point to decrease the inaccuracy) The SAMPLE_POINT is usually (SAMPLE_COUNT-1)/2 and remove the decimal.
A00A0030	**AUTOBAUD_REG**	8	**Auto Baud Monitor Register** the autobaud detection state ,it will not change until enable the autobaud_en again.
A00A0034	**RATEFIX_AD**	8	**Clock Rate Fix Register**
A00A0038	**AUTOBAUDSAMPLE**	8	**Auto Baud Sample Register** Since the system clock may change, autobaud sample duration should change as system clock changes. When system clock = 13MHz, autobaudsample = 6; when system clock = 26MHz, autobaudsample = 13;When system clock = 52MHz, autobaudsample = 27.
A00A003C	**GUARD**	8	**Guard time added register**
A00A0040	**ESCAPE_DAT**	8	**Escape character register**
A00A0044	**ESCAPE_EN**	8	**Escape enable register**
A00A0048	**SLEEP_EN**	8	**Sleep enable register**
A00A004C	**DMA_EN**	8	**DMA enable register**
A00A0050	**RXTRI_AD**	8	**Rx Trigger Address**
A00A0054	**FRACDIV_L**	8	**Fractional Divider LSB Address**
A00A0058	**FRACDIV_M**	8	**Fractional Divider MSB Address**
A00A005C	**FCR_RD**	8	**FIFO Control Register**

A00A0000 RBR RX Buffer Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									RBR							
Type									RU							
Reset									0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7:0	RBR	**Read-only register. The received data can be read by accessing this register.** Only when LCR[7] = 0.

A00A0000 THR TX Holding Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									**THR**							
Type									WO							
Reset									0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7:0	THR	**TX Holding Register. Write-only register. The data to be transmitted is written to this register, and then sent to the PC via serial communication.** only when LCR[7] = 0.

A00A0000 DLL Divisor Latch (LS) 01

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									**DLL**							
Type									RW							
Reset									0	0	0	0	0	0	0	1

Bit(s)	Name	Description
7:0	DLL	**divisor Latch low 8bit data.** *NOTE: modified when LCR[7]!=0

A00A0004 IER Interrupt Enable Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name											**XOFFI**			**ELSI**	**ETBEI**	**ERBFI**
Type											RW			RW	RW	RW
Reset											0			0	0	0

Bit(s)	Name	Description
5	XOFFI	**Masks an interrupt that is generated when an XOFF character is received.** Note: This interrupt is only enabled when software flow control is enabled. 0: Mask an interrupt that is generated when an XOFF character is received. 1: Unmask an interrupt that is generated when an XOFF character is received.
2	ELSI	**When set ("1"), an interrupt is generated if BI, FE, PE or OE (LSR[4:1]) becomes set.** 0: No interrupt is generated if BI, FE, PE or OE (LSR[4:1]) becomes set. 1: An interrupt is generated if BI, FE, PE or OE (LSR[4:1]) becomes set.
1	ETBEI	**When set ("1"), an interrupt is generated if the TX Holding Register is empty or the contents of the TX FIFO have been reduced to its Trigger Level.** 0: No interrupt is generated if the TX Holding Register is empty or the contents of the TX FIFO have been reduced to its Trigger Level. 1: An interrupt is generated if the TX Holding Register is empty or the contents of the TX FIFO have been reduced to its Trigger Level

Bit(s)	Name	Description
0	ERBFI	**When set ("1"), an interrupt is generated if RX Data is placed in the RX Buffer Register or the RX Trigger Level is reached.** 0: No interrupt is generated if RX Data is placed in the RX Buffer Register or the RX Trigger Level is reached. 1: An interrupt is generated if RX Data is placed in the RX Buffer Register or the RX Trigger Level is reached.

A00A0004 DLM Divisor Latch (MS) 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									DLM							
Type									RW							
Reset									0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7:0	DLM	**divisor Latch high 8bit data..** *NOTE: modified when LCR[7]!=0 Note: DLL & DLM can only be updated when DLAB(LCR[7]) is set to 1, Note to that division by 1 generates a BAUD signal that is constantly high. Note: DLL & DLM setting formula is {DLH,DLL}=(system clock frequency/baud_pulse/baud_rate). When RATE_FIX(RATEFIX_AD[0])=0, system clock frequency = 52MHz. When RATE_FIX(RATEFIX_AD[0])=1 and RATE_FIX(RATEFIX_AD[2])=0, system clock frequency = 26MHz. When RATE_FIX(RATEFIX_AD[0])=1 and RATE_FIX(RATEFIX_AD[2])=1, system clock frequency = 13MHz. For baud_pulse value refer to HIGH_SPEED(offset=24H) register. e.g. When 52MHz,default speed mode and 115200 baud rate, the {DLH,DLL}=52MHz/16/115200=28.

A00A0008 IIR Interrupt Identification Register 01

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									FIFOE		ID					
Type									RO		RU					
Reset									0	0	0	0	0	0	0	1

Bit(s)	Name	Description
7:6	FIFOE	
5:0	ID	**IIR[5:0] Priority Level Interrupt Source** 000001 - No interrupt pending 000110 1 Line Status Interrupt: BI, FE, PE or OE set in LSR. (Under IER[2]=1) 001100 2 RX Data Timeout: Timeout on character in RX FIFO. (Under IER[0]=1) 000100 3 RX Data Received: RX Data received or RX Trigger Level reached. (Under IER[0]=1) 000010 4 TX Holding Register Empty: 000000 5 Modem Status change: DCTS set in MSR. (Under IER[3]=1) TX Holding Register empty or TX FIFO Trigger Level reached. (Under IER[1]=1) 010000 6 Software Flow Control: XOFF Character received. (Under IER[5]=1) 100000 7 Hardware Flow Control: CTS or RTS Rising Edge. (Under IER[7]=1 or IER[6]=1) Line Status Interrupt: A RX Line Status Interrupt (IIR[5:0] == 000110b) is generated if ELSI (IER[2]) is set and any of BI, FE, PE or OE (LSR[4:1]) becomes set. The interrupt is cleared by reading the Line Status Register.

RX Data Timeout Interrupt:When virtual FIFO mode is disabled, RX Data Timeout Interrupt is generated if all of the following apply:
1. FIFO contains at least one character;
2. The most recent character was received longer than four character periods ago (including all start, parity and stop bits);
3. The most recent CPU read of the FIFO was longer than four character periods ago.
The timeout timer is restarted on receipt of a new byte from the RX Shift Register, or on a CPU read from the RX FIFO.
The RX Data Timeout Interrupt is enabled by setting EFRBI (IER[0]) to 1, and is cleared by reading RX FIFO.
When virtual FIFO mode is enabled, RX Data Timeout Interrupt is generated if all of the following apply:
1. FIFO is empty;
2. The most recent character was received longer than four character periods ago (including all start, parity and stop bits);
3. The most recent CPU read of the FIFO was longer than four character periods ago.
The timeout timer is restarted on receipt of a new byte from the RX Shift Register or reading DMA_EN register.
The RX Data Timeout Interrupt is enabled by setting EFRBI (IER[0]) to 1, and is cleared by reading DMA_EN register.

RX Data Received Interrupt: A RX Received interrupt (IER[5:0] == 000100b) is generated if EFRBI (IER[0]) is set and either RX Data is placed in the RX Buffer Register or the RX Trigger Level is reached. The interrupt is cleared by reading the RX Buffer Register or the RX FIFO (if enabled).

TX Holding Register Empty Interrupt: A TX Holding Register Empty Interrupt (IIR[5:0] = 000010b) is generated if ETRBI (IER[1]) is set and either the TX Holding Register is empty or the contents of the TX FIFO have been reduced to its Trigger Level. The interrupt is cleared by writing to the TX Holding Register or TX FIFO if FIFO enabled.

Modem Status Change Interrupt: A Modem Status Change Interrupt (IIR[5:0] = 000000b) is generated if EDSSI (IER[3]) is set and e DCTS (MSR[3:0]) becomes set. The interrupt is cleared by reading the Modem Status Register.

Software Flow Control Interrupt: A Software Flow Control Interrupt (IIR[5:0] = 010000b) is generated if Software Flow Control is enabled and XOFFI (IER[5]) becomes set, indicating that an XOFF character has been received. The interrupt is cleared by reading the Interrupt Identification Register.

Hardware Flow Control Interrupt: A Hardware Flow Control Interrupt (IER[5:0] = 100000b) is generated if Hardware Flow Control is enabled and either RTSI (IER[6]) or CTSI (IER[7]) becomes set indicating that a rising edge has been detected on either the RTS/CTS Modem Control line. The interrupt is cleared by reading the Interrupt Identification Register.

A00A0008 FCR FIFO Control Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									RFTL1_RFTL0		TFTL1_TFTL0			CLRT	CLRR	FIFOE
Type									WO		WO			WO	WO	WO
Reset									0	0	0	0		0	0	0

Bit(s)	Name	Description
7:6	RFTL1_RFTL0	**RX FIFO trigger threshold. (RX FIFO contains total 32 bytes.)** 0: 1 1: 6 2: 12 3: RXTRIG
5:4	TFTL1_TFTL0	**TX FIFO trigger threshold (TX FIFO contains total 32 bytes.)** 0: 1 1: 4 2: 8 3: 14

Bit(s)	Name	Description
2	CLRT	**control bit to clear tx fifo** 0: no effect 1: clear TX FIFO
1	CLRR	**control bit to clear rx fifo** 0: no effect 1: clear RX FIFO
0	FIFOE	**FIFO Enabled. This bit must be set to 1 for any of the other bits in the registers to have any effect.** 0: Disable both the RX and TX FIFOs. 1: Enable both the RX and TX FIFOs.

A00A0008 EFR Enhanced Feature Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name												ENABLE_E	SW_FLOW_CONT			
Type												RW	RW			
Reset												0	0	0	0	0

Bit(s)	Name	Description
4	ENABLE_E	**Enables enhancement feature** 0: Disabled. 1: Enabled.
3:0	SW_FLOW_CONT	**Software flow control bits.** 00xx: No TX Flow Control 01xx: No TX Flow Control 10xx: Transmit XON1/XOFF1 as flow control bytes xx00: No RX Flow Control xx01: No RX Flow Control xx10: Receive XON1/XOFF1 as flow control bytes

A00A000C LCR Line Control Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									DLAB	SB	SP	EPS	PEN	STB	WLS1_WLS0	
Type									RW	RW	RW	RW	RW	RW	RW	
Reset									0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7	DLAB	**Divisor Latch Access Bit.** 0: The RX and TX Registers are read/written at Address 0 and the IER register is read/written at Address 4. 1: The Divisor Latch LS is read/written at Address 0 and the Divisor Latch MS is read/written at Address 4.
6	SB	**Set Break** 0: No effect 1: SOUT signal is forced into the "0" state.
5	SP	**Stick Parity** 0: No effect. 1: The Parity bit is forced into a defined state, depending on the states of EPS and PEN: If EPS=1 & PEN=1, the Parity bit is set and checked = 0. If EPS=0 & PEN=1, the Parity bit is set and checked = 1.
4	EPS	**Even Parity Select**

		0: When EPS=0, an odd number of ones is sent and checked. 1: When EPS=1, an even number of ones is sent and checked.
3	PEN	**Parity Enable** 0: The Parity is neither transmitted nor checked. 1: The Parity is transmitted and checked.
2	STB	**Number of STOP bits** 0: One STOP bit is always added. 1: Two STOP bits are added after each character is sent; unless the character length is 5 when 1 STOP bit is added.
1:0	WLS1_WLS0	**Word Length Select.** 0: 5 bits 1: 6 bits 2: 7 bits 3: 8 bits

A00A0010 MCR Modem Control Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									XOFF_STATUS			Loop				
Type									RU			RW				
Reset									0			0				

Bit(s)	Name	Description
7	XOFF_STATUS	**This is a read-only bit.** 0: When an XON character is received. 1: When an XOFF character is received.
4	Loop	**Loop-back control bit.** 0: No loop-back is enabled. 1: Loop-back mode is enabled.

A00A0010 XON1 XON1 Char Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									XON1							
Type									RW							
Reset									0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7:0	XON1	**XON1 character for software flow control.** modified only when LCR=BF'h.

A00A0014 LSR Line Status Register 60

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									FIFOERR	TEMT	THRE	BI	FE	PE	OE	DR
Type									RU	RU	RU	RU	RU	RU	RU	RU
Reset									0	1	1	0	0	0	0	0

Bit(s)	Name	Description

7	FIFOERR	**RX FIFO Error Indicator.** 0: No PE, FE, BI set in the RX FIFO. 1: Set to 1 when there is at least one PE, FE or BI in the RX FIFO.
6	TEMT	**TX Holding Register (or TX FIFO) and the TX Shift Register are empty.** 0: Empty conditions below are not met. 1: If FIFOs are enabled, the bit is set whenever the TX FIFO and the TX Shift Register are empty. If FIFOs are disabled, the bit is set whenever TX Holding Register and TX Shift Register are empty.
5	THRE	**Indicates if there is room for TX Holding Register or TX FIFO is reduced to its Trigger Level.** 0: Reset whenever the contents of the TX FIFO are more than its Trigger Level (FIFOs are enabled), or whenever TX Holding Register is not empty(FIFOs are disabled). 1: Set whenever the contents of the TX FIFO are reduced to its Trigger Level (FIFOs are enabled), or whenever TX Holding Register is empty and ready to accept new data (FIFOs are disabled).
4	BI	**Break Interrupt.** 0: Reset by the CPU reading this register 1: If the FIFOs are disabled, this bit is set whenever the SIN is held in the 0 state for more than one transmission time (START bit + DATA bits + PARITY + STOP bits). If the FIFOs are enabled, this error is associated with a corresponding character in the FIFO and is flagged when this byte is at the top of the FIFO. When a break occurs, only one zero character is loaded into the FIFO: the next character transfer is enabled when SIN goes into the marking state and receives the next valid start bit.
3	FE	**Framing Error.** 0: Reset by the CPU reading this register 1: If the FIFOs are disabled, this bit is set if the received data did not have a valid STOP bit. If the FIFOs are enabled, the state of this bit is revealed when the byte it refers to is the next to be read.
2	PE	**Parity Error** 0: Reset by the CPU reading this register 1: If the FIFOs are disabled, this bit is set if the received data did not have a valid parity bit. If the FIFOs are enabled, the state of this bit is revealed when the referred byte is the next to be read.
1	OE	**Overrun Error.** 0: Reset by the CPU reading this register. 1: If the FIFOs are disabled, this bit is set if the RX Buffer was not read by the CPU before new data from the RX Shift Register overwrote the previous contents. If the FIFOs are enabled, an overrun error occurs when the RX FIFO is full and the RX Shift Register becomes full. OE is set as soon as this happens. The character in the Shift Register is then overwritten, but not transferred to the FIFO.
0	DR	**Data Ready.** 0: Cleared by the CPU reading the RX Buffer or by reading all the FIFO bytes. 1: Set by the RX Buffer becoming full or by the FIFO becoming no empty.

A00A0018 XOFF1 XOFF1 Char Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									XOFF1							
Type									RW							
Reset									0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7:0	XOFF1	**XOFF1 character for software flow control.** modified only when LCR=BF'h.

A00A001C SCR Scratch Register 00

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

Name									**SCR**							
Type									RW							
Reset									0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7:0	SCR	**A general purpose read/write register. After reset, its value is un-defined.** Modified when LCR != BFh.

A00A0020 AUTOBAUD_EN Auto Baud Detect Enable Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														**SLEEP_ACK_SEL**	**AUTOBAUD_SEL**	**AUTOBAUD_EN**
Type														RW	RW	RW
Reset														0	0	0

Bit(s)	Name	Description
2	SLEEP_ACK_SEL	**Sleep ack select when autobaud_en** 0: support sleep_ack when autobaud_en is opened . 1: not support sleep_ack when autobaud_en is opened .
1	AUTOBAUD_SEL	**Auto-baud select** 0: support standard baud rate detection . 1: support non_standard baud rate detection(support baud from 110 to 115200, it is recommend to use 52MHZ to auto fix).
0	AUTOBAUD_EN	**Auto-baud enable signal** 0: Auto-baud function disable 1: Auto-baud function enable (UARTn+0024h SPEED should be set 0) Note: when AUTOBAUD_EN is active, there should not A*/a* char before the auto baud char AT/at, if the A*/a* is Inevitable, the autobaud will fail and please disable the AUTOBAUD_EN to reset the autobaud feature and autobaud_en again.

A00A0024 HIGHSPEED High Speed Mode Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name															**SPEED**	
Type															RW	
Reset															0	0

Bit(s)	Name	Description

1:0	SPEED	**UART sample counter base**
		0: based on 16*baud_pulse, baud_rate = system clock frequency/16/{DLH, DLL}
		1: based on 8*baud_pulse, baud_rate = system clock frequency/8/{DLH, DLL}
		2: based on 4*baud_pulse, baud_rate = system clock frequency/4/{DLH, DLL}
		3: based on sampe_count * baud_pulse, baud_rate = system clock frequency / (sampe_count+1) / {DLM, DLL}

A00A0028 SAMPLE_COUNT Sample Counter Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									**SAMPLECOUNT**							
Type									RW							
Reset									0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7:0	SAMPLECOUNT	**Just be useful when HIGHSPEED mode = 3.**

A00A002C SAMPLE_POINTSample Point Register FF

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									**SAMPLEPOINT**							
Type									RW							
Reset									1	1	1	1	1	1	1	1

Bit(s)	Name	Description
7:0	SAMPLEPOINT	**The SAMPLE_POINT is usually (SAMPLE_COUNT-1)/2 and remove the decimal.** sample point , is effective only When HIGHSPEED=3

A00A0030 AUTOBAUD_REG Auto Baud Monitor Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									**BAUD_STAT**				**BAUD_RATE**			
Type									RU				RU			
Reset									0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7:4	BAUD_STAT	**Autobaud state(only true value in standard autobaud detection)** 0: Autobaud is detecting 1: AT_7N1 2: AT_7O1 3: AT_7E1 4: AT_8N1 5: AT_8O1 6: AT_8E1 7: at_7N1 8: at_7E1 9: at_7O1 10: at_8N1 11: at_8E1 12: at_8O1 13: Autobaud detection fails
3:0	BAUD_RATE	**Autobaud baud rate(only true value in standard autobaud detection)** 0: 115200 1: 57600 2: 38400 3: 19200 4: 9600 5: 4800 6: 2400 7: 1200 8: 300 9: 110

A00A0034 RATEFIX_AD Clock Rate Fix Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														**FREQ**	**AUTO**	**RATE**

														_SEL	BAUD_RATE_FIX	_FIX
Type														RW	RW	RW
Reset														0	0	0

Bit(s)	Name	Description
2	FREQ_SEL	**0: Select 26MHZ as system clock** 1: Select 13MHZ as system clock
1	AUTOBAUD_RATE_FIX	**0: Use 52MHZ as system clock for UART auto baud detect** 1: Use 26MHZ/13MHZ(depends on FREQ_SEL) as system clock for UART auto baud detect
0	RATE_FIX	**0: Use 52MHZ as system clock for UART TX/RX** 1: Use 26MHZ/13MHZ(depends on FREQ_SEL) as system clock for UART TX/RX

A00A0038 AUTOBAUDSAMPLE Auto Baud Sample Register 0D

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name											AUTOBAUDSAMPLE					
Type											RW					
Reset											0	0	1	1	0	1

Bit(s)	Name	Description
5:0	AUTOBAUDSAMPLE	**clk diveision for autobaud rate dectection.** for standard baud rate detection. system clk 52m :'d 27 system clk 26m :'d 13 system clk 13m :'d 6 for non-standard baud rate detection. :15.

A00A003C GUARD Guard time added register 0F

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name												GUARD_EN	GUARD_CNT			
Type												RW	RW			
Reset												0	1	1	1	1

Bit(s)	Name	Description
4	GUARD_EN	**Guard interval add enable signal.** 0: No guard interval added. 1: Add guard interval after stop bit.
3:0	GUARD_CNT	**Guard interval count value. Guard interval = (1/(system clock / div_step / div)) * GUARD_CNT.**

A00A0040 ESCAPE_DAT Escape character register FF

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									ESCAPE_DAT							
Type									RW							
Reset									1	1	1	1	1	1	1	1

Bit(s)	Name	Description
7:0	ESCAPE_DAT	**Escape character added before software flow control data and escape character, i.e. if tx data is xon (31h), with esc_en =1, uart transmits data as esc + CEh (~xon).**

A00A0044 ESCAPE_EN Escape enable register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																ESC_EN
Type																RW
Reset																0

Bit(s)	Name	Description
0	ESC_EN	**Add escape character in transmitter and remove escape character in receiver by UART.** 0: Do not deal with the escape character. 1: Add escape character in transmitter and remove escape character in receiver.

A00A0048 SLEEP_EN Sleep enable register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																SLEEP_EN
Type																RW
Reset																0

Bit(s)	Name	Description
0	SLEEP_EN	**For sleep mode issue** 0: Do not deal with sleep mode indicate signal 1: To activate hardware flow control or software control according to software initial setting when chip enters sleep mode. Releasing hardware flow when chip wakes up; but for software control, uart sends xon when awaken and when FIFO does not reach threshold level.

A00A004C DMA_EN DMA enable register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													FIFO_lsr_sel	TO_CNT_AUTORST	TX_DMA_EN	RX_DMA_EN
Type													RW	RW	RW	RW
Reset													0	0	0	0

Bit(s)	Name	Description
3	FIFO_lsr_sel	**fifo lsr mode selection** 0: lsr will hold the first line status error state until you read the lsr register. 1: lsr will update automatically.
2	TO_CNT_AUTORST	**Timeout counter auto reset register** 0: After RX timeout happen, SW shall reset the interrupt by reading UART 0x4C. 1: The timeout counter will be auto reset. Set this register when Rain's new DMA is used.

1	TX_DMA_EN	**TX_DMA mechanism enable signal** 0: Do not use DMA in TX. 1: Use DMA in TX. When this regusterr is enabled, the flow control is based on the DMA threshold, and generates a timeout interrupt for DMA.for DMA.
0	RX_DMA_EN	**RX_DMA mechanism enable signal** 0: Do not use DMA in RX 1: Use DMA in RX. When this regusterr is enabled, the flow control is based on the DMA threshold, and generates a timeout interrupt

A00A0050 RXTRI_AD Rx Trigger Address 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													RXTRIG			
Type													RW			
Reset													0	0	0	0

Bit(s)	Name	Description
3:0	RXTRIG	**When {rtm,rtl}=2'b11, The Rx FIFO threshold will be Rxtrig.** The value is suggested to be less than half of RX FIFO size, which is 32 Bytes.

A00A0054 FRACDIV_L Fractional Divider LSB Address 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									FRACDIV_L							
Type									RW							
Reset									0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7:0	FRACDIV_L	**Add sampling count (+1) from state data7 to data0, in order to contribute fractional divisor.only when high_speed==3.**

A00A0058 FRACDIV_M Fractional Divider MSB Address 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name															FRACDIV_M	
Type															RW	
Reset															0	0

Bit(s)	Name	Description
1:0	FRACDIV_M	**Add sampling count when in state stop to parity, in order to contribute fractional divisor.only when high_speed==3.**

A00A005C FCR_RD FIFO Control Register 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									RFTL1_RFTL0		TFTL1_TFTL0			CLRT	CLRR	FIFOE
Type									RO		RO			RO	RO	RO
Reset									0	0	0	0		0	0	0

Bit(s)	Name	Description
7:6	RFTL1_RFTL0	**RX FIFO trigger threshold. (RX FIFO contains total 32 bytes.)** 0: 1 1: 6 2: 12 3: RXTRIG
5:4	TFTL1_TFTL0	**TX FIFO trigger threshold (TX FIFO contains total 32 bytes.)** 0: 1 1: 4 2: 8 3: 14
2	CLRT	0: TX FIFO is not cleared 1: TX FIFO is cleared
1	CLRR	0: RX FIFO is not cleared 1: RX FIFOis cleared
0	FIFOE	**FIFO Enabled. This bit must be set to 1 for any of the other bits in the registers to have any effect.** 0: the RX and TX FIFOs are not enabled.. 1: RX and TX FIFOs are enabled.

3.10 I2C/SCCB Controller

3.10.1 General Description

I2C (Inter-IC)/SCCB (Serial Camera Control Bus) is a two-wire serial interface. The two signals are SCL and SDA. SCL is a clock signal that is driven by the master. SDA is a bi-directional data signal that can be driven by either the master or the slave. This generic controller supports the master role and conforms to the I2C specification.

3.10.1.1 Feature

- I2C compliant master mode operation
- Adjustable clock speed for LS/FS mode operation
- Supports 7-bit/10-bit addressing
- Supports high-speed mode
- Supports slave clock extension
- START/STOP/REPEATED START condition
- Manual transfer mode
- Multi-write per transfer (up to 8 data bytes for non-DMA mode)
- Multi-read per transfer (up to 8 data bytes for non-DMA mode)
- Multi-transfer per transaction
- Combined format transfer with length change capability
- Active drive/wired-and I/O configuration

3.10.1.2 Manual Transfer Mode

The controller offers one transfer mode, the manual mode.

When the manual mode is selected, in addition to the slave address register, the controller has a built-in 8byte deep FIFO which allows MCU to prepare up to 8 bytes of data for a write transfer, or read up to 8 bytes of data for a read transfer.

3.10.1.3 Transfer Format Support

This controller has been designed to be as generic as possible in order to support a wide range of devices that may utilize different combinations of transfer formats. Here are the transfer format types that can be supported through different software configuration:

Wording convention note

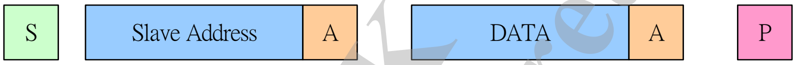
- Transfer = Anything encapsulated within a Start and Stop or Repeated Start.
- Transfer length = Number of bytes within the transfer
- Transaction = This is the top unit. Everything combined equals 1 transaction.
- Transaction length = Number of transfers to be conducted.

🟦 Master to slave dir

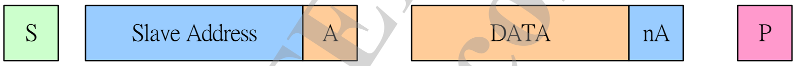
🟧 Slave to master dir

Single-byte access

Single Byte Write

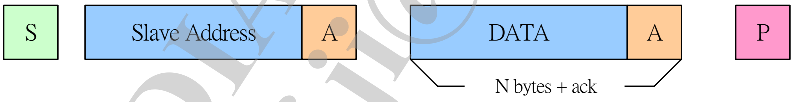


Single Byte Read

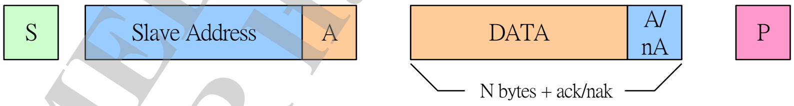


Multi-byte access

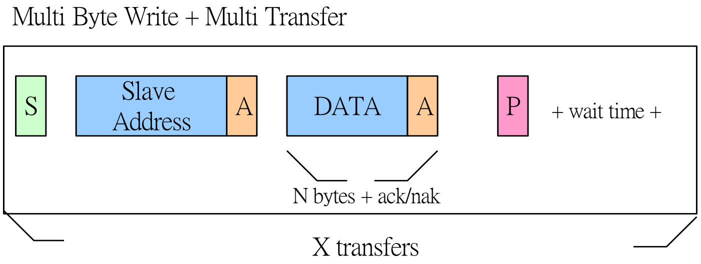
Multi Byte Write

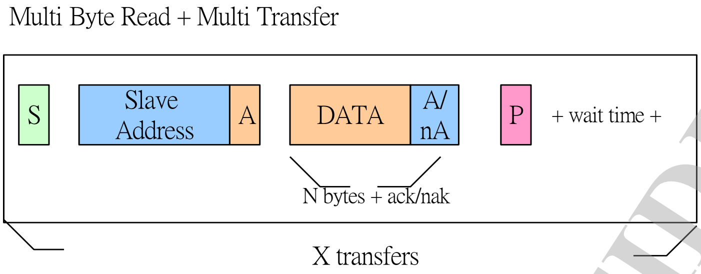


Multi Byte Read

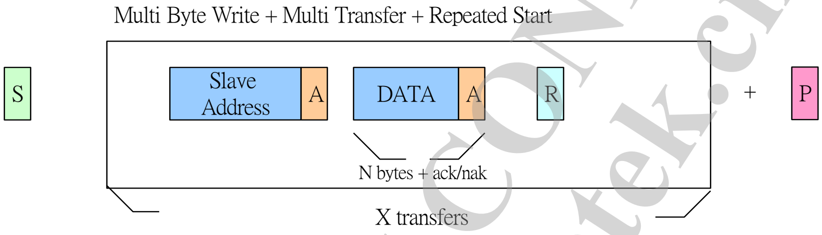


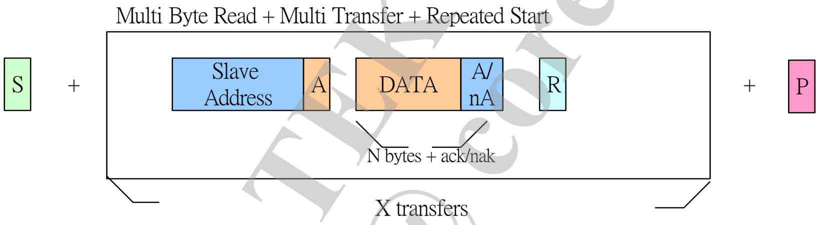
Multi-byte transfer + multi-transfer (same direction)





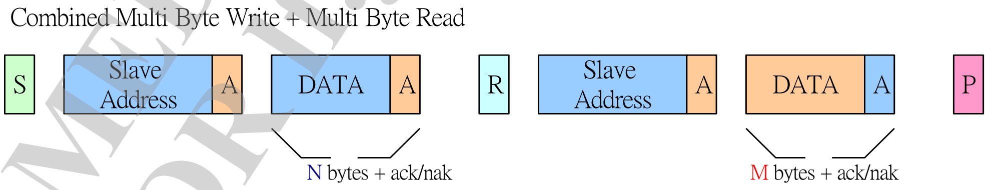
Multi-byte transfer + multi-transfer w RS (same direction)





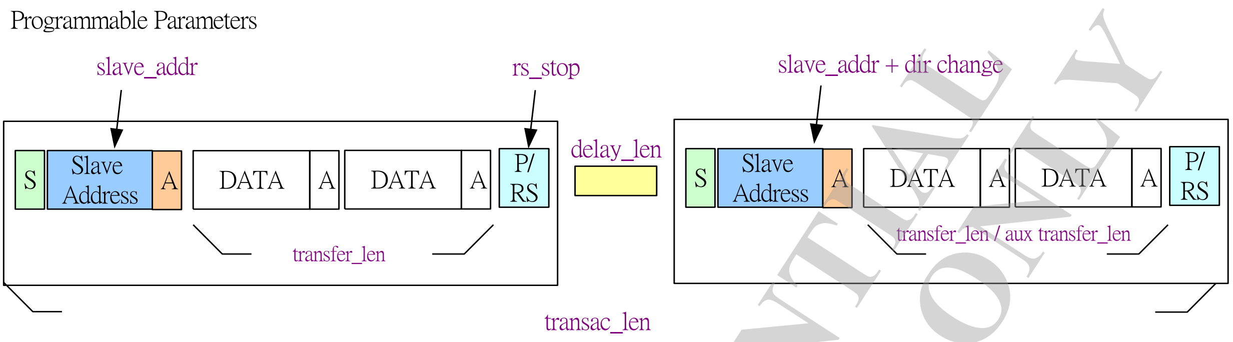
Combined write/read with Repeated Start (direction change)

Note: Only supports write and then read sequence. Read and then write is not supported.

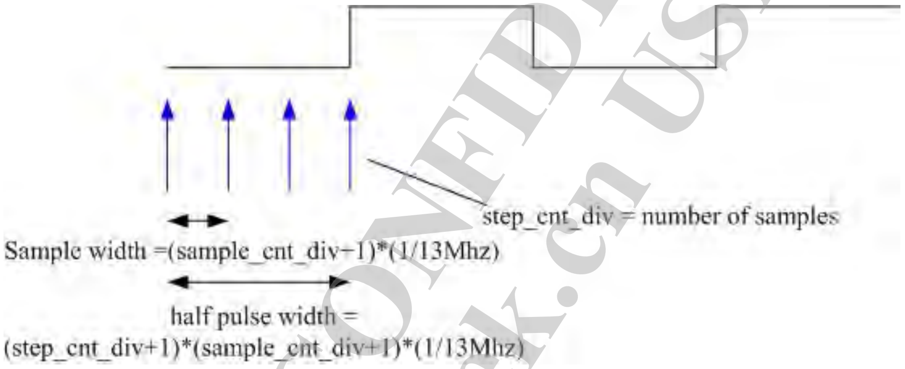


3.10.2 Programming Examples

Common transfer programmable parameters



Output waveform timing programmable parameters


step_cnt_div = number of samples
Sample width =(sample_cnt_div+1)*(1/13Mhz)
half pulse width =
(step_cnt_div+1)*(sample_cnt_div+1)*(1/13Mhz)

3.10.3 Register Definition

Module name: I2C_SCCB_Controller base address: (+A0120000h)

Address	Name	Width	Register function
A0120000	DATA_PORT	16	Data port register
A0120004	SLAVE_ADDR	16	Slave address register
A0120008	INTR_MASK	16	**Interrupt mask register** This register provides masks for the corresponding interrupt sources as indicated in the intr_stat register. 1 = Allow interrupt 0 = Disable interrupt *Note: While disabled, the corresponding interrupt will not be asserted, however intr_stat will still be updated with the status, i.e. mask does not affect intr_stat register values.*
A012000C	INTR_STAT	16	**Interrupt status register** When an interrupt is issued by the I2C controller, this register will need to be read by MCU to determine the cause for the interrupt. After this status has been read and appropriate actions are taken, the corresponding interrupt source will need to be written 1 to clear.
A0120010	CONTROL	16	Control register

Address	Name	Width	Register function
A0120014	TRANSFER_LEN	16	Transfer length register (number of bytes per transfer)
A0120018	TRANSAC_LEN	16	Transaction length register (number of transfers per transaction)
A012001C	DELAY_LEN	16	Inter delay length register
A0120020	TIMING	16	**Timing control register** LS/FS only. This register is used to control the output waveform timing. Each half pulse width, i.e. each high or low pulse, is equal to (step_cnt_div+1)*(sample_cnt_div + 1)/13MHz
A0120024	START	16	Start register
A0120030	FIFO_STAT	16	FIFO status register
A0120038	FIFO_ADDR_CLR	16	FIFO address clear register
A0120040	IO_CONFIG	16	**IO config register** This register is used to configure the I/O for the SDA and SCL lines to select between normal I/O mode, or open-drain mode to support wired-and bus.
A0120048	HS	16	**High speed mode register** This register contains options for supporting high speed operation features. Each HS half pulse width, i.e. each high or low pulse, is equal to (step_cnt_div+1)*(sample_cnt_div + 1)/13MHz
A0120050	SOFTRESET	16	Soft reset register
A0120060	SPARE	16	SPARE
A0120064	DEBUGSTAT	16	Debug status register
A0120068	DEBUGCTRL	16	Debug control register
A012006C	TRANSFER_LEN_AUX	16	Transfer length register (number of bytes per transfer)
A0120074	TIMEOUT	16	Timeout timing register

A0120000 DATA_PORT Data Port Register 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									DATA_PORT							
Type									RW							
Reset									0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
7:0	DATA_PORT	DATA_PORT	**FIFO access port** During master write sequences (slave_addr[0] = 0), this port can be written by APB, and during master read sequences (slave_addr[0] = 1), this port can be read by APB. *Note: Slave_addr must be set correctly before accessing FIFO.*

Bit(s)	Mnemonic	Name	Description
			For debugging only: If the fifo_apb_debug bit is set, FIFO can be read and written by the APB.

A0120004 SLAVE_ADDR Slave Address Register 00BF

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									SLAVE_ADDR							
Type									RW							
Reset									1	0	1	1	1	1	1	1

Bit(s)	Mnemonic	Name	Description
7:0	SLAVE_ADDR	SLAVE_ADDR	**Specifies the slave address of the device to be accessed** Bit 0 is defined by the I2C protocol as a bit that indicates the direction of transfer. 0: Master write 1: Master read

A0120008 INTR_MASK Interrupt Mask Register 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name												MASK_TIMEOUT		MASK_HS_NACKERR	MASK_ACKERR	MASK_TRANSAC_COMP
Type												RW		RW	RW	RW
Reset												0		0	0	0

Overview: This register provides masks for the corresponding interrupt sources as indicated in the intr_stat register. 1 = allow interrupt; 0 = disable interrupt Note that while disabled, the corresponding interrupt will not be asserted. However, intr_stat will still be updated with the status, i.e. mask does not affect intr_stat register values.

Bit(s)	Mnemonic	Name	Description
4	MASK_TIMEOUT	MASK_TIMEOUT	Setting this value to 0 will mask TIMEOUT interrupt signal.
2	MASK_HS_NACKERR	MASK_HS_NACKERR	Setting this value to 0 will mask HS_NACKERR interrupt signal.
1	MASK_ACKERR	MASK_ACKERR	Setting this value to 0 will mask ACK_ERR interrupt signal.
0	MASK_TRANSAC_COMP	MASK_TRANSAC_COMP	Setting this value to 0 will mask TRANSAC_COMP interrupt signal.

A012000C INTR_STAT Interrupt Status Register 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name												TIMEOUT	ARB_LOST	HS_NACKERR	ACKERR	TRANSAC_COMP
Type												W1C	W1C	W1C	W1C	W1C

Reset												0	0	0	0	0

Overview: When an interrupt is issued by the I2C controller, this register will need to be read by MCU to determine the cause for the interrupt. After this status has been read and appropriate actions are taken, the corresponding interrupt source will need to be written 1 to clear.

Bit(s)	Mnemonic	Name	Description
4	**TIMEOUT**	TIMEOUT_IRQ	This status is asserted if time-out is enabled and the timer expires. The internal master state machine will stop, and MCU will need to manually clear the state machine by either issuing software reset by disabling the transact_en bit. Time-out can be used to detect PCB or I2C malfunction.
3	**ARB_LOST**	SPARE	**Reserved**
2	**HS_NACKERR**	HS_NACKERR	This status is asserted if HS master code NACK error detection is enabled. If enabled, HS master code NACK err will cause transaction to end, and stop will be issued.
1	**ACKERR**	ACKERR	This status is asserted if ACK error detection is enabled. If enabled, ACKERR will cause transaction to end, and stop will be issued.
0	**TRANSAC_COMP**	TRANSAC_COMP	This status is asserted when a transaction is completed successfully.

A0120010 **CONTROL** **Control Register** **0000**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name								**TIMEOUT_EN**	**RESET_BUS_BUSY_EN**	**TRANSFER_LEN_CHANGE**	**ACKERR_DET_EN**	**DIR_CHANGE**	**CLK_EXT_EN**		**RS_STOP**	
Type								RW	RW	RW	RW	RW	RW		RW	
Reset								0	0	0	0	0	0		0	

Bit(s)	Mnemonic	Name	Description
8	**TIMEOUT_EN**	TIMEOUT_EN	**Enables time-out mechanism** When enabled, if SCL stays at 0 for too long period of time due to I2C slave holds SCL at 0 for too long or SCL stucks at 0 due to PCB issue, the master shall terminate the transaction, stop the internal state machine and assert time-out interrupt. MCU shall handle this case appropriately and reset the master and FIFO address before reissuing the transaction again. If this option is disabled, the HW timer will not count and nor expire forever. 0: Disable 1: Enable
7	**RESET_BUS_BUSY_EN**	SPARE	**Reserved**
6	**TRANSFER_LEN_CHANGE**	TRANSFER_LEN_CHANGE	**Specifies whether or not to change the transfer length after the fist transfer is completed** If enabled, the transfers after the first transfer will use the transfer_len_aux parameter. 0: Disable 1: Enable
5	**ACKERR_DET_EN**	ACKERR_DET_EN	**Enables slave ACK error detection** When enabled, if slave ACK error is detected, the master shall

Bit(s)	Mnemonic	Name	Description
			terminate the transaction by issuing a STOP condition and then assert the ACKERR interrupt. MCU handles this case appropriately then resets the FIFO address before reissuing transaction. If this option is disabled, the controller will ignore slave ACK error and keep on scheduled transaction. 0: Disable 1: Enable
4	**DIR_CHANGE**	DIR_CHANGE	**Combined transfer format, where the direction of transfer is to be changed from write to read after the FIRST RS condition** *Note: When set to 1, the transfers after the direction change will be based on the transfer_len_aux parameter.* 0: Disable 1: Enable
3	**CLK_EXT_EN**	CLK_EXT_EN	**I2C spec allows slaves to hold the SCL line low if it is not yet ready for further processing** Therefore, if this bit is set to 1, the master controller will enter a high wait state until the slave releases the SCL line.
1	**RS_STOP**	RS_STOP	**In LS/FS mode, this bit affects multi-transfer transaction only.** It controls whether or not the REPEATED-START condition is used between transfers. The last ending transfer always ends with a STOP. In HS mode, this bit must be set to 1. 0: Use STOP 1: Use REPEATED-START

A0120014 TRANSFER_LEN Transfer Length Register (Number of Bytes per Transfer) 0001

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													TRANSFER_LEN_AUX			
Type													RW			
Reset													0	0	0	1

Bit(s)	Mnemonic	Name	Description
3:0	**TRANSFER_LEN_AUX**	TRANSFER_LEN	**Indicates the number of data bytes to be transferred in 1 transfer unit (excluding slave address byte)** *Note: The value must be set to be bigger than 1; otherwise no transfer will take place.*

A0120018 TRANSAC_LEN Transaction Length Register (Number of Transfers per Transaction) 0001

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									TRANSAC_LEN							
Type									RW							
Reset									0	0	0	0	0	0	0	1

Bit(s)	Mnemonic	Name	Description
7:0	**TRANSAC_LEN**	TRANSAC_LEN	**Indicates the number of transfers to be transferred in 1 transaction** *Note: The value must be set to be bigger than 1; otherwise no*

Bit(s)	Mnemonic	Name	Description
			transfer will take place.

A012001C DELAY_LEN Inter Delay Length Register 0002

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									DELAY_LEN							
Type									RW							
Reset									0	0	0	0	0	0	1	0

Bit(s)	Mnemonic	Name	Description
7:0	**DELAY_LEN**	DELAY_LEN	**Sets up wait delay between consecutive transfers when RS_STOP bit is set to 0** Unit: Half the pulse width

A0120020 TIMING Timing Control Register 1303

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DATA_READ_ADJ	DATA_READ_TIME				SAMPLE_CNT_DIV					STEP_CNT_DIV					
Type	RW	RW				RW					RW					
Reset	0	0	0	1		0	1	1			0	0	0	0	1	1

Overview: LS/FS only. This register is used to control the output waveform timing. Each half pulse width, i.e. each high or low pulse, is equal to (step_cnt_div+1)*(sample_cnt_div + 1)/13MHz

Bit(s)	Mnemonic	Name	Description
15	**DATA_READ_ADJ**	DATA_READ_ADJ	When set to 1, data latch in sampling time during master reads are adjusted according to the DATA_READ_TIME value. Otherwise, by default, the data are latched in at half of the high pulse width point. This value must be set to be smaller than or equal to half the high pulse width.
14:12	**DATA_READ_TIME**	DATA_READ_TIME	This value is valid only when DATA_READ_ADJ is set to 1. This can be used to adjust so that the data are latched in at earlier sampling points (assuming data are settled by then).
10:8	**SAMPLE_CNT_DIV**	SAMPLE_CNT_DIV	Used for LS/FS only. This adjusts the width of each sample. Sample width = (sample_cnt_div +1)/13MHz
5:0	**STEP_CNT_DIV**	STEP_CNT_DIV	Specifies the number of samples per half pulse width, i.e. each high or low pulse

A0120024 START Start Register 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																START
Type																RW
Reset																0

Bit(s)	Mnemonic	Name	Description
0	**START**	START	**Starts the transaction on the bus** It is auto de-asserted at the end of the transaction.

A0120030 FIFO_STAT FIFO Status Register 0001

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RD_ADDR				WR_ADDR				FIFO_OFFSET						WR_FULL	RD_EMPTY
Type	RO				RU				RU						RO	RO
Reset	0	0	0	0	0	0	0	0	0	0	0	0			0	1

Bit(s)	Mnemonic	Name	Description
15:12	**RD_ADDR**	RD_ADDR	**Current RD address pointer** Only bit [2:0] have physical meanings.
11:8	**WR_ADDR**	WR_ADDR	**Current WR address pointer** Only bit [2:0] have physical meanings.
7:4	**FIFO_OFFSET**	FIFO_OFFSET	**wr_addr[3:0] - rd_addr[3:0]**
1	**WR_FULL**	WR_FULL	**Indicates FIFO is full**
0	**RD_EMPTY**	RD_EMPTY	**Indicates FIFO is empty**

A0120038 FIFO_ADDR_CLR FIFO Address Clear Register 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																FIFO_ADDR_CLR
Type																WO
Reset																0

Bit(s)	Mnemonic	Name	Description
0	**FIFO_ADDR_CLR**	FIFO_ADDR_CLR	When written with 1'b1, a 1 pulse fifo_addr_clr is generated to clear the FIFO address back to 0.

A0120040 IO_CONFIG IO Config Register 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													IDLE_OE_EN		SDA_IO_CONFIG	SCL_IO_CONFIG
Type													RW		RW	RW
Reset													0		0	0

Overview: This register is used to configure the I/O for the SDA and SCL lines to select between normal I/O mode or open-drain mode to support wired-and bus.

