12. I/O Interface

This part describes S905Y2's I/O interfaces from the following aspects:

- USB
- SDIO
- Transport Interface and Transport Stream Demux
- IR Remote
- SAR ADC
- I2C
- UART
- PWM

12.1 Universal Serial Bus

12.1.1 Overview

The chip integrates one USB XHCI OTG 2.0 ports, one USB3.0 and PCIe 2.0 combo interface up to 5Gbps, supports 2 configurations:

- 1 USB2.0 OTG + 1 USB 2.0 Host + 1 PCIe
- 1 USB2.0 OTG + 1 USB3.0 (No PCIe)

12.1.2 Features

The USB2.0 OTG controller features:

- Support for the following speeds: High-Speed (HS, 480-Mbps),Full-Speed (FS, 12-Mbps) and Low-Speed (LS, 1.5-Mbps) modes
- Multiple DMA/non DMA mode access support on the application side
- Supports up to 16 bidirectional endpoints, including control endpoint 0.
- Supports Session Request Protocol (SRP) and Host Negotiation Protocol (HNP)
- Supports up to 16 host channels.
- Supports ACA ID detector, refer to Amlogic USB2.0 OTG ID Detector Specification.pdf for detail

The USB2.0 Host controller features:

- Support for the following speeds: High-Speed (HS, 480-Mbps),Full-Speed (FS, 12-Mbps) and Low-Speed (LS, 1.5-Mbps) modes
- Multiple DMA/non DMA mode access support on the application side
- Supports up to 16 host channels.

The USB2.0 PHY features:

- Support for the following speeds: High-Speed (HS, 480-Mbps),Full-Speed (FS, 12-Mbps) and Low-Speed (LS, 1.5-Mbps) modes

The USB3.0 Host controller features:

- Support for the following speed: Super-Speed, High-Speed, Full-Speed and Low-Speed
- Compliant with the xHCI specification
- The USB3.0 SS PHY features:
- 5-Gbps SuperSpeed data transmission rate over 3-m USB 3 cable
- Integrated PHY includes transmitter, receiver, spread spectrum clock (SSC) generation, PLL, digital core, and electrostatic discharge (ESD) protection circuits
- Supports legacy Half-rate mode for power-saving

12.1.3 Register Description

Base address:

PHY 20: 0xff63_6000.

PHY 21: 0xff63_A000.

For the following registers, each register's final address= base address + offset *4.

reg32_00 0x00

Bit(s)	R/W	Default	Description
31	R/W	0	PLL_Fine_Tuning_2
30	R/W	1	PLL_Fine_Tuning_1
29	R/W	0	PLL_Fine_Tuning_0
28	R/W	0	PLL_Bypass_Enable
27	R/W	0	PLL_Lock_over_ride
26	R/W	0	pll_clock_divide_5
25	R/W	0	pll_clock_divide_4
24	R/W	0	pll_clock_divide_3
23	R/W	0	pll_clock_divide_2
22	R/W	0	pll_clock_divide_1
21	R/W	0	pll_clock_divide_0
20	R/W	0	refclk_multiplier_5
19	R/W	0	refclk_multiplier_4
18	R/W	1	refclk_multiplier_3
17	R/W	0	refclk_multiplier_2
16	R/W	1	refclk_multiplier_1
15	R/W	0	refclk_multiplier_0
14	R/W	1	reset_FS_LS_Clock_Divider
13	R/W	1	reset_HS_CDR
12	R/W	1	reset_FS_LS_CDR
11:10	R/W	0	reg32_00_11_10_reserved
9	R/W	0	BIAS_Power_Down
8	R/W	0	BGR_Power_Down
7	R/W	0	Calibration_Power_Down
6	R/W	0	HS_Disconnect_Power_Down

Bit(s)	R/W	Default	Description
5	R/W	0	HS_Squelch_Power_Down
4	R/W	0	FS_LS_RX_Power_Down
3	R/W	0	HS_RX_Power_Down
2	R/W	0	FS_LS_Driver_Power_Down
1	R/W	0	HS_TX_Driver_Power_Down
0	R/W	0	PLL_Power_Down

reg32_01 0x04

Bit(s)	R/W	Default	Description
31:24	R/W	0	Reserved
23:16	R/W	0	Reserved
15:10	R/W	0	Reserved
9	R/W	0	hs_en_mode
8	R/W	0	spare
7:4	R/W	0	Reserved
3:2	R/W	0	bypass_en
1:0	R/W	0x3	slew_control

reg32_02 0x08

Bit(s)	R/W	Default	Description
31	RO	0	Phy_status
30	RO	0	cal_en_flag
29:27	RO	0	reg32_02_29_27_reserved
26	RO	0	HS_Disconnect_Status
25	RO	0	HS_Squelch_Status
24	RO	0	PRBS_Sync_Out
23:16	RO	0	Calibration_code_Value_23_16