Bit(s)	Mnemonic	Name	Description
3	**IDLE_OE_E**	IDLE_OE_EN	0: Does not drive bus in idle state

Bit(s)	Mnemonic	Name	Description
	N		1: Drive bus in idle state
1	SDA_IO_CONFIG	SDA_IO_CONFIG	0: Normal tristate I/O mode 1: Open-drain mode
0	SCL_IO_CONFIG	SCL_IO_CONFIG	0: Normal tristate I/O mode 1: Open-drain mode

A0120048 HS High Speed Mode Register 0102

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name		HS_SAMPLE_CNT_DIV				HS_STEP_CNT_DIV				MASTER_CODE					HS_NACKERR_DET_EN	HS_EN
Type		RW				RW				RW					RW	RW
Reset		0	0	0		0	0	1		0	0	0			1	0

Overview: This register contains options for supporting high speed operation features Each HS half pulse width, i.e. each high or low pulse, is equal to (step_cnt_div+1)*(sample_cnt_div + 1)/13MHz.

Bit(s)	Mnemonic	Name	Description
14:12	HS_SAMPLE_CNT_DIV	HS_SAMPLE_CNT_DIV	When the high-speed mode is entered after the master code transfer is completed, the sample width will become dependent on this parameter.
10:8	HS_STEP_CNT_DIV	HS_STEP_CNT_DIV	When the high-speed mode is entered after the master code transfer is completed, the number of samples per half pulse width will become dependent on this value.
6:4	MASTER_CODE	MASTER_CODE	This is the 3 bit programmable value for the master code to be transmitted.
1	HS_NACKERR_DET_EN	HS_NACKERR_DET_EN	**Enables NACKERR detection during the master code transmission** When enabled, if NACK is not received after the master code is transmitted, the transaction will be terminated with a STOP condition.
0	HS_EN	HS_EN	**Enables high-speed transaction** *Note: rs_stop must be set to 1.*

A0120050 SOFTRESET Soft Reset Register 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																SOFT_RESET
Type																WO
Reset																0

Bit(s)	Mnemonic	Name	Description
0	SOFT_RESET	SOFT_RESET	When written with 1'b1, a 1 pulse soft reset is used as synchronous reset to reset the I2C internal hardware circuits.

A0120060 SPARE SPARE 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													SPARE			
Type													RW			
Reset													0	0	0	0

Bit(s)	Mnemonic	Name	Description
3:0	**SPARE**	SPARE	Reserved for future use

A0120064 DEBUGSTAT Debug Status Register 0020

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									BUS_BUSY	MASTER_WRITE	MASTER_READ	MASTER_STATE				
Type									RO	RO	RO	RO				
Reset									0	0	1	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
7	**BUS_BUSY**	SPARE	**Reserved**
6	**MASTER_WRITE**	MASTER_WRITE	**For debugging only** 1: Current transfer is in the master write dir.
5	**MASTER_READ**	MASTER_READ	**For debugging only** 1: Current transfer is in the master read dir.
4:0	**MASTER_STATE**	MASTER_STATE	**(For debugging only) Reads back the current master_state.** 0: Idle state 1: I2c master is preparing to send out the start bit, SCL=1, SDA=1. 2: I2C master is sending out the start bit, SCL=1, SDA=0. 3: I2C master/slave is preparing to transmit data bit, SCL=0, SDA=data bit. (Data bit can be changed when SCL=0.) 4: I2C master/slave is transmitting data bit, SCL=1, SDA=data bit. (Data bit is stable when SCL=1.) 5: I2C master/slave is preparing to transmit the ACK bit, SCL=0, SDA=ack. (The ACK bit can be changed when SCL=0.) 6: I2C master/slave is transmitting the ACK bit, SCL=1, SDA=0. (The ACK bit is stable when SCL=1.) 7: I2C master is preparing to send out stop bit or repeated-start bit, SCL=0, SDA=0/1. (0: Stop bit; 1: Repeated-start bit) 8: I2C master is sending out stop bit or repeated-start bit, SCL=1, SDA=1/0. (0: Repeated-start bit; 1: Stop bit) 9: I2C master is in delay start between two transfers, SCL=1, SDA=1. 10: I2C master is in FIFO wait state; For writing transaction, it means FIFO is empty and I2C master is waiting for DMA controller to write data into FIFo. For reading transaction, it means FIFO is full and I2C master is waiting for DMA controller to read data from FIFOo, SCL=0, SDA=don't care. 12: I2C master is preparing to send out data bit of master code. This state is used only in high-speed transaction, SCL=0, SDA=data bit of master code. (Data bit of master code can be changed when SCL=0.)

Bit(s)	Mnemonic	Name	Description
			13: I2C master is sending out data bit of master code. This state is used only in high-speed transaction, SCL=1, SDA=data bit of master code. (Data bit of master code is stable when SCL=1.) 14: I2C master/slave is preparing to transmit the NACK bit, SCL=0, SDA=nack bit. (The NACK bit can be changed when SCL=0.) This state is used only in high-speed transaction. 15: I2C master/slave is transmitting the NACK bit, SCL=1, SDA=1. This state is used only in high-speed transaction.

A0120068 DEBUGCTRL Debug Control Register 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name															APB_DEBUG_RD	FIFO_APB_DEBUG
Type															WO	RW
Reset															0	0

Bit(s)	Mnemonic	Name	Description
1	APB_DEBUG_RD	APB_DEBUG_RD	**Only valid when fifo_apb_debug is set to 1** Writing to this register will generate a 1 pulsed FIFO APB RD signal for reading the FIFO data.
0	FIFO_APB_DEBUG	FIFO_APB_DEBUG	**Used for trace 32 debugging** When using trace 32, and the memory map is shown, turning this bit on will block the normal APB read access. The APB read access to the FIFO will then be enabled by writing to apb_debug_rd. 0: Disable 1: Enable

A012006C TRANSFER_LEN_AUX Transfer Length Register (Number of Bytes per Transfer) 0001

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													TRANSFER_LEN			
Type													RW			
Reset													0	0	0	1

Bit(s)	Mnemonic	Name	Description
3:0	TRANSFER_LEN	TRANSFER_LEN_AUX	**Only valid when dir_change or transfer_len_change is set to 1. Indicates the number of data bytes to be transferred in 1 transfer unit (excluding slave address byte) for the transfers following the direction change or transfer_len_change** If dir_change =1, the first write transfer length will depend on transfer_len, while the second read transfer length will depend on transfer_len_aux. Dir change is always after the first transfer. Similarly, transfer length change is always after the first transfer. *Note: The value must be set to be bigger than 1; otherwise no transfer will take place.*

A0120074 **TIMEOUT** **Timeout Timing Register** **FFFF**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	TIMEOUT															
Type	RW															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit(s)	Mnemonic	Name	Description
15:0		TIMEOUT	**Indicates the number of steps to count before time-out** The time-out counter counts only when the time-out mechanism is enabled and has started a transaction.

3.11 I2C/SCCB Controller (I2C_SCCB_Controller_V18)

3.11.1 General Description

I2C (Inter-IC)/SCCB (Serial Camera Control Bus) is a two-wire serial interface. The two signals are SCL and SDA. SCL is a clock signal that is driven by the master. SDA is a bi-directional data signal that can be driven by either the master or the slave. This generic controller supports the master role and conforms to the I2C specification.

3.11.1.1 Feature

- I2C compliant master mode operation
- Adjustable clock speed for LS/FS mode operation
- Supports 7-bit/10-bit addressing
- Supports high-speed mode
- Supports slave clock extension
- START/STOP/REPEATED START condition
- Manual transfer mode
- Multi-write per transfer (up to 8 data bytes for non-DMA mode)
- Multi-read per transfer (up to 8 data bytes for non-DMA mode)
- Multi-transfer per transaction
- Combined format transfer with length change capability
- Active drive/wired-and I/O configuration

3.11.1.2 Manual Transfer Mode

The controller offers one transfer mode, the manual mode.

When the manual mode is selected, in addition to the slave address register, the controller has a built-in 8byte deep FIFO which allows MCU to prepare up to 8 bytes of data for a write transfer, or read up to 8 bytes of data for a read transfer.

3.11.1.3 Transfer Format Support

This controller has been designed to be as generic as possible in order to support a wide range of devices that may utilize different combinations of transfer formats. Here are the transfer format types that can be supported through different software configuration:

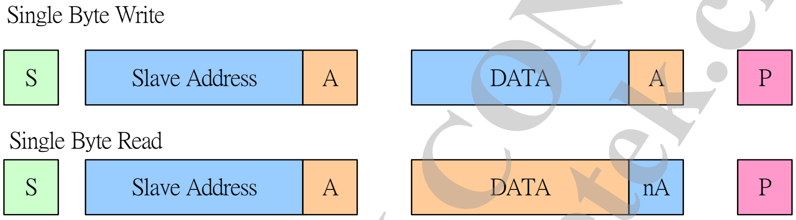
Wording convention note

- Transfer = Anything encapsulated within a Start and Stop or Repeated Start.
- Transfer length = Number of bytes within the transfer
- Transaction = This is the top unit. Everything combined equals 1 transaction.
- Transaction length = Number of transfers to be conducted.

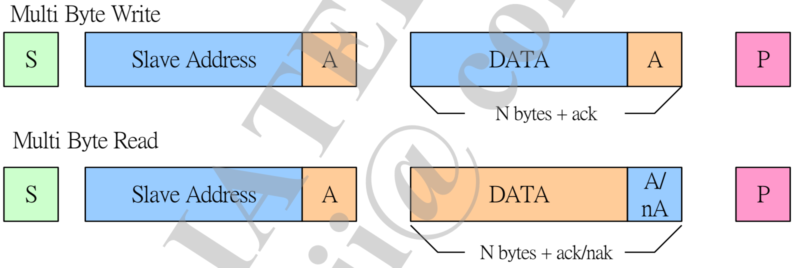
🟦 Master to slave dir

🟧 Slave to master dir

Single-byte access

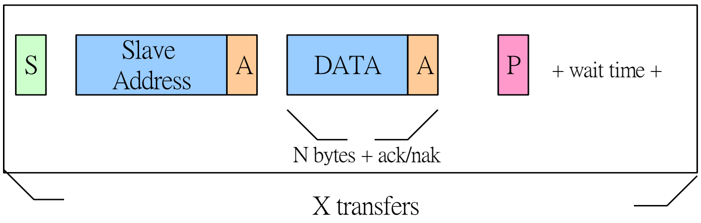


Multi-byte access

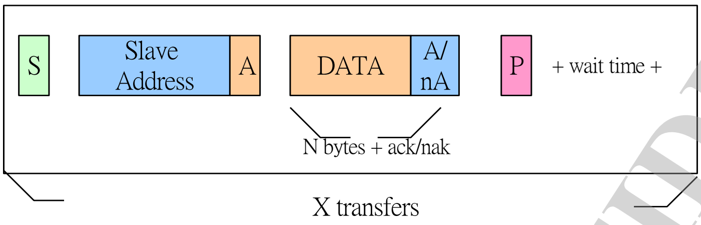


Multi-byte transfer + multi-transfer (same direction)

Multi Byte Write + Multi Transfer

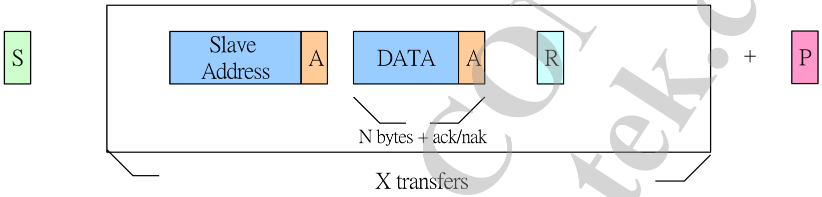


Multi Byte Read + Multi Transfer

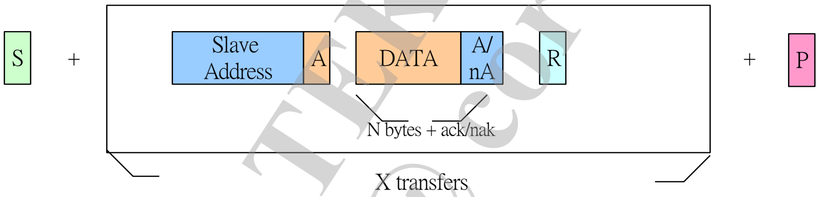


Multi-byte transfer + multi-transfer w RS (same direction)

Multi Byte Write + Multi Transfer + Repeated Start



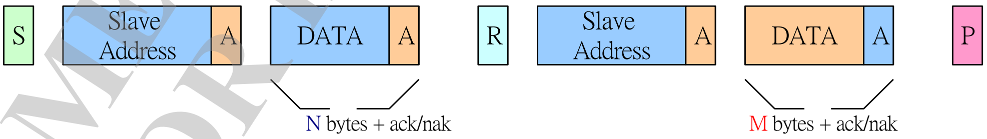
Multi Byte Read + Multi Transfer + Repeated Start



Combined write/read with Repeated Start (direction change)

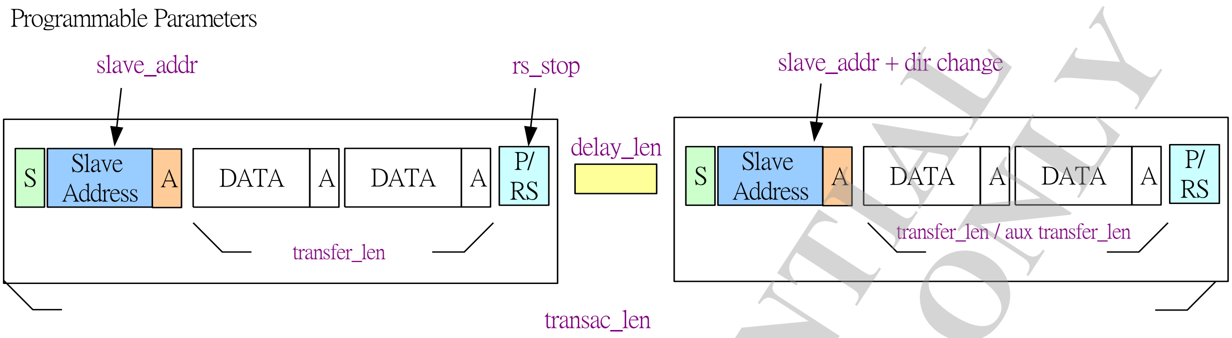
Note: Only supports write and then read sequence. Read and then write is not supported.

Combined Multi Byte Write + Multi Byte Read

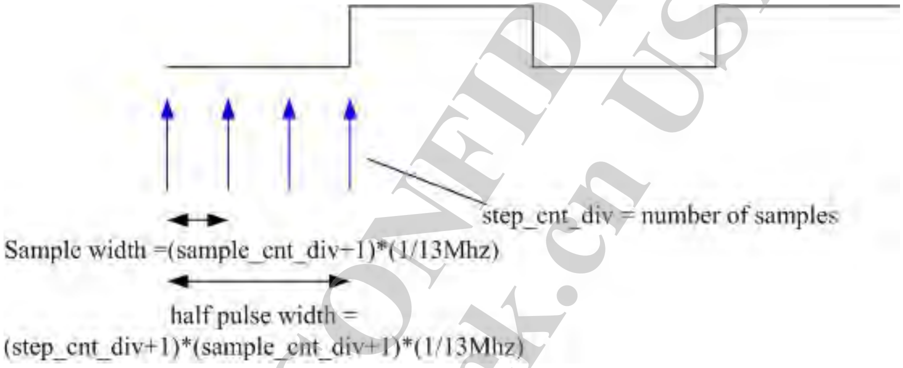


3.11.2 Programming Examples

Common transfer programmable parameters



Output waveform timing programmable parameters


step_cnt_div = number of samples
Sample width =(sample_cnt_div+1)*(1/13Mhz)
half pulse width =
(step_cnt_div+1)*(sample_cnt_div+1)*(1/13Mhz)

3.11.3 Register Definition

Module name: I2C_SCCB_Controller_V18 base address: (+A02A0000h)

Address	Name	Width	Register function
A02A0000	DATA_PORT	16	Data port register
A02A0004	SLAVE_ADDR	16	Slave address register
A02A0008	INTR_MASK	16	**Interrupt mask register** This register provides masks for the corresponding interrupt sources as indicated in the intr_stat register. 1 = Allow interrupt 0 = Disable interrupt *Note: While disabled, the corresponding interrupt will not be asserted, however intr_stat will still be updated with the status, i.e. mask does not affect intr_stat register values.*
A02A000C	INTR_STAT	16	**Interrupt status register** When an interrupt is issued by the I2C controller, this register will need to be read by MCU to determine the cause for the interrupt. After this status has been read and appropriate actions are taken, the corresponding interrupt source will need to be written 1 to clear.
A02A0010	CONTROL	16	Control register

Address	Name	Width	Register function
A02A0014	TRANSFER_LEN	16	Transfer length register (number of bytes per transfer)
A02A0018	TRANSAC_LEN	16	Transaction length register (number of transfers per transaction)
A02A001C	DELAY_LEN	16	Inter delay length register
A02A0020	TIMING	16	**Timing control register** LS/FS only. This register is used to control the output waveform timing. Each half pulse width, i.e. each high or low pulse, is equal to (step_cnt_div+1)*(sample_cnt_div + 1)/13MHz
A02A0024	START	16	Start register
A02A0030	FIFO_STAT	16	FIFO status register
A02A0038	FIFO_ADDR_CLR	16	FIFO address clear register
A02A0040	IO_CONFIG	16	**IO config register** This register is used to configure the I/O for the SDA and SCL lines to select between normal I/O mode, or open-drain mode to support wired-and bus.
A02A0048	HS	16	**High speed mode register** This register contains options for supporting high speed operation features. Each HS half pulse width, i.e. each high or low pulse, is equal to (step_cnt_div+1)*(sample_cnt_div + 1)/13MHz
A02A0050	SOFTRESET	16	Soft reset register
A02A0060	SPARE	16	SPARE
A02A0064	DEBUGSTAT	16	Debug status register
A02A0068	DEBUGCTRL	16	Debug control register
A02A006C	TRANSFER_LEN_AUX	16	Transfer length register (number of bytes per transfer)
A02A0074	TIMEOUT	16	Timeout timing register

A02A0000 DATA_PORT Data Port Register 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									DATA_PORT							
Type									RW							
Reset									0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
7:0	DATA_PORT	DATA_PORT	**FIFO access port** During master write sequences (slave_addr[0] = 0), this port can be written by APB, and during master read sequences (slave_addr[0] = 1), this port can be read by APB. *Note: Slave_addr must be set correctly before accessing FIFO.*

Bit(s)	Mnemonic	Name	Description
			For debugging only: If the fifo_apb_debug bit is set, FIFO can be read and written by the APB.

A02A0004 SLAVE_ADDR Slave Address Register 00BF

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									SLAVE_ADDR							
Type									RW							
Reset									1	0	1	1	1	1	1	1

Bit(s)	Mnemonic	Name	Description
7:0	**SLAVE_ADDR**	SLAVE_ADDR	**Specifies the slave address of the device to be accessed** Bit 0 is defined by the I2C protocol as a bit that indicates the direction of transfer. 0: Master write 1: Master read

A02A0008 INTR_MASK Interrupt Mask Register 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name												MASK_TIMEOUT		MASK_HS_NACKERR	MASK_ACKERR	MASK_TRANSAC_COMP
Type												RW		RW	RW	RW
Reset												0		0	0	0

Overview: This register provides masks for the corresponding interrupt sources as indicated in the intr_stat register. 1 = allow interrupt; 0 = disable interrupt. Note that while disabled, the corresponding interrupt will not be asserted. However, intr_stat will still be updated with the status, i.e. mask does not affect intr_stat register values.

Bit(s)	Mnemonic	Name	Description
4	**MASK_TIMEOUT**	MASK_TIMEOUT	Setting this value to 0 will mask TIMEOUT interrupt signal.
2	**MASK_HS_NACKERR**	MASK_HS_NACKERR	Setting this value to 0 will mask HS_NACKERR interrupt signal.
1	**MASK_ACKERR**	MASK_ACKERR	Setting this value to 0 will mask ACK_ERR interrupt signal.
0	**MASK_TRANSAC_COMP**	MASK_TRANSAC_COMP	Setting this value to 0 will mask TRANSAC_COMP interrupt signal.

A02A000C INTR_STAT Interrupt Status Register 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name												TIMEOUT	ARB_LOST	HS_NACKERR	ACKERR	TRANSAC_COMP
Type												W1C	W1C	W1C	W1C	W1C

Reset												0	0	0	0	0

Overview: When an interrupt is issued by the I2C controller, this register will need to be read by MCU to determine the cause for the interrupt. After this status has been read and appropriate actions are taken, the corresponding interrupt source will need to be written 1 to clear.

Bit(s)	Mnemonic	Name	Description
4	**TIMEOUT**	TIMEOUT_IRQ	This status is asserted if time-out is enabled and the timer expires. The internal master state machine will stop, and MCU will need to manually clear the state machine by either issuing software reset by disabling the transact_en bit. Time-out can be used to detect PCB or I2C malfunction.
3	**ARB_LOST**	SPARE	**Reserved**
2	**HS_NACKERR**	HS_NACKERR	This status is asserted if HS master code NACK error detection is enabled. If enabled, HS master code NACK err will cause transaction to end, and stop will be issued.
1	**ACKERR**	ACKERR	This status is asserted if ACK error detection is enabled. If enabled, ACKERR will cause transaction to end, and stop will be issued.
0	**TRANSAC_COMP**	TRANSAC_COMP	This status is asserted when a transaction is completed successfully.

A02A0010 CONTROL Control Register 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name								TIMEOUT_EN	RESET_BUS_BUSY_EN	TRANSFER_LEN_CHANGE	ACKERR_DET_EN	DIR_CHANGE	CLK_EXT_EN		RS_STOP	
Type								RW	RW	RW	RW	RW	RW		RW	
Reset								0	0	0	0	0	0		0	

Bit(s)	Mnemonic	Name	Description
8	**TIMEOUT_EN**	TIMEOUT_EN	**Enables time-out mechanism** When enabled, if SCL stays at 0 for too long period of time due to I2C slave holds SCL at 0 for too long or SCL stucks at 0 due to PCB issue, the master shall terminate the transaction, stop the internal state machine and assert time-out interrupt. MCU shall handle this case appropriately and reset the master and FIFO address before reissuing the transaction again. If this option is disabled, the HW timer will not count and nor expire forever. 0: Disable 1: Enable
7	**RESET_BUS_BUSY_EN**	SPARE	**Reserved**
6	**TRANSFER_LEN_CHANGE**	TRANSFER_LEN_CHANGE	**Specifies whether or not to change the transfer length after the fist transfer is completed** If enabled, the transfers after the first transfer will use the transfer_len_aux parameter. 0: Disable 1: Enable
5	**ACKERR_DET_EN**	ACKERR_DET_EN	**Enables slave ACK error detection** When enabled, if slave ACK error is detected, the master shall

Bit(s)	Mnemonic	Name	Description
			terminate the transaction by issuing a STOP condition and then assert the ACKERR interrupt. MCU handles this case appropriately then resets the FIFO address before reissuing transaction. If this option is disabled, the controller will ignore slave ACK error and keep on scheduled transaction. 0: Disable 1: Enable
4	**DIR_CHANGE**	DIR_CHANGE	**Combined transfer format, where the direction of transfer is to be changed from write to read after the FIRST RS condition** *Note: When set to 1, the transfers after the direction change will be based on the transfer_len_aux parameter.* 0: Disable 1: Enable
3	**CLK_EXT_EN**	CLK_EXT_EN	**I2C spec allows slaves to hold the SCL line low if it is not yet ready for further processing** Therefore, if this bit is set to 1, the master controller will enter a high wait state until the slave releases the SCL line.
1	**RS_STOP**	RS_STOP	**In LS/FS mode, this bit affects multi-transfer transaction only.** It controls whether or not the REPEATED-START condition is used between transfers. The last ending transfer always ends with a STOP. In HS mode, this bit must be set to 1. 0: Use STOP 1: Use REPEATED-START

A02A0014 **TRANSFER_LEN** **Transfer Length Register (Number of Bytes per Transfer)** **0001**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													TRANSFER_LEN_AUX			
Type													RW			
Reset													0	0	0	1

Bit(s)	Mnemonic	Name	Description
3:0	**TRANSFER_LEN_AUX**	TRANSFER_LEN	**Indicates the number of data bytes to be transferred in 1 transfer unit (excluding slave address byte)** *Note: The value must be set to be bigger than 1; otherwise no transfer will take place.*

A02A0018 **TRANSAC_LEN** **Transaction Length Register (Number of Transfers per Transaction)** **0001**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									TRANSAC_LEN							
Type									RW							
Reset									0	0	0	0	0	0	0	1

Bit(s)	Mnemonic	Name	Description
7:0	**TRANSAC_LEN**	TRANSAC_LEN	**Indicates the number of transfers to be transferred in 1 transaction** *Note: The value must be set to be bigger than 1; otherwise no*

Bit(s)	Mnemonic	Name	Description
			transfer will take place.

A02A001C DELAY_LEN Inter Delay Length Register 0002

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									DELAY_LEN							
Type									RW							
Reset									0	0	0	0	0	0	1	0

Bit(s)	Mnemonic	Name	Description
7:0	**DELAY_LEN**	DELAY_LEN	**Sets up wait delay between consecutive transfers when RS_STOP bit is set to 0** Unit: Half the pulse width

A02A0020 TIMING Timing Control Register 1303

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DATA_READ_ADJ	DATA_READ_TIME				SAMPLE_CNT_DIV					STEP_CNT_DIV					
Type	RW	RW				RW					RW					
Reset	0	0	0	1		0	1	1			0	0	0	0	1	1

Overview: LS/FS only. This register is used to control the output waveform timing. Each half pulse width, i.e. each high or low pulse, is equal to (step_cnt_div+1)*(sample_cnt_div + 1)/13MHz

Bit(s)	Mnemonic	Name	Description
15	**DATA_READ_ADJ**	DATA_READ_ADJ	When set to 1, data latch in sampling time during master reads are adjusted according to the DATA_READ_TIME value. Otherwise, by default, the data are latched in at half of the high pulse width point. This value must be set to be smaller than or equal to half the high pulse width.
14:12	**DATA_READ_TIME**	DATA_READ_TIME	This value is valid only when DATA_READ_ADJ is set to 1. This can be used to adjust so that the data are latched in at earlier sampling points (assuming data are settled by then).
10:8	**SAMPLE_CNT_DIV**	SAMPLE_CNT_DIV	Used for LS/FS only. This adjusts the width of each sample. Sample width = (sample_cnt_div +1)/13MHz
5:0	**STEP_CNT_DIV**	STEP_CNT_DIV	Specifies the number of samples per half pulse width, i.e. each high or low pulse

A02A0024 START Start Register 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																START
Type																RW
Reset																0

Bit(s)	Mnemonic	Name	Description
0	**START**	START	**Starts the transaction on the bus** It is auto de-asserted at the end of the transaction.

A02A0030 FIFO_STAT FIFO Status Register 0001

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RD_ADDR				WR_ADDR				FIFO_OFFSET						WR_FULL	RD_EMPTY
Type	RO				RU				RU						RO	RO
Reset	0	0	0	0	0	0	0	0	0	0	0	0			0	1

Bit(s)	Mnemonic	Name	Description
15:12	**RD_ADDR**	RD_ADDR	**Current RD address pointer** Only bit [2:0] have physical meanings.
11:8	**WR_ADDR**	WR_ADDR	**Current WR address pointer** Only bit [2:0] have physical meanings.
7:4	**FIFO_OFFSET**	FIFO_OFFSET	**wr_addr[3:0] - rd_addr[3:0]**
1	**WR_FULL**	WR_FULL	**Indicates FIFO is full**
0	**RD_EMPTY**	RD_EMPTY	**Indicates FIFO is empty**

A02A0038 FIFO_ADDR_CLR FIFO Address Clear Register 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																FIFO_ADDR_CLR
Type																WO
Reset																0

Bit(s)	Mnemonic	Name	Description
0	**FIFO_ADDR_CLR**	FIFO_ADDR_CLR	When written with 1'b1, a 1 pulse fifo_addr_clr is generated to clear the FIFO address back to 0.

A02A0040 IO_CONFIG IO Config Register 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													IDLE_OE_EN		SDA_IO_CONFIG	SCL_IO_CONFIG
Type													RW		RW	RW
Reset													0		0	0

Overview: This register is used to configure the I/O for the SDA and SCL lines to select between normal I/O mode or open-drain mode to support wired-and bus.

Bit(s)	Mnemonic	Name	Description
3	**IDLE_OE_E**	IDLE_OE_EN	0: Does not drive bus in idle state

Bit(s)	Mnemonic	Name	Description
	N		1: Drive bus in idle state
1	SDA_IO_CONFIG	SDA_IO_CONFIG	0: Normal tristate I/O mode 1: Open-drain mode
0	SCL_IO_CONFIG	SCL_IO_CONFIG	0: Normal tristate I/O mode 1: Open-drain mode

A02A0048 HS High Speed Mode Register 0102

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name		HS_SAMPLE_CNT_DIV				HS_STEP_CNT_DIV				MASTER_CODE					HS_NACKERR_DET_EN	HS_EN
Type		RW				RW				RW					RW	RW
Reset		0	0	0		0	0	1		0	0	0			1	0

Overview: This register contains options for supporting high speed operation features Each HS half pulse width, i.e. each high or low pulse, is equal to (step_cnt_div+1)*(sample_cnt_div + 1)/13MHz.

Bit(s)	Mnemonic	Name	Description
14:12	**HS_SAMPLE_CNT_DIV**	HS_SAMPLE_CNT_DIV	When the high-speed mode is entered after the master code transfer is completed, the sample width will become dependent on this parameter.
10:8	**HS_STEP_CNT_DIV**	HS_STEP_CNT_DIV	When the high-speed mode is entered after the master code transfer is completed, the number of samples per half pulse width will become dependent on this value.
6:4	**MASTER_CODE**	MASTER_CODE	This is the 3 bit programmable value for the master code to be transmitted.
1	**HS_NACKERR_DET_EN**	HS_NACKERR_DET_EN	**Enables NACKERR detection during the master code transmission** When enabled, if NACK is not received after the master code is transmitted, the transaction will be terminated with a STOP condition.
0	**HS_EN**	HS_EN	**Enables high-speed transaction** *Note: rs_stop must be set to 1.*

A02A0050 SOFTRESET Soft Reset Register 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																SOFT_RESET
Type																WO
Reset																0

Bit(s)	Mnemonic	Name	Description
0	**SOFT_RESET**	SOFT_RESET	When written with 1'b1, a 1 pulse soft reset is used as synchronous reset to reset the I2C internal hardware circuits.

A02A0060 SPARE SPARE 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													SPARE			
Type													RW			
Reset													0	0	0	0

Bit(s)	Mnemonic	Name	Description
3:0	**SPARE**	SPARE	Reserved for future use

A02A0064 DEBUGSTAT Debug Status Register 0020

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									BUS_BUSY	MASTER_WRITE	MASTER_READ	MASTER_STATE				
Type									RO	RO	RO	RO				
Reset									0	0	1	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
7	**BUS_BUSY**	SPARE	**Reserved**
6	**MASTER_WRITE**	MASTER_WRITE	**For debugging only** 1: Current transfer is in the master write dir.
5	**MASTER_READ**	MASTER_READ	**For debugging only** 1: Current transfer is in the master read dir.
4:0	**MASTER_STATE**	MASTER_STATE	**(For debugging only) Reads back the current master_state.** 0: Idle state 1: I2c master is preparing to send out the start bit, SCL=1, SDA=1. 2: I2C master is sending out the start bit, SCL=1, SDA=0. 3: I2C master/slave is preparing to transmit data bit, SCL=0, SDA=data bit. (Data bit can be changed when SCL=0.) 4: I2C master/slave is transmitting data bit, SCL=1, SDA=data bit. (Data bit is stable when SCL=1.) 5: I2C master/slave is preparing to transmit the ACK bit, SCL=0, SDA=ack. (The ACK bit can be changed when SCL=0.) 6: I2C master/slave is transmitting the ACK bit, SCL=1, SDA=0. (The ACK bit is stable when SCL=1.) 7: I2C master is preparing to send out stop bit or repeated-start bit, SCL=0, SDA=0/1. (0: Stop bit; 1: Repeated-start bit) 8: I2C master is sending out stop bit or repeated-start bit, SCL=1, SDA=1/0. (0: Repeated-start bit; 1: Stop bit) 9: I2C master is in delay start between two transfers, SCL=1, SDA=1. 10: I2C master is in FIFO wait state; For writing transaction, it means FIFO is empty and I2C master is waiting for DMA controller to write data into FIFo. For reading transaction, it means FIFO is full and I2C master is waiting for DMA controller to read data from FIFOo, SCL=0, SDA=don't care. 12: I2C master is preparing to send out data bit of master code. This state is used only in high-speed transaction, SCL=0, SDA=data bit of master code. (Data bit of master code can be changed when SCL=0.)

Bit(s)	Mnemonic	Name	Description
			13: I2C master is sending out data bit of master code. This state is used only in high-speed transaction, SCL=1, SDA=data bit of master code. (Data bit of master code is stable when SCL=1.) 14: I2C master/slave is preparing to transmit the NACK bit, SCL=0, SDA=nack bit. (The NACK bit can be changed when SCL=0.) This state is used only in high-speed transaction. 15: I2C master/slave is transmitting the NACK bit, SCL=1, SDA=1. This state is used only in high-speed transaction.

A02A0068 DEBUGCTRL Debug Control Register 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name															APB_DEBUG_RD	FIFO_APB_DEBUG
Type															WO	RW
Reset															0	0

Bit(s)	Mnemonic	Name	Description
1	**APB_DEBUG_RD**	APB_DEBUG_RD	**Only valid when fifo_apb_debug is set to 1** Writing to this register will generate a 1 pulsed FIFO APB RD signal for reading the FIFO data.
0	**FIFO_APB_DEBUG**	FIFO_APB_DEBUG	**Used for trace 32 debugging** When using trace 32, and the memory map is shown, turning this bit on will block the normal APB read access. The APB read access to the FIFO will then be enabled by writing to apb_debug_rd. 0: Disable 1: Enable

A02A006C TRANSFER_LEN_AUX Transfer Length Register (Number of Bytes per Transfer) 0001

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													TRANSFER_LEN			
Type													RW			
Reset													0	0	0	1

Bit(s)	Mnemonic	Name	Description
3:0	**TRANSFER_LEN**	TRANSFER_LEN_AUX	**Only valid when dir_change or transfer_len_change is set to 1. Indicates the number of data bytes to be transferred in 1 transfer unit (excluding slave address byte) for the transfers following the direction change or transfer_len_change** If dir_change =1, the first write transfer length will depend on transfer_len, while the second read transfer length will depend on transfer_len_aux. Dir change is always after the first transfer. Similarly, transfer length change is always after the first transfer. *Note: The value must be set to be bigger than 1; otherwise no transfer will take place.*

A02A0074 TIMEOUT Timeout Timing Register FFFF

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	TIMEOUT															
Type	RW															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit(s)	Mnemonic	Name	Description
15:0		TIMEOUT	**Indicates the number of steps to count before time-out** The time-out counter counts only when the time-out mechanism is enabled and has started a transaction.

3.12 Real Time Clock

3.12.1 General Descriptions

The Real-Time Clock (RTC) module provides time and data information, as well as 32.768kHz clock. By configuring pin XOSC32_ENB, the use of the 32k crystal can be determined, i.e. using a 32k crystal, or not to use a 32k crystal. The RTC block has an independent power supply. When the mobile handset is powered off, a dedicated regulator supplies the RTC block. If the main battery is not present, a backup supply such as a small mercury cell battery or a large capacitor is used. In addition to providing timing data, an alarm interrupt is generated and can be used to power up the baseband core. Regulator interrupts corresponding to seconds, minutes, hours and days can be generated whenever the time counter value reaches a maximum value (e.g., 59 for seconds and minutes, 23 for hours, etc.). The year span is supported up to 2127. The maximum day-of-month values, which depend on the leap year condition, are stored in the RTC block.

3.12.2 Register Definition

Module name: RTC Base address: (+A0710000h)

Address	Name	Width	Register function
A0710000	RTC_BBPU	16	**Baseband power up**
A071 0004	RTC_IRQ_STA	16	**RTC IRQ status** This register is fixed to 0 when RTC_POWERKEY1 & RTC_POWERKEY2 unmatch the correct values.
A071 0008	RTC_IRQ_EN	16	**RTC IRQ enable** This register is fixed to 0 when RTC_POWERKEY1 & RTC_POWERKEY2 unmatch the correct values.
A071 000C	RTC_CII_EN	16	**Counter increment IRQ enable** This register activates or de-activates the IRQ generation when the TC counter reaches its maximum value.
A071 0010	RTC_AL_MASK	16	**RTC alarm mask** The alarm condition for alarm IRQ generation depends on whether or not the corresponding bit in this register is masked. Warning: If you set all bits to 1 in

Address	Name	Width	Register function
			RTC_AL_MASK (i.e. RTC_AL_MASK=0x7f) and PWREN=1 in RTC_BBPU, it means the alarm will come every second, not disabled.
A071 0014	RTC_TC_SEC	16	**RTC seconds time counter register**
A071 0018	RTC_TC_MIN	16	**RTC minutes time counter register**
A071001C	RTC_TC_HOU	16	**RTC hours time counter register**
A0710020	RTC_TC_DOM	16	**RTC day-of-month time counter register**
A0710024	RTC_TC_DOW	16	**RTC day-of-week time counter register**
A0710028	RTC_TC_MTH	16	**RTC month time counter register**
A071002C	RTC_TC_YEA	16	**RTC year time counter register**
A0710030	RTC_AL_SEC	16	**RTC second alarm setting register**
A0710034	RTC_AL_MIN	16	**RTC minute alarm setting register**
A0710038	RTC_AL_HOU	16	**RTC hour alarm setting register**
A071003C	RTC_AL_DOM	16	**RTC day-of-month alarm setting register**
A0710040	RTC_AL_DOW	16	**RTC day-of-week alarm setting register**
A0710044	RTC_AL_MTH	16	**RTC month alarm setting register**
A0710048	RTC_AL_YEA	16	**RTC year alarm setting register**
A071004C	RTC_OSC32CON	16	**OSC32 control** The function is only active when RTC_POWERKEY1 & RTC_POWERKEY2 match the correct values.
A0710050	RTC_POWERKEY1	16	**RTC_POWERKEY1 register**
A0710054	RTC_POWERKEY2	16	**RTC_POWERKEY2 register**
A0710058	RTC_PDN1	16	**PDN1**
A071005C	RTC_PDN2	16	**PDN2**
A0710060	RTC_SPAR0	16	**Spare register for specific purpose**
A0710064	RTC_SPAR1	16	**Spare register for specific purpose**
A0710068	RTC_PROT	16	**Lock/unlock scheme to prevent RTC miswriting**
A071006C	RTC_DIFF	16	**One-time calibration offset** This register is fixed to 0 when RTC_POWERKEY1 & RTC_POWERKEY2 unmatch the correct values.
A0710070	RTC_CALI	16	**Repeat calibration offset** This register is fixed to 0 when RTC_POWERKEY1 & RTC_POWERKEY2 unmatch the correct values.
A0710074	RTC_WRTGR	16	**Enable transfers from core to RTC in queue**
A0710078	RTC_CON	16	**Other RTC control registers** Note: LPRST and LPEN are tied to 0 internally when RTC_POWERKEY1 & RTC_POWERKEY2 do not match the correct values. After changing RTC_CON, write WRTGR = 1 to make it take effect.

A0710000 RTC_BBPU Baseband Power Up 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne	KEY_BBPU									CBUSY	RELOAD	CLRPKY	AUTO	BBPU		PWREN
Type	WO									RO	WO	WO	RW	RW		RW
Reset	0	0	0	0	0	0	0	0		0	0	0	0	0		0

Bit (s)	Mnemonic	Name	Description
15:8	**KEY_BBPU**	KEY_BBPU	A bus write is acceptable only when KEY_BBPU=0x43.
6	**CBUSY**	CBUSY	The read/write channels between RTC/Core is busy. This bit indicates high after the software program sequence to anyone of RTC data registers and enables the transfer by RTC_WRTGR = 1. In addition, it is high after the reset from low to high due to RTC reload process.
5	**RELOAD**	RELOAD	**Reloads the values from RTC domain to core domain** Generally the RTC will reload and synchronize the data from RTC to core when being reset from 0 to 1. This bit can be treated as a debugging bit.
4	**CLRPKY**	CLRPKY	**Clears powerkey1 and powerkey2 at the same time** In some cases, the software may clear powerkey1 & powerkey2. BBWAKEUP depends on the matching specific patterns of powerkey1 and powerkey2. If any one of powerkey1 or powerkey2 or BBPU is cleared, BBWAKEUP will go low immediately. The software cannot program the other control bits without power. By programming RTC_BBPU with CLRPKY = 1 and BBPU = 0 condition, RTC can clear powerkey1, powerkey2 and BBPU at the same time.
3	**AUTO**	AUTO	**Controls if BBWAKEUP is automatically in the low state when SYSRST transitions from high to low.** 0: BBWAKEUP is not automatically in the low state when SYSRST# transitions are from high to low. 1: BBWAKEUP is automatically in the low state when SYSRST# transitions are from high to low. The function is only active when RTC_POWERKEY1 & RTC_POWERKEY2 match the correct values.
2	**BBPU**	BBPU	**Controls the power of PMU** If powerkey1 = A357h and powerkey2 = 67D2h, PMU takes on the value programmed by software; otherwise PMU is low. 0: Power down 1: Power on
0	**PWREN**	PWREN	0: RTC alarm has no action on power switch. 1: When an RTC alarm occurs, BBPU is set to 1 and the system is powered on by RTC alarm wakeup.

A0710004 RTC_IRQ_STA RTC IRQ Status 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne													LPSTA		TCSTA	ALSTA
Type													RC		RC	RC
Reset													0		0	0

Overview: This register is read-cleared and is fixed to 0 when RTC_POWERKEY1 and RTC_POWERKEY2 unmatch the correct values.

Bit (s)	Mnemonic	Name	Description
3	**LPSTA**	LPSTA	**Indicates the IRQ status and whether or not the LPD is asserted** (LPD function is either provided by XOSC32 or EOSC32[6], depending on XOSC32_ENB) 0: No IRQ occurred; the 32K clock is good. 1: IRQ occurred; the 32K clock stops. This can be masked by LP_EN or cleared by initializing LPD.
1	**TCSTA**	TCSTA	**Indicates the IRQ status and whether or not the tick condition has been met.** 0: No IRQ occurred; the tick condition has not been met. 1: IRQ occurred; the tick condition has been met.
0	**ALSTA**	ALSTA	**Indicates the IRQ status and whether or not the alarm condition has been met.** 0: No IRQ occurred; the alarm condition has not been met. 1: IRQ occurred; the alarm condition has been met.

A0710008 RTC_IRQ_EN RTC IRQ Enable 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne													**LP_EN**	**ONESHOT**	**TC_EN**	**AL_EN**
Type													RW	RW	RW	RW
Reset													0	0	0	0

Overview: This register is fixed to 0 when RTC_POWERKEY1 and RTC_POWERKEY2 unmatch the correct values.

Bit (s)	Mnemonic	Name	Description
3	**LP_EN**	LP_EN	**Enables the control bit for IRQ generation if low power detected (32k clock off).** 0: Disable IRQ generations 1: Enable LPD
2	**ONESHOT**	ONESHOT	**Controls automatic reset of AL_EN and TC_EN**
1	**TC_EN**	TC_EN	Enables the control bit for IRQ generation if the tick condition has been met. 0: Disable IRQ generations 1: Enable the tick time match interrupt. Clear the interrupt when ONESHOT is high upon generation of the corresponding IRQ.
0	**AL_EN**	AL_EN	**Enables the control bit for IRQ generation if the alarm condition has been met.** 0: Disable IRQ generations

Bit (s)	Mnemonic	Name	Description
			1: Enable the alarm time match interrupt. Clear the interrupt when ONESHOT is high upon generation of the corresponding IRQ.

A071000C RTC_CII_EN Counter Increment IRQ Enable 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne							1/8SECCII	1/4SECCII	1/2SECCII	YEACII	MTHCII	DOWCII	DOMCII	HOUCII	MINCII	SECCII
Type							RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Reset							0	0	0	0	0	0	0	0	0	0

Overview: This register activates or de-activates the IRQ generation when the TC counter reaches its maximum value.

Bit (s)	Mnemonic	Name	Description
9	**1/8SECCII**	SECCII_1_8	Set the bit to 1 to activate the IRQ at each 1/8 of a second update
8	**1/4SECCII**	SECCII_1_4	Set the bit to 1 to activate the IRQ at each 1/4 of a second update
7	**1/2SECCII**	SECCII_1_2	Set the bit to 1 to activate the IRQ at each 1/2 of a second update
6	**YEACII**	YEACII	Set the bit to 1 to activate the IRQ at each year update
5	**MTHCII**	MTHCII	Set the bit to 1 to activate the IRQ at each month update
4	**DOWCII**	DOWCII	Set the bit to 1 to activate the IRQ at each day-of-week update
3	**DOMCII**	DOMCII	Set the bit to 1 to activate the IRQ at each day-of-month update
2	**HOUCII**	HOUCII	Set the bit to 1 to activate the IRQ at each hour update
1	**MINCII**	MINCII	Set the bit to 1 to activate the IRQ at each minute update
0	**SECCII**	SECCII	Set this bit to 1 to activate the IRQ at each second update

A0710010 RTC_AL_MASK RTC Alarm Mask 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne										YEA_MSK	MTH_MSK	DOW_MSK	DOM_MSK	HOU_MSK	MIN_MSK	SEC_MSK
Type										RW	RW	RW	RW	RW	RW	RW
Reset										0	0	0	0	0	0	0

Overview: The alarm condition for alarm IRQ generation depends on whether or not the corresponding bit in this register is masked. Note that if all bits 1 in RTC_AL_MASK are set (i.e. RTC_AL_MASK = 0x7f) and PWREN = 1 in RTC_BBPU, it means the alarm will come every second, not disabled.

Bit (s)	Mnemonic	Name	Description
6	**YEA_MSK**	YEA_MSK	0: Condition (RTC_TC_YEA = RTC_AL_YEA) is checked to generate the alarm signal.

Bit (s)	Mnemonic	Name	Description
			1: Condition (RTC_TC_YEA = RTC_AL_YEA) is masked, i.e. the value of RTC_TC_YEA does not affect the alarm IRQ generation.
5	**MTH_MSK**	MTH_MSK	0: Condition (RTC_TC_MTH = RTC_AL_MTH) is checked to generate the alarm signal. 1: Condition (RTC_TC_MTH = RTC_AL_MTH) is masked, i.e. the value of RTC_TC_MTH does not affect the alarm IRQ generation.
4	**DOW_MSK**	DOW_MSK	0: Condition (RTC_TC_DOW = RTC_AL_DOW) is checked to generate the alarm signal. 1: Condition (RTC_TC_DOW = RTC_AL_DOW) is masked, i.e. the value of RTC_TC_DOW does not affect the alarm IRQ generation.
3	**DOM_MSK**	DOM_MSK	0: Condition (RTC_TC_DOM = RTC_AL_DOM) is checked to generate the alarm signal. 1: Condition (RTC_TC_DOM = RTC_AL_DOM) is masked, i.e. the value of RTC_TC_DOM does not affect the alarm IRQ generation.
2	**HOU_MSK**	HOU_MSK	0: Condition (RTC_TC_HOU = RTC_AL_HOU) is checked to generate the alarm signal. 1: Condition (RTC_TC_HOU = RTC_AL_HOU) is masked, i.e. the value of RTC_TC_HOU does not affect the alarm IRQ generation.
1	**MIN_MSK**	MIN_MSK	0: Condition (RTC_TC_MIN = RTC_AL_MIN) is checked to generate the alarm signal. 1: Condition (RTC_TC_MIN = RTC_AL_MIN) is masked, i.e. the value of RTC_TC_MIN does not affect the alarm IRQ generation.
0	**SEC_MSK**	SEC_MSK	0: Condition (RTC_TC_SEC = RTC_AL_SEC) is checked to generate the alarm signal. 1: Condition (RTC_TC_SEC = RTC_AL_SEC) is masked, i.e. the value of RTC_TC_SEC does not affect the alarm IRQ generation.

A0710014 RTC_TC_SEC RTC Seconds Time Counter Register 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne											**TC_SECOND**					
Type											RW					
Reset											0	0	0	0	0	0

Bit (s)	Mnemonic	Name	Description
5:0	**TC_SECOND**	TC_SECOND	**Second initial value for the time counter** Range: 0 ~ 59

A0710018 RTC_TC_MIN RTC Minutes Time Counter Register 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne											**TC_MINUTE**					
Type											RW					

Reset											0	0	0	0	0	0

Bit (s)	Mnemonic	Name	Description
5:0	**TC_MINUTE**	TC_MINUTE	**Minute initial value for the time counter** Range: 0 ~ 59

A071001C **RTC_TC_HOU** **RTC Hours Time Counter Register** **0000**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne												**TC_HOUR**				
Type												RW				
Reset												0	0	0	0	0

Bit (s)	Mnemonic	Name	Description
4:0	**TC_HOUR**	TC_HOUR	**Hour initial value for the time counter** Range: 0 ~ 23

A0710020 **RTC_TC_DOM** **RTC Day-of-month Time Counter Register** **0000**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne												**TC_DOM**				
Type												RW				
Reset												0	0	0	0	0

Bit (s)	Mnemonic	Name	Description
4:0	**TC_DOM**	TC_DOM	**Day-of-month initial value for the time counter** The day-of-month maximum value depends on the leap year condition, i.e. 2 LSB of year time counter are zeros.

A0710024 **RTC_TC_DOW** **RTC Day-of-week Time Counter Register** **0000**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne														**TC_DOW**		
Type														RW		
Reset														0	0	0

Bit (s)	Mnemonic	Name	Description
2:0	**TC_DOW**	TC_DOW	**Day-of-week initial value for the time counter** Range: 1 ~ 7

A0710028 **RTC_TC_MTH** **RTC Month Time Counter Register** **0000**

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

Mne													**TC_MONTH**			
Type													RW			
Reset													0	0	0	0

Bit (s)	Mnemonic	Name	Description
3:0	**TC_MONTH**	TC_MONTH	**Month initial value for the time counter** Range: 1 ~ 12

A071002C RTC_TC_YEA RTC Year Time Counter Register 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne										**TC_YEAR**						
Type										RW						
Reset										0	0	0	0	0	0	0

Bit (s)	Mnemonic	Name	Description
6:0	**TC_YEAR**	TC_YEAR	**Year initial value for the time counter** Range: 0 ~ 127 (2000-2127). The software can bias the year as multiples of 4 for the internal leap-year formula. Here are 3 examples: 2000 ~ 2127, 1972 ~ 2099 and 1904 ~ 2031. To simplify the process, the RTC hardware treats all 4-multiple as leap years. If the range you define includes non-leap 4-multiple year (e.g. 2100), please adjust it to the correct date by yourselves. (e.g. change Feb. 29th, 2100 to Mar. 1st, 2100). It is suggested to bias the range to be bigger than 1900 and smaller than 2100 to evade the manual adjustment, i.e. the bias values are suggested to be in the range of [-28,-96], that are (1972~ 2099) ~ (1904~ 2031). The formal leap formula: If year modulo 400 is 0 then leap Else if year modulo 100 is 0 then no_leap Else if year modulo 4 is 0 then leap Else no_leap

A0710030 RTC_AL_SEC RTC Second Alarm Setting Register 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne											**AL_SECOND**					
Type											RW					
Reset											0	0	0	0	0	0

Bit (s)	Mnemonic	Name	Description
5:0	**AL_SECOND**	AL_SECOND	**Second value of the alarm counter setting** Range: 0 ~ 59

A0710034 RTC_AL_MIN RTC Minute Alarm Setting Register 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne											AL_MINUTE					
Type											RW					
Reset											0	0	0	0	0	0

Bit (s)	Mnemonic	Name	Description
5:0	**AL_MINUTE**	AL_MINUTE	**Minute value of the alarm counter setting** Range: 0 ~ 59

A0710038 RTC_AL_HOU RTC Hour Alarm Setting Register 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne	NEW_SPARE0											AL_HOUR				
Type	RW											RW				
Reset	0	0	0	0	0	0	0	0				0	0	0	0	0

Bit (s)	Mnemonic	Name	Description
15:8	**NEW_SPARE0**	NEW_SPARE0	**New spare-register 0**
4:0	**AL_HOUR**	AL_HOUR	**Hour value of the alarm counter setting** Range: 0 ~ 23

A071003C RTC_AL_DOM RTC Day-of-month Alarm Setting Register 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne	NEW_SPARE1											AL_DOM				
Type	RW											RW				
Reset	0	0	0	0	0	0	0	0				0	0	0	0	0

Bit (s)	Mnemonic	Name	Description
15:8	**NEW_SPARE1**	NEW_SPARE1	**New spare-register 1**
4:0	**AL_DOM**	AL_DOM	**Day-of-month value of the alarm counter setting** The day-of-month maximum value depends on the leap year condition, i.e. 2 LSB of year time counter are zeros.

A0710040 RTC_AL_DOW RTC Day-of-week Alarm Setting Register 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne	NEW_SPARE2													AL_DOW		
Type	RW													RW		
Reset	0	0	0	0	0	0	0	0						0	0	0

Bit (s)	Mnemonic	Name	Description

Bit (s)	Mnemonic	Name	Description
15:8	**NEW_SPARE 2**	NEW_SPARE 2	**New spare-register 2**
2:0	**AL_DOW**	AL_DOW	**Day-of-week value of the alarm counter setting** Range: 1 ~ 7

A0710044 RTC_AL_MTH RTC Month Alarm Setting Register 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne	**NEW_SPARE3**												**AL_MONTH**			
Type	RW												RW			
Reset	0	0	0	0	0	0	0	0					0	0	0	0

Bit (s)	Mnemonic	Name	Description
15:8	**NEW_SPARE 3**	NEW_SPARE 3	New spare-register 3.
3:0	**AL_MONTH**	AL_MONTH	Month value of the alarm counter setting. Range: 1 ~ 12

A0710048 RTC_AL_YEA RTC Year Alarm Setting Register 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne	**NEW_SPARE4**									**AL_YEAR**						
Type	RW									RW						
Reset	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0

Bit (s)	Mnemonic	Name	Description
15:8	**NEW_SPARE 4**	NEW_SPARE 4	**New spare-register 4**
6:0	**AL_YEAR**	AL_YEAR	**Year value of the alarm counter setting** Range: 0 ~ 127 (2000-2127)

A071004C RTC_OSC32CON OSC32 Control 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne	**EOSC32_RSV**							**EMBCK_SEL_MODE**			**XOSC32_ENB**	**XOSCCALI**				
Type	RW							RW			RO	RW				
Reset	-	0	0	0	0	1	0	0	0	0	-	0	1	1	1	1

Overview: The function is only active when RTC_POWERKEY1 & RTC_POWERKEY2 match the correct values.

Bit (s)	Mnemonic	Name	Description
15:9	**OSC32_RSV**	OSC32_RSV	**OSC32 reserved bits** Keep it at 0x2.
8:6	**EMB_MODE**	EMBCK_SEL	**Mode setting for crystal removal case**

Bit (s)	Mnemonic	Name	Description
5	**XOSC32_ENB**	XOSC32_ENB	**XOSC32_ENB** Reads pin XOSC32_ENB configuration to know the 32k crystal usage. 0: Use 32k crystal 1: Does not use 32k crystal.
4:0	**XOSCCALI**	XOSCCALI	**Controls XOSC32/EOSC32 caliration** If XOSC32_ENB(RTC_OSC32CON[5]) = 0, XOSCCALI controls the bias current for 32k xtal in XOSC32. When powerkeys do not match, the default value is 0x7. If XOSC32_ENB(RTC_OSC32CON[5]) = 1, XOSCCALI is the trimming value for EOSC32. SW needs to find the best trimming value for EOSC32 when the 1st power-on by frequency meter. When powerkeys do not match, the default value is 0xf.

OSC32CON is not protected by RTC_PROT because RTC_PROT needs 32.768kHz clock to unlock. Similarly, to modify RTC_OSC32CON, writing RTC_WRTGR is not needed, neither. To protect the OSC32 control bits, follow the *update sequence* to update RTC_OSC32CON. After the updating sequence is completed, reload to acquire the internal RTC_OSC32CON. This register needs to be initialized **before** writing powerkeys match.

Update sequence:

Step 1: Write RTC_OSC32CON = 0x1a57 and wait until CBUSY=0
Step 2: Write RTC_OSC32CON = 0x2b68 and wait until CBUSY=0
Step 3: Write the real value you would like to write RTC_OSC32CON and wait until CBUSY=0
Step 4: Return to step 1 if you need to modify RTC_OSC32CON again.

Note: RTC_OSC32CON should be set before writing POWERKEY1 and POWERKEY2 to the correct value.

A0710050 RTC_POWERKEY1 RTC_POWERKEY1 Register 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne	**RTC_POWERKEY1**															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit (s)	Mnemonic	Name	Description
15:0	**RTC_POWERKEY1**	RTC_POWERKEY1	

A0710054 RTC_POWERKEY2 RTC_POWERKEY2 Register 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne	**RTC_POWERKEY2**															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit (s)	Mnemonic	Name	Description
15:0	**RTC_POWER KEY2**	RTC_POWER KEY2	

These register sets are used to determine if the real-time clock has been programmed by the software, i.e. the time value in real time clock is correct. When the real-time clock is first powered on, the register contents are all undefined, and therefore the time values shown are incorrect. The software needs to know if the real-time clock has been programmed. Hence, the two registers are defined to solve this power-on issue. After the software programs the correct value, the two register sets will not need to be updated. In addition to programming the correct time value, when the contents of the register sets are wrong, the interrupt will not be generated. Therefore, the-real time clock will not generate the interrupts before the software programs the registers, and unwanted interrupt due to wrong time value will not occur. The correct values of these two register sets are:

RTC_POWERKEY1	A357h
RTC_POWERKEY2	67D2h

A0710058 RTC_PDN1 PDN1 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne	**RTC_PDN1**															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit (s)	Mnemonic	Name	Description
15:0	**RTC_PDN1**	RTC_PDN1	Spare registers for software to keep the power-on and power-off state information

A071005C RTC_PDN2 PDN2 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne	**RTC_PDN2**															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit (s)	Mnemonic	Name	Description
15:0	**RTC_PDN2**	RTC_PDN2	Spare registers for software to keep power-on and power-off state information

A0710060 RTC_SPAR0 Spare Register For Specific Purpose 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne	**RTC_SPAR0**															
Type	RW															

Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit (s)	Mnemonic	Name	Description
15:0	**RTC_SPAR0**	RTC_SPAR0	Reserved for specific purposes

A0710064 RTC_SPAR1 Spare Register For Specific Purpose 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne	RTC_SPAR1															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit (s)	Mnemonic	Name	Description
15:0	**RTC_SPAR1**	RTC_SPAR1	Reserved for specific purposes

A0710068 RTC_PROT Lock/Unlock Scheme to Prevent RTC Miswriting 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne	RTC_PROT															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit (s)	Mnemonic	Name	Description

15:0	**RTC_PROT**	RTC_PROT	**Protects RTC write interface by RTC_PROT**
			Whether the RTC writing interface is enabled or not is decided by RTC_PROT contents. When RTC_POWERKEY1 & RTC_POWERKEY2 are not equal to the correct values, the RTC writing interface will always be enabled. However, when they match, the user has to perform the unlock flow to enable the writing interface.
			
			Unlock flow:
			Step1: *RTC_PROT=0x586a;
			Step2: *RTC_WRTGR=1;
			Step3: while(*RTC_BBPU & 0x40) {}; // Timeout period: 120usec
			Step4: *RTC_PROT=0x9136;
			Step5: *RTC_WRTGR=1;
			Step6: while(*RTC_BBPU & 0x40) {}; // Timeout period: 120usec
			
			Note: Always keep RTC in the unlock state in the power-on mode. Once the normal RTC content writing is completed, DO NOT modify the RTC_PROT content to lock RTC. The RTC_PROT contents will be cleared automatically when being powered off immediately.

A071006C RTC_DIFF One-time Calibration Offset 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne	**RESERVED**	**RESERVED**			**RTC_DIFF**											
Type	RW	RO			RW											
Reset	-	-			0	0	0	0	0	0	0	0	0	0	0	0

Overview: This register is fixed to 0 when RTC_POWERKEY1 and RTC_POWERKEY2 unmatch the correct values.

Bit (s)	Mnemonic	Name	Description
11:0	**RTC_DIFF**	RTC_DIFF	**Adjusts internal counter of RTC** It takes effect once and returns to 0 when done. In some cases, RTC is faster or slower than the standard. To change RTC_TC_SEC being coarse may cause alarm problem. RTC_DIFF provides a finer time unit. An internal 15-bit counter accumulates in each 32,768Hz clock. Entering a non-0 value to RTC_DIFF will cause the internal RTC counter to increase or decrease RTC_DIFF when RTC_DIFF changes to 0 again. RTC_DIFF represents 2's completement. For example, if you fill in 0xfff into RTC_DIFF, the internal counter will decrease by 1 when RTC_DIFF returns to 0. In other words, you can only use RTC_DIFF continuously if RTC_DIFF is equal to 0 now. *Note: RTC_DIFF ranges from 0x800 (-2048) to 0x7fd (2045). 0x7ff & 0x7fe are forbidden.*

A0710070 <u>RTC_CALI</u> Repeat Calibration Offset 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne	**RESERVED**	**RESERVED**	**RTC_CALI**													
Type	RO	RW	RW													
Reset	0	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: This register is fixed to 0 when RTC_POWERKEY1 & RTC_POWERKEY2 unmatch the correct values.

Bit (s)	Mnemonic	Name	Description
13:0	**RTC_CALI**	RTC_CALI	**Provides a repeated calibration scheme** RTC_CALI provides 7-bit calibration capability in 8-second duration, i.e. 5-bit calibration capability in each second. RTC_CALI represents 2's complement form for the user to adjust RTC increase or decrease Due to RTC_CALI is revealed in 8 seconds, the resolution is less than a 1/32768 clock. Avg. resolution: 1/32768/8 = 3.81us Avg. adjust range: -0.244~0.240ms/sec in 2's complement: -0x2000 ~ 0x1fff (-8192 ~ 8191)

A0710074 <u>RTC_WRTGR</u> Enable Transfers From Core to RTC in Queue 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne																WRTGR
Type																WO
Reset																0

Bit (s)	Mnemonic	Name	Description
0	**WRTGR**	WRTGR	**Enables the transfers from core to RTC** After you modify all the RTC registers and would like to change it, write 1 to RTC_WRTGR to trigger the transfer. The prior writing operation is queued in the core power domain. The pending data will not be transferred to the RTC domain until WRTGR = 1. After WRTGR=1, the pending data will be transferred to the RTC domain sequentially in order of register addresses, from low to high. For example: RTC_BBPU -> RTC_IRQ_EN -> RTC_CII_EN -> RTC_AL_MASK -> RTC_TC_SEC -> etc. CBUSY in RTC_BBPU is equal to 1 in the writing process. Observe CBUSY to determine when the transmission is completed.

A0710078 RTC_CON Other RTC Control Register 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne	LPSTA_RAW	RESERVED	RESERVED	POWEROF F_SEQ_EN	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	LPRST	LPEN		VBAT_LPSTA_RAW
Type	W1C	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW		RW
Reset																

Bit (s)	Mnemonic	Name	Description
15	**LPSTA_RAW**	LPSTA_RAW	**Raw status of LP_STA** Re-initialize LPD to clear this bit. *Note: This bit is always high before LPD initialization sequence after the first power-on.*
3	**LPRST**	LPRST	**Resets LPDETB** Only takes effect when LPEN = 1.
2:1	**LPEN**	LPEN	**Enables LPDETB** LP initialization sequence: 1. Write LPEN = 1, LPRST = 0. Write RTC_WRTGR = 1. wait cbusy down. 2. Write LPEN = 1, LPRST = 1. Write RTC_WRTGR = 1. wait cbusy down. 3. Write LPEN = 1, LPRST = 0. Write RTC_WRTGR = 1. wait cbusy down.
0	**VBAT_LPSTA_RAW**	VBAT_LPSTA_RAW	**Indicates the battery has been in LP state** Software needs to clear this bit for the next time use 0: VBAT has not been in LP state. 1: VBAT has been in LP state. *Note: VBAT LP state = VBAT < 2.5V*

When Vcore always exists, the software can trust the registers and wait for the LP interrupt from RTC if the 32.768 kHz clock stopped or has been stopped.

However, nothing can be trusted after the battery is off (Vrtc may drop). In every boot time, the software checks if LPSTA_RAW = 1. (LP_STA = LPSTA_RAW & LP_IRQ_EN) If true, the RTC contents will no longer be trusted just like powerkeys do not match. You have to initialize the RTC contents in this case.

3.13 Auxiliary ADC Unit

The auxiliary ADC unit is used to monitor the status of the battery and charger, to identify the plugged peripheral, and to perform temperature measurement.

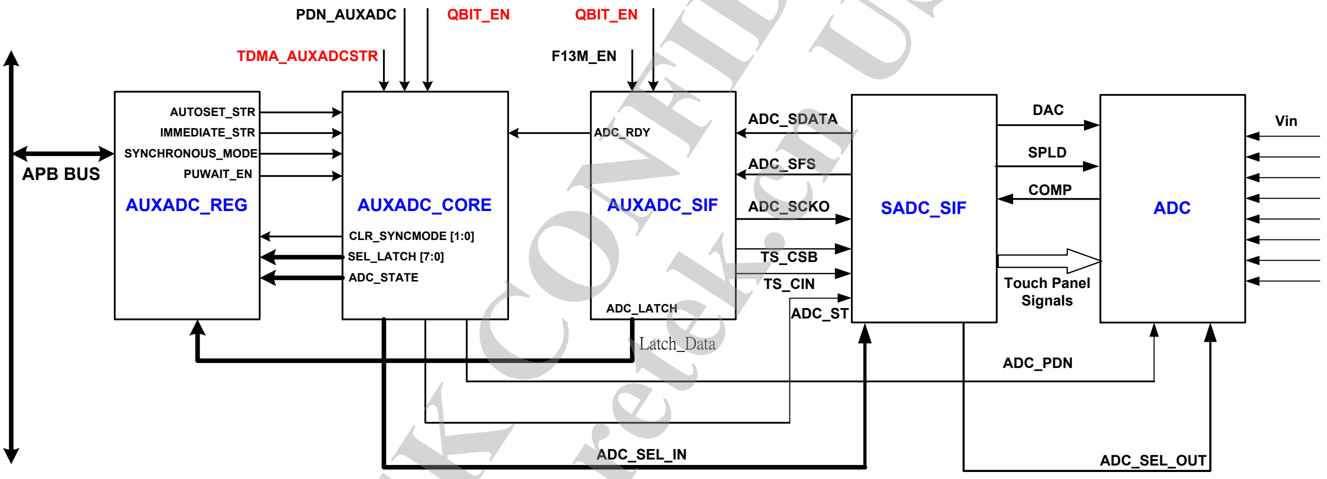


Figure 38. AUXADC architecture

Each channel operates in one of the two modes: immediate mode or timer-triggered mode. The mode of each channel can be individually selected through register AUXADC_CON0. For example, if the flag SYN0 in register AUXADC_CON0 is set, channel 0 will be set in the timer-triggered mode. Otherwise, the channel will operate in the immediate mode.

In the immediate mode, the A/D converter samples the value once only when the flag in the AUXADC_CON1 register is set. For example, if the flag IMM0 in AUXADC_CON1 is set, the A/D converter will sample the data for channel 0. The IMM flags must be cleared and set again to initialize another sampling.

The value sampled for channel 0 is stored in register AUXADC_DAT0, and the value for channel 1 is stored in register AUXADC_DAT1, and so on.

If the AUTOSET flag in register AUXADC_CON3 is set, the auto-sample function will be enabled. The A/D converter samples the data for the channel in which the corresponding data register is read. For example, in the case where the SYN1 flag is not set, the AUTOSET flag is set. When the data register AUXADC_DAT0 is read, the A/D converter will sample the next value for channel 1 immediately.

If multiple channels are selected at the same time, the task will be performed sequentially on every selected channel. For example, if AUXADC_CON1 is set to 0x3f, i.e. 6 channels are selected, the state machine in the unit will start sampling from channel 6 to channel 0 and save the values of each input channel in respective registers. The same process also applies to the timer-triggered mode.

In the timer-triggered mode, the A/D converter samples the value for the channels in which the corresponding SYN flags are set when the TDMA timer counts to the value specified in register TDMA_AUXEV1 placed in the TDMA timer. For example, if AUXADC_CON0 is set to 0x3f, the 6 channels will be selected to be in the timer-triggered mode. The state machine will sample the 6 channels sequentially and save the values in registers from AUXADC_DAT0 to AUXADC_DAT5, as it does in the immediate mode.

AUTOCLR*n* in register AUXADC_CON3 is set when it is intended to sample only once after setting up the timer-triggered mode. If the AUTOCLR1 flag is set, after the data for the channels in the timer-triggered mode are stored, the SYN*n* flags in register AUXADC_CON0 will be cleared.

The uses of the immediate mode and timer-triggered mode are mutually exclusive in terms of individual channels.

There are only two external pins (channel 4 ~ 5) for voltage detection. Other channels (0 ~ 3) are for battery voltage, battery current, charger and battery temperature respectively. Channel 9 is used for audio.

Touch panel

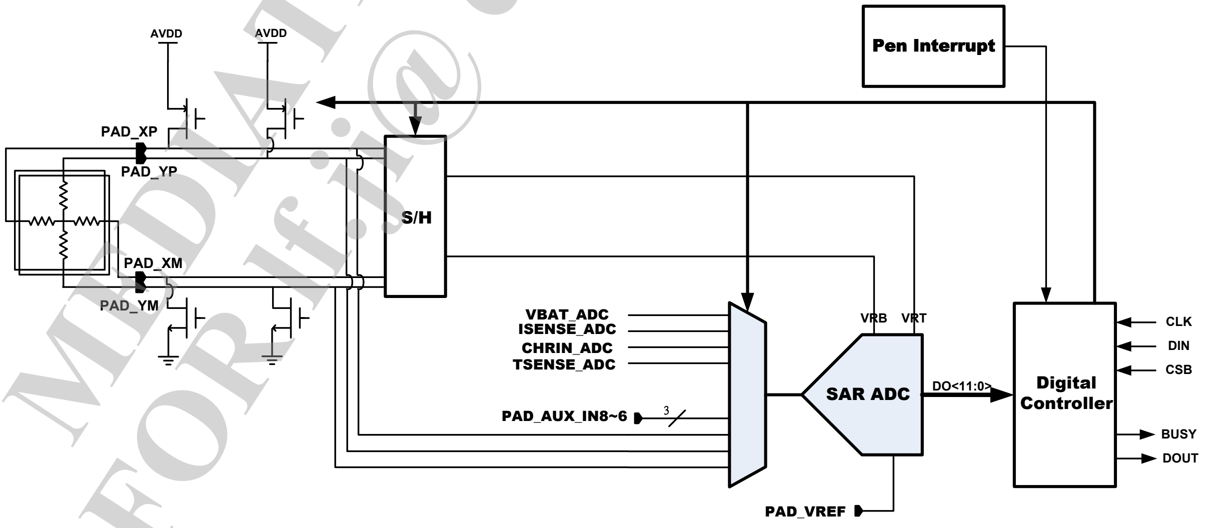


Figure 39. Touch panel circuit structure

Besides the normal sampling of external input voltage, the AUXADC includes the sampling of the touch panel function. For specified axis, the software should program AUX_TS_CMD first then trigger the sample of touch panel in register AUX_TS_CON. The touch panel sampling waveform is shown as the following. After the software polls the status bit in register AUXADC_CON3 to know that the touch panel sample is finished, the software can read back the specified axis value from register AUX_TS_DAT0.

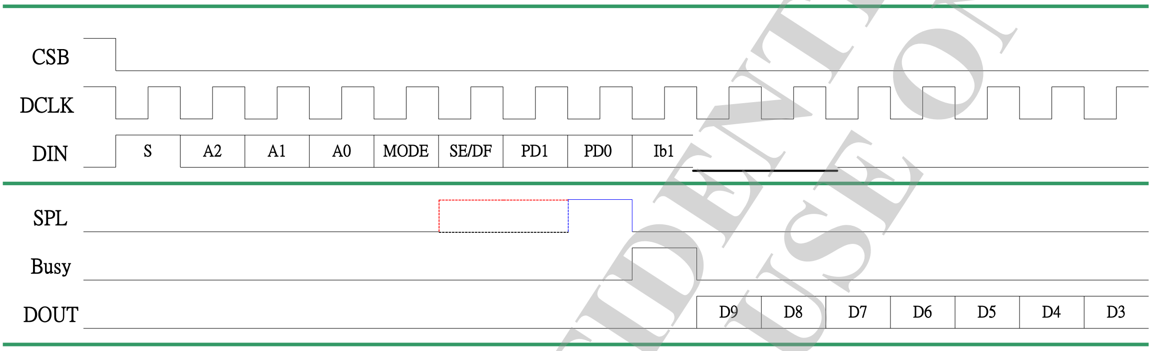


Figure 40. Touch panel sampling waveform

S: Start bit
A2 ~ A0: Addressing bits
Mode: 10-bit or 8-bit
SE/DF: Single end or differential mode
PD1 ~ 0: Power down command

These values are defined in register AUX_TS_CMD. The table below shows the relationship between AUX_TS_CMD and touch panel control signals.

Table 49. Relationship between commands and touch panel control signals

A2	A1	A0	SE/DFB	*CHN_SEL*	ADC In	X switches	Y switches	+REF	-REF
0	0	1	0	C	X+	OFF	ON	Y+	Y-
0	1	0	0	F	X-	OFF	ON	Y+	Y-
0	1	1	0	C	X+	X+ OFF X- ON	Y+ ON Y- OFF	Y+	X-
1	0	0	0	E	Y-	X+ OFF X- ON	Y+ ON Y- OFF	Y+	X-
1	0	1	0	D	Y+	ON	OFF	X+	X-
1	1	0	0	E	Y-	ON	OFF	X+	X-

Table 2. AUXADC channel description

AuxADC Channel ID	Description
Channel 0	VBAT

AuxADC Channel ID	Description
Channel 1	ISENSE
Channel 2	CHRIN
Channel 3	BATON (BATtemp)
Channel 4	AUXIN4 (external)
Channel 5	ACCDET (external)
Channel 9	ClassAB
Channel 12	XP / External
Channel 13	YP / External
Channel 14	YM / External
Channel 15	XM / External

3.13.1 Register Definition

Module name: AUXADC Base address: (+A0790000h)

Address	Name	Width	Register Function
A0790000	AUXADC_CON0	16	**AuxiliaryADC Control Register 0** These bits define whether the corresponding channel is sampled or not in the timer-triggered mode. It is associated with the timing offset register TDMA_AUXEV1. It supports multiple flags. The flags can be automatically cleared after those channels are sampled if AUTOCLR1 in register AUXADC_CON3 is set.
A0790004	AUXADC_CON1	16	**AuxiliaryADC Control Register 1** These bits are set individually to sample the data for the corresponding channel. It supports multiple flags.
A079000C	AUXADC_CON3	16	**AuxiliaryADC Control Register 3**
A0790010	AUXADC_DAT0	16	**AuxiliaryADC Channel 0 Register (VBAT)**
A0790014	AUXADC_DAT1	16	**AuxiliaryADC Channel 1 Register (ISENSE)**
A0790018	AUXADC_DAT2	16	**AuxiliaryADC Channel 2 Register (CHRIN)**
A079001C	AUXADC_DAT3	16	**AuxiliaryADC Channel 3 Register (VBATTMP)**
A0790020	AUXADC_DAT4	16	**AuxiliaryADC Channel 4 Register (External)**
A0790024	AUXADC_DAT5	16	**AuxiliaryADC Channel 5 Register (External/ACCDET)**
A0790034	AUXADC_DAT9	16	**AuxiliaryADC Channel 9 Register (ClassAB)**
A0790040	AUXADC_DAT12	16	**AuxiliaryADC Channel 12 Register (External)**
A0790044	AUXADC_DAT13	16	**AuxiliaryADC Channel 13 Register (External)**
A0790048	AUXADC_DAT14	16	**AuxiliaryADC Channel 14 Register (External)**
A079004C	AUXADC_DAT15	16	**AuxiliaryADC Channel 15 Register (External)**
A0790054	AUX_TS_CMD0	16	**Touch Screen Sample Command 0**
A0790058	AUX_TS_CON	16	**Touch Screen Control**
A079005C	AUX_TS_DAT0	16	**Touch Screen Sample DATA 0**
A0790070	AUXADC_DAT_ZCV	16	**AuxiliaryADC ZCV Sample DATA**
A07900D0	AUXADC_CON4	16	**AuxiliaryADC Control Register 4**

Address	Name	Width	Register Function
A07900D4	AUX_TS_CMD1	**16**	**Touch Screen Sample Command 1**
A07900D8	AUX_TS_DAT1	16	**Touch Screen Sample DATA 1**

A0790000 AUXADC_CON0 AuxiliaryADC Control Register 0 — 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**SYN15**	**SYN14**	**SYN13**	**SYN12**			**SYN9**				**SYN5**	**SYN4**	**SYN3**	**SYN2**	**SYN1**	**SYN0**
Type	R/W	R/W	R/W	R/W			R/W				R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0			0				0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
15	**SYN15**	SYN15	**Channel 15 sync mode** 0: The channel is not selected. 1: The channel is selected.
14	**SYN14**	SYN14	**Channel 14 sync mode** 0: The channel is not selected. 1: The channel is selected.
13	**SYN13**	SYN13	**Channel 13 sync mode** 0: The channel is not selected. 1: The channel is selected.
12	**SYN12**	SYN12	**Channel 12 sync mode** 0: The channel is not selected. 1: The channel is selected.
9	**SYN9**	SYN9	**Channel 9 sync mode** 0: The channel is not selected. 1: The channel is selected.
5	**SYN5**	SYN5	**Channel 5 sync mode** 0: The channel is not selected. 1: The channel is selected.
4	**SYN4**	SYN4	**Channel 4 sync mode** 0: The channel is not selected. 1: The channel is selected.
3	**SYN3**	SYN3	**Channel 3 sync mode** 0: The channel is not selected. 1: The channel is selected.
2	**SYN2**	SYN2	**Channel 2 sync mode** 0: The channel is not selected. 1: The channel is selected.
1	**SYN1**	SYN1	**Channel 1 sync mode** 0: The channel is not selected. 1: The channel is selected.
0	**SYN0**	SYN0	**Channel 0 sync mode** 0: The channel is not selected. 1: The channel is selected.

A0790004 AUXADC_CON1 AuxiliaryADC Control Register 1 — 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**IMM15**	**IMM14**	**IMM13**	**IMM12**			**IMM9**				**IMM5**	**IMM4**	**IMM3**	**IMM2**	**IMM1**	**IMM0**
Type	R/W	R/W	R/W	R/W			R/W				R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0			0				0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description

15	**IMM15**	IMM15	**Channel 15 immediate mode** 0: The channel is not selected. 1: The channel is selected.
14	**IMM14**	IMM14	**Channel 14 immediate mode** 0: The channel is not selected. 1: The channel is selected.
13	**IMM13**	IMM13	**Channel 13 immediate mode** 0: The channel is not selected. 1: The channel is selected.
12	**IMM12**	IMM12	**Channel 12 immediate mode** 0: The channel is not selected. 1: The channel is selected.
9	**IMM9**	IMM9	**Channel 9 immediate mode** 0: The channel is not selected. 1: The channel is selected.
5	**IMM5**	IMM5	**Channel 5 immediate mode** 0: The channel is not selected. 1: The channel is selected.
4	**IMM4**	IMM4	**Channel 4 immediate mode** 0: The channel is not selected. 1: The channel is selected.
3	**IMM3**	IMM3	**Channel 3 immediate mode** 0: The channel is not selected. 1: The channel is selected.
2	**IMM2**	IMM2	**Channel 2 immediate mode** 0: The channel is not selected. 1: The channel is selected.
1	**IMM1**	IMM1	**Channel 1 immediate mode** 0: The channel is not selected. 1: The channel is selected.
0	**IMM0**	IMM0	**Channel 0 immediate mode** 0: The channel is not selected. 1: The channel is selected.

A079000C **AUXADC_CON3** **AuxiliaryADC Control Register 3** **0010**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**AUTOSET**				**RSV**		**AUTOCLR1**		**SOFT_RST**			**BYPASS_SLP_ZCV_TRIGGER**				**AUXADC_STA**
Type	R/W				R/W		R/W		R/W			R/W				RO
Reset	0				0		0		0			1				0

Bit(s)	Mnemonic	Name	Description
15	**AUTOSET**	AUTOSET	**Defines the auto-sample mode of the module. In auto-sample mode, each channel with its sample register being read can start sampling immediately without configuring the control register AUXADC_CON1 again.**
11	**RSV**	RSV	**Please keep 1'b0**
9	**AUTOCLR1**	AUTOCLR1	**Defines the auto-clear mode of the module for event 1. In the auto-clear mode, each timer-triggered channel acquires samples of specified channels once the SYNn bit in register AUXADC_CON0 is set. The SYNn bits are automatically cleared and the channel is not enabled again by the timer event except when the SYNn flags are set again.** 0: The automatic clear mode is not enabled. 1: The automatic clear mode is enabled.

7	**SOFT_RST**	SOFT_RST	**Software reset AUXADC state machine** 0: Normal function 1: Reset AUXADC state machine
4	**BYPASS_SLP_ZCV_TRIGGER**	BYPASS_SLP_ZCV_TRIGGER	**Bypass zcv triggering after sleep mode setting** 0: trigger zcv measuring after sleep mode 1: bypass zcv measuring trigger after sleep mode
0	**AUXADC_STA**	AUXADC_STA	**Defines the state of the module** 0: This module is idle. 1: This module is busy.

A0790010 AUXADC_DAT0 AuxiliaryADC Channel 0 Register (VBAT) 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							DAT0									
Type							RO									
Reset							0	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
9:0	**DAT0**	DAT0	**Sampled data for channel0**

A0790014 AUXADC_DAT1 AuxiliaryADC Channel 1 Register (ISENSE) 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							DAT1									
Type							RO									
Reset							0	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
9:0	**DAT1**	DAT1	**Sampled data for channel1**

A0790018 AUXADC_DAT2 AuxiliaryADC Channel 2 Register (CHRIN) 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							DAT2									
Type							RO									
Reset							0	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
9:0	**DAT2**	DAT2	**Sampled data for channel2**

A079001C AUXADC_DAT3 AuxiliaryADC Channel 3 Register (VBATTMP) 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							DAT3									
Type							RO									
Reset							0	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
9:0	**DAT3**	DAT3	**Sampled data for channel3**

A0790020 AUXADC_DAT4 AuxiliaryADC Channel 4 Register (External) 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							**DAT4**									
Type							RO									
Reset							0	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
9:0	**DAT4**	DAT4	**Sampled data for channel4**

A0790024 AUXADC_DAT5 AuxiliaryADC Channel 5 Register (External/ACCDET) 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							**DAT5**									
Type							RO									
Reset							0	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
9:0	**DAT5**	DAT5	**Sampled data for channel5**

A0790034 AUXADC_DAT9 AuxiliaryADC Channel 9 Register (ClassAB) 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							**DAT9**									
Type							RO									
Reset							0	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
9:0	**DAT9**	DAT9	**Sampled data for channel9**

A0790040 AUXADC_DAT12 AuxiliaryADC Channel 12 Register (External) 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							**DAT12**									
Type							RO									
Reset							0	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
9:0	**DAT12**	DAT12	**Sampled data for channel12**

A0790044 AUXADC_DAT13 AuxiliaryADC Channel 13 Register (External) 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							**DAT13**									
Type							RO									
Reset							0	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
9:0	**DAT13**	DAT13	**Sampled data for channel13**

A0790048 AUXADC_DAT14 AuxiliaryADC Channel 14 Register (External) 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							DAT14									
Type							RO									
Reset							0	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
9:0	**DAT14**	DAT14	**Sampled data for channel14**

A079004C AUXADC_DAT15 AuxiliaryADC Channel 15 Register (External) 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							DAT15									
Type							RO									
Reset							0	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
9:0	**DAT15**	DAT15	**Sampled data for channel15**

A0790054 AUX_TS_CMD0 Touch Screen Sample Command 0 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	TS_SPL	TS_MAGIC_KEY								ADDRESS			MODE	SEDF	PD	
Type	R/W	WO								R/W			R/W	R/W	R/W	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
15	**TS_SPL**	TS_SPL	**Touch screen sample trigger** For the touch screen status, please refer to PMU datasheet : "AUX_CON6" register bit 14 "ts_status" 0: No action 1: When software writes 1'b1, AUXADC will trigger the touch screen process. After the sampling process of touch screen is finished, this bit will be disserted.
14:7	**TS_MAGIC_KEY**	TS_MAGIC_KEY	**The TS commands in AUX_TS_CMD0 can only take effect when the TS_MAGIC_KEY matches the correct value.** TS_MAGIC_KEY =0xaa .
6:4	**ADDRESS**	ADDRESS	**Defines which x or y or z data will be sampled** 001: Y position 011: Z1 position 100: Z2 position 101: X position Others: Reserved
3	**MODE**	MODE	**Selects sample resolution** 0: 10-bit 1: 8-bit
2	**SEDF**	SEDF	**Selects mode** 0: Differential mode 1: Single-end mode
1:0	**PD**	PD	**Power-down control for analog IRQ signal and touch screen sample control signal**

00: Turn on Y-_drive signal and PDN_sh_ref
01: Turn on PDN_IRQ and PDN_sh_ref
10: Reserved
11: Turn on PDN_IRQ

A0790058 AUX_TS_CON Touch Screen Control 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																TS_SPL
Type																R/W
Reset																0

Bit(s)	Mnemonic	Name	Description
0	**TS_SPL**	TS_SPL	**Touch screen sample trigger** For the touch screen status, please refer to PMU datasheet : "AUX_CON6" register bit 14 "ts_status" 0: No action 1: When software writes 1'b1, AUXADC will trigger the touch screen process. After the sampling process of touch screen is finished, this bit will be disserted.

A079005C AUX_TS_DAT0 Touch Screen Sample DATA 0 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	TS_SPL						TS_DAT									
Type	RO						RO									
Reset	0						0	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
15	**TS_SPL**	TS_SPL	**Touch screen sample trigger** For the touch screen status, please refer to PMU datasheet : "AUX_CON6" register bit 14 "ts_status"
9:0	**TS_DAT**	TS_DAT	**Touch screen sample result data**

A0790070 AUXADC_DAT_ZCV AuxiliaryADC ZCV Sample DATA 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							DAT_ZCV									
Type							RO									
Reset							0	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
9:0	**DAT_ZCV**	DAT_ZCV	**Sampled data for ZCV**

A07900D0 AUXADC_CON4 AuxiliaryADC Control Register 4 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	TS_SPL	AUXADC_STA					SYN9				SYN5	SYN4	SYN3	SYN2	SYN1	SYN0
Type	RO	RO					RO				RO	RO	RO	RO	RO	RO
Reset	0	0					0				0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
15	**TS_SPL**	TS_SPL	**Touch screen sample trigger** For the touch screen status, please refer to PMU datasheet : "AUX_CON6" register bit 14 "ts_status" 0: No action 1: When software writes 1'b1, AUXADC will trigger the touch screen process. After the sampling process of touch screen is finished, this bit will be disserted.
14	**AUXADC_STA**	AUXADC_STA	**Defines the state of the module** 0: This module is idle. 1: This module is busy.
9	**SYN9**	SYN9	**Channel 9 sync mode** 0: The channel is not selected. 1: The channel is selected.
5	**SYN5**	SYN5	**Channel 5 sync mode** 0: The channel is not selected. 1: The channel is selected.
4	**SYN4**	SYN4	**Channel 4 sync mode** 0: The channel is not selected. 1: The channel is selected.
3	**SYN3**	SYN3	**Channel 3 sync mode** 0: The channel is not selected. 1: The channel is selected.
2	**SYN2**	SYN2	**Channel 2 sync mode** 0: The channel is not selected. 1: The channel is selected.
1	**SYN1**	SYN1	**Channel 1 sync mode** 0: The channel is not selected. 1: The channel is selected.
0	**SYN0**	SYN0	**Channel 0 sync mode** 0: The channel is not selected. 1: The channel is selected.

A07900D4 AUX_TS_CMD1 Touch Screen Sample Command 1 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name											ADDRESS		MODE	SEDF	PD	
Type											R/W		R/W	R/W	R/W	
Reset										0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
6:4	**ADDRESS**	ADDRESS	**Defines which x or y or z data will be sampled** 001: Y position 011: Z1 position 100: Z2 position 101: X position Others: Reserved
3	**MODE**	MODE	**Selects sample resolution** 0: 10-bit 1: 8-bit
2	**SEDF**	SEDF	**Selects mode** 0: Differential mode 1: Single-end mode
1:0	**PD**	PD	**Power-down control for analog IRQ signal and touch screen sample control signal**

00: Turn on Y-_drive signal and PDN_sh_ref
01: Turn on PDN_IRQ and PDN_sh_ref
10: Reserved
11: Turn on PDN_IRQ

A07900D8 AUX_TS_DAT1 Touch Screen Sample DATA 1 0000

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							**TS_DAT**									
Type							RO									
Reset							0	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
9:0	**TS_DAT**	TS_DAT	**Touch screen sample result data**

※ Please refer to **“*PMU*”** datasheet for other AuxADC related settings:

- Sampling cycle control: Please see “AUX_CON4” and “AUX_CON5”

3.13.2 General Programming Guide

All register writes will occur in sequence. However, due to synchronization time, any reads after writes need to be delayed by three dummy reads.

3.13.3 Usage Programming Guide

There are two modes to program the AUXADC to sample: immediate mode and synchronous mode. The following are notes on programming each mode:

Immediate mode sampling is accomplished by programming AUXADC_CON1 with the channels to be sampled. After programming, it is necessary to perform three dummy reads on AUXADC_CON3. After the dummy reads, the next read of AUXADC_CON3 will be valid. After sampling is done, it is necessary to program AUXADC_CON1 back to zero before sampling again.

Synchronous mode sampling is accomplished by programming AUXADC_CON0 with the channels to be sampled. Then it is necessary to program TDMA_EVTENA7 to 0x2. After the sample is done, TDMA_EVTENA7 must be programmed to 0x0 with waiting of two frames before sampling again.