Bit(s)	R/W	Default	Description
15:8	RO	0	Calibration_code_Value_15_8
7:0	RO	0	Calibration_code_Value_7_0

reg32_03 0x0c

Bit(s)	R/W	Default	Description
31:24	R/W	0	Reserved
23:16	R/W	0	Reserved
15:8	R/W	0	Reserved
7:4	R/W	0x2	Reserved
3:2	R/W	0x1	hsdic_ref
1:0	R/W	0	squelch_ref

reg32_04 0x10

Bit(s)	R/W	Default	Description
31:30	R/W	0	i_c2l_bias_trim_3_2
29:28	R/W	0	i_c2l_bias_trim_1_0
27	R/W	0	TEST_Bypass_mode_enable
26	RO	0	i_c2l_cal_done
25	R/W	0	i_c2l_cal_reset_n
24	R/W	0	i_c2l_cal_en
23:16	R/W	0	Calibration_code_Value_23_16
15:8	R/W	0	Calibration_code_Value_15_8
7:0	R/W	0	Calibration_code_Value_7_0

reg32_05 0x14

Bit(s)	R/W	Default	Description
31:24	R/W	0	i_c2l_obs_7_0

Bit(s)	R/W	Default	Description
23:22	R/W	0	reg32_05_23_22_reserved
21:20	R/W	0	i_c2l_pll_perfcfg_21_20
19	R/W	0x1	i_c2l_pll_perfcfg_19
18:16	R/W	0	i_c2l_pll_perfcfg_18_16
15:12	R/W	0	i_c2l_pll_perfcfg_15_12
11:10	R/W	0	i_c2l_pll_perfcfg_11_10
9	R/W	0x1	i_c2l_pll_perfcfg_9
8	R/W	0x1	i_c2l_pll_perfcfg_8
7:4	R/W	0x7	i_c2l_pll_perfcfg_7_4
3:2	R/W	0x3	i_c2l_pll_perfcfg_3_2
1:0	R/W	0	i_c2l_pll_perfcfg_1_0

reg32_06 0x18

Bit(s)	R/W	Default	Description
31	R/W	0x1	hub_extra_bit_cntr
30:24	R/W	0	cntr_timeout
23	R/W	0	Internal_loopback
22	R/W	0	reg32_06_22_reserved
21	R/W	0	PCS_Reset_Transmit_State_machine
20	R/W	0	PCS_Reset_Receive_State_machine
19:16	R/W	0xf	fsls_farend_device_disconnect_micro_second_count_11_8
15:12	R/W	0xa	reg32_06_15_12_reserved
11:8	R/W	0	bypass_disc_cntr_3_0
7:0	R/W	0x17	PCS_microsecond_timer_done_count_value_7_0

reg32_07 0x1c

Bit(s)	R/W	Default	Description
31:24	RO	0	Prbs_Error_count
23:21	R/W	0	reg32_07_23_21_reserved
20:17	R/W	0xf	RX_ERROR_Turn_Around_Timer_Count
16	R/W	0	acceptable_bit_drops
15	R/W	0	host_tristate
14:12	R/W	0x4	fs_ls_minimum_count
11:8	R/W	0xf	cntr_done_value_7_4
7:4	R/W	0xf	cntr_done_value_3_0
3:0	R/W	0	HS_CDR_internal_tap_select

reg32_08 0x20

Bit(s)	R/W	Default	Description
31:24	R/W	0	Custom_Pattern_2
23:16	R/W	0	Custom_Pattern_1
15:8	R/W	0x4	Custom_Pattern_0
7	R/W	0	Enable_RX_ERROR_Timeout_Mode
6	R/W	0	reset_us_timer
5	R/W	0	PRBS_ERROR_Insert
4	R/W	0	PRBS_comparison_enable
3	R/W	0	PRBS_Enable
2:0	R/W	0	pattern

reg32_09 0x24

Bit(s)	R/W	Default	Description
31:24	R/W	0	Custom_Pattern_6
23:16	R/W	0	Custom_Pattern_5
15:8	R/W	0	Custom_Pattern_4