3.13.4 Performance Programming Guide

For details on adjusting the performance of ADC sampling, please refer to registers AUX_CON4, AUX_CON5 and AUX_CON6 in the ***PMU*** datasheet.

3.13.5 AUXADC PDN

AUXADC is located in A-die. Due to limitation, one of UART1,2,3 clocks must be turned on in order to use AUXADC. Please clear the corresponding PDN bits to enable the AUXADC clock (bit 2 or 3 of ACFG_CLK_CG).

3.13.6 Notice

The 4 TP pins – PAD_XP, PAD_XM, PAD_YP, PAD_YM, are used as EINT pins when no R-touch is used in the system. Therefore, please make sure the pin settings are correct.

3.14 USB Device Controller

3.14.1 General Description

This chip provides a USB function interface which complies with Universal Serial Bus Specification Rev 1.1. The USB device controller supports only full-speed (12Mbps) operation. The chip can make use of this widely available USB interface to transmit/receive data with USB hosts, typically PC/laptop. There are 6 endpoints in the USB device controller besides the mandatory control endpoint, 4 of which are for IN transactions and 2 endpoints are for OUT transactions. Word, half-word, and byte access are all allowed for loading and unloading the FIFO. The controller features 4 DMA channels to accelerate the data transfer for ACL and SCO data streams. The features of the endpoints are:

1. Endpoint 0: The control endpoint feature 64 bytes FIFO and accommodates maximum packet size of up to 64 bytes. DMA transfer is not supported.

2. IN endpoint 1: It features 64 bytes FIFO and accommodates maximum packet size of up to 64 bytes. GDMA Channel 4 Write Transfer is supported.

3. IN endpoint 2: It features 64 bytes FIFO and accommodates maximum packet size of up to 64 bytes. GDMA Channel 6 Write Transfer is supported.

4. IN endpoint 3: It features 16-byte FIFO and accommodates maximum packet size of 16 bytes. GDMA Transfer is not supported.

5. IN endpoint 4: It features 16-byte FIFO and accommodates maximum packet size of 16 bytes. GDMA Transfer is not supported.

6. OUT endpoint 1: It features 64 bytes FIFO and accommodates maximum packet size of 64 bytes. GDMA Channel 5 Read Transfer is supported.

7. OUT endpoint 2: It features 64 bytes FIFO and accommodates maximum packet size of 64 bytes. GDMA Channel 7 Read Transfer is supported.

For each endpoint except for the control endpoint, when the packet size is smaller than half the size of the FIFO, at most 2 packets can be buffered.

This unit is highly software configurable. All endpoints except for the control endpoint can be configured to be a bulk, interrupt or isochronous endpoints. Composite devices are also supported. IN endpoint 1 and OUT endpoint 1 share the same endpoint number but they can be used separately. So is the situation for the endpoint 2.

The USB device uses the cable-powered feature for the transceiver but only drains little current. An internal pull-up resistor is integrated across Vbus and D+ signal. The switch on/off of the pull-up resistor can be configured through the internal register. Two additional external serial resistors might be needed to place on the output of D+ and D- signals to make the output impedance equivalent to 28~44Ohm.

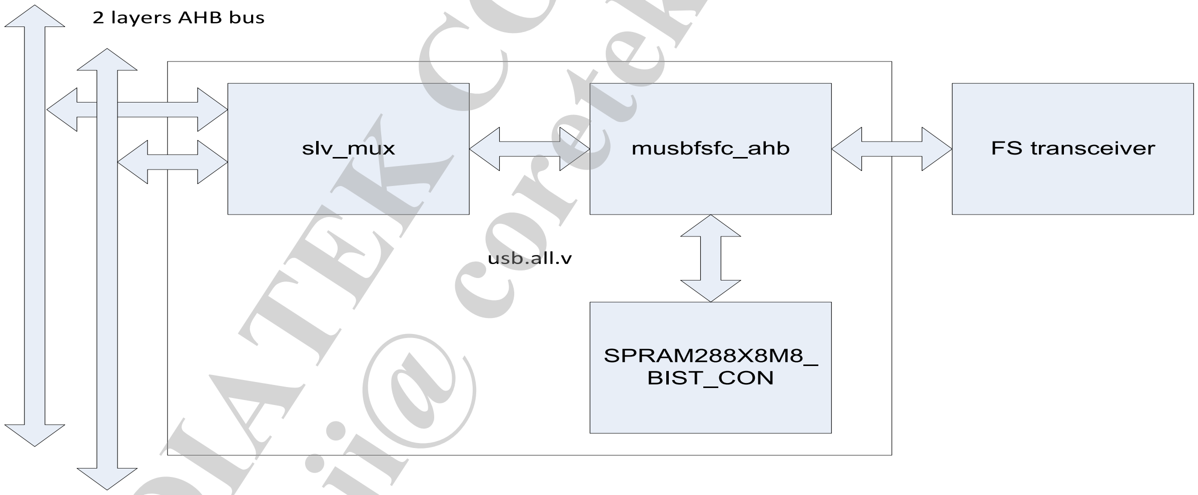


Figure 1. USB11 controller system diagram

3.14.2 Terminology

RW: Writable, Readable.
RO: Read-only. Value never changes.
WO: Write-only.
W1: Write-once. Readable.
RU: Read-only but value updated by the design.
W1C: Readable. Write 1 to bitwise-clear.

RC: Clear on read.
A1: Auto-set by the design. Can be read and write 0 to clear.
A0: Auto-cleared by the design. Can be read and write 1 to set.
DC: Don't care.
OTHER: Others. Mixed attribute. Refer to bit description.
RSV: Reserved. The read/write behavior to this bit is undefined.

3.14.3 Register Definition

00 USB_FADDR **USB Function Address Register** **00**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									**UPD**	**FADDR**						
Type									RW	RW						
Reset									0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7	UPD	**This is an 8bit register that should be written with the functions 7-bit address (received through a SET_ADDRESS description). It is then used for decoding the function address in subsequent token packets.** When set by the MCU, the core will wait for an SOF token from the time INPKTRDY is set before sending the packet
6:0	FADDR	**Function address of device**

01 USB_POWER **USB Power Control Register** **00**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									**ISOUPDATE**			**SWRSTENAB**	**RESET**	**RESUME**	**SUSPMODE**	**SUSPENAB**
Type									RW			RU	RU	RW	RU	RW
Reset									0			0	0	0	0	0

Bit(s)	Name	Description
7	ISOUPDATE	When set by the MCU, the core will wait for an SOF token from the time INPKTRDY is set before sending the packet
4	SWRSTENAB	**Set by the MCU to enable the mode in which the device can only be reset by the software after reset signals are detected on the bus.** In case the software is delayed by other high priority processes and cannot make it to read the command from the buffer before the hardware reset the device after the reset signal is detected on the bus, the command will be lost. That is why the software reset mode is effective. When the flag is enabled, the hardware state machine cannot reset itself but by the software. In that sense, the software and hardware can keep synchronous detecting the reset signal.
3	RESET	**The read-only bit is set when Reset signaling is present on the bus**
2	RESUME	**Set by the MCU to generate Resume signaling when the function is in the suspend mode.** The MCU should clear this bit after 10ms (maximum 15ms) to end Resume signaling
1	SUSPMODE	**Set by the USB core when the Suspend mode is entered** Cleared when the Resume bit of this register is set.
0	SUSPENAB	**Set by the MCU to enable device into the Suspend mode when Suspend**

Bit(s)	Name	Description
		signaling is received on the bus

0002 [USB_INTRIN]() **USB IN Endpoints Interrupt Register** **0000**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name												**EP4_IN**	**EP3_IN**	**EP2_IN**	**EP1_IN**	**EP0**
Type												RC	RC	RC	RC	RC
												0	0	0	0	0

Bit(s)	Name	Description
4	EP4_IN	IN Endpoint 4 interrupt event
3	EP3_IN	IN Endpoint 3 interrupt event
2	EP2_IN	IN Endpoint 2 interrupt event
1	EP1_IN	IN Endpoint 1 interrupt event
0	EP0	Endpoint 0 interrupt event

0004 [USB_INTROUT]() **USB OUT Endpoints InterruptRegister** **0000**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														**EP2_OUT**	**EP1_OUT**	
Type														RC	RC	
Reset														0	0	

Bit(s)	Name	Description
2	EP2_OUT	OUT Endpoint 2 interrupt event
1	EP1_OUT	OUT Endpoint 1 interrupt event

06 [USB_INTRUSB]() **USB General Interrupt Register** **00**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name												**POWERDWN**	**SOF**	**RESET**	**RESUME**	**SUSPEND**
Type												RC	RC	RC	RC	RC
Reset												0	0	0	0	0

Bit(s)	Name	Description
4	POWERDWN	**Set at SUSPMODE and LineState is JState.** The programmer should have de bounce scheme in SW code when using this interrupt.
3	SOF	**Set at the start of each frame**
2	RESET	**Set when Reset signaling is detected on the bus**
1	RESUME	**Set when Resume signaling is detected on the bus while the USB core is**

Bit(s)	Name	Description
		in suspend mode
0	SUSPEND	Set when Suspend signaling is detected on the bus

0007 USB_INTRINE USB IN Endpoints Interrupt Enable Register 00FF

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name												EP4_INE	EP3_INE	EP2_INE	EP1_INE	EP0_E
Type												RW	RW	RW	RW	RW
Reset												1	1	1	1	1

Bit(s)	Name	Description
4	EP4_INE	1b0: Disable IN Endpoint 4 interrupt event 1b1: Enable IN Endpoint 4 interrupt event
3	EP3_INE	1b0: Disable IN Endpoint 3 interrupt event 1b1: Enable IN Endpoint 3 interrupt event
2	EP2_INE	1b0: Disable IN Endpoint 2 interrupt event 1b1: Enable IN Endpoint 2 interrupt event
1	EP1_INE	1b0: Disable IN Endpoint 1 interrupt event 1b1: Enable IN Endpoint 1 interrupt event
0	EP0_E	1b0: Disable IN Endpoint 0 interrupt event 1b1: Enable IN Endpoint 0 interrupt event

0009 USB_INTROUTE USB OUT Endpoints Interrupt Enable Register 00FE

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														EP2_OUTE	EP1_OUTE	
Type														RW	RW	
Reset														1	1	

Bit(s)	Name	Description
2	EP2_OUTE	1b0: Disable OUT Endpoint 2 interrupt event 1b1: Enable OUT Endpoint 2 interrupt event
1	EP1_OUTE	1b0: Disable OUT Endpoint 1 interrupt event 1b1: Enable OUT Endpoint 1 interrupt event

0B INTRUSBE USB General Interrupt Enable Register 06

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name												POWERDWN_E	SOF_E	RESET_E	RESUM_E	SUSPEND_E
Type												RW	RW	RW	RW	RW
Reset												0	0	1	1	0

Bit(s)	Name	Description
4	POWERDWN_E	Enables power-down interrupt

Bit(s)	Name	Description
3	SOF_E	Enables SOF interrupt
2	RESET_E	Enables reset/babble interrupt
1	RESEUM_E	Enables resume interrupt
0	SUSPEND_E	Enables suspend interrupt

0C **USB_FRAME1** **USB Frame Count #1 Register** **00**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									NUML							
Type									RU							
Reset									0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7:0	NUML	The lower 8 bits of the frame number

0D **USB_FRAME2** **USB Frame Count #2 Register** **00**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														NUMH		
Type														RU		
Reset														0	0	0

Bit(s)	Name	Description
2:0	NUMH	The upper 3 bits of the frame number

0E **USB_INDEX** **USB Endpoint Register Index** **00**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													INDEX			
Type													RW			
Reset													0	0	0	0

The register determines which endpoint control/status registers are to be accessed at addresses USB+10h to USB+17h. Each IN endpoint and OUT endpoint has its own set of control/status registers. Only one set of IN control/status and one set of OUT control/status registers appear in the memory map at a time. Before accessing the control/status registers of an endpoint, the endpoint number should be written to the USB_INDEX register to ensure that the correct control/status registers appear in the memory map.

Bit(s)	Name	Description
3:0	INDEX	Index of the endpoint

0F **USB_RSTCTRL** **USB Reset Control** **00**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									SWRST				RSTCNTR			
Type									RW				RW			

Reset									0				0	0	0	0

Bit(s)	Name	Description
7	SWRST	If the flag SWRSTENAB in register USB_POWER is set to 1, the software enable mode will be enabled, and the device can be reset by writing 1 to this flag.
3:0	RSTCNTR	Signifies the duration of the reset operation after reset signal is detected on the bus. It is only enabled when software reset is not enabled. If the value is 0, the duration will be 2.5us. Otherwise, the duration will be this value multiplied by 341 then added by 2.5 in us unit. The range consequently starts from 2.5us to 5122.5us.

11 USB_EP0_CSR **USB Control/Status Register for Endpoint 0** **00**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									**SSETUPEND**	**SOUTPKTRDY**	**SENDSTALL**	**SETUPEND**	**DATAEND**	**SENTSTALL**	**INPKTRDY**	**OUTPKTRDY**
Type									A0	A0	A0	RU	A0	A1	A0	RU
Reset									0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7	SSETUPEND	**The MCU writes 1 to this bit to clear the SETUPEND bit.** It is cleared automatically. Only active when a transaction is started.
6	SOUTPKTRDY	**The MCU writes 1 to this bit to clear the OUTPKTRDY bit.** It is cleared automatically. Only active when an OUT transaction is started
5	SENDSTALL	**The MCU writes 1 to this bit to terminate the current transaction.** The STALL handshake will be transmitted, and this bit will be cleared automatically.
4	SETUPEND	**This bit will be set when a control transaction ends before the DATAEND bit is set.** An interrupt will be generated and FIFO flushed at this time. The bit is cleared by the MCU writing 1 to the SSETUPEND bit.
3	DATAEND	**The MCU sets this bit:** 1. When setting INPKTRDY for the last data packet. 2. When clearing OUTPKTRDY after unloading the last data packet. 3. When setting INPKTRDY for a zero length data packet. It is cleared automatically.
2	SENTSTALL	**This bit is set when a STALL handshake is transmitted.** The MCU should clear this bit by writing 0 to it.
1	INPKTRDY	**The MCU sets up this bit after loading a data packet into the FIFO.** It is cleared automatically when the data packet is transmitted. An interrupt is generated when this bit is set
0	OUTPKTRDY	**This bit is set when a data packet is received.** An interrupt is generated when this bit is set. The MCU clears this bit by setting up the SOUTPKTRDY bit.

16 USB_EP0_COUNT **EP0 Received Bytes Count Register** **00**

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

Name										EP0_COUNT						
Type										RU						
Reset										0	0	0	0	0	0	0

The register indicates the number of received data bytes in the endpoint 0. The value returned is valid while the OUTPKTRDY bit of the USB_EP0_CSR register is set. The register is active when the USB_INDEX register is set to 0.

Bit(s)	Name	Description
6:0	EP0_COUNT	Number of received data bytes in the endpoint 0

10 USB_EP_INMAXP USB Maximum Packet Size Register for IN Endpoint 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									MAXP							
Type									RW							
Reset									0	0	0	0	0	0	0	0

The register holds the maximum packet size for transactions through the currently selected IN endpoint - in units of one byte. When setting up the value, the programmer should note the constraints placed by the USB Specification on the packet size for bulk interrupt and isochronous transactions in full speed operation. There is an INMAXP register for each IN endpoint except for endpoint 0. The registers are active when the USB_INDEX register is set to 1, 2, 3, and 4 respectively.

The value written to this register should match the wMaxPacketSize field of the standard endpoint descriptor for the associated endpoint. A mismatch may cause unexpected results. If a value is bigger than the configured IN FIFO size for the endpoint written to the register, the value will be automatically changed to the IN FIFO size. If the value written to the register is smaller than or equal to half the IN FIFO size, two IN packets can be buffered. The configured IN FIFO size for the endpoint 1, 2, and 3, are 16 bytes, 64 bytes, and 64 bytes respectively.

The register is reset to 0. If the register is changed after the packets are sent from the endpoint, the endpoint IN FIFO should be completely flushed after writing the new value to the register.

Bit(s)	Name	Description
7:0	MAXP	Maximum packet size (unit: byte)

11 USB_EP_INCSR1 USB Control/Status Register #1 for IN Endpoint 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									ABORTPKT_EN	CLRDATATOG	SENTSTALL	SENDSTALL	FLUSHFIFO	UNDERRUN	FIFONOTEMPTY	INPKTRDY
Type									RW	A0	A1	RW	A0	A1	RU	A0
Reset									0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7	ABORTPKT_EN	When MCU writes 1 to ABORTPKT_EN, FLUSHFIFO switches to abort the packet function. This bit should be enabled before FLUSHFIFO is set. If FLUSHFIFO is set and ABORTPKT_EN is enabled the data loaded into FIFO will be discarded. After

Bit(s)	Name	Description
		the packet is aborted, EP will issue an interrupt. The programmer should wait for this interrupt to make sure the packet is aborted
6	CLRDATATOG	The MCU writes 1 to this bit to reset the endpoint IN data toggle to 0
5	SENTSTALL	The bit is set when a STALL handshake is transmitted. The FIFO is flushed and the INPKTRDY bit is cleared. The MCU should clear this bit by writing 0 to this bit.
4	SENDSTALL	The MCU writes 1 to this bit to issue a STALL handshake to an IN token. The MCU clears this bit to terminate the stall condition
3	FLUSHFIFO	The MCU writes 1 to this bit to flush the next packet to be transmitted from the endpoint IN FIFO. The FIFO pointer is reset and the INPKTRDY bit is cleared. If the FIFO contains two packets, FLUSHFIFO will need to be set twice to completely clear the FIFO. FLUSHFIFO should only be used when INPKTRDY is set. At other times, it may cause data corruption. If ABORTPKT_EN is enabled and this bit is set , the function of this bit will become ABORTPKT to abort the next packet to be transmitted from the endpoint IN FIFO and does not need to set INPKTRDY. It is the same with the FLUSHFIFO function. This bit is only active when the endpoint is idle.
2	UNDERRUN	In isochronous mode, this bit is set when a zero length data packet is sent after receiving an IN token with the INPKTRDY bit not set. In Bulk/Interrupt mode, this bit is set when a NAK is returned in response to an IN token. The MCU should clear this bit by writing a 0 to this bit
1	FIFONOTEMPTY	This bit is set when there is at least 1 packet in the IN FIFO
0	INPKTRDY	The MCU sets up this bit after loading a data packet into the FIFO. Only active when an IN transaction is started. It is cleared automatically when a data packet is transmitted. An interrupt will be generated (if enabled) when the bit is cleared

12 USB_EP_INCSR2 USB Control/Status Register #2 for IN Endpoint 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									AUTOSET	ISO	MODE	DMAENAB	FRCDATATOG			
Type									RW	RW	RW	RW	RW			
Reset									0	0	0	0	0			

Bit(s)	Name	Description
7	AUTOSET	If the MCU sets up the bit, INPKTRDY will be automatically set when the data of the maximum packet size (value in INMAXP) is loaded into the IN FIFO. If a packet of smaller than the maximum packet size is loaded, INPKTRDY will have to be set manually. When 2 packets are in the IN FIFO, INPKTRDY will also be automatically set when the first packet is sent if the second packet is the maximum packet size
6	ISO	The MCU sets up this bit to enable the IN endpoint for isochronous transfer and clears it to enable the IN endpoint for bulk/interrupt transfers
5	MODE	The MCU sets up this bit to enable the endpoint direction as IN and clears it to enable the endpoint direction as OUT. It is valid only when the same endpoint FIFO is used for both IN and OUT transactions.
4	DMAENAB	The MCU sets up this bit to enable the DMA request for the IN endpoint.
3	FRCDATATOG	The MCU sets up this bit to force the endpoints IN data toggle to switch after each data packet is sent regardless of whether an ACK has been received. This can be used by interrupt IN endpoints which are used to communicate

Bit(s)	Name	Description
		rate feedback for isochronous endpoints

13	USB_EP_OUTMAXP	USB Maximum Racket Size Register for OUT Endpoint	00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									MAXP							
Type									RW							
Reset									0	0	0	0	0	0	0	0

This register holds the maximum packet size for transactions through the currently selected OUT endpoint (unit: byte). When setting up this value, the programmer should note the constraints placed by the USB specification on packet sizes for bulk, interrupt, and isochronous transactions in full speed operations. There is an OUTMAXP register for each OUT endpoint except for endpoint 0. The registers are active when the USB_INDEX register is set to 1 and 2 respectively.

The value written to this register should match the wMaxPacketSize field of the standard endpoint descriptor for the associated endpoint. A mismatch may cause unexpected results. The total amount of data represented by the value written to this register must not exceed the FIFO size for the OUT endpoint and should not exceed half the FIFO size if double buffering is required. If a value bigger than the configured OUT FIFO size for the endpoint is written to the register, the value will be automatically changed to the OUT FIFO size. If the value written to the register is smaller than or equal to half the OUT FIFO size, two OUT packets can be buffered. The configured IN FIFO size for the endpoint 1, 2, and 3 are both 16, 64, and 64 bytes, respectively.

Bit(s)	Name	Description
7:0	MAXP	Maximum packet size (unit: byte)

14	USB_EP_OUTCSR1	USB Control/Status Register #1 for OUT Endpoint	00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									CLRDTATOG	SENTSTALL	SENDSTALL	FLUSHFIFO	DATAERROR	OVERRUN	FIFOFULL	RXPKTRDY
Type									A0	A1	RW	A0	RU	A1	RU	A1
Reset									0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7	CLRDTATOG	The MCU writes 1 to this bit to reset the endpoint data toggle to 0.
6	SENTSTALL	The bit is set when a STALL handshake is transmitted. The MCU should clear this bit by writing 0 to it.
5	SENDSTALL	The MCU writes 1 to this bit to issue a STALL handshake. The MCU clears this bit to terminate the stall condition. This bit will have no effect if the OUT endpoint is in the isochronous mode
4	FLUSHFIFO	The MCU writes 1 to this bit to flush the next packet to be read from the endpoint OUT FIFO. If the FIFO contains two packets, FLUSHFIFO will need to be set twice to completely clear the FIFO. FLUSHFIFO should only be used when OUTPKTRDY is set. At other times, it may cause data corruption.
3	DATAERROR	The bit is set when OUTPKTRDY is set if the data packet has a CRC or bit

Bit(s)	Name	Description
		stuff error. It is cleared when OUTPKTRDY is cleared. This bit is only valid in the isochronous mode.
2	OVERRUN	The bit will be set if an OUT packet cannot be loaded into the OUT FIFO. The MCU should clear the bit by writing 0 to it. This bit is only valid in the isochronous mode.
1	FIFOFULL	This bit is set when no more packets can be loaded into the OUT FIFO
0	RXPKTRDY	The bit is set when a data packet has been received. The MCU should clear (write 0 to it) the bit when the packet is unloaded from the OUT FIFO. An interrupt will be generated when the bit is set. When the receiving null packet is received, OUTPKTRDY will be set after USB_INTROUT1 is high.

15 **USB_EP_OUTCSR2** **USB Control/Status Register #2 for OUT Endpoint** **00**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									**AUTOCLEAR**	**ISO**	**DMAENAB**	**DMAMODE**				
Type									RW	RW	RW	RW				
Reset									0	0	0	0				

Bit(s)	Name	Description
7	AUTOCLEAR	If the MCU sets up this bit, the OUTPKTRDY bit will be automatically cleared when a packet of OUTMAXP bytes is unloaded from the OUT FIFO. When packets of smaller than the maximum packet size are unloaded, OUTPKTRDY will have to be cleared manually.
6	ISO	The MCU sets up this bit to enable the OUT endpoint for isochronous transfers and clears it to enable the OUT endpoint for bulk/interrupt transfers
5	DMAENAB	The MCU sets up this bit to enable the DMA request for the OUT endpoint
4	DMAMODE	Two modes of DMA operation are supported: DMA mode 0 in which a DMA request is generated for all received packets, together with an interrupt (if enabled). DMA mode 1 in which a DMA request (but no interrupt) is generated for OUT packets of size OUTMAXP bytes and an interrupt (but no DMA request) is generated for OUT packets of any other size. The MCU sets up the bit to select DMA mode 1 and clears this bit to select DMA mode 0.

16 **USB_EP_COUNT1** **USB OUT Endpoint Byte Counter Register LSB Part for Endpoint** **00**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									**NUML**							
Type									RU							
Reset									0	0	0	0	0	0	0	0

The register holds the lower 8 bits of the number of received data bytes in the packet in the FIFO associated with the currently selected OUT endpoint. The value returned is valid while OUTPKTRDY in register USB_OUTCSR1 is set. The registers are active when the USB_INDEX register is set to 1 and 2 respectively.

Bit(s)	Name	Description

Bit(s)	Name	Description
7:0	NUML	Lower 8 bits of the number of received data bytes for the OUT endpoint

17 **USB_EP_COUNT2** **USB OUT Endpoint Byte Counter Register MSB Part for Endpoint** **00**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														NUMH		
Type														RU		
Reset														0	0	0

The register holds the upper 3 bits of the number of received data bytes in the packet in the FIFO associated with the currently selected OUT endpoint. The value returned is valid while OUTPKTRDY in register USB_EP_OUTCSR1 is set. The registers are active when the USB_INDEX register is set to 1 and 2 respectively.

Bit(s)	Name	Description
2:0	NUMH	Upper 8 bits of the number of received data bytes for the OUT endpoint.

20 **USB_EP0_FIFO_DB0** **USB Endpoint 0 FIFO Register DB0** **00**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									FIFO0_DB0							
Type									Other							
Reset									0	0	0	0	0	0	0	0

The register provides MCU access to the FIFO for the endpoint 0. Writing to this register will load data to the FIFO for the endpoint 0. Reading this register unloads data from the FIFO for the endpoint 0.

The register provides word, half word, and byte mode accesses. If word or half word accesses are performed, the less significant byte will correspond to the prior byte to load to or unload from the FIFO.

Bit(s)	Name	Description
7:0	FIFO0_DB0	The first byte to be loaded to or unloaded from the FIFO.

21 **USB_EP0_FIFO_DB1** **USB Endpoint 0 FIFO Register DB1** **00**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									FIFO0_DB1							
Type									Other							
Reset									0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7:0	FIFO0_DB1	The second byte to be loaded to or unloaded from the FIFO.

22 **USB_EP0_FIFO_** **USB Endpoint 0 FIFO Register DB2** **00**

DB2

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									FIFO0_DB2							
Type									Other							
Reset									0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7:0	FIFO0_DB2	The third byte to be loaded to or unloaded from the FIFO.

23 USB_EP0_FIFO_DB3 **USB Endpoint 0 FIFO Register DB3** 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									FIFO0_DB3							
Type									Other							
Reset									0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7:0	FIFO0_DB3	The forth byte to be loaded to or unloaded from the FIFO.

24 USB_EP1_FIFO_DB0 **USB Endpoint 1 FIFO Register DB0** 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									FIFO1_DB0							
Type									Other							
Reset									0	0	0	0	0	0	0	0

The register provides MCU access to the FIFO for the endpoint 1. Writing to this register will load data to the FIFO for the endpoint 1. Reading this register will unload data from the FIFO for the endpoint 1.

The register provides word, half word, and byte mode accesses. If word or half word accesses are performed, the less significant byte will correspond to the prior byte to load to or unload from the FIFO.

Bit(s)	Name	Description
7:0	FIFO1_DB0	The first byte to be loaded to or unloaded from the FIFO.

25 USB_EP1_FIFO_DB1 **USB Endpoint 1 FIFO Register DB1** 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									FIFO1_DB1							
Type									Other							
Reset									0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7:0	FIFO1_DB1	The second byte to be loaded to or unloaded from the FIFO.

26 USB_EP1_FIFO_DB2 **USB Endpoint 1 FIFO Register DB2** 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									**FIFO1_DB2**							
Type									Other							
Reset									0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7:0	FIFO1_DB2	The third byte to be loaded to or unloaded from the FIFO.

27 USB_EP1_FIFO_DB3 **USB Endpoint 1 FIFO Register DB3** 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									**FIFO1_DB3**							
Type									Other							
Reset									0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7:0	FIFO1_DB3	The forth byte to be loaded to or unloaded from the FIFO.

28 USB_EP2_FIFO_DB0 **USB Endpoint 2 FIFO Register DB0** 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									**FIFO2_DB0**							
Type									Other							
Reset									0	0	0	0	0	0	0	0

The register provides MCU access to the FIFO for the endpoint 2. Writing to this register will load data to the FIFO for the endpoint 2. Reading this register will
unloads data from the FIFO for the endpoint 2.unloads data from the FIFO for the endpoint 2.
The register provides word, half word, and byte mode accesses. If word or half word accesses are performed, the less significant byte will correspond to the prior byte to load to or unload from the FIFO.

Bit(s)	Name	Description
7:0	FIFO2_DB0	The first byte to be loaded to or unloaded from the FIFO.

29 USB_EP2_FIFO_DB1 **USB Endpoint 2 FIFO Register DB1** 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									**FIFO2_DB1**							
Type									Other							
Reset									0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7:0	FIFO2_DB1	The second byte to be loaded to or unloaded from the FIFO.

2A USB_EP2_FIFO_DB2 USB Endpoint 2 FIFO Register DB2 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									FIFO2_DB2							
Type									Other							
Reset									0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7:0	FIFO2_DB2	The third byte to be loaded to or unloaded from the FIFO.

2B USB_EP2_FIFO_DB3 USB Endpoint 2 FIFO Register DB3 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									FIFO2_DB3							
Type									Other							
Reset									0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7:0	FIFO2_DB3	The forth byte to be loaded to or unloaded from the FIFO.

2C USB_EP3_FIFO_DB0 USB Endpoint 3 FIFO Register DB0 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									FIFO3_DB0							
Type									Other							
Reset									0	0	0	0	0	0	0	0

The register provides MCU access to the FIFO for the endpoint 3. Writing to this register will load data to the FIFO for the endpoint 3. Reading this register will unload data from the FIFO for the endpoint 3.

The register provides word, half word, and byte mode accesses. If word or half word accesses are performed, the less significant byte will correspond to the prior byte to load to or unload from the FIFO.

Bit(s)	Name	Description
7:0	FIFO3_DB0	The first byte to be loaded to or unloaded from the FIFO.

2D USB_EP3_FIFO_DB1 USB Endpoint 3 FIFO Register DB1 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									FIFO3_DB1							
Type									Other							
Reset									0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7:0	FIFO3_DB1	The second byte to be loaded to or unloaded from the FIFO.

2E USB_EP3_FIFO_DB2 **USB Endpoint 3 FIFO Register DB2** 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									**FIFO3_DB2**							
Type									Other							
Reset									0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7:0	FIFO3_DB2	The third byte to be loaded to or unloaded from the FIFO.

2F USB_EP3_FIFO_DB3 **USB Endpoint 3 FIFO Register DB3** 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									**FIFO3_DB3**							
Type									Other							
Reset									0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7:0	FIFO3_DB3	The forth byte to be loaded to or unloaded from the FIFO.

30 USB_EP4_FIFO_DB0 **USB Endpoint 4 FIFO Register DB0** 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									**FIFO4_DB0**							
Type									Other							
Reset									0	0	0	0	0	0	0	0

The register provides MCU access to the FIFO for the endpoint 4. Writing to this register will load data to the FIFO for the endpoint 4. Reading this register will unload data from the FIFO for the endpoint 4.

The register provides word, half word, and byte mode accesses. If word or half word accesses are performed, the less significant byte will correspond to the prior byte to load to or unload from the FIFO.

Bit(s)	Name	Description
7:0	FIFO4_DB0	The first byte to be loaded to or unloaded from the FIFO.

31 USB_EP4_FIFO_DB1 **USB Endpoint 4 FIFO Register DB1** 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									**FIFO4_DB1**							
Type									Other							
Reset									0	0	0	0	0	0	0	0

Bit(s)	Name	Description

Bit(s)	Name	Description
7:0	FIFO4_DB1	The second byte to be loaded to or unloaded from the FIFO.

32 USB_EP4_FIFO_DB2 **USB Endpoint 4 FIFO Register DB2** 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									FIFO4_DB2							
Type									Other							
									0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7:0	FIFO4_DB2	The third byte to be loaded to or unloaded from the FIFO.

33 USB_EP4_FIFO_DB3 **USB Endpoint 4 FIFO Register DB3** 00

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									FIFO4_DB3							
Type									Other							
Reset									0	0	0	0	0	0	0	0

Bit(s)	Name	Description
7:0	FIFO4_DB3	The forth byte to be loaded to or unloaded from the FIFO.

240 USB_CON **USB PHY Control** 20

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name											NULLPKT_FIX				DMPULLUP	DPPULLUP
Type											RW				RW	RW
Reset											1				0	0

Bit(s)	Name	Description
5	NULLPKT_FIX	If NULLPKT_FIX is set to 1, the USB controller will not issue a DMAreq when a null packet is received.
1	DMPULLUP	Pull-up enabling pin. Enables the pull-up 1.5KOhm pull-up on D pin as a full speed device by setting it to high
0	DPPULLUP	Pull-up enabling pin. Enables the pull-up 1.5KOhm pull-up on D+ pin as a full speed device by setting it to high

3.14.4 System Integration Guide

3.14.4.1 USB device configuration

The target audience of this section is the software engineer.
The USB device controller features one control endpoint and 6 other endpoints. The configuration of interfaces and endpoints can be accommodated by software for specific functions, basically supporting Bluetooth HCI, and device firmware upgrade.
Bluetooth HCI transport layer defines the configuration of endpoints and interfaces.

✓ One voice channel with 16-bit encoding.

Endpoint	Endpoint type	Max. packet size (bytes)	Max. bandwidth (bytes/ms)	Min. bandwidth (bytes/ms)	Double buffer in controller	Generic DMA
Endpoint 0 (command)	Control	64			No	No
Endpoint 1 IN	Bulk (IN)	64		1024	No	Yes
Endpoint 2 OUT	Bulk (OUT)	64		1024	No	Yes
Endpoint 2 IN	Bulk (IN)	64	1024		No	Yes
Endpoint 2 OUT	Bulk (OUT)	64	1024		No	Yes
Endpoint 3 IN	Interrupt (IN)	16		16	No	No
Endpoint 4 IN	Interrupt (IN)	16		16	No	No

*When the maximum packet size is smaller than one half of the device FIFO size (64 bytes), the double buffer will be automatically enabled by hardware.

The pull-up resister on the USB transceiver is initially disconnected when boot-up. No external resistor is required. The software should enable it after performing the configuration of the USB device controller.

3.14.4.2 System Infrastructure Configuration

The clock, interrupt and DMA are defined as the system infrastructure. It requires several steps to bring up the USB. Those steps should be done in sequence to prevent from malfunction.

Power-on

1. Enable USB PLL.

2. Enable USB clock after USB PLL is settled.

3. Unmask the USB interrupt in the interrupt controller.

4. Enable the pull-up resistor.

The USB device controller can generate the interrupt when conditions are met as defined in USB_INTRINE, USB_INTROUTE, and USB_INTRUSBE.
The generic DMA controller is used to move data from or to the USB device controller. The USB device controller will use at most 4 DMA channels for ACL and SCO. The user should use the half-channel DMA since only the half channel DMA has the hardware flow control.
The USB FIFO provides byte and word accesses to the read/write port of USB_EP0_FIFO, USB_EP1_FIFO, USB_EP2_FIFO, USB_EP3_FIFO and USB_EP4_FIFO. If the data buffer allocated in memory is word aligned, the user can enable word transfer in the DMA controller. If the data buffer allocated in memory is not word aligned, the user should set to byte aligned and set B2W in DMAx_CON to 1 to enable fast byte-to-word transfer.

3.14.4.3 Power On/Off USB PHY and Controller Sequence

Power-on sequence after plug-in

1. Turn on Vusb(PHY 3.3v power) – The control register is in PMIC document.

2. Turn on USB AHB clock(78MHz) – The control register is in config document.

3. Turn on internal 48MHz PLL – the control register is in clock document.

4. Wait for 50 usec. (PHY 3.3v power stable time)

5. Turn on USB PHY BIAS current control → reg[USB+08C1h] bit3 = 1. (RG_USB11_FSLS_ENBGRI).

6. Wait for 10 usec.

7. Set up D+ pull up register for connecting Host → reg[USB+0240h] bit 0=1(PUB)

Power-off sequence after plug-out

1. Release D+ pull up register for disconnecting Host → Setting reg[USB+0240h] bit 0=0 (PUB)

2. Turn off USB PHY BIAS current control → reg[USB+08C1h] bit3 = 0. (RG_USB11_FSLS_ENBGRI).

3. Turn off Vusb (PHY 3.3v power) – The control register is in PMIC document

3.15 Accessory Detector

3.15.1 General Description

The hardware accessory detector (ACCDET) detects plug-in/out of multiple types of external components. Based on the suggested circuit (see **Figure 41**), this design supports 3 types of external components, which are microphone, hook-switch and TV-OUT line. This design uses the internal 2-bit comparator to separate external components. The de-bounce scheme is also supported to resist uncertain input noises. When the plug-in/out state is stable, the PWM unit of ACCDET will enable the comparator, MBIAS and threshold voltage of the comparator periodically for the plugging detection. With suitable PWM settings, very low-power consumption can be achieved when the detection feature is enabled. In order to compensate the delay between the detection login and comparator, the delay enabling scheme is adopted. Given the suitable delay number compared to the rising edge of PWM high pulse, the stable plugging state can be prorogated to digital detection logic. Then the correct plugging state can be detected and reported.

Figure 42 shows the state machine without TV-OUT mode. The state machine is executed by the software to control the ACCDET design. The ACCDET design will send one interrupt to acknowledge the software after the ACCDET input state is changed and the duration of the state is longer than de-bounce time. The software needs to read out the memorized ACCDET input state and follow the recommend state machine to program the register in it.

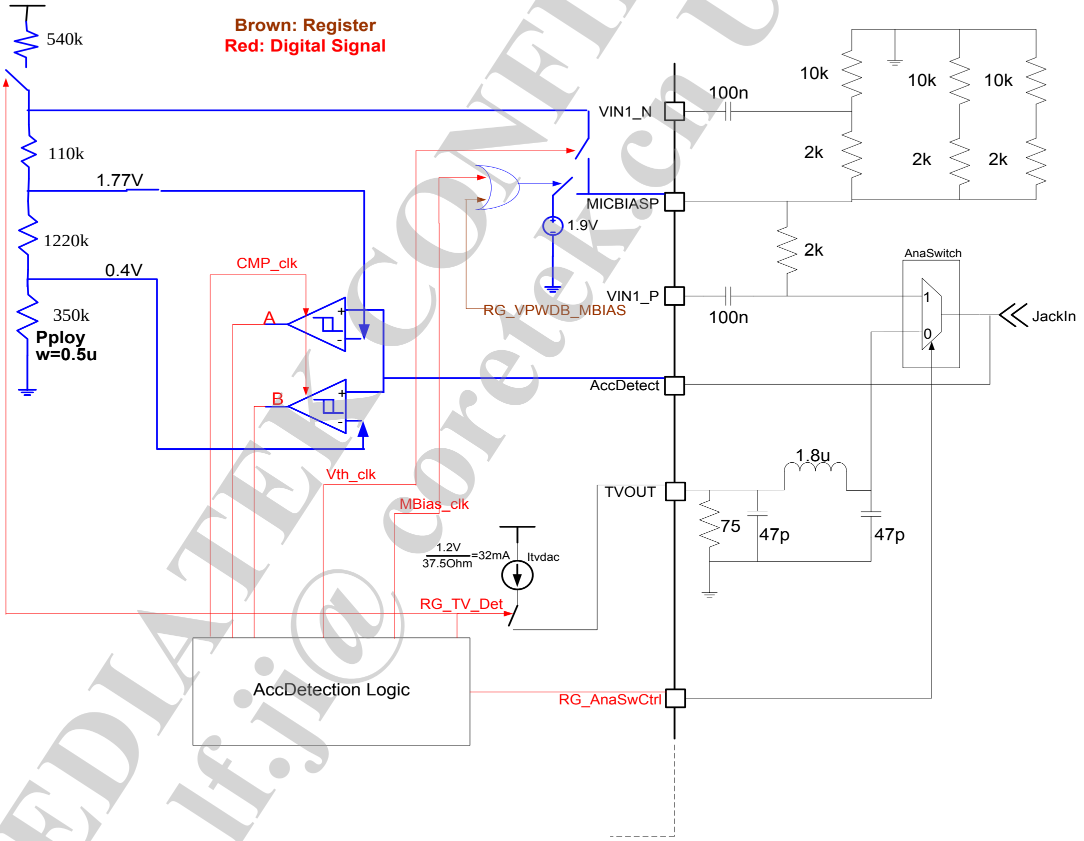


Figure 41. Suggested Accessory Detection Circuit.

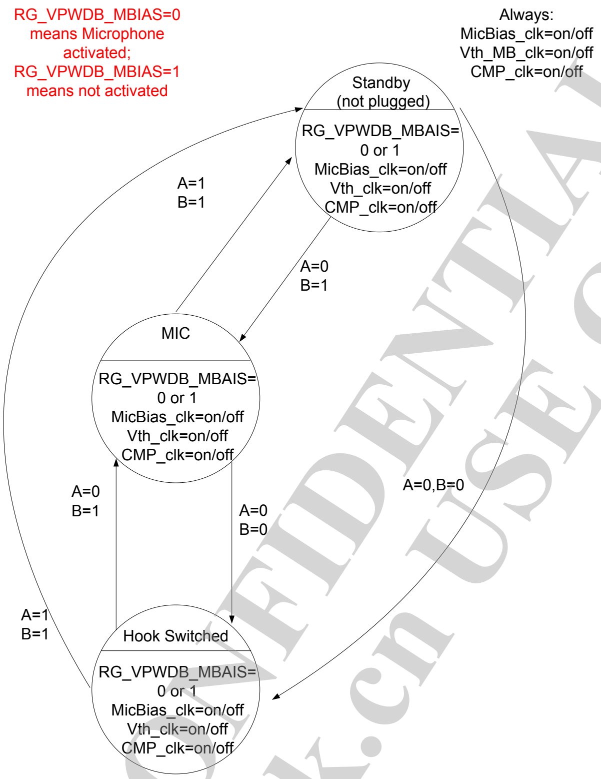


Figure 42. The Sate machine between Microphone and Hook-Switch plug-in/out change.

3.15.2 Pulse Width Modulation

The ACCDET design also provides one Pulse-Width-Modulation (PWM) to enable the comparator, microphone's bias current and the threshold voltage of the comparator periodically. With suitable PWM and settings for delayed enabling, the ACCDET can achieve very low power consumption and accurate plug-in/out detection. Figure 43 shows a timing diagram example of such PWM design. The output from PWM keeps being at "0" until the value of the counter is smaller than the programmed threshold.

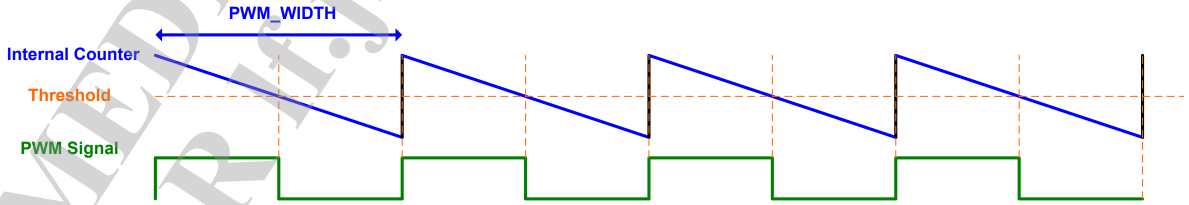


Figure 43. PWM waveform.

3.15.3 Register Definition

Module name: ACCDET base address: (+A0750000h)

Address	Name	Width	Register function
A0750000	ACCDET_RSTB	32	ACCDET software reset register
A0750004	ACCDET_CTRL	32	ACCDET control register
A0750008	ACCDET_STATE_SWCTRL	32	ACCDET state switch control register
A075000C	ACCDET_PWM_WIDTH	32	ACCDET PWM width register
A0750010	ACCDET_PWM_THRESH	32	ACCDET PWM threshold register
A0750024	ACCDET_EN_DELAY_NUM	32	ACCDET enable delay number register
A0750028	ACCDET_PWM_IDLE_VALUE	32	ACCDET PWM IDLE value register
A075002C	ACCDET_DEBOUNCE0	32	ACCDET debounce0 register
A0750030	ACCDET_DEBOUNCE1	32	ACCDET debounce1 register
A0750038	ACCDET_DEBOUNCE3	32	ACCDET debounce3 register
A075003C	ACCDET_IRQ_STS	32	ACCDET interrupt status register
A0750040	ACCDET_CURR_IN	32	ACCDET current input status register
A0750044	ACCDET_SAMPLE_IN	32	ACCDET sampled input status register
A0750048	ACCDET_MEMOIZED_IN	32	ACCDET memorized input status register
A075004C	ACCDET_LAST_MEMOIZED_IN	32	ACCDET last memorized input status register
A0750050	ACCDET_FSM_STATE	32	ACCDET FSM status register
A0750054	ACCDET_CURR_DEBOUNCE	32	ACCDET current de-bounce status register
A0750058	ACCDET_VERSION	32	ACCDET version code
A075005C	ACCDET_IN_DEFAULT	32	default value of accdet_in

A0750000 ACCDET_RSTB ACCDET Software Reset Register 00000001

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Mne																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne																RSTB
Type																RW
Reset																1

Overview: After applying the setting to register, software reset is necessary for state initialization. Without this process, ACCDET may detect incorrect plug state.

Bit(s)	Mnemonic	Name	Description
0	RSTB	RSTB	**Set to 0 to reset the ACCDET unit and set to 1 after the reset process is finished.** This software reset will clear ACCDET's enable signal but keep all ACCDET's settings. After the reset process, ACCDET will return to the IDLE state.

A0750004 ACCDET_CTRL ACCDET Control Register 00000001

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Mne																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne																ACCDET_EN
Type																RW
Reset																1

Bit(s)	Mnemonic	Name	Description
0	ACCDET_EN	EN	Set to 1 to enable the ACCDET unit.

A0750008 ACCDET_STATE_SWCTRL ACCDET State Switch Control Register 00000001

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Mne																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne												MBIAS_PWM_EN	VTH_PWM_EN	CMP_PWM_EN		
Type												RW	RW	RW		
Reset												0	0	0		

Bit(s)	Mnemonic	Name	Description
4	MBIAS_PWM_EN	MBIAS_PWM_EN	Enables PWM of ACCDET MBIAS unit
3	VTH_PWM_EN	VTH_PWM_EN	Enables PWM of ACCDET voltage threshold unit
2	CMP_PWM_EN	CMP_PWM_EN	Enables PWM of ACCDET comparator

A075000C ACCDET_PWM_WIDTH ACCDET PWM Width Register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Mne																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne	PWM_WIDTH															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
15:0	PWM_WIDTH	PWM_WIDTH	**ACCDET PWM width** It is PWM max. counter value. It will be the initial value for the internal counter. The PWM internal counter always counts down to zero to finish one compete period, and the value of the internal counter will return to the value of PWM_WIDTH. PWM output frequency = (32k/PWM_WIDTH) Hz.

A0750010 ACCDET_PWM_THRESH ACCDET PWM Threshold Register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Mne																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne	PWM_THRESH															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit(s)	Mnemonic	Name	Description
15:0	PWM_THRESH	PWM_THRESH	**ACCDET PWM threshold** When the internal counter value is bigger than or equal to PWM_THRESH, the PWM output signal will be "0". When the internal counter is smaller than PWM_THRESH, the PWM output signal will be "1". PWM output duty cycle = (PWM_THRESH)x(1/32) ms.

Figure 44 shows the PWM waveform with register value present.

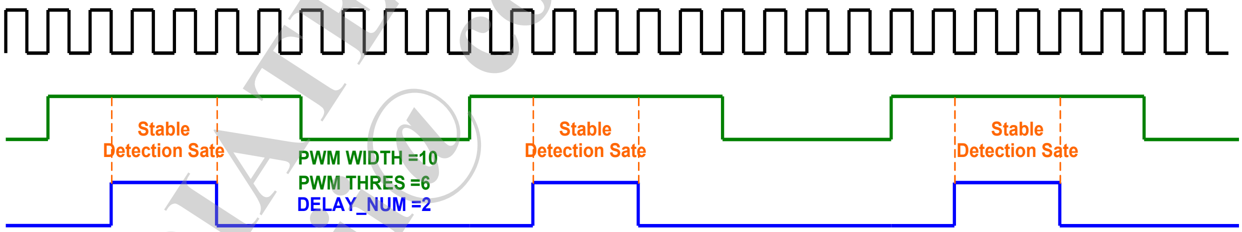


Figure 44. PWM waveform with register value present

A0750024 ACCDET_EN_DELAY_NUM ACCDET Enable Delay Number Register 00000101

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Mne																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne	FALL_DELA	RISE_DELAY_NUM														

	Y_NUM															
Type	RW	RW														
Reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1

Bit(s)	Mnemonic	Name	Description
15	**FALL_DELAY_NUM**	FALL_DELAY_NUM	**Falling delay cycle compared to CMP PWM waveform** In order to make sure the plug state is stable after disabling ACCDET, the suitable delay cycle is necessary. This number indicates the clock cycle number between the point when the digital part of ACCDET stops receiving accdet_in and the point when the analog part of ACCDET stops working.
14:0	**RISE_DELAY_NUM**	RISE_DELAY_NUM	**Rising delay cycle compared to PWM waveform** In order to make sure the plug state is stable before activating ACCDET, the suitable delay cycle is necessary. This number indicates the clock cycle number between the point when the analog part of ACCDET starts working and the point when the digital part of ACCDET starts receiving stable accdet_in. This number should be fine tuned depending on different project requirements.

A0750028 ACCDET_PWM_IDLE_VALUE ACCDET PWM IDLE Value Register 00000001

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Mne																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne														**MBIAS**	**VTH**	**CMP**
Type														RW	RW	RW
Reset														0	0	1

Bit(s)	Mnemonic	Name	Description
2	**MBIAS**	MBIAS	IDLE value of MBIAS PWM
1	**VTH**	VTH	IDLE value of VTH PWM
0	**CMP**	CMP	IDLE value of CMP PWM

A075002C ACCDET_DEBOUNCE0 ACCDET Debounce0 Register 00000010

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Mne																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne	**DEBOUNCE0**															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Overview: This register defines the waiting period before plug-in/out or release events are considered stable. If the de-bounce setting is too small, the plug-in/out will be too sensitive and detect too many unexpected plug-ins/outs. The suitable de-bounce time setting must be adjusted for the user's demand.

Bit(s)	Mnemonic	Name	Description
15:0	DEBOUNCE0	DEBOUNCE0	**De-bounce time control of the next state = 2'b00** De-bounce time = DEBOUNCE/32 ms

A0750030 ACCDET_DEBOUNCE1 ACCDET Debounce1 Register 00000010

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Mne																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne	DEBOUNCE1															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Overview: This register defines the waiting period before plug-in/out or release events are considered stable. If the de-bounce setting is too small, the plug-in/out will be too sensitive and detect too many unexpected plug-ins/outs. The suitable de-bounce time setting must be adjusted for the user's demand.

Bit(s)	Mnemonic	Name	Description
15:0	DEBOUNCE1	DEBOUNCE1	**De-bounce time control of the next state = 2'b01** De-bounce time = DEBOUNCE/32 ms

A0750034 ACCDET_DEBOUNCE2 ACCDET Debounce2 Register 00000010

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Mne																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne	DEBOUNCE2															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Overview: This register defines the waiting period before plug-in/out or release events are considered stable. If the de-bounce setting is too small, the plug-in/out will be too sensitive and detect too many unexpected plug-ins/outs. The suitable de-bounce time setting must be adjusted for the user's demand.

Bit(s)	Mnemonic	Name	Description
15:0	DEBOUNCE2	DEBOUNCE2	**De-bounce time control of the next state = 2'b10** De-bounce time = DEBOUNCE/32 ms

A0750038 ACCDET_DEBOUNCE3 ACCDET Debounce3 Register 00000010

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Mne																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne	DEBOUNCE3															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Overview: This register defines the waiting period before plug-in/out or release events are considered stable. If the de-bounce setting is too small, the plug-in/out will be too sensitive and detect too many unexpected plug-ins/outs. The suitable de-bounce time setting must be adjusted for the user's demand.

Bit(s)	Mnemonic	Name	Description
15:0	DEBOUNCE3	DEBOUNCE3	**De-bounce time control of the next state = 2'b11** De-bounce time = DEBOUNCE/32 ms

A075003C ACCDET_IRQ_STS ACCDET Interrupt Status Register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Mne																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne								IRQ_CLR								IRQ
Type								RW								RW
Reset								0								0

Overview: When the interrupt of ACCDET is asserted, IRQ_CLR must be set to 1 to clear the interrupt status. This bit will pause all activities in the ACCDET design until both interrupt status and IRQ_CLR are cleared. The software should write 1 to IRQ_CLR first to clear the interrupt (IRQ). After that, if pclk gating is enabled, the software should read ACCDET_IRQ_STS again to make IRQ_CLR self-reset to 0.

Bit(s)	Mnemonic	Name	Description
8	IRQ_CLR	IRQ_CLR	**Clears interrupt status of ACCDET unit**
0	IRQ	IRQ	**Interrupt status of ACCDET unit** Because this register will be cleared by hardware, the interrupt edge-sensitive scheme should be adopted for this design.

A0750040 ACCDET_CURR_IN ACCDET Current Input Status Register 00000003

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Mne																
Type																
Reset																

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne															CURR_IN	
Type															RO	
Reset															1	1

Bit(s)	Mnemonic	Name	Description
1:0	**CURR_IN**	CURR_IN	Current input status of ACCDET unit

A0750044 ACCDET_SAMPLE_IN ACCDET Sampled Input Status Register 00000003

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Mne																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne															SAMPLE_IN	
Type															RO	
Reset															1	1

Bit(s)	Mnemonic	Name	Description
1:0	**SAMPLE_IN**	SAMPLE_IN	**Samples input status of ACCDET unit** When the plug-in/out state is changed, the ACCDET unit will do sampling.

A0750048 ACCDET_MEMOIZED_IN ACCDET Memorized Input Status Register 00000003

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Mne																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne															MEMORIZED_IN	
Type															RO	
Reset															1	1

Bit(s)	Mnemonic	Name	Description
1:0	**MEMORIZED_IN**	MEMORIZED_IN	**Memorized input status of ACCDET unit** When the plug-in/out states is changed and held longer than the debounce time, the ACCDET unit will save the sampled input state to the memorized state. The interrupt will also be asserted to acknowledge the software.

A075004C ACCDET_LAST_MEMOIZED_IN ACCDET Last Memorized Input Status Register 00000003

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

Mne																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne															LAST_MEMORIZED_IN	
Type															RO	
Reset															1	1

Bit(s)	Mnemonic	Name	Description
1:0	**LAST_MEMORIZED_IN**	LAST_MEMORIZED_IN	Last memorized input status of ACCDET unit

A0750050 ACCDET_FSM_STATE ACCDET FSM Status Register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Mne																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne														FSM_STATE		
Type														RO		
Reset														0	0	0

Bit(s)	Mnemonic	Name	Description
2:0	**FSM_STATE**	FSM_SATE	**State of ACCDET unit finite-state-machine** 0: ACCDET_IDLE 1: ACCDET_SAMPLE 2: ACCDET_DEBOUNCE 3: ACCDET_CHECK 4: ACCDET_MEMORIZED 5: ACCDET_IRQ

A0750054 ACCDET_CURR_DEBOUNCE ACCDET Current De-bounce Status Register 00000004

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Mne																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mne	CURR_DEBOUNCE															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

Bit(s)	Mnemonic	Name	Description
15:0	**CURR_DEBOUNCE**	CURR_DEBOUNCE	Currently used de-bounce time setting

A0750058 ACCDET_VERSION ACCDET Version Code 000000 03

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name															ACCDET_VERSION	
Type															RO	
Reset															1	1

Bit(s)	Mnemonic	Name	Description
1:0	ACCDET_VERSION	ACCDET_VERSION	Version code for ACCDET

A075005C ACCDET_IN_DEFAULT Default Value of accdet_in 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name												ACCDET_IN_DEFAULT_REFRESH_EN			ACCDET_IN_DEFAULT	
Type												RW			RW	
Reset												0			0	0

Overview: The default value of sample_accdet_in and memorised_accdet_in can be set by software instead of using the default value set by hardware(i.e. 3). ACCDET_DEFAULT_REFRESH_EN is the enable bit controlling whether to use this additional function. The value of sample_accdet_in and memorized_accdet_in will change when accdet_en rises from low to high. *Note that if software reset is applied when accdet_en is high, the default value of sample_accdet_in and memorized_accdet_in will also be loaded when the software reset is de-asserted.*

Bit(s)	Mnemonic	Name	Description
4	ACCDET_IN_DEFAULT_REFRESH_EN	ACCDET_IN_DEFAULT_REFRESH_EN	**Enable signal for whether to load accdet_in_default** 0: accdet_in_default will not be loaded. 1: accdet_in_default will be loaded.
1:0	ACCDET_IN_DEFAULT	ACCDET_IN_DEFAULT	**Default value of accdet_in set by software**

3.16 SD Memory Card Controller (MSDC0)

3.16.1 Introduction

The controller fully supports the SD memory card bus protocol as defined in SD Memory Card Specification Part 1 Physical Layer Specification version 2.0.

Furthermore, the controller also partially supports the SDIO card specification version 2.0. However, the controller can only be configured as the host of the SD memory card. Hereafter, the controller is also abbreviated as the SD controller. The following are the main features of the controller.

- Interface with MCU by APB bus
- 16/32-bit access on APB bus
- 16/32-bit access for control registers
- 32-bit access for FIFO
- Built-in 32 bytes FIFO buffers for transmit and receive, FIFO is shared for transmit and receive
- Built-in CRC circuit
- CRC generation can be disabled
- DMA supported
- Interrupt capabilities
- Data rate up to 48 Mbps in serial mode, 48x4 Mbps in parallel model, the module is targeted at 48 MHz operating clock
- Serial clock rate on SD bus is programmable
- Card detection capabilities during sleep mode
- Controllability of power for memory card
- Does not support SPI mode for SD memory card
- Does not support multiple SD memory cards

3.16.2 Overview

3.16.2.1 Pin Assignment

The following lists pins required for the SD memory card. Table 50 shows how the pins are shared. Note that all I/O pads have embedded both pull-up and pull-down resistors because they are shared by the SD memory card. The pull-down resistors for these pins can be used for power saving. If optimal pull-up or pull-down resistors are required on the system board, all embedded pull-up and pull-down resistors can be disabled by programming the corresponding control registers. The VDDPD pin is used for power saving. Power for the SD memory card can be shut down by programming the corresponding control register. The WP (Write Protection) pin is used to detect the status of the Write Protection Switch on the SD memory card.

Table 50. Sharing of pins for SD memory card controller

No.	Name	Type	MMC	SD	Description

No.	Name	Type	MMC	SD	Description
1	SD_CLK	O	CLK	CLK	Clock
2	SD_DAT3	I/O/PP		CD/DAT3	Data Line [Bit 3]
3	SD_DAT0	I/O/PP	DAT0	DAT0	Data Line [Bit 0]
4	SD_DAT1	I/O/PP		DAT1	Data Line [Bit 1]
5	SD_DAT2	I/O/PP		DAT2	Data Line [Bit 2]
6	SD_CMD	I/O/PP	CMD	CMD	Command Or Bus State
7	SD_PWRON	O			VDD ON/OFF
8	SD_WP	I			Write Protection Switch in SD
9	SD_INS	I	VSS2	VSS2	Card Detection

3.16.2.2 Card Detection

For SD memory card, detection of card insertion/removal by hardware is supported, and a dedicated pin “INS” is used to perform card insertion and removal for SD. The pin “INS” will be connected to the pin “VSS2” of a SD connector (see **Figure 45**).

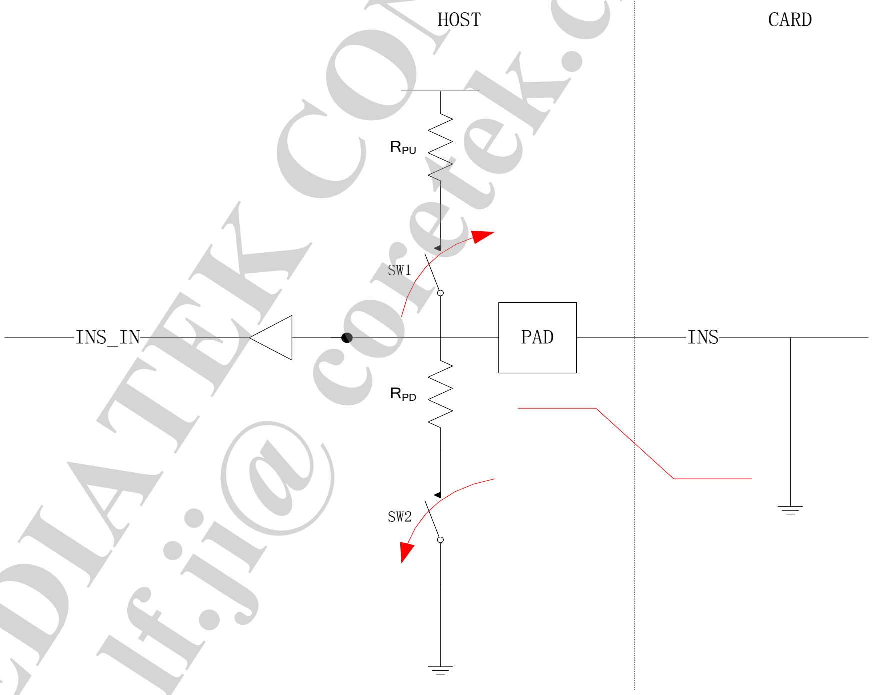


Figure 45. Card detection for SD memory card

3.16.3 Register Definition

Module name: MSDC0 base address: (+A0130000h)

Address	Name	Width	Register function

Address	Name	Width	Register function
A0130000	**MSDC_CFG**	32	**SD memory card controller configuration register** For general configuration of the SD controller. *Note: MSDC_CFG[31:16] can be accessed by 16-bit APB bus access.*
A0130004	**MSDC_STA**	32	**SD memory card controller status register** Contains the status of FIFO, interrupts and data requests.
A0130008	**MSDC_INT**	32	**SD memory card controller interrupt register** Contains the status of interrupts. Note that the register still shows the status of interrupt even though the interrupt is disabled, that is, register bit INTEN of register MSDC_CFG is set to 0. It implies that software interrupt can be implemented by polling register bit INT of register MSDC_STA and this register. However, if hardware interrupt is desired, be sure to clear the register before setting up register bit INTEN of register MSDC_CFG to 1, or undesired hardware interrupt arisen from the previous interrupt status may take place.
A013000C	**MSDC_PS**	32	**SD memory card pin status register** Used for card detection. When the memory card controller and system are powered on, the power for the memory card will still be off unless the power is supplied by the PMIC. Meanwhile, the pad for card detection defaults to pull down when the system is powered on. The scheme of card detection for MS is the same as that for SD. For detecting card insertion, first pull up the INS pin and then enable card detection and the input pin at the same time. After 32 cycles of controller clock, the status of pin changes will emerge. To detect card removal, simply keep enabling card detection and the input pin.
A0130010	**MSDC_DAT**	32	**SD memory card controller data register** Reads/Writes data from/to FIFO inside SD controller. Data access unit: 32 bits
A0130014	**MSDC_IOCON**	32	**SD memory card controller IO control register** Specifies output driving capability and slew rate of IO pads for MSDC. The reset value is suggested setting. If the output driving capability of pins DAT0, DAT1, DAT2 and DAT3 is too large, it is possible to arise ground bounce and thus result in glitch on SCLK. The actual driving current will depend on the PAD type selected for the chip.
A0130018	**MSDC_IOCON1**	32	**SD memory card controller IO control register 1**
A0130020	**SDC_CFG**	32	**SD memory card controller configuration register** Configures the SD memory card controller when it is configured as the host of SD. If the controller is configured as the host of memory stick, the contents of the register will have no impact on the operation of the

Address	Name	Width	Register function
			controller. *Note: SDC_CFG[31:16] can be accessed by 16-bit APB bus access.*
A0130024	SDC_CMD	32	**SD memory card controller command register** Defines a SD memory card command and its attribute. Before the SD controller issues a transaction onto the SD bus, application shall specify other relative settings such as argument for command. After application writes the register, the SD controller will issue the corresponding transaction onto SD serial bus. If the command is GO_IDLE_STATE, the controller will have serial clock on SD bus run 128 cycles before issuing the command.
A0130028	SDC_ARG	32	**SD memory card controller argument register** Contains argument of the SD memory card command.
A013002C	SDC_STA	32	**SD memory card controller status register** Contains various statuses of SD controller as the controller is configured as the host of SD memory card.
A0130030	SDC_RESP0	32	**SD memory card controller response register 0** Contains parts of the last SD memory card bus response. See descriptions of register field SDC_RESP3.
A0130034	SDC_RESP1	32	**SD memory card controller response register 1** Contains parts of the last SD memory card bus response. See descriptions of register field SDC_RESP3.
A0130038	SDC_RESP2	32	**SD memory card controller response register 2** Contains parts of the last SD memory card bus response. See descriptions of register field SDC_RESP3.
A013003C	SDC_RESP3	32	**SD memory card controller response register 3** Contains parts of the last SD memory card bus response. Register fields SDC_RESP0, SDC_RESP1, SDC_RESP2 and SDC_RESP3 compose the last SD memory card bus response. For response of type R2, that is, response of the command ALL_SEND_CID, SEND_CSD and SEND_CID, only bit 127 to 0 of the response token are stored in register field SDC_RESP0, SDC_RESP1, SDC_RESP2 and SDC_RESP3. For responses of other types, only bit 39 to 8 of the response token are stored in register field SDC_RESP0.
A0130040	SDC_CMDSTA	32	**SD memory card controller command status register** Contains the status of SD controller during command execution and that of SD bus protocol after command execution when the SD controller is configured as the host of SD memory card. The register will also be used as the interrupt source. The register is cleared when

Address	Name	Width	Register function
			being read. Meanwhile, if interrupt is enabled and thus interrupt caused by the register is generated, reading the register will de-assert the interrupt.
A0130044	**SDC_DATSTA**	32	**SD memory card controller data status register** Contains the status of SD controller during data transfer on DAT line(s) when the SD controller is configured as the host of SD memory card. The register is also used as the interrupt source. The register is cleared when being read. Meanwhile, if interrupt is enabled and thus interrupt caused by the register is generated, reading the register will de-assert the interrupt.
A0130048	**SDC_CSTA**	32	**SD memory card status register** After commands with R1 and R1b response this register containing the status of the SD card, it will be used as the response interrupt source. In all register fields, logic high indicates error, and logic low indicates there is no error. The register is cleared when being read. Meanwhile, if interrupt is enabled and thus interrupt caused by the register is generated, reading the register will de-assert the interrupt.
A013004C	**SDC_IRQMASK0**	32	**SD memory card IRQ mask register 0** Contains parts of SD memory card interrupt mask register. See the descriptions of register SDC_IRQMASK1 for reference. The register masks interrupt sources from register SDC_CMDSTA and SDC_DATSTA. IRQMASK[3:0] is for SDC_CMDSTA, and IRQMASK[18:16] for SDC_DATSTA. Note that IRQMASK[18] masks SDC_DATSTA[9:2] together. '1' in some bits of the register masks the corresponding interrupt source with the same bit position. For example, if IRQMASK[0] is 1, then the interrupt source from register field CMDRDY of register SDC_CMDSTA will be masked. '0' in some bits does not cause interrupt mask on the corresponding interrupt source from register SDC_CMDSTA and SDC_DATSTA.
A0130050	**SDC_IRQMASK1**	32	**SD memory card IRQ mask register 1** Contains parts of SD memory card interrupt mask register. Registers SDC_IRQMASK1 and SDC_IRQMASK0 compose the SD memory card interrupt mask register. The register masks interrupt sources from register SDC_CSTA. '1' in some bits of the register masks the corresponding interrupt source with the same bit position. For example, if IRQMASK[63] is 1, then interrupt source from register field OUT_OF_RANGE of register SDC_ CSTA will be masked. '0' in some bit does not cause interrupt mask on the corresponding interrupt source from register SDC_CSTA.
A0130054	**SDIO_CFG**	32	**SDIO configuration register** Configures functions for SDIO.

Address	Name	Width	Register function
A0130058	**SDIO_STA**	32	**SDIO status register** Identifies if thers is SDIO interrupt during the interrupt period on data line.
A0130080	**CLK_RED**	32	**CLK latch configuration register** Configures the MSDC sample data/response clock. *Note: When MSDC_IOCON[19] = 1, the host will latch response; otherwise MSDC FSM will handle the response from PAD directly.*
A0130098	**DAT_CHECKSUM**	32	**MSDC Rx data checksum register** Competes the checksum value of Rx read data

A0130000 MSDC_CFG SD Memory Card Controller Configuration Register 04000020

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name					FIFOTHD						VDDPD	RCDEN	DIRQEN	PINEN	DMAEN	INTEN
Type					RW						RW	RW	RW	RW	RW	RW
Reset					0		0	0			0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	SCLKF								SCLKON	CRED	STDBY	CLKSRC		NOCRC	RST	MSDC
Type	RW								RW	RW	RW	RW		RW	W1C	RW
Reset	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0

Overview: For general configuration of the SD controller. Note that MSDC_CFG[31:16] can be accessed by 16-bit APB bus access.

Bit(s)	Mnemonic	Name	Description
27:24	**FIFOTHD**	FIFOTHD	**FIFO threshold** The register field determines when to issue a DMA request. For write transactions, DMA requests will be asserted if the number of free entries in FIFO are bigger than or equal to the value in the register field. For read transactions, DMA requests will be asserted if the number of valid entries in FIFO are bigger than or equal to the value in the register field. The register field must be set according to the setting of data transfer count in DMA burst mode. If single mode for DMA transfer is used, the register field shall be set to 0b0001. 0000: Invalid 0001: Threshold value is 1. 0010: Threshold value is 2. 0011~01111: ... 1000: Threshold value is 8. Others: Invalid
21	**VDDPD**	VDDPD	**Controls output pin VDDPD used for power saving** Output pin VDDPD controls the power for memory card. 0: Output pin VDDPD outputs logic low. The power for memory card will be turned off. 1: Output pin VDDPD outputs logic high. The power for memory card will be turned on.
20	**RCDEN**	RCDEN	**Controls output pin RCDEN used for card identification process when the controller is for SD memory card** Its output controls the pull-down resistor on the system board to

Bit(s)	Mnemonic	Name	Description
			connect to or disconnect from signal CD/DAT3. 0: The output pin RCDEN outputs logic low. 1: The output pin RCDEN outputs logic high.
19	**DIRQEN**	DIRQEN	**Enables data request interrupt** The register bit is used to control if data request is used as an interrupt source. 0: Data request is not used as an interrupt source. 1: Data request is used as an interrupt source.
18	**PINEN**	PINEN	**Enables pin interrupt** The register bit is used to control if the pin for card detection is used as an interrupt source. 0: The pin for card detection is not used as an interrupt source. 1: The pin for card detection is used as an interrupt source.
17	**DMAEN**	DMAEN	**Enables DMA** *Note: If DMA capability is disabled, the application software must poll the status of register MSDC_STA to check on any data transfer request. If DMA is desired, the register bit must be set up before command register is written.* 0: DMA request induced by various conditions is disabled, no matter the controller is configured as the host of either SD memory card or memory stick. 1: DMA request induced by various conditions is enabled, no matter the controller is configured as the host of either SD memory card or memory stick.
16	**INTEN**	INTEN	**Enables interrupt** *Note: If interrupt capability is disabled, the application software must poll the status of register MSDC_STA to check on any interrupt request.* 0: Interrupt induced by various conditions is disabled, no matter the controller is configured as the host of either SD memory card or memory stick. 1: Interrupt induced by various conditions is enabled, no matter the controller is configured as the host of either SD memory card or memory stick.
15:8	**SCLKF**	SCLKF	**Controls clock frequency of serial clock on SD bus and denotes clock frequency of SD bus serial clock as fslave and clock frequency of the SD controller as fhost which is 98.3 or 96.2 MHz** *Note: The allowed maximum frequency of fslave is 49.15MHz.* While changing the clock rate, "1T clock period before change + 1T clock period after change" is required for HW signal to re-synchronize. 00000000b: fslave =(1/2)*fhost 00000001b: fslave = [1/(4*1)]*fhost 00000010b: fslave = [1/(4*2)]*fhost 00000011b: fslave = [1/(4*3)]*fhost 00000100b~11111110b: ... 11111111b: fslave = [1/(4*255)]*fhost
7	**SCLKON**	SCLKON	**Serial clock always on** For debugging. 0: Serial clock not always on 1: Serial clock always on
6	**CRED**	CRED	**Rising edge data** The register bit is used to determine the serial data input is latched at the falling edge or rising edge of the serial clock. The default

Bit(s)	Mnemonic	Name	Description
			setting is at the rising edge. If the serial data have bad timing, set the register bit to 1. When the memory card has bad timing on returned read data, set the register bit to 1. 0: Serial data input is latched at the rising edge of serial clock. 1: Serial data input is latched at the falling edge of serial clock.
5	**STDBY**	STDBY	**Standby mode** If the module is powered down, operating clock to the module will be stopped. At the same time, the clock to card detection circuitry will also be stopped. If detection on memory card insertion and removal is desired, write 1 to the register bit. If interrupt for detection on memory card insertion and removal is enabled, the interrupt will take place whenever the memory is inserted or removed. 0: Standby mode is disabled. 1: Standby mode is enabled.
4:3	**CLKSRC**	CLKSRC	**Specifies which clock is used as source clock of memory card** 00 : MPLL/5.5MHz clock 01 : MPLL/7MHz dock (this divider is default off, before switch controller source clock to this clock, you should switch controller source clock to 26Mhz first, and then enable divider MPLL/7, after a moment you can switch controller source clock to MPLL/ 7 to void source dock glitch. For detail setting please refer to confgsys document) 10 : MPLL/8MHz dock 11 : MPLL/10MHz dock For phone 00 : 94.5MHz dock Need to keep BT_APP_DIV_EN= 1'b0 in CLK_CONDA[15]. 01 : 74.3MHz dock NOTE: Need to set POWERFUL_DIV_EN1 = 1'b1 first in CLK_CONDA[10]. 10 : 65MHz dock 11 : Forbidden For BT app. 00 : Forbidden 01 : 89.1MHz dock NOTE: Need to set POWERFUL_DIV_EN1 = 1'b1 first in CLK_CONDA[10]. 10 78MHz dock 11 : 62.4MHz clock NOTE: Need to set POWERFUL_DIV_EN2 = 1'b1 first in CLK_CONDA[9].
2	**NOCRC**	NOCRC	**Disable CRC** '1' indicates data transfer without CRC is desired. For write data block, the data are transmitted without CRC. For read data block, CRC will not be checked. For testing purpose. 0: Data transfer with CRC is desired. 1: Data transfer without CRC is desired.
1	**RST**	RST	**Software reset** Writing 1 to the register bit will cause internal synchronous reset of SD controller but will not reset register settings, RST should only be set when RST equal to 0. 0: Read 0 stands for the reset process is finished. 1: Write 1 to reset SD controller.
0	**MSDC**	MSDC	**Configures the controller as SD memory card mode** CLK/CMD/DAT line is pulled low when SD memory card mode is disable. 0: Disable SD memory card 1: Enable SD memory card

A0130004 MSDC_STA SD Memory Card Controller Status Register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	BUSY	FIFOCLR							FIFOCNT				INT	DRQ	BE	BF
Type	R	W1C							RO				RO	RO	RO	RO
Reset	0	0							0	0	0	0	0	0	0	0

Overview: The register contains the status of FIFO, interrupts and data requests.

Bit(s)	Mnemonic	Name	Description
15	**BUSY**	BUSY	**Status of the controller** If the controller is in busy state, the register bit will be 1; otherwise 0. 0: The controller is in busy state. 1: The controller is in idle state.
14	**FIFOCLR**	FIFOCLR	**Clears FIFO** **Writing 1 to the register bit will cause the content of FIFO clear and reset the status of FIFO controller.** 0: Read 0 stands for the FIFO clear process is finished. 1: Write 1 to clear the content of FIFO clear and reset the status of FIFO controller.
7:4	**FIFOCNT**	FIFOCNT	**FIFO count** The register field shows how many valid entries are there in FIFO. 0000: 0 valid entry in FIFO 0001: 1 valid entry in FIFO 0010: 2 valid entries in FIFO 0011~0111: ... 1000: 8 valid entries in FIFO
3	**INT**	INT	**Indicates if there is any interrupt existing** When there is interrupt existing, the register bit will still be active even if register bit INTEN in register MSDC_CFG is disabled. The SD controller can interrupt MCU by issuing interrupt request to the interrupt controller, or the software/application will poll the register endlessly to check if there is any interrupt request existing in the SD controller. When register bit INTEN in register MSDC_CFG is disabled, the second method is used. For read commands, it is possible that time-out error takes place. The software can read the status register to check if the time-out error takes place without OS time tick support or data request asserted. *Note: The register bit will be cleared when register MSDC_INT is read.* 0: No interrupt request existing. 1: Interrupt request exists.
2	**DRQ**	DRQ	**Indicates if any data transfer is required** When a data transfer is required, the register bit will still be active even if register bit DIRQEN in register MSDC_CFG is disabled. Data transfer can be achieved by DMA channel alleviating MCU loading, or by polling the register bit to check if any data transfer is requested. When register bit DIRQEN in register MSDC_CFG is disabled, the second method is used. 0: No DMA request existing. 1: DMA request exists.
1	**BE**	BE	**Indicates if FIFO in SD controller is empty** 0: FIFO in SD controller is not empty.

Bit(s)	Mnemonic	Name	Description
			1: FIFO in SD controller is empty.
0	**BF**	BF	**Indicates if FIFO in SD controller is full** 0: FIFO in SD controller is not full. 1: FIFO in SD controller is full.

A0130008 MSDC_INT SD Memory Card Controller Interrupt Register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									**SDIOIRQ**	**SDR1BIRQ**		**SDMCIRQ**	**SDDATIRQ**	**SDCMDIRQ**	**PINIRQ**	**DIRQ**
Type									RC	RC		RC	RC	RC	RC	RC
Reset									0	0		0	0	0	0	0

Overview: The register contains the status of interrupts. Note that the register still show status of interrupt even though interrupt is disabled, that is, register bit INTEN of register MSDC_CFG is set to 0. It implies that software interrupt can be implemented by polling register bit INT of register MSDC_STA and this register. However, if hardware interrupt is desired, be sure to clear the register before setting register bit INTEN of register MSDC_CFG to 1, or undesired hardware interrupt arisen from the previous interrupt status may take place.

Bit(s)	Mnemonic	Name	Description
7	**SDIOIRQ**	SDIOIRQ	**SDIO interrupt** The register bit indicates if there is any interrupt for SDIO existing. Whenever an interrupt for SDIO exists, the register bit will be set to 1 if the interrupt is enabled. It will be reset when the register is read. 0: No SDIO interrupt 1: Interrupt for SDIO exists.
6	**SDR1BIRQ**	SDR1BIRQ	**SD R1b response interrupt** The register bit will be active when a SD command with R1b response is finished and the DAT0 line is transited from busy to idle state. Single block write commands with R1b response will cause interrupt when the command is completed either successfully or with CRC error. However, multi-block write commands with R1b response do not cause interrupt because multi-block write commands are always stopped by STOP_TRANS commands. STOP_TRANS commands (with R1b response) behind multi-block write commands will cause interrupt. Single block read command with R1b response will cause interrupt when the command is completed, but multi-block read commands do not. *Note: STOP_TRANS commands (with R1b response) behind multi-block read commands will cause interrupt.* 0: No interrupt for SD R1b response. 1: Interrupt for SD R1b response exists.
4	**SDMCIRQ**	SDMCIRQ	**SD memory card interrupt** The register bit indicates if there is any interrupt for SD memory card existing. Whenever an interrupt for SD memory card exists, i.e. any bit in register SDC_CSTA is active, the register bit will be set to 1 if interrupt is enabled. It will be reset when the register is read. *Note: This bit will not trigger MSDC hardware interrupt.* 0: No SD memory card interrupt

Bit(s)	Mnemonic	Name	Description
			1: SD memory card interrupt exists.
3	**SDDATIRQ**	SDDATIRQ	**SD bus DAT interrupt** The register bit indicates if there is any interrupt for SD DAT line existing. Whenever interrupt for SD DAT line exists, i.e. any bit in register SDC_ DATSTA is active, the register bit will be set to 1 if interrupt is enabled. It will be reset when the register is read. 0: No SD DAT line interrupt 1: SD DAT line interrupt exists.
2	**SDCMDIRQ**	SDCMDIRQ	**SD bus CMD interrupt** The register bit indicates if there is any interrupt for SD CMD line existing. Whenever interrupt for SD CMD line exists, i.e. any bit in the register SDC_CMDSTA is active, the register bit will be set to 1 if interrupt is enabled. It will be reset when the register is read. 0: No SD CMD line interrupt 1: SD CMD line interrupt exists.
1	**PINIRQ**	PINIRQ	**Pin change interrupt** The register bit indicates if there is any interrupt for memory card insertion/removal existing. Whenever the memory card is inserted or removed and card detection interrupt is enabled, i.e. register bit PINEN in register MSDC_CFG is set to 1, the register bit will be set to 1. It will be reset when the register is read. 0: Otherwise 1: Card is inserted or removed.
0	**DIRQ**	DIRQ	**Data request interrupt** The register bit indicates if there is any interrupt for data request existing. Whenever data request exists and data request as an interrupt source is enabled, i.e. register bit DIRQEN in register MSDC_CFG is set to 1, the register bit will be active. It will be reset when being read. For software, data requests can be recognized by polling register bit DRQ or by data request interrupt. Data request interrupts will be generated every FIFOTHD data transfers. 0: No data request interrupt 1: Data request interrupt occurs.

A013000C MSDC_PS SD Memory Card Pin Status Register 00000008

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name								**CMD**	**DAT**							
Type								RO	RO							
Reset								0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**CDDEBOUNCE**											**PINCHG**	**PIN0**	**POEN0**	**PIEN0**	**CDEN**
Type	RW											RC	RO	RW	RW	RW
Reset	0	0	0	0								0	1	0	0	0

Overview: The register is used for card detection. When the memory card controller and system are powered on, , the power for the memory card will still be off unless the power is supplied by the PMIC. Meanwhile, the pad for card detection defaults to pull-down when the system is powered on. The scheme of card detection for MS is the same as that for SD. To detect card insertion, first pull up the INS pin and then enable card detection and the input pin at the same time. After 32 cycles of controller clock, the status of pin changes will emerge. To detect card removal, simply keep enabling card detection and the input pin.

Bit(s)	Mnemonic	Name	Description
24	**CMD**	CMD	**Memory card/SDIO card/MMC card command lines**
23:16	**DAT**	DAT	**Memory card/SDIO card/MMC card data lines**
15:12	**CDDEBOUNCE**	CDDEBOUNCE	**Specifies the time interval for card detection de-bounce** Default value: 0. It means the de-bounce interval is 32-cycle time at 32kHz. The interval can extend one cycle time at 32kHz by increasing the counter by 1.
4	**PINCHG**	PINCHG	**Pin change** The register bit indicates the status of card insertion/removal. If the memory card is inserted or removed, the register bit will be set to 1 no matter pin change interrupt is enabled or not. It will be cleared when the register is read. 0: Otherwise 1: Card is inserted or removed.
3	**PIN0**	PIN0	**Shows the value of input pin for card detection** 0: The value of input pin for card detection is logic low. 1: The value of input pin for card detection is logic high.
2	**POEN0**	POEN0	**Controls output of input pin for card detection** 0: Output of input pin for card detection is disabled. 1: Output of input pin for card detection is enabled.
1	**PIEN0**	PIEN0	**Controls input pin for card detection** 0: Input pin for card detection is disabled. 1: Input pin for card detection is enabled.
0	**CDEN**	CDEN	**Enables card detection** The register bit is used to enable or disable card detection. 0: Card detection is disabled. 1: Card detection is enabled.

A0130010 MSDC_DAT SD Memory Card Controller Data Register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DATA[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DATA[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: The register is used to read/write data from/to FIFO inside the SD controller. Data access unit: 32 bits.

Bit(s)	Mnemonic	Name	Description
31:0	**DATA**	DATA	**Reads/Writes data from/to FIFO inside SD controller** Data access unit: 32 bits

A0130014 MSDC_IOCON SD Memory Card Controller IO Control Register 010000C3

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name							SAMPLEDLY		FIXDLY		SAMPON	CRCDIS	CMDSEL	INTLH		DSW
Type							RW		RW		RW	RW	RW	RW		RW

Reset							0	1	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**CMDRE**					**HIGH_SPEED**	**DMABURST**		**SRCFG1**	**SRCFG0**	**ODCCFG1**			**ODCCFG0**		
Type	RW					RW	RW		RW	RW	RW			RW		
Reset	0					0	0	0	1	1	0	0	0	0	1	1

Overview: The register specifies the output driving capability and slew rate of IO pads for MSDC. The reset value is suggested setting. If the output driving capability of pins DAT0, DAT1, DAT2 and DAT3 is too large, it is possible to arise ground bounce and thus result in glitch on SCLK. The actual driving current depends on the PAD type selected for the chip.

Bit(s)	Mnemonic	Name	Description
25:24	**SAMPLEDLY**	SAMPLEDLY	**Used for SW to select the turn-around delay cycle between write data end bit and CRC status for SD card** 00: 0-T delay 01: 1-T delay 10: 2-T delay 11: 3-T delay
23:22	**FIXDLY**	FIXDLY	**Used for SW to select the delay cycle after clock fix high for the host controller to SD card** 00: 0-T delay 01: 1-T delay 10: 2-T delay 11: 3-T delay
21	**SAMPON**	SAMPON	**Data sample enabling always on** The bit is suggested to be set to 1 when the feedback clock is used and to 0 when the multiple phase clock is used. 0: Data sample enabling not always on 1: Data sample enabling always on
20	**CRCDIS**	CRCDIS	**Switches off data CRC check for SD read data** 0: CRC check is on. 1: CRC check is off.
19	**CMDSEL**	CMDSEL	**Determines whether the host should delay 1-T to latch response from card** 0: Host latches response without 1-T delay 1: Host latches response with 1-T delay.
18:17	**INTLH**	INTLH	**Selects latch timing for SDIO multi-block read interrupt** 00: Host latches INT at the second backend clock after the end bit of the current data block from card is received. (Default) 01: Host latches INT at the first backend clock after the end bit of the current data block from card is received. 10: Host latches INT at the second backend clock after the end bit of the current data block from card is received. 11: Host latches INT at the third backend clock after the end bit of the current data block from card is received.
16	**DSW**	DSW	**Determines whether the host should latch data with 1-T delay or not** For SD card, this bit is suggested to be 0. For MSPRO cards, it is suggested to be 1. 0: Host latches the data with 1-T delay. 1: Host latches the data without 1-T delay.
15	**CMDRE**	CMDRE	**Determines whether the host should latch response token (sent from card on CMD line) at rising edge or falling edge of serial**

Bit(s)	Mnemonic	Name	Description
			clock (T.B.D this bit is un-useful) 0: Host latches response at rising edge of serial clock. 1: Host latches response at falling edge of serial clock.
10	**HIGH_SPEED**	HIGH_SPEED	**For high-speed mode when internal sample clock is used** High-speed mode means the SD/MMC serial bus clock rate is bigger than 25MHz. The default speed mode means that the SD/MMC serial bus clock rate is bigger than 25MHz. 0: Default speed 1: High speed
9:8	**DMABURST**	DMABURST	**Used for SW to select burst type when data are transferred by DMA** *Note: Only single mode can support non-4N bytes data transfer in read operation.* 00: Single mode 01: 4-beat incrementing burst 10: 8-beat incrementing burst 11: Reserved
7	**SRCFG1**	SRCFG1	**Output driving capability for pins DAT0, DAT1, DAT2 and DAT3** 0: Fast slew rate 1: Slow slew rate
6	**SRCFG0**	SRCFG0	**Output driving capability for pins CMD/BS and SCLK** 0: Fast slew rate 1: Slow slew rate
5:3	**ODCCFG1**	ODCCFG1	**Output driving capability for pins DAT0, DAT1, DAT2 and DAT3** 000: 4mA 001: 8mA 010: 12mA 011: 16mA
2:0	**ODCCFG0**	ODCCFG0	**Output driving capability for pins CMD/BS and SCLK** 000: 4mA 001: 8mA 010: 12mA 011: 16mA

A0130018 MSDC_IOCON1 SD Memory Card Controller IO Control Register 1 00022022

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name		**PRCFG_CK**	**PRVAL_CK**			**PRCFG_CM**	**PRVAL_CM**			**PRCFG_DA**	**PRVAL_DA**					
Type		RW	RW			RW	RW			RW	RW					
Reset		0	1	0		0	0	0		0	1	0				

Bit(s)	Mnemonic	Name	Description
14	**PRCFG_CK**	PRCFG_CK	**Pull-up/down register configuration for pin CK** Default value: 0 0: Pull-up resistor in the I/O pad of pin CK is enabled. 1: Pull-down resistor in the I/O pad of pin CK is enabled.

Bit(s)	Mnemonic	Name	Description
13:12	**PRVAL_CK**	PRVAL_CK	**Pull-up/down register value for pin CLK** Default value: 10 00: Pull-up/down resistor in the I/O pad of pin CLK are all disabled. 01: Pull-up/down resistor in the I/O pad of pin CLK value is 47k. 10: Pull-up/down resistor in the I/O pad of pin CLK value is 47k. 11: Pull-up/down resistor in the I/O pad of pin CLK value is 23.5k.
10	**PRCFG_CM**	PRCFG_CM	**Pull-up/down register configuration for pin CM** Default value is 0. 0: Pull-up resistor in the I/O pad of pin CM is enabled. 1: Pull-down resistor in the I/O pad of pin CM is enabled.
9:8	**PRVAL_CM**	PRVAL_CM	**Pull-up/down register value for pin CMD/BS** Default value: 00 00: Pull-up/down resistor in the I/O pad of pin CMD/BS are all disabled. 01: Pull-up/down resistor in the I/O pad of pin CMD/BS value is 47k. 10: Pull-up/down resistor in the I/O pad of pin CMD/BS value is 47k. 11: Pull-up/down resistor in the I/O pad of pin CMD/BS value is 23.5k.
6	**PRCFG_DA**	PRCFG_DA	**Pull-up/down register configuration for pin DAT0, DAT1, DAT2 and DAT3** Default value: 0 0: Pull-up resistor in the I/O pad of pin DAT is enabled. 1: Pull-down resistor in the I/O pad of pin DAT is enabled.
5:4	**PRVAL_DA**	PRVAL_DA	**Pull-up/down register value for pin DAT0, DAT1, DAT2 and DAT3** Default value: 10 00: Pull-up/ down resistor in the I/O pad of pin DAT are all disabled. 01: Pull-up/down resistor in the I/O pad of pin DAT value is 47k. 10: Pull-up/down resistor in the I/O pad of pin DAT value is 47k. 11: Pull-up/down resistor in the I/O pad of pin DAT value is 23.5k.