7:0	R/W	0	Custom_Pattern_3

reg32_10 0x28

Bit(s)	R/W	Default	Description
31:24	R/W	0	Custom_Pattern_10
23:16	R/W	0	Custom_Pattern_9
15:8	R/W	0	Custom_Pattern_8
7:0	R/W	0	Custom_Pattern_7

reg32_11 0x2c

Bit(s)	R/W	Default	Description
31:24	R/W	0	Custom_Pattern_14
23:16	R/W	0	Custom_Pattern_13
15:8	R/W	0	Custom_Pattern_12
7:0	R/W	0	Custom_Pattern_11

reg32_12 0x30

Bit(s)	R/W	Default	Description
31:24	R/W	0	Custom_Pattern_18
23:16	R/W	0	Custom_Pattern_17
15:8	R/W	0	Custom_Pattern_16
7:0	R/W	0	Custom_Pattern_15

reg32_13 0x34

Bit(s)	R/W	Default	Description
31	R/W	0	reg32_13_31_reserved
30	R/W	0	i_c2l_fsls_rx_en
29	R/W	0	i_c2l_hs_rx_en
28	R/W	0	i_c2l_fs_oe

Bit(s)	R/W	Default	Description
27	R/W	0	i_c2l_hs_oe
26	R/W	0	i_c2l_ls_en
25	R/W	0	i_c2l_fs_en
24	R/W	0	i_c2l_hs_en
23	R/W	0	Bypass_Host_Disconnect_Enable
22	R/W	0	Bypass_Host_Disconnect_Value
21	R/W	0	Clear_Hold_HS_disconnect
20:16	R/W	0xc	minimum_count_for_sync_detection
15	R/W	0	Update_PMA_signals
14	R/W	0	load_stat
13:8	R/W	0	reg32_13_13_8_reserved
7:0	R/W	0	Custom_Pattern_19

reg32_14 0x38

Bit(s)	R/W	Default	Description
31:24	R/W	0	reg32_14_31_24_reserved
23:16	R/W	0	Bypass_ctrl_15_8
15:8	R/W	0	Bypass_ctrl_7_0
7	R/W	0	reg32_14_7_reserved
6	R/W	0	i_c2l_assert_single_enable_zero
5	R/W	0	i_c2l_data_16_8
4	R/W	0	pg_rstn
3:2	R/W	0	i_rpu_sw2_en
1	R/W	0	i_rpu_sw1_en
0	R/W	0	i_rpd_en

reg32_15 0x3c

Bit(s)	R/W	Default	Description
31:29	R/W	0	reg32_15_31_29_reserved
28:16	R/W	0xfa0	ms_4_cntr
15:8	R/W	0x3c	non_se0_cntr
7:0	R/W	0x3c	se0_cntr

reg32_16 0x40

Bit(s)	R/W	Default	Description
31	RO	0	usb2_mppll_lock_dig
30	RO	0	usb2_mppll_lock
29	R/W	0	usb2_mppll_reset
28	R/W	0	usb2_mppll_en
27	R/W	0x1	usb2_mppll_fast_lock
26	R/W	0	usb2_mppll_lock_f
25:24	R/W	0x1	usb2_mppll_lock_long
23	R/W	0	usb2_mppll_dco_sdm_en
22	R/W	0x1	usb2_mppll_load
21	R/W	0	usb2_mppll_sdm_en
20	R/W	0	usb2_mppll_tdc_mode
19:15	R/W	0	reg32_16_19_15_reserved
14:10	R/W	0x1	usb2_mppll_n
9	R/W	0	reg32_16_9_reserved
8:0	R/W	0x14	usb2_mppll_m

reg32_17 0x44

Bit(s)	R/W	Default	Description
31:28	R/W	0x9	usb2_mppll_filter_pvt1

Bit(s)	R/W	Default	Description
27:24	R/W	0x2	usb2_mppll_filter_pvt2
23	R/W	0	usb2_mppll_filter_mode
22:20	R/W	0x7	usb2_mppll_lambda0
19:17	R/W	0x7	usb2_mppll_lambda1
16	R/W	0	usb2_mppll_fix_en
15:14	R/W	0	reg32_17_15_14_reserved
13:0	R/W	0	usb2_mppll_frac_in

reg32_18 0x48

Bit(s)	R/W	Default	Description
31	R/W	0	usb2_mppll_acg_range
30:29	R/W	0x3	usb2_mppll_adj_ldo
28:26	R/W	0x3	usb2_mppll_alpha
25:24	R/W	0x1	usb2_mppll_bb_mode
23:22	R/W	0x1	usb2_mppll_bias_adj
21:19	R/W	0x3	usb2_mppll_data_sel
18:16	R/W	0x3	usb2_mppll_rou
15:14	R/W	0	usb2_mppll_pfd_gain
13	R/W	0x1	usb2_mppll_dco_clk_sel
12	R/W	0	usb2_mppll_dco_m_en
11:6	R/W	0x27	usb2_mppll_lk_s
5:2	R/W	0x9	usb2_mppll_lk_w
1:0	R/W	0x1	usb2_mppll_lkw_sel