A0130020 **SDC_CFG** **SD Memory Card Controller Configuration Register** **00008000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**DTOC**								**WDOD**				**SDIO**		**MDLEN**	**SIEN**
Type	RW								RW				RW		RW	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0		0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**BSYDLY**				**BLKLEN**											
Type	RW				RW											
Reset	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: The register is used to configure the SD memory card controller when it is configured as the host of SD. If the controller is configured as the host of memory stick, the contents of the register will have no impact on the operation of the controller. Note that SDC_CFG[31:16] can be accessed by 16-bit APB bus access.

Bit(s)	Mnemonic	Name	Description
31:24	**DTOC**	DTOC	**Data time-out counter** The period from finish of the initial host read command or the last read data block in a multiple block read operation to the start bit of the next read data block requires at least two serial clock cycles. The counter is used to extend the period (Read Data Access Time) in unit

Bit(s)	Mnemonic	Name	Description
			of 65,536 serial clock. See the register field descriptions of register bit RDINT for reference. 00000000: Extend 65,536 more serial clock cycles 00000001: Extend 65,536x2 more serial clock cycles 00000010: Extend 65,536x3 more serial clock cycles 00000011~11111110: ... 11111111: Extend 65,536x 256 more serial clock cycles
23:20	**WDOD**	WDOD	**Write data output delay** The period from finish of the response for the initial host write command or the last write data block in a multiple block write operation to the start bit of the next write data block requires at least two serial clock cycles. The register field is used to extend the period (Write Data Output Delay) in unit of one serial clock. 0000: No extension 0001: Extend 1 more serial clock cycle 0010: Extend 2 more serial clock cycles 0011~1110: ... 1111: Extend 15 more serial clock cycle
19	**SDIO**	SDIO	**Enables SDIO** 0: Disable SDIO mode 1: Enable SDIO mode
17	**MDLEN**	MDLEN	**Enables multiple data line** The register can be enabled only when SD memory card is applied and detected by software application. It is the responsibility of the application to program the bit correctly when an multi-media card is applied. If an multi-media card is applied and 4-bit data line is enabled, the 4 bits will be output every serial clock. Therefore, data integrity will fail. 0: Disable 4-bit data line 1: Enable 4-bit data line
16	**SIEN**	SIEN	**Enables serial interface** It should be enabled as soon as possible before any command. 0: Disable serial interface for SD 1: Enable serial interface for SD
15:12	**BSYDLY**	BSYDLY	**Only valid for the commands with R1b response** If the command has a response of R1b type, the SD controller must monitor the data line 0 for card busy status from the bit time that is two serial clock cycles after the command end bit to check if the operation in SD memory card have finished. The register field is used to expand the time between the command end bit and end of detection period to detect card busy status. If time is up and there is no card busy status on data line 0, the controller will abandon the detection. 0000: No extension 0001: Extend 1 more serial clock cycle 0010: Extend 2 more serial clock cycles 0011~1110: ... 1111: Extend 15 more serial clock cycle
11:0	**BLKLEN**	BLKLEN	**Block length** The register field is used to define the length of one block in unit of byte in a data transaction. The maximum value of block length is 2,048 bytes. 000000000000: Reserved 000000000001: Block length is 1 byte. 000000000010: Block length is 2 bytes.

Bit(s)	Mnemonic	Name	Description
			000000000011~011111111110: ... 011111111111: Block length is 2,047 bytes. 100000000000: Block length is 2,048 bytes.

A0130024 SDC_CMD SD Memory Card Controller Command Register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																CMDFAIL
Type																RW
Reset																0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	INTC	STOP	RW	DTYPE		IDRT	RSPTYP			BREAK	CMD					
Type	RW	RW	RW	RW		RW	RW			RW	RW					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: The register defines a SD memory card command and its attribute. Before the SD controller issues a transaction onto the SD bus, application shall specify other relative settings such as argument for command. After application writes the register, the SD controller will issue the corresponding transaction onto the SD serial bus. If the command is GO_IDLE_STATE, the controller will have serial clock on SD bus run 128 cycles before issuing the command.

Bit(s)	Mnemonic	Name	Description
16	**CMDFAIL**	CMDFAIL	**If 4-bit SDIO mode is enabled and when CMD/DAT error occurrs, set this bit to select whether to "wait stop command" or "wait data state machine idle".** 0: Wait stop command 1: Wait data state machine idle
15	**INTC**	INTC	**Indicates if the command is GO_IRQ_STATE** If the command is GO_IRQ_STATE, the period between command token and response token will not be limited. 0: The command is not GO_IRQ_STATE. 1: The command is GO_IRQ_STATE.
14	**STOP**	STOP	**Indicates if the command is a stop transmission command** 0: The command is not a stop transmission command. 1: The command is a stop transmission command.
13	**RW**	RW	**Defines the command is a read command or write command** The register bit is valid only when the command causes a transaction with data token. 0: The command is a read command. 1: The command is a write command.
12:11	**DTYPE**	DTYPE	**Defines data token type for the command** 00: No data token for the command 01: Single block transaction 10: Multiple block transaction, i.e. the command is a multiple block read or write command. 11: Stream operation. It can only be used when an multi-media card is applied.
10	**IDRT**	IDRT	**Identification response time** The register bit indicates if the command has a response with NID (i.e. 5 serial clock cycles as defined in SD Memory Card

Bit(s)	Mnemonic	Name	Description
			Specification Part 1 Physical Layer Specification version 1.0) response time. The register bit is valid only when the command has a response token. Thus the register bit must be set to 1 for CMD2 (ALL_SEND_CID) and ACMD41 (SD_APP_OP_CMD). 0: Otherwise 1: The command has a response with NID response time.
9:7	**RSPTYP**	RSPTYP	**Defines response type for the command** For commands with R1 and R1b response, register SDC_CSTA (not SDC_STA) updates after response token is received. This register SDC_CSTA contains the status of the SD, and it can be used as a response interrupt source. *Note: If CMD7 is used with all 0's RCA, then RSPTYP must be "000". Command "GO_TO_IDLE" also has RSPTYP='000'.* 000: There is no response for the command, e.g. broadcast command without response and GO_INACTIVE_STATE command. 001: The command has R1 response. R1 response token is 48-bit. 010: The command has R2 response. R2 response token is 136-bit. 011: The command has R3 response. Even though R3 is 48-bit response, it does not contain CRC checksum. 100: The command has R4 response. R4 response token is 48-bit. (only for MMC) 101: The command has R5 response. R5 response token is 48-bit. (only for MMC) 110: The command has R6 response. R6 response token is 48-bit. 111: The command has R1b response. If the command has a response of R1b type, SD controller must monitor the data line 0 for card busy status from the bit time that is 2 or 4 serial clock cycles after the command end bit to check if the operation in SD memory card has finished. There are two cases for detection of card busy status. The first case is that the host stops the data transmission during an active write data transfer. The card will assert busy signal after the stop transmission command end bit followed by 4 serial clock cycles. The second case is that the card is in idle state or receiving a stop transmission command between data blocks when multiple block write command is in progress. The register bit will be valid only when the command has a response token.
6	**BREAK**	BREAK	**Aborts pending MMC GO_IRQ_MODE command** It is only valid for a pending GO_IRQ_MODE command waiting for MMC interrupt response. 0: Other fields are valid. 1: Break a pending MMC GO_IRQ_MODE command in the controller. Other fields are invalid.
5:0	**CMD**	CMD	**SD memory card command** Total 6 bits.

A0130028 **SDC_ARG** **SD Memory Card Controller Argument Register** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**ARG[31:16]**															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**ARG[15:0]**															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: The register contains the argument of the SD memory card command.

Bit(s)	Mnemonic	Name	Description
31:0	**ARG**	ARG	Contains argument of the SD memory card command.

A013002C SDC_STA SD Memory Card Controller Status Register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**WP**											**FEDATBUSY**	**FECMDBUSY**	**BEDATBUSY**	**BECMDBUSY**	**BESDCBUSY**
Type	RO											RO	RO	RO	RO	RO
Reset	0											0	0	0	0	0

Overview: The register contains various statuses of SD controller as the controller is configured as the host of SD memory card.

Bit(s)	Mnemonic	Name	Description
15	**WP**	WP	**Detects the status of write protection switch on SD memory card** The register bit shows the status of write protection switch on SD memory card. There is no default reset value. Pin WP (Write Protection) is only useful when the controller is configured for SD memory card. 1: Write protection switch on, i.e. memory card is desired to be write-protected. 0: Write protection switch off, i.e. memory card is writable.
4	**FEDATBUSY**	FEDATBUSY	**Indicates if there is any transmission going on DAT line on SD bus** This bit indicates directly the CMD line at card clock domain. For those commands without data but still involving DAT line, the register bit is useless. For example, if an erase command is issued, checking if the register bit is 0 before issuing the next command with data will not guarantee that the controller is idle. In this case, use register bit BESDCBUSY. 0: No transmission is going on DAT line on SD bus. 1: There is transmission going on DAT line on SD bus.
3	**FECMDBUSY**	FECMDBUSY	**Indicates if there is any transmission going on CMD line on SD bus** This bit indicates directly the CMD line at card clock domain. 0: No transmission is going on CMD line on SD bus. 1: There is transmission going on CMD line on SD bus.
2	**BEDATBUSY**	BEDATBUSY	**Indicates if there is any transmission going on DAT line on SD bus** 0: Backend SDC controller gets the info that no transmission is going on DAT line on SD bus. 1: Backend SDC controller gets the info that there is transmission going on DAT line on SD bus.
1	**BECMDBUS**	BECMDBUSY	**Indicates if there is any transmission going on CMD line on SD**

Bit(s)	Mnemonic	Name	Description
	Y		**bus** This bit shows backend controller's CMD busy state. The busy state is synced from card clock domain to bus clock domain. 0: Backend SDC controller gets the info that no transmission is going on CMD line on SD bus. 1: Backend SDC controller gets the info that there is transmission going on CMD line on SD bus.
0	**BESDCBUSY**	BESDCBUSY	**Indicates if SD controller is busy, i.e. is there any transmission going on CMD or DAT line on SD bus** This bit shows backend controller's SDC busy state. The busy state is synced from card clock domain to bus clock domain. 0: Backend SD controller is idle. 1: Backend SD controller is busy.

A0130030 SDC_RESP0 SD Memory Card Controller Response Register 0 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**RESP[31:0][31:16]**															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**RESP[31:0][15:0]**															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: The register contains parts of the last SD memory card bus response. See descriptions of register field SDC_RESP3.

Bit(s)	Mnemonic	Name	Description
31:0	**RESP[31:0]**	RESP_31_0	

A0130034 SDC_RESP1 SD Memory Card Controller Response Register 1 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**RESP[63:32][31:16]**															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**RESP[63:32][15:0]**															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: The register contains parts of the last SD memory card bus response. See descriptions of register field SDC_RESP3.

Bit(s)	Mnemonic	Name	Description
31:0	**RESP[63:32]**	RESP_63_32	

A0130038 SDC_RESP2 SD Memory Card Controller Response Register 2 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RESP[95:64][31:16]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RESP[95:64][15:0]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: The register contains parts of the last SD memory card bus response. See descriptions of register field SDC_RESP3.

Bit(s)	Mnemonic	Name	Description
31:0	**RESP[95:64]**	RESP_95_64	

A013003C SDC_RESP3 SD Memory Card Controller Response Register 3 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RESP[127:96][31:16]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RESP[127:96][15:0]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: The register contains parts of the last SD memory card bus response. Register fields SDC_RESP0, SDC_RESP1, SDC_RESP2 and SDC_RESP3 compose the last SD memory card bus response. For response of type R2, i.e. response of commands ALL_SEND_CID, SEND_CSD and SEND_CID, only bit 127 to 0 of the response token are stored in register fields SDC_RESP0, SDC_RESP1, SDC_RESP2 and SDC_RESP3. For response of other types, only bit 39 to 8 of the response token are stored in register field SDC_RESP0.

Bit(s)	Mnemonic	Name	Description
31:0	**RESP[127:96]**	RESP_127_96	

A0130040 SDC_CMDSTA SD Memory Card Controller Command Status Register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														RSPCRCERR	CMDTO	CMDRDY
Type														RC	RC	RC
Reset														0	0	0

Overview: The register contains the status of SD controller during command execution and that of SD bus protocol after command execution when the SD controller is configured as the host of SD memory card. The register can also be used as an interrupt source. The register is cleared w hen being read. Meanw hile, if the interrupt is enabled and thus interrupt caused by the register is generated, reading the register w ill de-assert the interrupt.

Bit(s)	Mnemonic	Name	Description
2	**RSPCRCERR**	RSPCRCERR	**CRC error on CMD detected** '1' indicates the SD controller detects a CRC error after reading a response from the CMD line. 0: Otherw ise 1: SD controller detects a CRC error after reading a response from the CMD line.
1	**CMDTO**	CMDTO	**Time-out on CMD detected** '1' indicates the SD controller detects a time-out condition w hile waiting for a response on the CMD line. 0: Otherw ise 1: SD controller detects a timeout condition w hile w aiting for a response on the CMD line.
0	**CMDRDY**	CMDRDY	**For command without response, the register bit will be 1 once the command is completed on SD bus** For command w ith response, the register bit w ill be 1 w henever the command is issued onto the SD bus and its corresponding response is received w ithout CRC error. 0: Otherw ise 1: Command w ith/w ithout response is finished successfully w ithout CRC error.

A0130044 SDC_DATSTA SD Memory Card Controller Data Status Register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							DATCRCERR								DATTO	BLKDONE
Type							RC								RC	RC
Reset							0	0	0	0	0	0	0	0	0	0

Overview: The register contains the status of SD controller during data transfer on DAT line(s) w hen the SD controller is configured as the host of SD memory card. The register can be used as an interrupt source. The register is cleared w hen being read. Meanw hile, if the interrupt is enabled and thus interrupt caused by the register is generated, reading the register w ill de-assert the interrupt.

Bit(s)	Mnemonic	Name	Description
9:2	**DATCRCERR**	DATCRCERR	**CRC error on DAT detected** '1' indicates that the SD controller detects a CRC error for bit n after reading a block of data from the DAT line or SD signals a CRC error after w riting a block of data to the DAT line. 0: Otherw ise 1: SD controller detects a CRC error after reading a block of data from the DAT line or SD signals a CRC error after writing a block of data to the DAT line. *Note: n is 7 ~ 1 for 8-bits mode. Each bit is read and cleared*

Bit(s)	Mnemonic	Name	Description
			individually.
1	**DATTO**	DATTO	**Time-out on DAT detected** A '1' indicates that the SD controller detects a time-out condition while waiting for data token on the DAT line. 0: Otherwise 1: SD controller detects a time-out condition while waiting for data token on the DAT line.
0	**BLKDONE**	BLKDONE	**Indicates the status of data block transfer** 0: Otherwise 1: A data block is successfully transferred.

A0130048 SDC_CSTA SD Memory Card Status Register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**CSTA [31:0][31:16]**															
Type	RC															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**CSTA [31:0][15:0]**															
Type	RC															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: After commands with R1 and R1b respond this register containing the status of the SD card, it will be used as a response interrupt source. In all register fields, logic high indicates error, and logic low indicates no error. The register is cleared when being read. Meanwhile, if the interrupt is enabled and thus interrupt caused by the register is generated, reading the register will de-assert the interrupt.

Bit(s)	Mnemonic	Name	Description
31:0	**CSTA [31:0]**	CSTA_31_0	CSTA31: OUT_OF_RANGE. The command's argument is out of the allowed range for this card. CSTA30: ADDRESS_ERROR. A misaligned address that does not match the block length is used in the command. CSTA29: BLOCK_LEN_ERROR. The transferred block length is not allowed for this card, or the number of transferred bytes does not match the block length. CSTA28: ERASE_SEQ_ERROR. An error in the sequence of erase commands occurs. CSTA27: ERASE_PARAM. An invalid selection of write-blocks for erase occurs. CSTA26: WP_VIOLATION. Attempt to program a write-protected block. CSTA25: Reserved. Return to 0. CSTA24: LOCK_UNLOCK_FAILED. Set when a sequence or password error is detected in lock/unlock card command or if there is an attempt to access a locked card. CSTA23: COM_CRC_ERROR. The CRC check of the previous command fails. CSTA22: ILLEGAL_COMMAND. Command not legal for the card state. CSTA21: CARD_ECC_FAILED. Card internal ECC is applied but fails to correct the data. CSTA20: CC_ERROR. Internal card controller error.

CSTA19: ERROR. A general or unknown error occurs during the operation.

CSTA18: UNDERRUN. The card cannot sustain data transfer in stream read mode.

CSTA17: OVERRUN. The card cannot sustain data programming in stream write mode.

CSTA16: CID/CSD_OVERWRITE. It can be either one of the following errors: 1) The CID register has been already written and cannot be overwritten; 2) The read-only section of the CSD does not match the card; 3) An attempt to reverse the copy (set as the original) or permanent WP (unprotected) bits is made.

CSTA[15: 4]: Reserved. Return to 0.

CSTA3: AKE_SEQ_ERROR. Error in the sequence of authentication process

CSTA[2: 0]: Reserved. Return to 0.

A013004C SDC_IRQMASK0 SD Memory Card IRQ Mask Register 0 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	IRQMASK [31:0][31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	IRQMASK [31:0][15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: The register contains parts of SD memory card interrupt mask register. See the register descriptions of register SDC_IRQMASK1 for reference. The register masks interrupt sources from register SDC_CMDSTA and SDC_DATSTA. IRQMASK[3:0] is for SDC_CMDSTA, and IRQMASK[18:16] for SDC_DATSTA. Note that IRQMASK[18] masks SDC_DATSTA[9:2] together. '1' in some bits of the register masks the corresponding interrupt source with the same bit position. For example, if IRQMASK[0] is 1, then the interrupt source from register field CMDRDY of register SDC_CMDSTA will be masked. '0' in some bits does not cause interrupt mask on the corresponding interrupt source from registers SDC_CMDSTA and SDC_DATSTA.

Bit(s)	Mnemonic	Name	Description
31:0	IRQMASK [31:0]	IRQMASK_31_0	

A0130050 SDC_IRQMASK1 SD Memory Card IRQ Mask Register 1 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	IRQMASK [63:32][31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	IRQMASK [63:32][15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: The register contains parts of SD memory card interrupt mask register. Registers SDC_IRQMASK1 and SDC_IRQMASK0 compose the SD memory card interrupt mask register. The register masks interrupt sources from register SDC_CSTA. '1' in some bits of the register masks the corresponding interrupt source with the same bit position. For example, if IRQMASK[63] is 1, then the interrupt source from register field OUT_OF_RANGE of register SDC_ CSTA will be masked. '0' in some bits does not cause interrupt mask on the corresponding interrupt source from register SDC_CSTA.

Bit(s)	Mnemonic	Name	Description
31:0	**IRQMASK [63:32]**	IRQMASK_63_32	

A0130054 SDIO_CFG SDIO Configuration Register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name											**DISSEL**		**INTCSEL**	**DSBSEL**		**INTEN**
Type											RW		RW	RW		RW
Reset											0		0	0		0

Overview: The register is used to configure functions for SDIO.

Bit(s)	Mnemonic	Name	Description
5	**DISSEL**	DISSEL	**Selects data block interrupt source** 0: The host detects SDIO interrupt during interrupt period between two data blocks of multiple block data access. 1: The host ignores SDIO interrupt during interrupt period between two data blocks of multiple block data access.
3	**INTCSEL**	INTCSEL	**Selects interrupt control** 0: The host detects DAT1 low as SDIO interrupt. 1: The host detects DAT3/DAT2/DAT1/DAT0 4'b1101 as SDIO interrupt.
2	**DSBSEL**	DSBSEL	**Selects data block start bit** 0: Use data line 0 as start bit of data block. Other data lines are ignored. 1: Start bit of a data block is received only when all data line 0-3 become low.
0	**INTEN**	INTEN	**Enables interrupt for SDIO** 0: Disable 1: Enable

A0130058 SDIO_STA SDIO Status Register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																**IRQ**
Type																RO
Reset																0

Overview: This register is used to identify if there is SDIO interrupt during the interrupt period on data line.

Bit(s)	Mnemonic	Name	Description
0	**IRQ**	IRQ	**SDIO interrupt exists on the data line.** For example, when in the interrupt period or the 1 -bit data line mode and DAT1/5 goes low from high, this bit will become 1 from 0. If DAT1/5 goes high from low, this bit will become 0 from 1. 0: There is no SDIO interrupt existing on the data line. 1: There is SDIO interrupt existing on the data line.

A0130080 CLK_RED CLK Latch Configuration Register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name			**CMD_RED**													
Type			RW													
Reset			0													
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name			**DAT_RED**						**CLKPAD_RED**	**CLK_LATCH**						
Type			RW						RW	RW						
Reset			0						0	0						

Overview: The register is used to configure the MSDC sample data/response clock. Note that only when MSDC_IOCON[19] = 1 will the host latch response; otherwise MSDC FSM will handle the response from PAD directly.

Bit(s)	Mnemonic	Name	Description
29	**CMD_RED**	CMD_RED	**Determines the command response from card output is latched at falling edge or rising edge of internal clock** Only effective when CLK_LATCH = 1 0: Internal clock rising edge to latch response 1: Internal clock falling edge to latch response
13	**DAT_RED**	DAT_RED	**Determines the input data from card output is latched at falling edge or rising edge of internal sample clock** Only effective when CLK_LATCH = 1 0: Internal clock rising edge to latch data 1: Internal clock falling edge to latch data
7	**CLKPAD_RED**	CLKPAD_RED	**Determines the input data from card is latched at falling edge or rising edge of the feedback clock from pad** The suggested setting is 0 when SD serial clock is lower than 25MHz. The suggestion setting is 1 when SD serial clock is higher than 25MHz. The suggestion setting is 0 for MMC card no matter the serial clock rate is high speed or default speed. Only effective when CLK_LATCH = 0. 0: Internal feedback clock rising edge to latch data/response 1: Internal feedback clock falling edge to latch data/response
6	**CLK_LATCH**	CLK_LATCH	**Determines which clock to latch data from card** The suggested setting is 1 if SCLKF in register field MSDC_CFG is 0x0. Otherwise, the suggested setting is 0. 0: Internal feedback clock is used to latch data/response from card.

Bit(s)	Mnemonic	Name	Description
			1: Internal clock is used to latch data/response from card.

A0130098 DAT_CHECKSUM MSDC Rx Data Checksum Register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DAT_CHECKSUM[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DAT_CHECKSUM[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: The register is used to compete the checksum value of Rx read data

Bit(s)	Mnemonic	Name	Description
31:0	**DAT_CHECKSUM**	DAT_CHECKSUM	The checksum algorithm is 32-bit XOR.

3.17 SD Memory Card Controller (MSDC1)

3.17.1 Introduction

The controller fully supports the SD memory card bus protocol as defined in SD Memory Card Specification Part 1 Physical Layer Specification version 2.0.

Furthermore, the controller also partially supports the SDIO card specification version 2.0. However, the controller can only be configured as the host of the SD memory card. Hereafter, the controller is also abbreviated as the SD controller. The following are the main features of the controller.

- Interface with MCU by APB bus
- 16/32-bit access on APB bus
- 16/32-bit access for control registers
- 32-bit access for FIFO
- Built-in 32 bytes FIFO buffers for transmit and receive, FIFO is shared for transmit and receive
- Built-in CRC circuit
- CRC generation can be disabled
- DMA supported
- Interrupt capabilities
- Data rate up to 48 Mbps in serial mode, 48x4 Mbps in parallel model, the module is targeted at 48 MHz operating clock
- Serial clock rate on SD bus is programmable
- Card detection capabilities during sleep mode
- Controllability of power for memory card
- Does not support SPI mode for SD memory card

- Does not support multiple SD memory cards

3.17.2 Overview

3.17.2.1 Pin Assignment

The following lists pins required for the SD memory card. Table 51 shows how the pins are shared. Note that all I/O pads have embedded both pull-up and pull-down resistors because they are shared by the SD memory card. The pull-down resistors for these pins can be used for power saving. If optimal pull-up or pull-down resistors are required on the system board, all embedded pull-up and pull-down resistors can be disabled by programming the corresponding control registers. The VDDPD pin is used for power saving. Power for the SD memory card can be shut down by programming the corresponding control register. The WP (Write Protection) pin is used to detect the status of the Write Protection Switch on the SD memory card.

Table 51. Sharing of pins for SD memory card controller

No.	Name	Type	MMC	SD	Description
1	SD_CLK	O	CLK	CLK	Clock
2	SD_DAT3	I/O/PP		CD/DAT3	Data Line [Bit 3]
3	SD_DAT0	I/O/PP	DAT0	DAT0	Data Line [Bit 0]
4	SD_DAT1	I/O/PP		DAT1	Data Line [Bit 1]
5	SD_DAT2	I/O/PP		DAT2	Data Line [Bit 2]
6	SD_CMD	I/O/PP	CMD	CMD	Command Or Bus State
7	SD_PWRON	O			VDD ON/OFF
8	SD_WP	I			Write Protection Switch in SD
9	SD_INS	I	VSS2	VSS2	Card Detection

3.17.2.2 Card Detection

For SD memory card, detection of card insertion/removal by hardware is supported, and a dedicated pin "INS" is used to perform card insertion and removal for SD. The pin "INS" will be connected to the pin "VSS2" of a SD connector (see **Figure 45**).

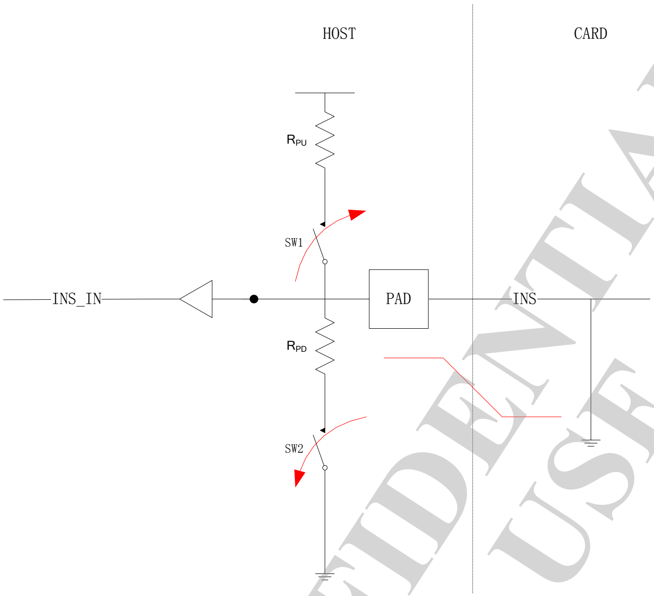


Figure 46. Card detection for SD memory card

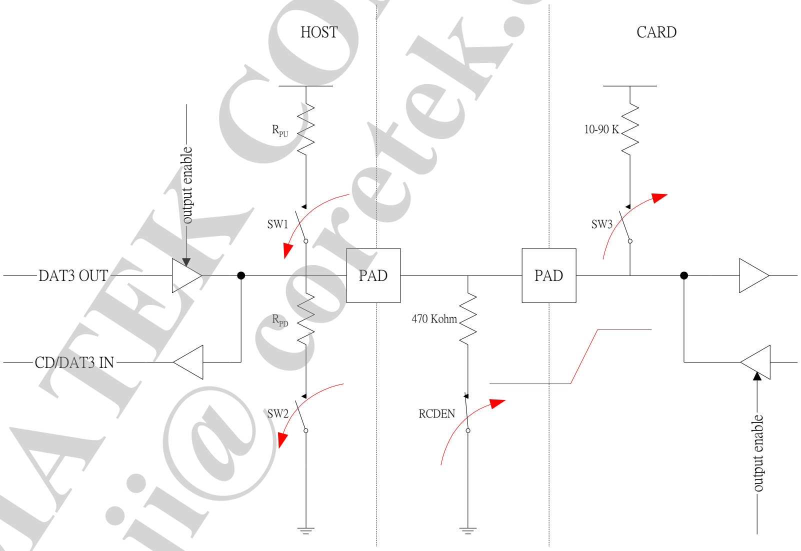


Figure 47. Card detection for SD memory card (Scheme 2)

3.17.3 Register Definition

Module name: MSDC1 base address: (+A0270000h)

Address	Name	Width	Register function
A0270000	**MSDC_CFG**	32	**SD memory card controller configuration register** For general configuration of the SD controller. *Note: MSDC_CFG[31:16] can be accessed by 16-bit APB bus access.*
A0270004	**MSDC_STA**	32	**SD memory card controller status register**

Address	Name	Width	Register function
			Contains the status of FIFO, interrupts and data requests.
A0270008	**MSDC_INT**	32	**SD memory card controller interrupt register** Contains the status of interrupts. Note that the register still shows the status of interrupt even though the interrupt is disabled, that is, register bit INTEN of register MSDC_CFG is set to 0. It implies that software interrupt can be implemented by polling register bit INT of register MSDC_STA and this register. However, if hardware interrupt is desired, be sure to clear the register before setting up register bit INTEN of register MSDC_CFG to 1, or undesired hardware interrupt arisen from the previous interrupt status may take place.
A027000C	**MSDC_PS**	32	**SD memory card pin status register** Used for card detection. When the memory card controller and system are powered on, the power for the memory card will still be off unless the power is supplied by the PMIC. Meanwhile, the pad for card detection defaults to pull down when the system is powered on. The scheme of card detection for MS is the same as that for SD. For detecting card insertion, first pull up the INS pin and then enable card detection and the input pin at the same time. After 32 cycles of controller clock, the status of pin changes will emerge. To detect card removal, simply keep enabling card detection and the input pin.
A0270010	**MSDC_DAT**	32	**SD memory card controller data register** Reads/Writes data from/to FIFO inside SD controller. Data access unit: 32 bits
A0270014	**MSDC_IOCON**	32	**SD memory card controller IO control register** Specifies output driving capability and slew rate of IO pads for MSDC. The reset value is suggested setting. If the output driving capability of pins DAT0, DAT1, DAT2 and DAT3 is too large, it is possible to arise ground bounce and thus result in glitch on SCLK. The actual driving current will depend on the PAD type selected for the chip.
A0270018	**MSDC_IOCON1**	32	**SD memory card controller IO control register 1**
A0270020	**SDC_CFG**	32	**SD memory card controller configuration register** Configures the SD memory card controller when it is configured as the host of SD. The register is used to configure the SD memory card controller when it is configured as the host of SD memory card. If the controller is configured as the host of memory stick, the contents of the register will have no impact on the operation of the controller. *Note: SDC_CFG[31:16] can be accessed by 16-bit APB bus access.*
A0270024	**SDC_CMD**	32	**SD memory card controller command register**

Address	Name	Width	Register function
			Defines a SD memory card command and its attribute. Before the SD controller issues a transaction onto the SD bus, application shall specify other relative settings such as argument for command. After application writes the register, the SD controller will issue the corresponding transaction onto SD serial bus. If the command is GO_IDLE_STATE, the controller will have serial clock on SD bus run 128 cycles before issuing the command.
A0270028	**SDC_ARG**	32	**SD memory card controller argument register** Contains argument of the SD memory card command.
A027002C	**SDC_STA**	32	**SD memory card controller status register** Contains various statuses of SD controller as the controller is configured as the host of SD memory card.
A0270030	**SDC_RESP0**	32	**SD memory card controller response register 0** Contains parts of the last SD memory card bus response. See descriptions of register field SDC_RESP3.
A0270034	**SDC_RESP1**	32	**SD memory card controller response register 1** Contains parts of the last SD memory card bus response. See descriptions of register field SDC_RESP3.
A0270038	**SDC_RESP2**	32	**SD memory card controller response register 2** Contains parts of the last SD memory card bus response. See descriptions of register field SDC_RESP3.
A027003C	**SDC_RESP3**	32	**SD memory card controller response register 3** Contains parts of the last SD memory card bus response. Register fields SDC_RESP0, SDC_RESP1, SDC_RESP2 and SDC_RESP3 compose the last SD memory card bus response. For response of type R2, that is, response of the command ALL_SEND_CID, SEND_CSD and SEND_CID, only bit 127 to 0 of the response token are stored in register field SDC_RESP0, SDC_RESP1, SDC_RESP2 and SDC_RESP3. For responses of other types, only bit 39 to 8 of the response token are stored in register field SDC_RESP0.
A0270040	**SDC_CMDSTA**	32	**SD memory card controller command status register** Contains the status of SD controller during command execution and that of SD bus protocol after command execution when the SD controller is configured as the host of SD memory card. The register will also be used as the interrupt source. The register is cleared when being read. Meanwhile, if interrupt is enabled and thus interrupt caused by the register is generated, reading the register will de-assert the interrupt.
A0270044	**SDC_DATSTA**	32	**SD memory card controller data status register** Contains the status of SD controller during data

Address	Name	Width	Register function
			transfer on DAT line(s) when the SD controller is configured as the host of SD memory card. The register is also used as the interrupt source. The register is cleared when being read. Meanwhile, if interrupt is enabled and thus interrupt caused by the register is generated, reading the register will de-assert the interrupt.
A0270048	**SDC_CSTA**	32	**SD memory card status register** After commands with R1 and R1b response this register containing the status of the SD card, it will be used as the response interrupt source. In all register fields, logic high indicates error, and logic low indicates there is no error. The register is cleared when being read. Meanwhile, if interrupt is enabled and thus interrupt caused by the register is generated, reading the register will de-assert the interrupt.
A027004C	**SDC_IRQMASK0**	32	**SD memory card IRQ mask register 0** Contains parts of SD memory card interrupt mask register. See the descriptions of register SDC_IRQMASK1 for reference. The register masks interrupt sources from register SDC_CMDSTA and SDC_DATSTA. IRQMASK[3:0] is for SDC_CMDSTA, and IRQMASK[18:16] for SDC_DATSTA. Note that IRQMASK[18] masks SDC_DATSTA[9:2] together. ‘1’ in some bits of the register masks the corresponding interrupt source with the same bit position. For example, if IRQMASK[0] is 1, then the interrupt source from register field CMDRDY of register SDC_CMDSTA will be masked. '0' in some bits does not cause interrupt mask on the corresponding interrupt source from register SDC_CMDSTA and SDC_DATSTA.
A0270050	**SDC_IRQMASK1**	32	**SD memory card IRQ mask register 1** Contains parts of SD memory card interrupt mask register. Registers SDC_IRQMASK1 and SDC_IRQMASK0 compose the SD memory card interrupt mask register. The register masks interrupt sources from register SDC_CSTA. '1' in some bits of the register masks the corresponding interrupt source with the same bit position. For example, if IRQMASK[63] is 1, then interrupt source from register field OUT_OF_RANGE of register SDC_ CSTA will be masked. '0' in some bit does not cause interrupt mask on the corresponding interrupt source from register SDC_CSTA.
A0270054	**SDIO_CFG**	32	**SDIO configuration register** Configures functions for SDIO.
A0270058	**SDIO_STA**	32	**SDIO status register** Identifies if thers is SDIO interrupt during the interrupt period on data line.
A0270080	**CLK_RED**	32	**CLK latch configuration register** Configures the MSDC sample data/response clock.

Address	Name	Width	Register function
			Note: When MSDC_IOCON[19] = 1, the host will latch response; otherwise MSDC FSM will handle the response from PAD directly.
A0270098	<u>DAT_CHECKSUM</u>	32	**MSDC Rx data checksum register** Competes the checksum value of Rx read data

A0270000 **<u>MSDC_CFG</u>** **SD Memory Card Controller Configuration Register** **04000020**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name					**FIFOTHD**						**VDDPD**	**RCDEN**	**DIRQEN**	**PINEN**	**DMAEN**	**INTEN**
Type					RW						RW	RW	RW	RW	RW	RW
Reset					0	1	0	0			0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**SCLKF**								**SCLKON**	**CRED**	**STDBY**	**CLKSRC**		**NOCRC**	**RST**	**MSDC**
Type	RW								RW	RW	RW	RW		RW	W1C	RW
Reset	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0

Overview: For general configuration of the SD controller. Note that MSDC_CFG[31:16] can be accessed by 16-bit APB bus access.

Bit(s)	Mnemonic	Name	Description
27:24	**FIFOTHD**	FIFOTHD	**FIFO threshold** The register field determines when to issue a DMA request. For write transactions, DMA requests will be asserted if the number of free entries in FIFO are bigger than or equal to the value in the register field. For read transactions, DMA requests will be asserted if the number of valid entries in FIFO are bigger than or equal to the value in the register field. The register field must be set according to the setting of data transfer count in DMA burst mode. If single mode for DMA transfer is used, the register field shall be set to 0b0001. 0000: Invalid 0001: Threshold value is 1. 0010: Threshold value is 2. 0011~01111: ... 1000: Threshold value is 8. Others: Invalid
21	**VDDPD**	VDDPD	**Controls output pin VDDPD used for power saving** Output pin VDDPD controls the power for memory card. 0: Output pin VDDPD outputs logic low. The power for memory card will be turned off. 1: Output pin VDDPD outputs logic high. The power for memory card will be turned on.
20	**RCDEN**	RCDEN	**Controls output pin RCDEN used for card identification process when the controller is for SD memory card** Its output controls the pull-down resistor on the system board to connect to or disconnect from signal CD/DAT3. 0: The output pin RCDEN outputs logic low. 1: The output pin RCDEN outputs logic high.
19	**DIRQEN**	DIRQEN	**Enables data request interrupt** The register bit is used to control if data request is used as an

Bit(s)	Mnemonic	Name	Description
			interrupt source. 0: Data request is not used as an interrupt source. 1: Data request is used as an interrupt source.
18	**PINEN**	PINEN	**Enables pin interrupt** The register bit is used to control if the pin for card detection is used as an interrupt source. 0: The pin for card detection is not used as an interrupt source. 1: The pin for card detection is used as an interrupt source.
17	**DMAEN**	DMAEN	**Enables DMA** *Note: If DMA capability is disabled, the application software must poll the status of register MSDC_STA to check on any data transfer request. If DMA is desired, the register bit must be set up before command register is written.* 0: DMA request induced by various conditions is disabled, no matter the controller is configured as the host of either SD memory card or memory stick. 1: DMA request induced by various conditions is enabled, no matter the controller is configured as the host of either SD memory card or memory stick.
16	**INTEN**	INTEN	**Enables interrupt** *Note: If interrupt capability is disabled, the application software must poll the status of register MSDC_STA to check on any interrupt request.* 0: Interrupt induced by various conditions is disabled, no matter the controller is configured as the host of either SD memory card or memory stick. 1: Interrupt induced by various conditions is enabled, no matter the controller is configured as the host of either SD memory card or memory stick.
15:8	**SCLKF**	SCLKF	**Controls clock frequency of serial clock on SD bus and denotes clock frequency of SD bus serial clock as fslave and clock frequency of the SD controller as fhost which is 98.3 or 96.2 MHz** *Note: The allowed maximum frequency of fslave is 49.15MHz.* While changing the clock rate, "1T clock period before change + 1T clock period after change" is required for HW signal to re-synchronize. 00000000b: fslave =(1/2)*fhost 00000001b: fslave = [1/(4*1)]*fhost 00000010b: fslave = [1/(4*2)]*fhost 00000011b: fslave = [1/(4*3)]*fhost 00000100b~11111110b: ... 11111111b: fslave = [1/(4*255)]*fhost
7	**SCLKON**	SCLKON	**Serial clock always on** For debugging. 0: Serial clock not always on 1: Serial clock always on
6	**CRED**	CRED	**Rising edge data** The register bit is used to determine the serial data input is latched at the falling edge or rising edge of the serial clock. The default setting is at the rising edge. If the serial data have bad timing, set the register bit to 1. When the memory card has bad timing on returned read data, set the register bit to 1. 0: Serial data input is latched at the rising edge of serial clock. 1: Serial data input is latched at the falling edge of serial clock.

Bit(s)	Mnemonic	Name	Description
5	**STDBY**	STDBY	**Standby mode** If the module is powered down, operating clock to the module will be stopped. At the same time, the clock to card detection circuitry will also be stopped. If detection on memory card insertion and removal is desired, write 1 to the register bit. If interrupt for detection on memory card insertion and removal is enabled, the interrupt will take place whenever the memory is inserted or removed. 0: Standby mode is disabled. 1: Standby mode is enabled.
4:3	**CLKSRC**	CLKSRC	**Specifies which clock is used as source clock of memory card** 00 : MPLL/5.5MHz clock 01 : MPLL/7MHz clock (this divider is default off, before switch controller source clock to this clock, you should switch controller source clock to 26Mhz first, and then enable divider MPLL/7, after a moment you can switch controller source clock to MPLL/ 7 to void source clock glitch. For detail setting please refer to confgsys document) 10 : MPLL/8MHz clock 11 : MPLL/10MHz clock For phone 00 : 94.5MHz clock Need to keep BT_APP_DIV_EN= 1'b0 in CLK_CONDA[15]. 01 : 74.3MHz clock NOTE: Need to set POWERFUL_DIV_EN1 = 1'b1 first in CLK_CONDA[10]. 10 : 65MHz clock 11 : Forbidden For BT app. 00 : Forbidden 01 : 89.1MHz clock NOTE: Need to set POWERFUL_DIV_EN1 = 1'b1 first in CLK_CONDA[10]. 10 78MHz clock 11 : 62.4MHz clock NOTE: Need to set POWERFUL_DIV_EN2 = 1'b1 first in CLK_CONDA[9].
2	**NOCRC**	NOCRC	**Disable CRC** '1' indicates data transfer without CRC is desired. For write data block, the data are transmitted without CRC. For read data block, CRC will not be checked. For testing purpose. 0: Data transfer with CRC is desired. 1: Data transfer without CRC is desired.
1	**RST**	RST	**Software reset** Writing 1 to the register bit will cause internal synchronous reset of SD controller but will not reset register settings, RST should only be set when RST equal to 0. 0: Read 0 stands for the reset process is finished. 1: Write 1 to reset SD controller.
0	**MSDC**	MSDC	**Configures the controller as SD memory card mode** CLK/CMD/DAT line is pulled low when SD memory card mode is disable. 0: Disable SD memory card 1: Enable SD memory card

A0270004 **MSDC_STA** **SD Memory Card Controller Status Register** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Name	BUSY	FIFOCLR							FIFOCNT				INT	DRQ	BE	BF
Type	R	W1C							RO				RO	RO	RO	RO
Reset	0	0							0	0	0	0	0	0	0	0

Overview: The register contains the status of FIFO, interrupts and data requests.

Bit(s)	Mnemonic	Name	Description
15	**BUSY**	BUSY	**Status of the controller** If the controller is in busy state, the register bit will be 1; otherwise 0. 0: The controller is in busy state. 1: The controller is in idle state.
14	**FIFOCLR**	FIFOCLR	**Clears FIFO** **Writing 1 to the register bit will cause the content of FIFO clear and reset the status of FIFO controller.** 0: Read 0 stands for the FIFO clear process is finished. 1: Write 1 to clear the content of FIFO clear and reset the status of FIFO controller.
7:4	**FIFOCNT**	FIFOCNT	**FIFO count** The register field shows how many valid entries are there in FIFO. 0000: 0 valid entry in FIFO 0001: 1 valid entry in FIFO 0010: 2 valid entries in FIFO 0011~0111: ... 1000: 8 valid entries in FIFO
3	**INT**	INT	**Indicates if there is any interrupt existing** When there is interrupt existing, the register bit will still be active even if register bit INTEN in register MSDC_CFG is disabled. The SD controller can interrupt MCU by issuing interrupt request to the interrupt controller, or the software/application will poll the register endlessly to check if there is any interrupt request existing in the SD controller. When register bit INTEN in register MSDC_CFG is disabled, the second method is used. For read commands, it is possible that time-out error takes place. The software can read the status register to check if the time-out error takes place without OS time tick support or data request asserted. *Note: The register bit will be cleared when register MSDC_INT is read.* 0: No interrupt request existing. 1: Interrupt request exists.
2	**DRQ**	DRQ	**Indicates if any data transfer is required** When a data transfer is required, the register bit will still be active even if register bit DIRQEN in register MSDC_CFG is disabled. Data transfer can be achieved by DMA channel alleviating MCU loading, or by polling the register bit to check if any data transfer is requested. When register bit DIRQEN in register MSDC_CFG is disabled, the second method is used. 0: No DMA request existing. 1: DMA request exists.
1	**BE**	BE	**Indicates if FIFO in SD controller is empty** 0: FIFO in SD controller is not empty. 1: FIFO in SD controller is empty.
0	**BF**	BF	**Indicates if FIFO in SD controller is full** 0: FIFO in SD controller is not full. 1: FIFO in SD controller is full.

A0270008 MSDC_INT SD Memory Card Controller Interrupt Register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									SDIOIRQ	SDR1BIRQ		SDMCIRQ	SDDATIRQ	SDCMDIRQ	PINIRQ	DIRQ
Type									RC	RC		RC	RC	RC	RC	RC
Reset									0	0		0	0	0	0	0

Overview: The register contains the status of interrupts. Note that the register still show status of interrupt even though interrupt is disabled, that is, register bit INTEN of register MSDC_CFG is set to 0. It implies that software interrupt can be implemented by polling register bit INT of register MSDC_STA and this register. However, if hardware interrupt is desired, be sure to clear the register before setting register bit INTEN of register MSDC_CFG to 1, or undesired hardware interrupt arisen from the previous interrupt status may take place.

Bit(s)	Mnemonic	Name	Description
7	**SDIOIRQ**	SDIOIRQ	**SDIO interrupt** The register bit indicates if there is any interrupt for SDIO existing. Whenever an interrupt for SDIO exists, the register bit will be set to 1 if the interrupt is enabled. It will be reset when the register is read. 0: No SDIO interrupt 1: Interrupt for SDIO exists.
6	**SDR1BIRQ**	SDR1BIRQ	**SD R1b response interrupt** The register bit will be active when a SD command with R1b response is finished and the DAT0 line is transited from busy to idle state. Single block write commands with R1b response will cause interrupt when the command is completed either successfully or with CRC error. However, multi-block write commands with R1b response do not cause interrupt because multi-block write commands are always stopped by STOP_TRANS commands. STOP_TRANS commands (with R1b response) behind multi-block write commands will cause interrupt. Single block read command with R1b response will cause interrupt when the command is completed, but multi-block read commands do not. *Note: STOP_TRANS commands (with R1b response) behind multi-block read commands will cause interrupt.* 0: No interrupt for SD R1b response. 1: Interrupt for SD R1b response exists.
4	**SDMCIRQ**	SDMCIRQ	**SD memory card interrupt** The register bit indicates if there is any interrupt for SD memory card existing. Whenever an interrupt for SD memory card exists, i.e. any bit in register SDC_CSTA is active, the register bit will be set to 1 if interrupt is enabled. It will be reset when the register is read. *Note: This bit will not trigger MSDC hardware interrupt.* 0: No SD memory card interrupt 1: SD memory card interrupt exists.
3	**SDDATIRQ**	SDDATIRQ	**SD bus DAT interrupt** The register bit indicates if there is any interrupt for SD DAT line existing. Whenever interrupt for SD DAT line exists, i.e. any bit in register SDC_ DATSTA is active, the register bit will be set to 1 if

Bit(s)	Mnemonic	Name	Description
			interrupt is enabled. It will be reset when the register is read. 0: No SD DAT line interrupt 1: SD DAT line interrupt exists.
2	**SDCMDIRQ**	SDCMDIRQ	**SD bus CMD interrupt** The register bit indicates if there is any interrupt for SD CMD line existing. Whenever interrupt for SD CMD line exists, i.e. any bit in the register SDC_CMDSTA is active, the register bit will be set to 1 if interrupt is enabled. It will be reset when the register is read. 0: No SD CMD line interrupt 1: SD CMD line interrupt exists.
1	**PINIRQ**	PINIRQ	**Pin change interrupt** The register bit indicates if there is any interrupt for memory card insertion/removal existing. Whenever the memory card is inserted or removed and card detection interrupt is enabled, i.e. register bit PINEN in register MSDC_CFG is set to 1, the register bit will be set to 1. It will be reset when the register is read. 0: Otherwise 1: Card is inserted or removed.
0	**DIRQ**	DIRQ	**Data request interrupt** The register bit indicates if there is any interrupt for data request existing. Whenever data request exists and data request as an interrupt source is enabled, i.e. register bit DIRQEN in register MSDC_CFG is set to 1, the register bit will be active. It will be reset when being read. For software, data requests can be recognized by polling register bit DRQ or by data request interrupt. Data request interrupts will be generated every FIFOTHD data transfers. 0: No data request interrupt 1: Data request interrupt occurs.

A027000C MSDC_PS SD Memory Card Pin Status Register 00000008

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name								**CMD**	**DAT**							
Type								RO	RO							
Reset								0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**CDDEBOUNCE**											**PINCHG**	**PIN0**	**POEN0**	**PIEN0**	**CDEN**
Type	RW											RC	RO	RW	RW	RW
Reset	0	0	0	0								0	1	0	0	0

Overview: The register is used for card detection. When the memory card controller and system are powered on, , the power for the memory card will still be off unless the power is supplied by the PMIC. Meanwhile, the pad for card detection defaults to pull-down when the system is powered on. The scheme of card detection for MS is the same as that for SD. To detect card insertion, first pull up the INS pin and then enable card detection and the input pin at the same time. After 32 cycles of controller clock, the status of pin changes will emerge. To detect card removal, simply keep enabling card detection and the input pin.

Bit(s)	Mnemonic	Name	Description
24	**CMD**	CMD	**Memory card/SDIO card/MMC card command lines**
23:16	**DAT**	DAT	**Memory card/SDIO card/MMC card data lines**
15:12	**CDDEBOUN**	CDDEBOUNCE	**Specifies the time interval for card detection de-bounce**

Bit(s)	Mnemonic	Name	Description
	CE		Default value: 0. It means the de-bounce interval is 32-cycle time at 32kHz. The interval can extend one cycle time at 32kHz by increasing the counter by 1.
4	**PINCHG**	PINCHG	**Pin change** The register bit indicates the status of card insertion/removal. If the memory card is inserted or removed, the register bit will be set to 1 no matter pin change interrupt is enabled or not. It will be cleared when the register is read. 0: Otherwise 1: Card is inserted or removed.
3	**PIN0**	PIN0	**Shows the value of input pin for card detection** 0: The value of input pin for card detection is logic low. 1: The value of input pin for card detection is logic high.
2	**POEN0**	POEN0	**Controls output of input pin for card detection** 0: Output of input pin for card detection is disabled. 1: Output of input pin for card detection is enabled.
1	**PIEN0**	PIEN0	**Controls input pin for card detection** 0: Input pin for card detection is disabled. 1: Input pin for card detection is enabled.
0	**CDEN**	CDEN	**Enables card detection** The register bit is used to enable or disable card detection. 0: Card detection is disabled. 1: Card detection is enabled.

A0270010 MSDC_DAT SD Memory Card Controller Data Register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DATA[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DATA[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: The register is used to read/write data from/to FIFO inside the SD controller. Data access unit: 32 bits.

Bit(s)	Mnemonic	Name	Description
31:0	**DATA**	DATA	**Reads/Writes data from/to FIFO inside SD controller** Data access unit: 32 bits

A0270014 MSDC_IOCON SD Memory Card Controller IO Control Register 010000C3

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name							SAMPLEDLY		FIXDLY		SAMPON	CRCDIS	CMDSEL	INTLH		DSW
Type							RW		RW		RW	RW	RW	RW		RW
Reset							0	1	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	CMDRE					HIGH_SPEE	DMABURST		SRCFG1	SRCFG0	ODCCFG1			ODCCFG0		

						D										
Type	RW					RW	RW		RW	RW	RW			RW		
Reset	0					0	0	0	1	1	0	0	0	0	1	1

Overview: The register specifies the output driving capability and slew rate of IO pads for MSDC. The reset value is suggested setting. If the output driving capability of pins DAT0, DAT1, DAT2 and DAT3 is too large, it is possible to arise ground bounce and thus result in glitch on SCLK. The actual driving current depends on the PAD type selected for the chip.

Bit(s)	Mnemonic	Name	Description
25:24	**SAMPLEDLY**	SAMPLEDLY	**Used for SW to select the turn-around delay cycle between write data end bit and CRC status for SD card** 00: 0-T delay 01: 1-T delay 10: 2-T delay 11: 3-T delay
23:22	**FIXDLY**	FIXDLY	**Used for SW to select the delay cycle after clock fix high for the host controller to SD card** 00: 0-T delay 01: 1-T delay 10: 2-T delay 11: 3-T delay
21	**SAMPON**	SAMPON	**Data sample enabling always on** The bit is suggested to be set to 1 when the feedback clock is used and to 0 when the multiple phase clock is used. 0: Data sample enabling not always on 1: Data sample enabling always on
20	**CRCDIS**	CRCDIS	**Switches off data CRC check for SD read data** 0: CRC check is on. 1: CRC check is off.
19	**CMDSEL**	CMDSEL	**Determines whether the host should delay 1-T to latch response from card** 0: Host latches response without 1-T delay 1: Host latches response with 1-T delay.
18:17	**INTLH**	INTLH	**Selects latch timing for SDIO multi-block read interrupt** 00: Host latches INT at the second backend clock after the end bit of the current data block from card is received. (Default) 01: Host latches INT at the first backend clock after the end bit of the current data block from card is received. 10: Host latches INT at the second backend clock after the end bit of the current data block from card is received. 11: Host latches INT at the third backend clock after the end bit of the current data block from card is received.
16	**DSW**	DSW	**Determines whether the host should latch data with 1-T delay or not** For SD card, this bit is suggested to be 0. For MSPRO cards, it is suggested to be 1. 0: Host latches the data with 1-T delay. 1: Host latches the data without 1-T delay.
15	**CMDRE**	CMDRE	**Determines whether the host should latch response token (sent from card on CMD line) at rising edge or falling edge of serial clock** (T.B.D this bit is un-useful) 0: Host latches response at rising edge of serial clock.

Bit(s)	Mnemonic	Name	Description
			1: Host latches response at falling edge of serial clock.
10	**HIGH_SPEED**	HIGH_SPEED	**For high-speed mode when internal sample clock is used** High-speed mode means the SD/MMC serial bus clock rate is bigger than 25MHz. The default speed mode means that the SD/MMC serial bus clock rate is bigger than 25MHz. 0: Default speed 1: High speed
9:8	**DMABURST**	DMABURST	**Used for SW to select burst type when data are transferred by DMA** *Note: Only single mode can support non-4N bytes data transfer in read operation.* 00: Single mode 01: 4-beat incrementing burst 10: 8-beat incrementing burst 11: Reserved
7	**SRCFG1**	SRCFG1	**Output driving capability for pins DAT0, DAT1, DAT2 and DAT3** 0: Fast slew rate 1: Slow slew rate
6	**SRCFG0**	SRCFG0	**Output driving capability for pins CMD/BS and SCLK** 0: Fast slew rate 1: Slow slew rate
5:3	**ODCCFG1**	ODCCFG1	**Output driving capability for pins DAT0, DAT1, DAT2 and DAT3** 000: 4mA 001: 8mA 010: 12mA 011: 16mA
2:0	**ODCCFG0**	ODCCFG0	**Output driving capability for pins CMD/BS and SCLK** 000: 4mA 001: 8mA 010: 12mA 011: 16mA

A0270018 MSDC_IOCON1 SD Memory Card Controller IO Control Register 1 00022022

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name		PRCFG_CK	PRVAL_CK			PRCFG_CM	PRVAL_CM			PRCFG_DA	PRVAL_DA					
Type		RW	RW			RW	RW			RW	RW					
Reset		0	1	0		0	0	0		0	1	0				

Bit(s)	Mnemonic	Name	Description
14	**PRCFG_CK**	PRCFG_CK	**Pull-up/down register configuration for pin CK** Default value: 0 0: Pull-up resistor in the I/O pad of pin CK is enabled. 1: Pull-down resistor in the I/O pad of pin CK is enabled.
13:12	**PRVAL_CK**	PRVAL_CK	**Pull-up/down register value for pin CLK** Default value: 10

Bit(s)	Mnemonic	Name	Description
			00: Pull-up/down resistor in the I/O pad of pin CLK are all disabled. 01: Pull-up/down resistor in the I/O pad of pin CLK value is 47k. 10: Pull-up/down resistor in the I/O pad of pin CLK value is 47k. 11: Pull-up/down resistor in the I/O pad of pin CLK value is 23.5k.
10	**PRCFG_CM**	PRCFG_CM	**Pull-up/down register configuration for pin CM** Default value is 0. 0: Pull-up resistor in the I/O pad of pin CM is enabled. 1: Pull-down resistor in the I/O pad of pin CM is enabled.
9:8	**PRVAL_CM**	PRVAL_CM	**Pull-up/down register value for pin CMD/BS** Default value: 00 00: Pull-up/down resistor in the I/O pad of pin CMD/BS are all disabled. 01: Pull-up/down resistor in the I/O pad of pin CMD/BS value is 47k. 10: Pull-up/down resistor in the I/O pad of pin CMD/BS value is 47k. 11: Pull-up/down resistor in the I/O pad of pin CMD/BS value is 23.5k.
6	**PRCFG_DA**	PRCFG_DA	**Pull-up/down register configuration for pin DAT0, DAT1, DAT2 and DAT3** Default value: 0 0: Pull-up resistor in the I/O pad of pin DAT is enabled. 1: Pull-down resistor in the I/O pad of pin DAT is enabled.
5:4	**PRVAL_DA**	PRVAL_DA	**Pull-up/down register value for pin DAT0, DAT1, DAT2 and DAT3** Default value: 10 00: Pull-up/ down resistor in the I/O pad of pin DAT are all disabled. 01: Pull-up/down resistor in the I/O pad of pin DAT value is 47k. 10: Pull-up/down resistor in the I/O pad of pin DAT value is 47k. 11: Pull-up/down resistor in the I/O pad of pin DAT value is 23.5k.

A0270020 SDC_CFG SD Memory Card Controller Configuration Register 00008000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DTOC								WDOD				SDIO		MDLEN	SIEN
Type	RW								RW				RW		RW	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0		0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	BSYDLY				BLKLEN											
Type	RW				RW											
Reset	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: The register is used to configure the SD memory card controller when it is configured as the host of SD. The register is used to configure the SD memory card controller when it is configured as the host of SD memory card. If the controller is configured as the host of memory stick, the contents of the register will have no impact on the operation of the controller. Note that SDC_CFG[31:16] can be accessed by 16-bit APB bus access.

Bit(s)	Mnemonic	Name	Description
31:24	**DTOC**	DTOC	**Data time-out counter** The period from finish of the initial host read command or the last read data block in a multiple block read operation to the start bit of the next read data block requires at least two serial clock cycles. The counter is used to extend the period (Read Data Access Time) in unit of 65,536 serial clock. See the register field descriptions of register

Bit(s)	Mnemonic	Name	Description
			bit RDINT for reference. 00000000: Extend 65,536 more serial clock cycles 00000001: Extend 65,536x2 more serial clock cycles 00000010: Extend 65,536x3 more serial clock cycles 00000011~11111110: ... 11111111: Extend 65,536x 256 more serial clock cycles
23:20	**WDOD**	WDOD	**Write data output delay** The period from finish of the response for the initial host write command or the last write data block in a multiple block write operation to the start bit of the next write data block requires at least two serial clock cycles. The register field is used to extend the period (Write Data Output Delay) in unit of one serial clock. 0000: No extension 0001: Extend 1 more serial clock cycle 0010: Extend 2 more serial clock cycles 0011~1110: ... 1111: Extend 15 more serial clock cycle
19	**SDIO**	SDIO	**Enables SDIO** 0: Disable SDIO mode 1: Enable SDIO mode
17	**MDLEN**	MDLEN	**Enables multiple data line** The register can be enabled only when SD memory card is applied and detected by software application. It is the responsibility of the application to program the bit correctly when an multi-media card is applied. If an multi-media card is applied and 4-bit data line is enabled, the 4 bits will be output every serial clock. Therefore, data integrity will fail. 0: Disable 4-bit data line 1: Enable 4-bit data line
16	**SIEN**	SIEN	**Enables serial interface** It should be enabled as soon as possible before any command. 0: Disable serial interface for SD 1: Enable serial interface for SD
15:12	**BSYDLY**	BSYDLY	**Only valid for the commands with R1b response** If the command has a response of R1b type, the SD controller must monitor the data line 0 for card busy status from the bit time that is two serial clock cycles after the command end bit to check if the operation in SD memory card have finished. The register field is used to expand the time between the command end bit and end of detection period to detect card busy status. If time is up and there is no card busy status on data line 0, the controller will abandon the detection. 0000: No extension 0001: Extend 1 more serial clock cycle 0010: Extend 2 more serial clock cycles 0011~1110: ... 1111: Extend 15 more serial clock cycle
11:0	**BLKLEN**	BLKLEN	**Block length** The register field is used to define the length of one block in unit of byte in a data transaction. The maximum value of block length is 2,048 bytes. 000000000000: Reserved 000000000001: Block length is 1 byte. 000000000010: Block length is 2 bytes. 000000000011~011111111110: ...

Bit(s)	Mnemonic	Name	Description
			011111111111: Block length is 2,047 bytes. 100000000000: Block length is 2,048 bytes.

A0270024 SDC_CMD SD Memory Card Controller Command Register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																CMDFAIL
Type																RW
Reset																0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	INTC	STOP	RW	DTYPE		IDRT	RSPTYP			BREAK	CMD					
Type	RW	RW	RW	RW		RW	RW			RW	RW					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: The register defines a SD memory card command and its attribute. Before the SD controller issues a transaction onto the SD bus, application shall specify other relative settings such as argument for command. After application writes the register, the SD controller will issue the corresponding transaction onto the SD serial bus. If the command is GO_IDLE_STATE, the controller will have serial clock on SD bus run 128 cycles before issuing the command.

Bit(s)	Mnemonic	Name	Description
16	**CMDFAIL**	CMDFAIL	**If 4-bit SDIO mode is enabled and when CMD/DAT error occurrs, set this bit to select whether to "wait stop command" or "wait data state machine idle".** 0: Wait stop command 1: Wait data state machine idle
15	**INTC**	INTC	**Indicates if the command is GO_IRQ_STATE** If the command is GO_IRQ_STATE, the period between command token and response token will not be limited. 0: The command is not GO_IRQ_STATE. 1: The command is GO_IRQ_STATE.
14	**STOP**	STOP	**Indicates if the command is a stop transmission command** 0: The command is not a stop transmission command. 1: The command is a stop transmission command.
13	**RW**	RW	**Defines the command is a read command or write command** The register bit is valid only when the command causes a transaction with data token. 0: The command is a read command. 1: The command is a write command.
12:11	**DTYPE**	DTYPE	**Defines data token type for the command** 00: No data token for the command 01: Single block transaction 10: Multiple block transaction, i.e. the command is a multiple block read or write command. 11: Stream operation. It can only be used when an multi-media card is applied.
10	**IDRT**	IDRT	**Identification response time** The register bit indicates if the command has a response with NID (i.e. 5 serial clock cycles as defined in SD Memory Card Specification Part 1 Physical Layer Specification version 1.0)

Bit(s)	Mnemonic	Name	Description
			response time. The register bit is valid only w hen the command has a response token. Thus the register bit must be set to 1 for CMD2 (ALL_SEND_CID) and ACMD41 (SD_APP_OP_CMD). 0: Otherw ise 1: The command has a response w ith NID response time.
9:7	**RSPTYP**	RSPTYP	**Defines response type for the command** For commands w ith R1 and R1b response, register SDC_CSTA (not SDC_STA) updates after response token is received. This register SDC_CSTA contains the status of the SD, and it can be used as a response interrupt source. *Note: If CMD7 is used with all 0's RCA, then RSPTYP must be "000". Command "GO_TO_IDLE" also has RSPTYP='000'.* 000: There is no response for the command, e.g. broadcast command w ithout response and GO_INACTIVE_STATE command. 001: The command has R1 response. R1 response token is 48-bit. 010: The command has R2 response. R2 response token is 136-bit. 011: The command has R3 response. Even though R3 is 48-bit response, it does not contain CRC checksum. 100: The command has R4 response. R4 response token is 48-bit. (only for MMC) 101: The command has R5 response. R5 response token is 48-bit. (only for MMC) 110: The command has R6 response. R6 response token is 48-bit. 111: The command has R1b response. If the command has a response of R1b type, SD controller must monitor the data line 0 for card busy status from the bit time that is 2 or 4 serial clock cycles after the command end bit to check if the operation in SD memory card has finished. There are tw o cases for detection of card busy status. The first case is that the host stops the data transmission during an active w rite data transfer. The card w ill assert busy signal after the stop transmission command end bit follow ed by 4 serial clock cycles. The second case is that the card is in idle state or receiving a stop transmission command betw een data blocks w hen multiple block w rite command is in progress. The register bit w ill be valid only w hen the command has a response token.
6	**BREAK**	BREAK	**Aborts pending MMC GO_IRQ_MODE command** It is only valid for a pending GO_IRQ_MODE command w aiting for MMC interrupt response. 0: Other fields are valid. 1: Break a pending MMC GO_IRQ_MODE command in the controller. Other fields are invalid.
5:0	**CMD**	CMD	**SD memory card command** Total 6 bits.

A0270028 SDC_ARG SD Memory Card Controller Argument Register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**ARG[31:16]**															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**ARG[15:0]**															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: The register contains the argument of the SD memory card command.

Bit(s)	Mnemonic	Name	Description
31:0	**ARG**	ARG	Contains argument of the SD memory card command.

A027002C SDC_STA SD Memory Card Controller Status Register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**WP**											**FEDATBUSY**	**FECMDBUSY**	**BEDATBUSY**	**BECMDBUSY**	**BESDCBUSY**
Type	RO											RO	RO	RO	RO	RO
Reset	0											0	0	0	0	0

Overview: The register contains various statuses of SD controller as the controller is configured as the host of SD memory card.

Bit(s)	Mnemonic	Name	Description
15	**WP**	WP	**Detects the status of write protection switch on SD memory card** The register bit shows the status of write protection switch on SD memory card. There is no default reset value. Pin WP (Write Protection) is only useful when the controller is configured for SD memory card. 1: Write protection switch on, i.e. memory card is desired to be write-protected. 0: Write protection switch off, i.e. memory card is writable.
4	**FEDATBUSY**	FEDATBUSY	**Indicates if there is any transmission going on DAT line on SD bus** This bit indicates directly the CMD line at card clock domain. For those commands without data but still involving DAT line, the register bit is useless. For example, if an erase command is issued, checking if the register bit is 0 before issuing the next command with data will not guarantee that the controller is idle. In this case, use register bit BESDCBUSY. 0: No transmission is going on DAT line on SD bus. 1: There is transmission going on DAT line on SD bus.
3	**FECMDBUSY**	FECMDBUSY	**Indicates if there is any transmission going on CMD line on SD bus** This bit indicates directly the CMD line at card clock domain. 0: No transmission is going on CMD line on SD bus. 1: There is transmission going on CMD line on SD bus.
2	**BEDATBUSY**	BEDATBUSY	**Indicates if there is any transmission going on DAT line on SD bus** 0: Backend SDC controller gets the info that no transmission is going on DAT line on SD bus. 1: Backend SDC controller gets the info that there is transmission going on DAT line on SD bus.
1	**BECMDBUS**	BECMDBUSY	**Indicates if there is any transmission going on CMD line on SD**

Bit(s)	Mnemonic	Name	Description
	Y		**bus** This bit shows backend controller's CMD busy state. The busy state is synced from card clock domain to bus clock domain. 0: Backend SDC controller gets the info that no transmission is going on CMD line on SD bus. 1: Backend SDC controller gets the info that there is transmission going on CMD line on SD bus.
0	**BESDCBUSY**	BESDCBUSY	**Indicates if SD controller is busy, i.e. is there any transmission going on CMD or DAT line on SD bus** This bit shows backend controller's SDC busy state. The busy state is synced from card clock domain to bus clock domain. 0: Backend SD controller is idle. 1: Backend SD controller is busy.

A0270030 SDC_RESP0 SD Memory Card Controller Response Register 0 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RESP[31:0][31:16]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RESP[31:0][15:0]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: The register contains parts of the last SD memory card bus response. See descriptions of register field SDC_RESP3.

Bit(s)	Mnemonic	Name	Description
31:0	**RESP[31:0]**	RESP_31_0	

A0270034 SDC_RESP1 SD Memory Card Controller Response Register 1 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RESP[63:32][31:16]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RESP[63:32][15:0]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: The register contains parts of the last SD memory card bus response. See descriptions of register field SDC_RESP3.

Bit(s)	Mnemonic	Name	Description
31:0	**RESP[63:32]**	RESP_63_32	

A0270038 SDC_RESP2 SD Memory Card Controller Response Register 2 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RESP[95:64][31:16]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RESP[95:64][15:0]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: The register contains parts of the last SD memory card bus response. See descriptions of register field SDC_RESP3.

Bit(s)	Mnemonic	Name	Description
31:0	**RESP[95:64]**	RESP_95_64	

A027003C SDC_RESP3 SD Memory Card Controller Response Register 3 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RESP[127:96][31:16]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RESP[127:96][15:0]															
Type	RO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: The register contains parts of the last SD memory card bus response. Register fields SDC_RESP0, SDC_RESP1, SDC_RESP2 and SDC_RESP3 compose the last SD memory card bus response. For response of type R2, i.e. response of commands ALL_SEND_CID, SEND_CSD and SEND_CID, only bit 127 to 0 of the response token are stored in register fields SDC_RESP0, SDC_RESP1, SDC_RESP2 and SDC_RESP3. For response of other types, only bit 39 to 8 of the response token are stored in register field SDC_RESP0.

Bit(s)	Mnemonic	Name	Description
31:0	**RESP[127:96]**	RESP_127_96	

A0270040 SDC_CMDSTA SD Memory Card Controller Command Status Register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														RSPCRCERR	CMDTO	CMDRDY
Type														RC	RC	RC
Reset														0	0	0

Overview: The register contains the status of SD controller during command execution and that of SD bus protocol after command execution when the SD controller is configured as the host of SD memory card. The register can also be used as an interrupt source. The register is cleared when being read. Meanwhile, if the interrupt is enabled and thus interrupt caused by the register is generated, reading the register will de-assert the interrupt.

Bit(s)	Mnemonic	Name	Description
2	**RSPCRCERR**	RSPCRCERR	**CRC error on CMD detected** '1' indicates the SD controller detects a CRC error after reading a response from the CMD line. 0: Otherwise 1: SD controller detects a CRC error after reading a response from the CMD line.
1	**CMDTO**	CMDTO	**Time-out on CMD detected** '1' indicates the SD controller detects a time-out condition while waiting for a response on the CMD line. 0: Otherwise 1: SD controller detects a timeout condition while waiting for a response on the CMD line.
0	**CMDRDY**	CMDRDY	**For command without response, the register bit will be 1 once the command is completed on SD bus** For command with response, the register bit will be 1 whenever the command is issued onto the SD bus and its corresponding response is received without CRC error. 0: Otherwise 1: Command with/without response is finished successfully without CRC error.

A0270044 **SDC_DATSTA** **SD Memory Card Controller Data Status Register** **00000000**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name							**DATCRCERR**								**DATTO**	**BLKDONE**
Type							RC								RC	RC
Reset							0	0	0	0	0	0	0	0	0	0

Overview: The register contains the status of SD controller during data transfer on DAT line(s) when the SD controller is configured as the host of SD memory card. The register can be used as an interrupt source. The register is cleared when being read. Meanwhile, if the interrupt is enabled and thus interrupt caused by the register is generated, reading the register will de-assert the interrupt.

Bit(s)	Mnemonic	Name	Description
9:2	**DATCRCERR**	DATCRCERR	**CRC error on DAT detected** '1' indicates that the SD controller detects a CRC error for bit n after reading a block of data from the DAT line or SD signals a CRC error after writing a block of data to the DAT line. 0: Otherwise 1: SD controller detects a CRC error after reading a block of data from the DAT line or SD signals a CRC error after writing a block of data to the DAT line. *Note: n is 7 ~ 1 for 8-bits mode. Each bit is read and cleared*

Bit(s)	Mnemonic	Name	Description
			individually.
1	**DATTO**	DATTO	**Time-out on DAT detected** A '1' indicates that the SD controller detects a time-out condition while waiting for data token on the DAT line. 0: Otherwise 1: SD controller detects a time-out condition while waiting for data token on the DAT line.
0	**BLKDONE**	BLKDONE	**Indicates the status of data block transfer** 0: Otherwise 1: A data block is successfully transferred.

A0270048 SDC_CSTA SD Memory Card Status Register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	**CSTA [31:0][31:16]**															
Type	RC															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	**CSTA [31:0][15:0]**															
Type	RC															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: After commands with R1 and R1b respond this register containing the status of the SD card, it will be used as a response interrupt source. In all register fields, logic high indicates error, and logic low indicates no error. The register is cleared when being read. Meanwhile, if the interrupt is enabled and thus interrupt caused by the register is generated, reading the register will de-assert the interrupt.

Bit(s)	Mnemonic	Name	Description
31:0	**CSTA [31:0]**	CSTA_31_0	CSTA31: OUT_OF_RANGE. The command's argument is out of the allowed range for this card. CSTA30: ADDRESS_ERROR. A misaligned address that does not match the block length is used in the command. CSTA29: BLOCK_LEN_ERROR. The transferred block length is not allowed for this card, or the number of transferred bytes does not match the block length. CSTA28: ERASE_SEQ_ERROR. An error in the sequence of erase commands occurs. CSTA27: ERASE_PARAM. An invalid selection of write-blocks for erase occurs. CSTA26: WP_VIOLATION. Attempt to program a write-protected block. CSTA25: Reserved. Return to 0. CSTA24: LOCK_UNLOCK_FAILED. Set when a sequence or password error is detected in lock/unlock card command or if there is an attempt to access a locked card. CSTA23: COM_CRC_ERROR. The CRC check of the previous command fails. CSTA22: ILLEGAL_COMMAND. Command not legal for the card state. CSTA21: CARD_ECC_FAILED. Card internal ECC is applied but fails to correct the data. CSTA20: CC_ERROR. Internal card controller error.

CSTA19: ERROR. A general or unknown error occurs during the operation.

CSTA18: UNDERRUN. The card cannot sustain data transfer in stream read mode.

CSTA17: OVERRUN. The card cannot sustain data programming in stream write mode.

CSTA16: CID/CSD_OVERWRITE. It can be either one of the following errors: 1) The CID register has been already written and cannot be overwritten; 2) The read-only section of the CSD does not match the card; 3) An attempt to reverse the copy (set as the original) or permanent WP (unprotected) bits is made.

CSTA[15: 4]: Reserved. Return to 0.

CSTA3: AKE_SEQ_ERROR. Error in the sequence of authentication process

CSTA[2: 0]: Reserved. Return to 0.

A027004C SDC_IRQMASK0 SD Memory Card IRQ Mask Register 0 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	IRQMASK [31:0][31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	IRQMASK [31:0][15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: The register contains parts of SD memory card interrupt mask register. See the register descriptions of register SDC_IRQMASK1 for reference. The register masks interrupt sources from register SDC_CMDSTA and SDC_DATSTA. IRQMASK[3:0] is for SDC_CMDSTA, and IRQMASK[18:16] for SDC_DATSTA. Note that IRQMASK[18] masks SDC_DATSTA[9:2] together. '1' in some bits of the register masks the corresponding interrupt source with the same bit position. For example, if IRQMASK[0] is 1, then the interrupt source from register field CMDRDY of register SDC_CMDSTA will be masked. '0' in some bits does not cause interrupt mask on the corresponding interrupt source from registers SDC_CMDSTA and SDC_DATSTA.

Bit(s)	Mnemonic	Name	Description
31:0	IRQMASK [31:0]	IRQMASK_31_0	

A0270050 SDC_IRQMASK1 SD Memory Card IRQ Mask Register 1 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	IRQMASK [63:32][31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	IRQMASK [63:32][15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: The register contains parts of SD memory card interrupt mask register. Registers SDC_IRQMASK1 and SDC_IRQMASK0 compose the SD memory card interrupt mask register. The register masks interrupt sources from register SDC_CSTA. '1' in some bits of the register masks the corresponding interrupt source with the same bit position. For example, if IRQMASK[63] is 1, then the interrupt source from register field OUT_OF_RANGE of register SDC_ CSTA will be masked. '0' in some bits does not cause interrupt mask on the corresponding interrupt source from register SDC_CSTA.

Bit(s)	Mnemonic	Name	Description
31:0	**IRQMASK [63:32]**	IRQMASK_63_32	

A0270054 SDIO_CFG SDIO Configuration Register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name											**DISSEL**		**INTCSEL**	**DSBSEL**		**INTEN**
Type											RW		RW	RW		RW
Reset											0		0	0		0

Overview: The register is used to configure functions for SDIO.

Bit(s)	Mnemonic	Name	Description
5	**DISSEL**	DISSEL	**Selects data block interrupt source** 0: The host detects SDIO interrupt during interrupt period between two data blocks of multiple block data access. 1: The host ignores SDIO interrupt during interrupt period between two data blocks of multiple block data access.
3	**INTCSEL**	INTCSEL	**Selects interrupt control** 0: The host detects DAT1 low as SDIO interrupt. 1: The host detects DAT3/DAT2/DAT1/DAT0 4'b1101 as SDIO interrupt.
2	**DSBSEL**	DSBSEL	**Selects data block start bit** 0: Use data line 0 as start bit of data block. Other data lines are ignored. 1: Start bit of a data block is received only when all data line 0-3 become low.
0	**INTEN**	INTEN	**Enables interrupt for SDIO** 0: Disable 1: Enable

A0270058 SDIO_STA SDIO Status Register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
Type																
Reset																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																**IRQ**
Type																RO
Reset																0

Overview: This register is used to identify if there is SDIO interrupt during the interrupt period on data line.

Bit(s)	Mnemonic	Name	Description
0	**IRQ**	IRQ	**SDIO interrupt exists on the data line.** For example, when in the interrupt period or the 1 -bit data line mode and DAT1/5 goes low from high, this bit will become 1 from 0. If DAT1/5 goes high from low, this bit will become 0 from 1. 0: There is no SDIO interrupt existing on the data line. 1: There is SDIO interrupt existing on the data line.

A0270080 CLK_RED CLK Latch Configuration Register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name			**CMD_RED**													
Type			RW													
Reset			0													
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name			**DAT_RED**						**CLKPAD_RED**	**CLK_LATCH**						
Type			RW						RW	RW						
Reset			0						0	0						

Overview: The register is used to configure the MSDC sample data/response clock. Note that only when MSDC_IOCON[19] = 1 will the host latch response; otherwise MSDC FSM will handle the response from PAD directly.

Bit(s)	Mnemonic	Name	Description
29	**CMD_RED**	CMD_RED	**Determines the command response from card output is latched at falling edge or rising edge of internal clock** Only effective when CLK_LATCH = 1 0: Internal clock rising edge to latch response 1: Internal clock falling edge to latch response
13	**DAT_RED**	DAT_RED	**Determines the input data from card output is latched at falling edge or rising edge of internal sample clock** Only effective when CLK_LATCH = 1 0: Internal clock rising edge to latch data 1: Internal clock falling edge to latch data
7	**CLKPAD_RED**	CLKPAD_RED	**Determines the input data from card is latched at falling edge or rising edge of the feedback clock from pad** The suggested setting is 0 when SD serial clock is lower than 25MHz. The suggestion setting is 1 when SD serial clock is higher than 25MHz. The suggestion setting is 0 for MMC card no matter the serial clock rate is high speed or default speed. Only effective when CLK_LATCH = 0. 0: Internal feedback clock rising edge to latch data/response 1: Internal feedback clock falling edge to latch data/response
6	**CLK_LATCH**	CLK_LATCH	**Determines which clock to latch data from card** The suggested setting is 1 if SCLKF in register field MSDC_CFG is 0x0. Otherwise, the suggested setting is 0. 0: Internal feedback clock is used to latch data/response from card.

Bit(s)	Mnemonic	Name	Description
			1: Internal clock is used to latch data/response from card.

A0270098 DAT_CHECKSUM MSDC Rx Data Checksum Register 00000000

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DAT_CHECKSUM[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DAT_CHECKSUM[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Overview: The register is used to compete the checksum value of Rx read data

Bit(s)	Mnemonic	Name	Description
31:0	**DAT_CHECKSUM**	DAT_CHECKSUM	The checksum algorithm is 32-bit XOR.

3.18 BTIF

3.18.1 General Description

Bluetooth Interface (BTIF) is designed in SOC (BT+GSM) as the UART interface between the BT chip and baseband chip. As in the UART design, BTIF is an APB slave which transmits or receives data by MCU access or through DMA/VFIFO.

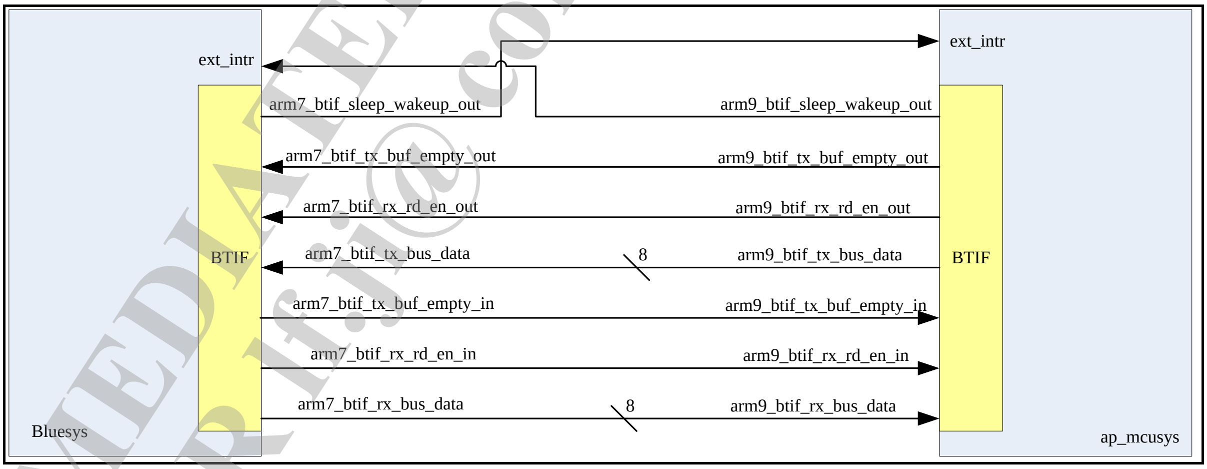


Figure 48. Interface connection between BT and baseband system

3.18.2 Register Definition

BTIF+0000h Rx Buffer Register BTIF_RBR

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									**RBR[7:0]**							
Type									RO							

RBR Rx buffer register. A read-only register. The received data can be read by accessing this register. This register is valid only when BTIF_FAKELCR[7] (0x0C) is 0.

BTIF+0000h Tx Holding Register BTIF_THR

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									**THR[7:0]**							
Type									WO							

THR Tx holding register. A write-only register. The data to be transmitted are written to this register and sent to the Bluetooth via BTIF. This register is valid only when BTIF_FAKELCR[7] (0x0C) is 0.

BTIF+0004h Interrupt Enable Register BTIF_IER

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name															**TXEEN**	**RXFEN**
Type															W/R	W/R
Reset															0	0

This register is valid only when BTIF_FAKELCR[7] is 0.

TXEEN Enables Tx empty interrupt. When set to 1, an interrupt will be generated if the Tx holding register is empty.

0 No interrupt will be generated if the Tx holding register is empty.

1 An interrupt will be generated if the Tx holding register is empty

RXFEN Enables Rx full interrupt. When set to 1, an interrupt will be generated if the Rx buffer contains data.

0 No interrupt will be generated if the Rx buffer contains data.

1 An interrupt will be generated if the Rx buffer contains data.

BTIF+0008h Interrupt Identification Register BTIF_IIR

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													ID2	ID1	ID0	NINT
Type													RO	RO	RO	RO
Reset													0	0	0	1

This register is valid only when BTIF_FAKELCR (0x0C) is not 0xBF.

IIR Identifies if there are pending interrupts. The following table lists the IIR[5:0] codes associated with the possible interrupts:

Table 52 IIR[5:0] codes associated with the possible interrupts

IIR[3:0]	Priority level	Interrupt	Source
0001	-	No pending interrupt	
0100	1	Rx data received	Rx data received
1100	2	Rx data time-out	Time-out on character in Rx buffer
0010	3	Tx holding register empty	Tx holding register empty.

Rx data received interrupt

A Rx received interrupt (IIR[3:0] = 0x04) is generated when RXFEN (IER[0]) is set and Rx data are placed in the Rx buffer register. The interrupt is cleared by reading the Rx buffer register.

Rx data time-out interrupt

The Rx data time-out interrupt will be generated if all of the following conditions are applied:

1. Rx buffer is empty.

2. The most recent character is received longer than (RTOCNT*bclk period*4).

3. RXFEN (IER[0]) is set to 1.

The time-out timer is restarted upon receipt of a new byte from the Rx shift register. This interrupt is only valid while VFIFO is used. This register is cleared by reading the VFIFO status register (0x4C).

Tx holding register empty

A Tx holding register empty interrupt (IIR[3:0] = 0x02) is generated when TXEEN(IER[1]) is set and no data are placed in the Tx holding register. This interrupt is cleared by writing data into BTIF_THR (0x00).

BTIF+0008h FIFO_CTRL BTIF_FIFOCTRL

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														CLRT	CLRR	
Type														WO	WO	
Reset														0	0	

This register is valid only when BTIF_FAKELCR (0x0C) is not 0xBF.

CLRT Clears transmit FIFO. This bit is self-clearing.

0 Leave Tx FIFO intact.

1 Clear all the bytes in Tx FIFO.

CLRR Clears receive FIFO. This bit is self-clearing.

0 Leave Rx FIFO intact.

1 Clear all the bytes in Rx FIFO.

BTIF+000Ch FAKE LCR BTIF_FAKELCR

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									FAKELCR[7:0]							
Type									R/W							
Reset									0	0	0	0	0	0	0	0

FAKELCR This register is added to synchronize the software control method of UART. When FAKELCR[7] is 1, RBR(0x00), THR(0x00) and IER(0x04) will not be readable/writable. When FAKELCR is 0xBF, RBR(0x00), THR(0x00), IER(0x04), IIR(0x08) and LSR(0x14) will not be readable/writable.

BTIF+0014h Line Status Register BTIF_LSR

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name										TEMT	THRE					DR
Type										RO	RO					RO
Reset										1	1					0

LSR Line status register. Readable when LCR != 0xBF.

TEMT Tx holding register is empty.

0 Empty conditions are not met.

1 This bit is set when the Tx holding register is empty.

THRE Indicates if Tx FIFO is reduced to its trigger level

- **0** Reset whenever the contents of Tx FIFO are more than its trigger level (FIFOs are enabled)
- **1** Set whenever the contents of Tx FIFO are reduced to its trigger level (FIFOs are enabled)

DR Data Ready

- **0** Cleared by reading the Rx buffer.
- **1** Set by the Rx buffer becoming full.

BTIF+0048h Sleep Enable Register BTIF_SLEEP_EN

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																SLEEP_EN
Type																R/W
Reset																0

SLEEP_EN For sleep mode issue

- **0** Does not deal with sleep mode indication signal
- **1** Activate flow control according to software initial settings when the chip enters the sleep mode. Release hardware flow when the chip wakes up.

BTIFn+004Ch DMA Enable Register BTIF_DMA_EN

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														TO_CNT_AUTORST	TX_DMA_EN	RX_DMA_EN
Type														R/W	R/W	R/W
Reset														0	0	0

RX_DMA_EN RX_DMA mechanism enabling signal

- **0** Does not use DMA in Rx.
- **1** Use DMA in Rx. When this register is enabled, the flow control is based on the DMA threshold and generates a time-out interrupt

TX_DMA_EN TX_DMA mechanism enabling signal

- **0** Does not use DMA in Tx.
- **1** Use DMA in Tx. When this register is enabled, the flow control is based on the DMA threshold and generates a time-out interrupt for DMA.

TO_CNT_AUTORST Time-out counter auto reset register

0 After Rx time-out takes place, the software shall reset the interrupt by reading BTIF 0x4C.

1 The time-out counter will be auto reset.

BTIF+0054h Rx Time-out Count BTIF_RTOCNT

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									**RTOCNT[7:0]**							
Type									R/W							
Reset									0x40							

RTOCNT Used for Rx time-out interrupt. The Rx time-out interrupt will be generated when:

1. RXFEN (0x04[0]) is set to 1.
2. Rx buffer is empty.
3. The most recent character is received longer than (RTOCNT*bclk period*4).

BTIF+0060h TRX_TRIGGER_LEVEL BTIF_TRI_LVL

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name									**BTIF_LOOP**	**RX_TRI_LVL**			**TX_TRI_LVL**			
Type									R/W	R/W			R/W			
Reset									0x0	0x5			0x0a			

TX_TRI_LVL Used for Tx FIFO trigger threshold. THRE(0x14[5]) will be set if the data in the TXFIFO are less than TX_TRI_LVL.

RX_TRI_LVL Used for Rx FIFO trigger threshold. A Rx trigger interrupt (IIR(0x08]) = 4) might be set if the data in the RXFIFO are more than RX_TRI_LVL. The output flow control signal will also be set if the data in the RXFIFO are more than RX_TRI_LVL.

BTIF_LOOP Enables BTIF loop back mode. The data output from Tx will be received by Rx.

BTIF+0064h SLEEP_WAKEUP BTIF_WAK

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																**SLEEP_WAKE**

Type																WO
Reset																1

SLEEP_WAKE ARM9 side btif_sleep_wakeup_in_b is connected to eint[16] (ARM9 has eint[19:0]. ARM7 side btif_sleep_wakeup_in_b is connected to eint[0] (ARM7 has eint[3:0])

BTIF+0068h ASYNC_WAIT_TIME BTIF_WAT_TIME

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name											**WAT_TIME_2**			**WAT_TIME_1**		
Type											R/W			R/W		
Reset											0X2			0x2		

ASYNC_WAIT_TIME Sets up waiting time of RX read-out.

WAT_TIME_1 The first level of wait time.

WAT_TIME_2 The second level of wait time.

Notes: The value of WAT_TIME_1/ WAT_TIME_1 cannot be smaller than 0x2.

BTIF+006Ch NEW_HANDSHAKE BTIF_HANDSHAKE

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name														**RTO_EXT**	**HIGH_SPEED_EN**	**HANDSHAKE_EN**
Type														R/W	R/W	R/W
Reset														0	0	1

NEW_HANDSHAKE The default value of handshake is 0. The function of handshake is disabled. The function of BTIF has limitation. The ratio of bclk cannot be bigger than 2; otherwise, two system data transmissions will be wrong. If the value of handshake is 1, the ratio of bclk will be free.

HANDSHAKE_EN Enables handshake mode.

high_speed_en Enables high speed mode. Reserved.

RTO_EXT Extends the value of RX time-out counter (16*rto_time).

4 GPS

4.1 RF Part

4.1.1 LNA/Mixer

Upon receiving RF input signal in through either multi-GNSS antenna to internal LNA or external antenna and LNA, the mixer down converts the amplified signal (GPS/Galileo=1575.42MHz, Beidou=1561.098-MHz, GLONASS=1601.71-MHz). The current chip provides 2 configurations to choose from, which are high-gain LNA and low-gain LNA. The high-gain LNA is used for low-cost solution without external LNA. The low-gain LNA offers high linearity to allow high external LNA gain, with much worsen noise figure performance. In the application with external LNA, the external LNA gain ranging from 15 to 20 dB is recommended. The down-conversion mixer is single-ended passive mixer with current mode interface between the mixer and multi-modes low pass filter.

4.1.2 VCO/Synthesizer

The entire frequency synthesizer includes crystal oscillator, VCO, divider, phase frequency detector (PFD), charge pump (CP) and loop filter which are all integrated on the MT2503A chip. Upon power-on, VCO is auto-calibrated to its required sub-band. The synthesizer adopts fractional-N sigma-delta PLL topology, which supports 12.6 to 40MHz reference clock frequencies.

4.1.3 LPF

The current-mode LPF supports multiple modes for different GNSS combinations. The LPF also provides 26dB gain-control range, with approximately 2dB per step.

4.1.4 ADC

The differential IF signal is being quantized by a high performance ADC. The sampling clock can be provided from divided clock from LO.

4.2 Digital Part

4.2.1 ARM7EJ-S

The ARM7EJ-S processor provides flexibility necessary for building Java-enabled, real-time embedded devices requiring small size, low-power and high performance. It builds on the features and benefits of the established ARM7TDMI core and is delivered in synthesizable form. ARM7EJ-S is supported by a wide variety of development tools and can run at speeding up to 158 MHz.

ARM7EJ-S includes a JTAG interface which provides a standard development and debugging interface. The interface can connect to a variety of off-the-shelf emulators. The emulators provide single-step, trap and access to all the internal registers of the digital part of MT2503A.

4.2.2 Cache

MT2503A provides cache to speed up program execution and reduce external flash access times. It supports up to 64 Kbits cache buffer and can be used as internal memory when it is not fully used.

4.2.3 Boot ROM

The embedded boot ROM provides a function of loading a set of user code through the host interface into SRAM. The host interface (UART/SPI/I2C) is decided by strap control.

4.2.4 Real Time Clock (RTC)

MT2503A provides very low leakage battery backed-up memory, which contains all the necessary multi-GNSS information for quick start-up and a small amount of user configuration variables. There is a built-in 1.1 volts LDO for RTC domain and it can be bypassed while an external LDO is applied. The RTC LDO is a voltage regulator having very low quiescent current. The small ceramic capacitor can be used as the output capacitor, and the stable operation region ranges from very light load (~=0) to about 3 mA.

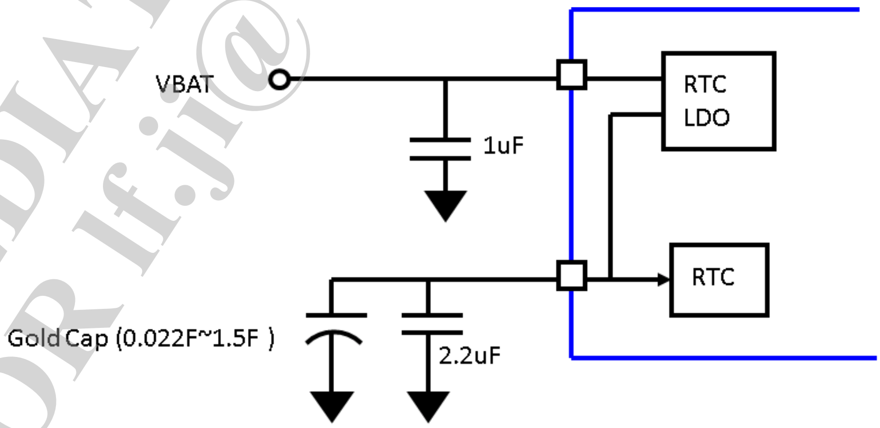


Figure 49. RTC with internal RTC LDO application circuit 1

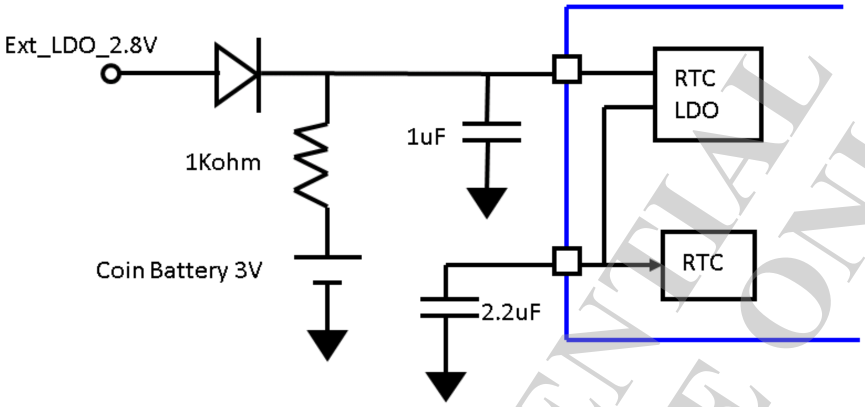


Figure 50. RTC with internal RTC LDO application circuit 2

4.2.5 SMPS

A built-in switching mode power supply provides 1.8 volts power supply for the digital 1.1 volts CLDO and RF input power. In the active mode, SMPS is operated in the PWM/PFM automatic mode. In the power saving mode, SMPS is operated with reduced switching frequency in the PFM mode. The recommended L/C value is 1 uH / 4.7 uF.

4.2.6 Timer function

The timer function supports a time tick generation of 31.25 ms resolution. With the 24-bit counter, the period of timer is from 31.25 ms to 524,287 s.

4.2.7 GPIO in RTC domain

The “32K_OUT” pin in RTC domain can output 32.768 KHz clock which can be used to support low clock rate operation mode for some applications or peripherals that need an external clock source. This pin can also be programmed to be an input pin to receive the signal from an external accelerator sensor IC to be the wake-up signal of MT2503A when it is in the low-power mode.

4.2.8 Low power detection

A low power detection circuit is implemented. Whenever the independent power source (AVDD11_RTC) becomes low voltage, the low power detection circuit will detect this condition and use an indicator signal (output high in normal condition and low in low-power condition) to reflect this condition.

4.2.9 Clock module

The clock module generates all internal clocks required by processor, correlator, internal memory, bus interface and so on. The referenced input clock is generated from the RF block. For system flexibility and maximum power saving, it supports various power management modes.

4.2.10 Reset controller

The built-in reset controller generates reset signals for all digital blocks. It has power-on reset feature and hardware trapping function. The power-on reset level is 2.7 ± 0.1 volts. The software reset function for different circuit blocks are also included for flexible applications.

In Figure 51, the voltage drop time T_{drop_vbat} and T_{drop_cldo} depend on the capacitance connection of their power net. But $T_{drop\_vbat} > T_{drop\_cldo}$ should be guaranteed for the correct operation of reset behavior during power off sequence. It is strongly recommend using external LDOs without output discharged function or make sure $T_{drop\_vbat} > 100$ ms.

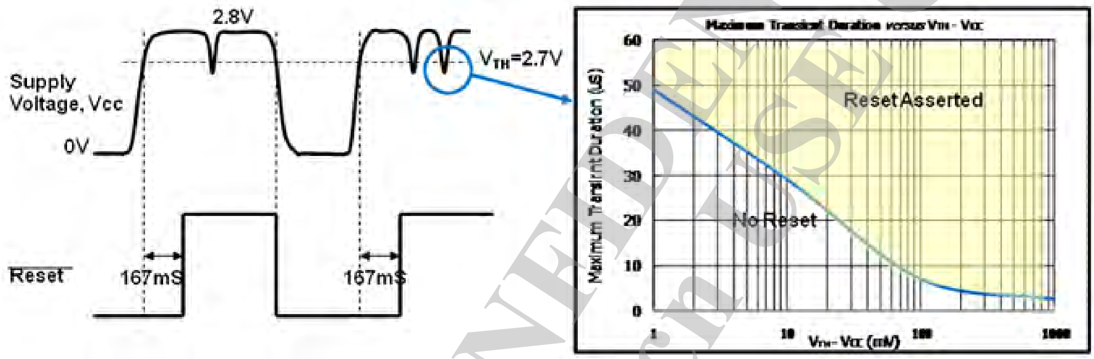


Figure 51. Power on reset diagram

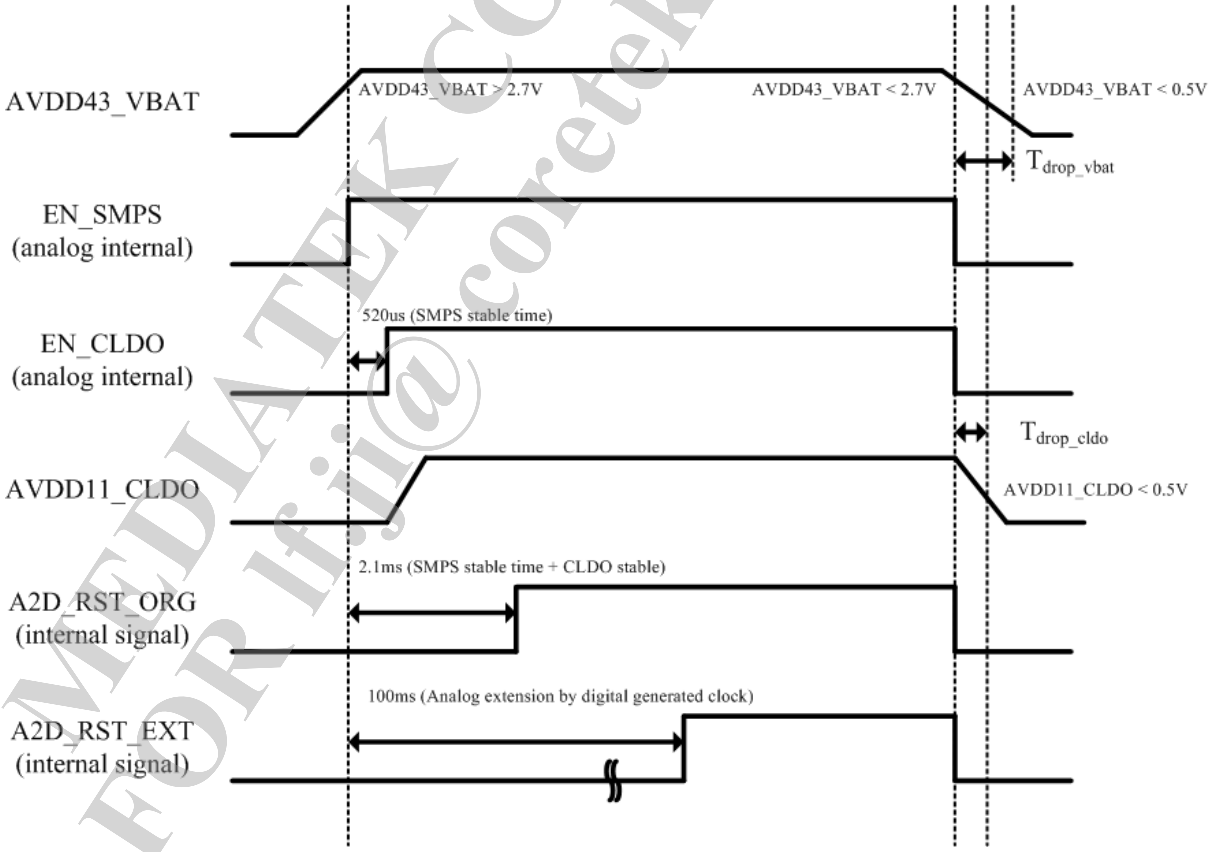


Figure 52. Power on/off reset behavior

4.2.11 Host interface

MT2503A supports 3 different host interfaces, which are UART, SPI, and I2C. The interface used as the host interface is determined by strap pins. Note that SPI and I2C support firmware update only for now.

4.2.11.1 UART

UART is the abbreviation of "Universal Asynchronous Receiver/Transmitter". MT2503A has 3 full duplex serial ports. It is used for serial data communication. A UART converts bytes of data to and from asynchronous start-stop bit streams represented as binary electrical impulses.

There are several functions in MT2503A related to UART communication, such as UART data transmission/receive and NMEA sentences input/output. In general, UART0 is as NMEA output and PMTK command input, UART1 as RTCM input. You can adjust the UART2 port as desired. The receiver (RX) and transmitter (TX) side of every port contains a 16-byte FIFO, but only UART0 has 256 bytes of URAM. The bit rates are selectable and range from 4.8 to 921.6 kbps. UART provides signal or message outputs.

4.2.11.2 SPI (by request)

The serial peripheral interface port manages the communication between digital BB and external devices. MT2503A supports both master and slave modes. Only 4 bytes of register in the master mode can be transferred. The slave has 4-byte-register mode or URAM mode. In the URAM mode, the transmitted and received data size is 256 bytes. The clock phase and clock polarity are selectable. MT2503A supports manual or automatic indicator for data transfer in the slave mode.

4.2.11.3 I2C (by request)

The I2C interface is mainly connected to external devices. MT2503A supports multi-master and slave modes. Both modes have 256-byte URAM mode and 8-byte FIFO mode for transmitting and receiving data. The multi-master mode supports 7-bit and 10-bit address modes up to 400 Kb/s fast mode and 3.4 Mb/s high-speed mode. In additions, MT2503A supports manual or automatic indicator for data transfer in the slave mode. Device addresses in the slave mode are programmable and support fast mode and high-speed mode data transmission and reception.

4.2.12 Interrupt control unit

The interrupt control unit manages all internal and external sources of interrupts, which include timer, watch-dog, all interfaces such as UART, I2C and SPI and external user interrupt pins. These interrupt sources can be wake-up events in the power saving mode.

4.2.13 Flash

An external SPI serial flash up to 128 Mb is supported. Specific MTK Flash Tool is also supported for downloading firmware into the internal flash.

4.2.14 GPIO unit

GPIO is the abbreviation of “General-Purpose Input/Output”. MT2503A supports a variety of peripherals through maximum 16 GPIO programmable ports. The unit manages all GPIO lines and supports a simple control interface. GPIO provides signal or message outputs.

4.2.15 PPS

The PPS (Pulse Per Second) signal is provided through designated output pin for many external applications. The pulse is not only limited to being active every second but also allowed to set up the required duration, frequency and active high/low by programming user-defined settings.

4.2.16 ECLK

ECLK is a clock input pin for introducing an external clock signal to MT2503A and obtaining the relation between the external clock and GPS local clock. With precise external clock input, the clock drift of the GPS local clock can be correctly estimated. Therefore, the doppler search range is narrowed down accordingly. The technology is beneficial to speeding up the satellite acquisition process. Particularly in the cold start case, due to limited priori information about the satellite’s location and local clock uncertainty, a receiver will execute a search in full frequency range. Consequently, a longer acquisition time is expected. However, the ECLK technology is able to reduce the frequency uncertainty so that the search process will be completed in a short time. Efficient acquisition and lower power consumption are attained by the ECLK technology.

4.2.17 SYNC

SYNC is a time stamp signal input pin for introducing an external timing to the GPS receiver and obtaining the relation between the external timing and the GPS receiver local timing. With precise external timing input and the established relation, the GPS time of week (TOW) can be correctly estimated in the GPS receiver. The technology is beneficial for time to first fix (TTFF), particularly in weak signal environments. In hot starts, with priori information about the GPS receiver’s location and satellite ephemeris data, the GPS receiver uses the correct GPS TOW to accurately predict the signal code chip/phase. Therefore, the code search range can be narrowed down accordingly. Hence, fast TTFF is achieved by the SYNC technology.

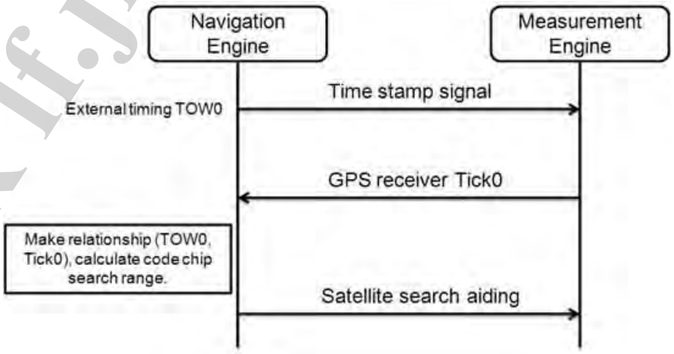


Figure 53. Flow diagram of SYNC function

4.2.18 Power scheme

Internal SMPS is used as the source power of the internal RF/BB LDO. It is also used as 1.8 volts I/O power, external TCXO/LNA voltage source via built-in TCXO switch. The internal SMPS can switch to the LDO mode to supply power to each of the about block

The minimum/maximum input voltage of AVDD43_VBAT and AVDD43_DCV is 2.8/4.3 volts.

The power-on reset voltage threshold of AVDD43_VBAT is 2.7 ± 0.1 volts. The maximum TLDO drop out voltage at half load (25 mA) is 0.2 volts. If one external LDO is used to provide power to MT2503A, the 3.3 volts external LDO will be recommended after taking TLDO drop-out into consideration.

The power efficiency in SMPS mode will be better than that in the internal LDO mode.

I/O supports 1.8 and 2.8 volts. The power comes from SMPS output for 1.8 volts application or TLDO output (AVDD28_TLDO) for 2.8 volts application.

The power for internal flash comes from AVDD28_TLDO.

TCXO power is from AVDD_TCXO_SW that can select either from AVDD28_TLDO (2.8V) or from AVDD28_CLDO (1.8V) by setting up power-on strap.

RTC LDO input power comes from backup battery or uses coin battery.

Here are 3 power schemes: low power (Figure 54), low cost (Figure 55) and external PMU (Figure 56).

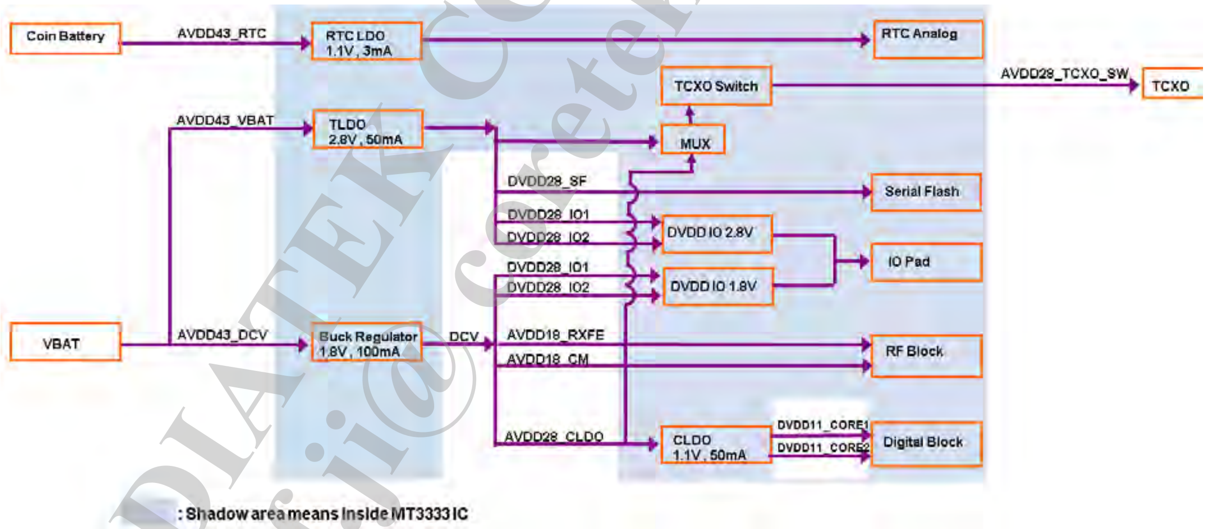


Figure 54. Power supply connection (low power)

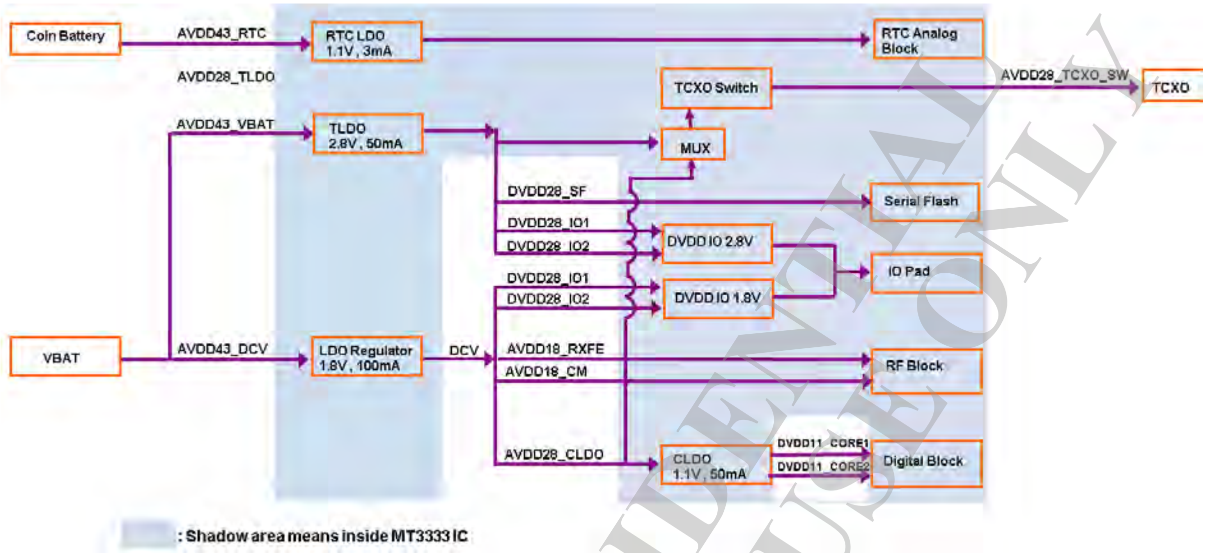


Figure 55. Power supply connection (low cost)

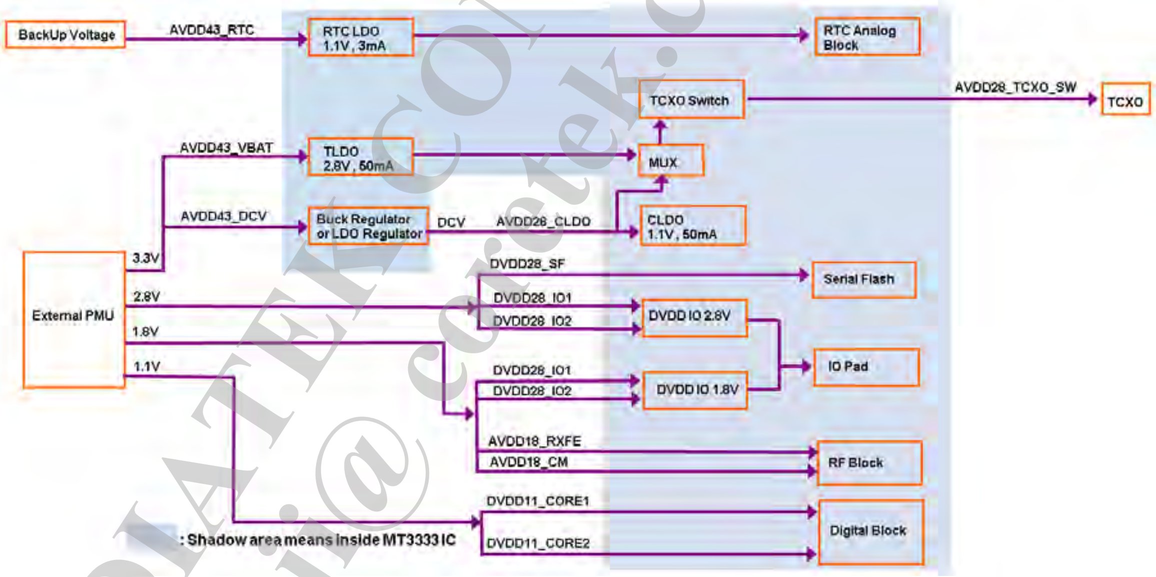


Figure 56. Power supply connection (external LDO)

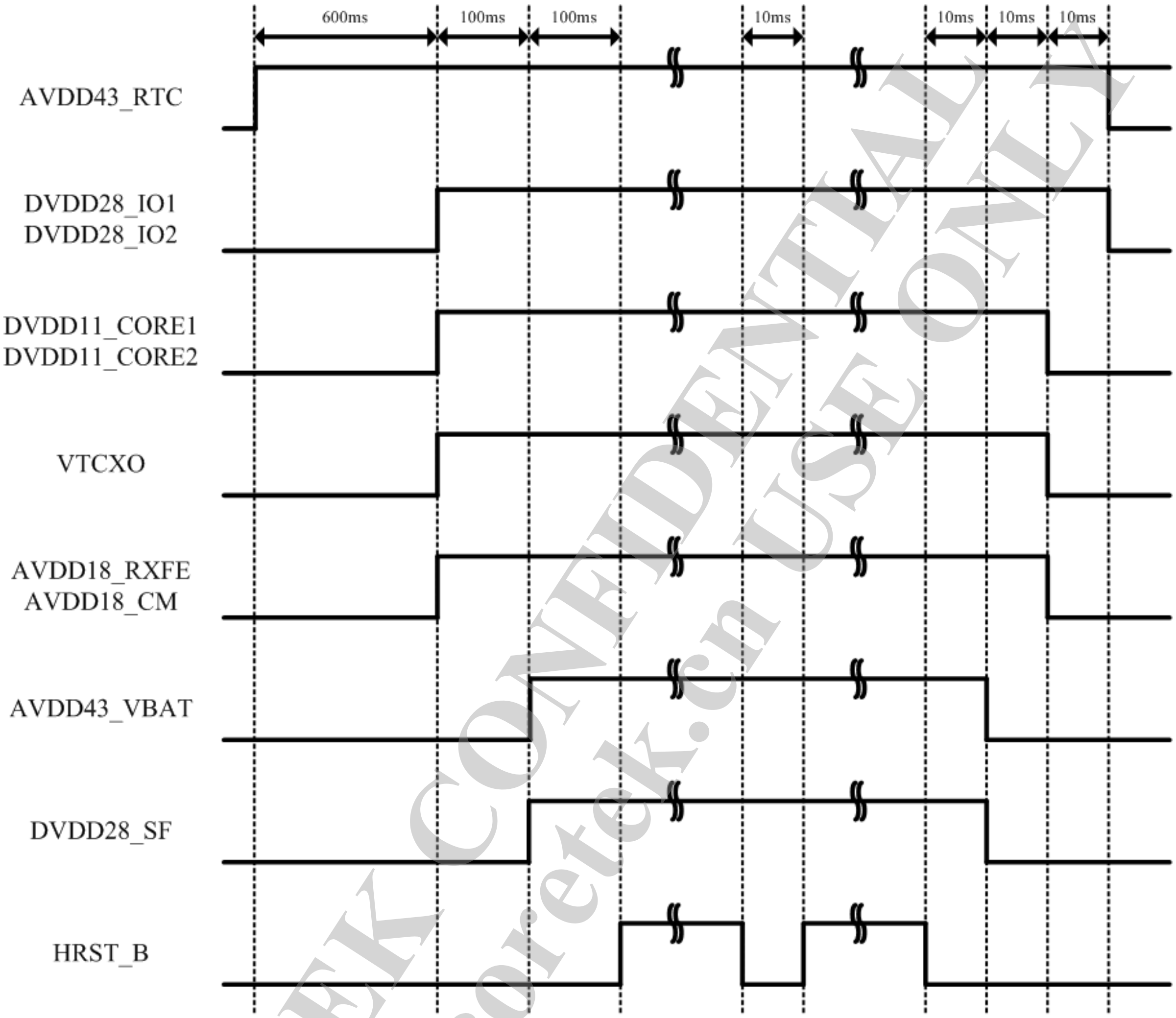


Figure 57. Power on/off sequence for external LDO mode

4.3 Electrical Characteristics

4.3.1 DC characteristics

4.3.1.1 Absolute maximum ratings

Symbol	Parameter	Rating	Unit
AVDD43_DCV	SMPS power supply	-0.3 ~ 4.3	V
AVDD43_VBAT	2.8 volts TLDO power supply	-0.3 ~ 4.3	V
AVDD28_CLDO	1.1 volts CLDO power supply	-0.3 ~ 3.6	V
DVDD28_SF	Embedded flash power supply	-0.3 ~ 3.6	V
DVDD28_IO1	IO 2.8/1.8 volts power supply	-0.3 ~ 3.6	V

DVDD28_IO2			
DVDD11_CORE1 DVDD11_CORE2	Baseband 1.1 volts power supply	-0.3 ~ 1.21	V
AVDD43_RTC	RTC 1.1 volts LDO power supply	-0.3 ~ 4.3	V
AVDD18_RXFE	1.8 volts supply for RF core circuits	-0.3 ~ 3.6	V
AVDD18_CM		-0.3 ~ 3.6	V
T_{STG}	Storage temperature	-50 ~ +125	°C
T_A	Operating temperature	-45 ~ +85	°C

4.3.1.2 Recommended operating conditions

Symbol	Parameter	Min.	Typ.	Max.	Unit
AVDD43_DCV	SMPS power supply	2.8	3.3	4.3	V
AVDD43_VBAT	2.8 volts TLDO power supply	2.8	3.3	4.3	V
DVDD11_CORE1 DVDD11_CORE2	1.1 volts baseband core power	0.99	1.1	1.21	V
DVDD28_IO1 DVDD28_IO2	2.8 volts digital I/O power	2.52	2.8	3.08	V
	1.8 volts digital I/O power	1.62	1.8	1.98	V
DVDD28_SF	Embedded flash power supply	2.7	2.8	3.6	V
AVDD18_RXFE	1.35 volts supply for RF core circuits in bypass mode	1.3	1.35	1.98	V
	1.8 volts supply for RF core circuits in LDO mode	1.62	1.8	3.08	V
AVDD18_CM	1.35 volts supply for common RF block in bypass mode	1.3	1.35	1.98	V
	1.8V volts supply for common RF block in LDO mode	1.62	1.8	3.08	V
T_A	Operating temperature	-40	25	85	˚C
T_j	Commercial junction operating temperature	0	25	115	˚C
	Industry junction operating temperature	-40	25	125	˚C

4.3.1.3 General DC characteristics

Symbol	Parameter	Condition	Min.	Max.	Unit
I_{IL}	Input low current	No pull-up or down	-1	1	uA
I_{IH}	Input high current	No pull-up or down	-1	1	uA
I_{OZ}	Tri-state leakage current		-10	10	uA

4.3.1.4 DC electrical characteristics for 2.8 volts operation

Symbol	Parameter	Condition	Min.	Typ.	Max.	Unit
VDD	Supply voltage of core power		0.99	1.1	1.21	V
VDDIO	Supply voltage of IO power		2.52	2.8	3.08	V
V_{IL}	Input lower voltage	LVTTL	-0.3	-	0.25*VDDIO	V
V_{IH}	Input high voltage		0.75*VDDIO	-	VDDIO+0.3	V
V_{OL}	Output low voltage	VDDIO = min I_{OL} = -2 mA	-	-	0.15*VDDIO	V
V_{OH}	Output high voltage	VDDIO = min I_{OH} = -2 mA	0.85*VDDIO	-	-	V
R_{PU}	Input pull-up resistance	VDDIO = typ Vinput = 0 V	40	85	190	KΩ
R_{PD}	Input pull-down resistance	VDDIO = typ Vinput = 2.8 V	40	85	190	KΩ

4.3.1.5 DC electrical characteristics for 1.8 volts operation

Symbol	Parameter	Condition	Min.	Typ.	Max.	Unit
VDD	Supply voltage of core power		0.99	1.1	1.21	V
VDDIO	Supply voltage of IO power		1.62	1.8	1.98	V
V_{IL}	Input lower voltage	LVTTL	-0.3	-	0.25*VDDIO	V
V_{IH}	Input high voltage		0.75*VDDIO	-	VDDIO+0.3	V
V_{OL}	Output low voltage	VDDIO = min I_{OL} = -2 mA	-	-	0.15*VDDIO	V
V_{OH}	Output high voltage	VDDIO = min I_{OH} = -2 mA	0.85*VDDIO	-	-	V
R_{PU}	Input pull-up resistance	VDDIO = typ Vinput = 0 V	70	150	320	KΩ
R_{PD}	Input pull-down resistance	VDDIO = typ Vinput = 1.8 V	70	150	320	KΩ

4.3.1.6 DC electrical characteristics for 1.1 volts operation (for FORCE_ON and 32K_OUT)

Symbol	Parameter	Condition	Min.	Typ.	Max.	Unit

Symbol	Parameter	Condition	Min.	Typ.	Max.	Unit
VDD	Supply voltage of core power		0.99	1.1	1.21	V
VDDIO	Supply voltage of IO power		0.99	1.1	1.21	V
V_{IL}	Input lower voltage	LVTTL	-0.3	-	0.25*VDDIO	V
V_{IH}	Input high voltage		0.75*VDDIO	-	VDDIO+0.3	V
V_{OL}	Output low voltage	VDDIO = min I_{OL} = -2 mA	-	-	0.15*VDDIO	V
V_{OH}	Output high voltage	VDDIO = min I_{OH} = -2 mA	0.85*VDDIO	-	-	V
R_{PU}	Input pull-up resistance	VDDIO = typ Vinput = 0 V	130		560	KΩ
R_{PD}	Input pull-down resistance	VDDIO = typ Vinput = 1.1 V	130		560	KΩ

4.3.2 Analog related characteristics

4.3.2.1 SMPS DC characteristics

Symbol	Parameter	Min.	Typ.	Max.	Unit	Note
AVDD43_DCV	SMPS input supply voltage	2.8	3.3	4.3	V	
DCV	SMPS output	1.74	1.84	1.94	V	
I_{cc}	SMPS output current	-	-	100	mA	
ΔV_PWM	Ripple of PWM mode	-	-	40	mV	With L=1uH, C=4.7uF
ΔV_PFM	Ripple of PFM mode	-	-	90	mV	With L=1uH, C=4.7uF

4.3.2.2 TCXO LDO DC characteristics

Symbol	Parameter	Min.	Typ.	Max.	Unit	Note
AVDD43_VBAT	TCXO LDO input supply voltage	2.8	3.3	4.3	V	Will change to bypass mode under 3.1 volts
AVDD28_TLDO	TCXO LDO output	2.71	2.8	2.89	V	
I_{cc}	LDO output current	-	-	50	mA	Not include external devices
	PSRR-30 KHz	35	-	-	dB	Co = 1 uF, ESR = 0.05, Iload = 25 mA
	Load regulation	-84	10	84	mV	

4.3.2.3 TCXO SWITCH DC characteristics

Symbol	Parameter	Min.	Typ.	Max.	Unit	Note

Symbol	Parameter	Min.	Typ.	Max.	Unit	Note
AVDD_TCXO_SW	TCXO switch output voltage @ TCXO switch input = AVDD28_TLDO	2.66	-	-	V	
AVDD_TCXO_SW	TCXO switch output voltage @ TCXO switch input = AVDD28_CLDO	1.71	-	-	V	
I_{max}	TCXO SWITCH current limit	-	-	30	mA	

4.3.2.4 1.1 volts core LDO DC characteristics

Symbol	Parameter	Min.	Typ.	Max.	Unit	Note
AVDD28_CLDO	1.2 volts LDO input supply voltage	1.62	1.8	3.08	V	
AVDD11_CLDO	1.1 volts LDO output	1.05	1.12	1.2	V	
I_{cc}	LDO output current	-	-	50	mA	
	Load regulation	-	-	-	mV	

4.3.2.5 1.1 volts RTC LDO DC characteristics

Symbol	Parameter	Min.	Typ.	Max.	Unit	Note
AVDD43_RTC	RTC LDO input supply voltage	2	4	4.3	V	
AVDD11_RTC	RTC LDO output	0.99	1.1	1.21	V	
I_{cc}	LDO output current	-	-	3	mA	
I_{leak}	Leakage current	2.2	10	-	uA	Including LDO and RTC domain circuit

4.3.2.6 32 KHz crystal oscillator (XOSC32)

Symbol	Parameter	Min.	Typ.	Max.	Unit	Note
AVDD11_RTC	Analog power supply	0.99	-	1.21	V	
Dcyc	Duty cycle	-	50	-	%	

4.3.3 RF related characteristics

4.3.3.1 DC electrical characteristics for RF part

Symbol	Parameter	Min.	Typ.	Max.	Unit
I_{cc}(GPS+GLONASS)	Total supply current:	-	8.9	-	mA

4.3.3.2 RX chain (GPS+GLONASS mode)

Parameter	Condition	Min.	Typ.	Max.	Unit
RF input frequency		-	1575.4	-	MHz
LO frequency		-	1588.6	--	MHz
LO leakage	Measured at balun matching network input at LNA high gain	-	-70	-	dBm
Input return loss	Differential input and external matched to 50Ω source using balun matching network for all gain	-10	-	-	dB
Gain (Av) (integrated average over Fc+-4M)	High current mode with max PGA gain	80	76	70	dB
	Low current mode with max PGA gain	-	64		dB
PGA Gain range		-	24	-	dB
PGA Gain step		-	2	-	dB
NF (integrated average over Fc+-4M)	High current mode with max PGA gain	-	2.2	-	dB

4.3.3.3 Crystal oscillator (XO)

Symbol	Parameter	Min.	Typ.	Max.	Unit
F_{tcxo}	TCXO oscillation frequency	12.6	16.368	40	MHz
V_{tcxo}	TCXO output swing	0.8	1.2	-	Vpp

4.4 Interface Characteristics

4.4.1 JTAG interface timing

Description	Symbol	Min.	Max.	Unit	Note
TDI input setup to rising TCK	T1	0.35T	-	ns	1
TDI input hold from rising TCK	T2	0.15T	-	ns	1
TMS input setup to rising TCK	T1	0.35T	-	ns	1
TMS input hold from rising TCK	T2	0.15T	-	ns	1
Rising TCK to TDO valid	T3	-	0.5T	ns	1
TDO hold from rising TCK	T4	0	-	ns	1

Note: The maximal condition of JTAG clock cycle (TCK) is 50 MHz.

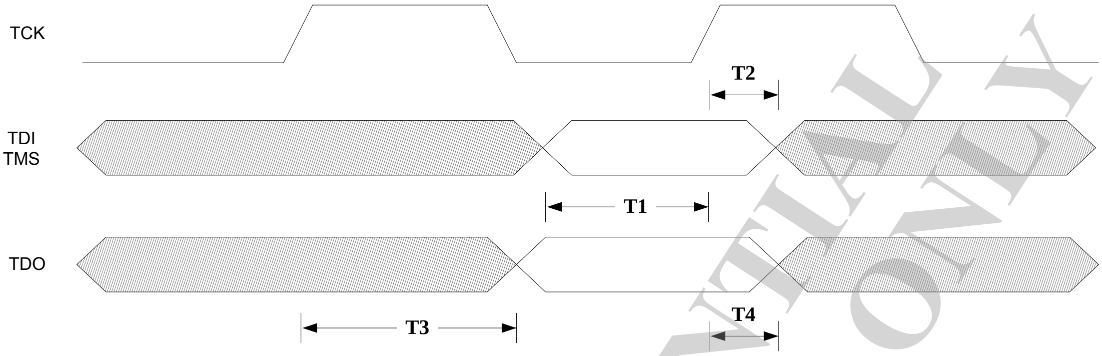


Figure 58. Timing diagram of JTAG interface

4.4.2 RS-232 interface timing

Baudrate required (bps)	Programmed baudrate (bps)	Baudrate error (%)	Baudrate error (%) 3
4,800	4,800.000	0.0000	0.002
9,600	9,600.000	0.0000	0.002
14,400	14,408.451	0.0587	0.0567
19,200	19,164.319	0.0587	0.0567
38,400	38,422.535	0.0587	0.0567
57,600	57,633.803	0.0587	0.0567
115,200	115,267.606	0.0587	0.0567
230,400	230,535.211	0.0587	0.0567
460,800	454,666.667	-1.3310	-1.3330
921,600	909,333.333	-1.3310	-1.3330

Notes:

1. UART baud-rate settings with UART_CLK frequency = 16.368 MHz (UART_CLK uses the reference clock of the system).
2. The baudrate error is optimized. Each baudrate needs to adjust counter to obtain the optimized error.

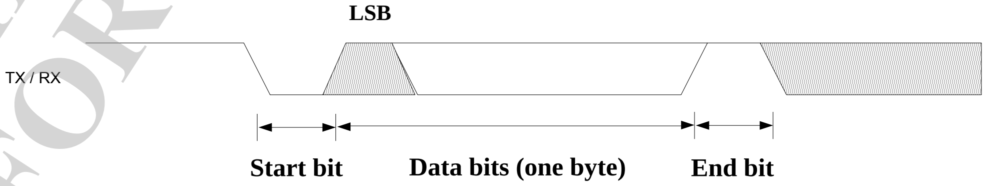


Figure 59. Timing diagram of RS-232 interface

4.4.3 SPI interface timing

Description	Symbol	Min.	Max.	Unit	Note
SCS# setup time	T1	0.5T	-	ns	1
SCS# hold time	T2	0.5T	-	ns	1
SO setup time	T3	0.5T – 3t	0.5T - 2t	ns	1, 2
SO hold time	T4	0.5T + 2t	0.5T + 3t	ns	1, 2
SIN setup time	T5	3t	-	ns	1, 2
SIN hold time	T6	10	-	ns	1

Notes:

1. The condition of SPI clock cycle (T) is (SPI_IPLL/12) MHz ~ (rf_clk/1,020) MHz.
2. t indicates the period of SPI controller clock, which is SPI_IPLL clock or rf_clk.

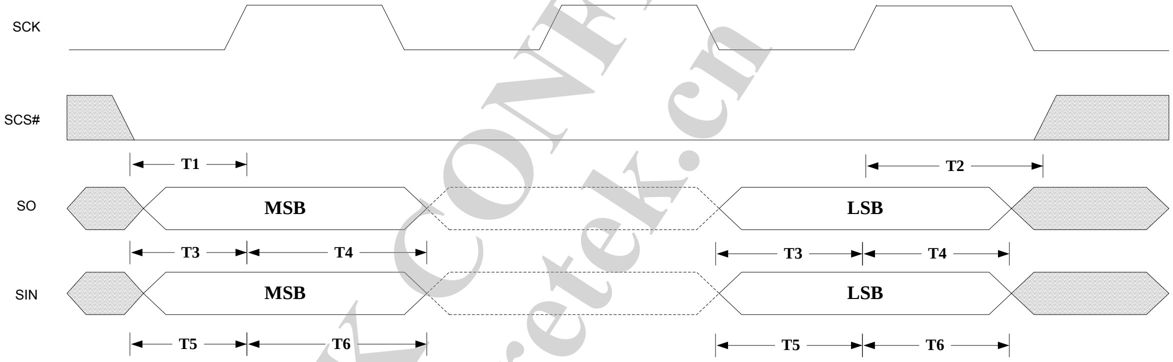


Figure 60. Timing diagram of SPI interface

4.4.4 I2C interface timing

Symbol	Period
T1	(MM_CNT_PHASE_VAL0+1)/TCXO_CLK
T2	(MM_CNT_PHASE_VAL1+1)/TCXO_CLK
T3	(MM_CNT_PHASE_VAL2+1)/TCXO_CLK
T4	(MM_CNT_PHASE_VAL3+1)/TCXO_CLK

Note: The condition of I2C clock cycle (I2C_CLK) is (TCXO_CLK/4) MHz ~ (TCXO_CLK/(MM_CNT+4)) MHz. The MM_CNT is sum of MM_CNT_PHASE_VAL0, MM_CNT_PHASE_VAL1, MM_CNT_PHASE_VAL2 and MM_CNT_PHASE_VAL3 in full speed mode.

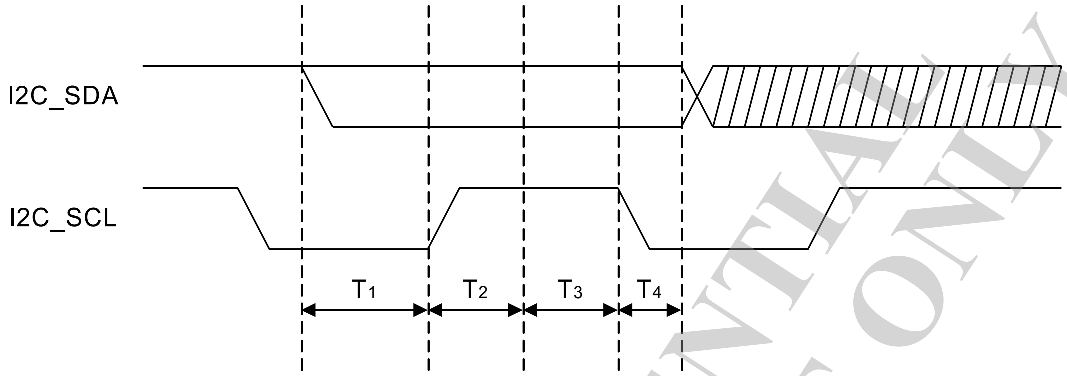


Figure 61.Timing diagram of HOST I2C interface