reg32_19 0x4c

Bit(s)	R/W	Default	Description
31	RO	0	usb2_mppll_lock_dig

Bit(s)	R/W	Default	Description
30	RO	0	usb2_mppll_lock
29:10	RO	0	reg32_19_29_10_reserved
9:0	RO	0	usb2_mppll_reg_out

reg32_20 0x50

Bit(s)	R/W	Default	Description
31	R/W	0	bypass_cal_done_r5
30:29	R/W	0	usb2_bgr_dbg_1_0
28:24	R/W	0	usb2_bgr_vref_4_0
23:22	R/W	0	reg32_20_23_22_reserved
21	R/W	0	usb2_bgr_start
20:16	R/W	0	usb2_bgr_adj_4_0
15:14	R/W	0	usb2_edgedrv_trim_1_0
13	R/W	0	usb2_edgedrv_en
12:9	R/W	0xf	usb2_dmon_sel_3_0
8	R/W	0	usb2_dmon_en
7	R/W	0	bypass_otg_det
6	R/W	0	usb2_cal_code_r5
5	R/W	0	usb2_amon_en
4	R/W	0x1	usb2_otg_vbusdet_en
3:1	R/W	0x4	usb2_otg_vbus_trim_2_0
0	R/W	0	usb2_otg_iddet_en

reg32_21 0x54

Bit(s)	R/W	Default	Description
31:26	R/W	0	reg32_21_31_26_reserved
25:20	R/W	0	bypass_utmi_reg

Bit(s)	R/W	Default	Description
19:16	R/W	0	bypass_utmi_cntr
15:6	R/W	0	reg32_21_15_6_reserved
5:4	R/W	0x2	usb2_otg_aca_trim_1_0
3	R/W	0	usb2_tx_strg_pd
2	R/W	0x1	usb2_otg_aca_en
1	R/W	0x1	usb2_cal_ack_en
0	R/W	0	usb2_bgr_force

reg32_22 0x58

Bit(s)	R/W	Default	Description
31:6	RO	0	reg32_22_31_6_reserved
5:3	RO	0	usb2_otg_aca_iddig
2	RO	0	usb2_otg_vbus_vld
1	RO	0	usb2_otg_sess_vld
0	RO	0x2	usb2_otg_id_dig

reg32_23 0x5c

Bit(s)	R/W	Default	Description
31:16	RO	0	test_bus_data_int_15_0
15:8	R/W	0	reg32_23_15_8_reserved
7	R/W	0	orw_test_bus_en
6:1	R/W	0	orw_test_bus_sel_5_0
0	R/W	0	orw_usb2_bgr_en

12.2 PCIe

PCIe module includes PCIe controller and PCIe PHY.

PCIe controller features:

- Up to x16 Gen1, Gen2, Gen3, or Gen4 lanes (x1, x2, x4, x8, or x16)
- 32, 64, 128, or 256-bit Internal Datapath Operating at 62.5, 125, 250, 500, or 1000 MHz
- Automatic Integration With Most PHY types (8-bit, 16-bit, 32-bit, and 64-bit PIPE Width per Lane)

PCIe PHY features:

- 5-Gbps data transmission rate
- Supports one, two, three, or four PIPEs mapped onto 1, 2, 4, 8, or 16 lanes

12.3 Inter-Integrated Circuit (I2C)

12.3.1 Overview

Inter-Integrated Circuit (IIC or I2C) is a multi-slave serial communication bus between ICs. S905Y2 integrates the I2C interface and signals allowing communications with other I2C peripheral devices.

12.3.2 Features

The I^2C Master Module has the following features:

- Support for 7-bit and 10-bit addressable devices
- Programmable bus speed including standard speed (100kBits/s) and fast speed (400kBits/sec)
- Error transfer detection
- “Transfer complete” indication by polling or interrupt (Interrupts handled by the ISA module. See the ISA module for details).
- Internal buffer holding up to 8 bytes for transfer (in either direction)
- Flexible architecture allowing the software to dictate the format of the I^2C bit streams
- Manual setting of the I^2C bus to accommodate a software only mode

12.3.3 Register Description

For I2C module in EE domain, each register final address = 0x FF805000 + offset * 4 for master mode, final address = 0x FF806000 + offset * 4 for slave mode.

I2C_M_0_CONTROL_REG 0x7c00

Bit(s)	R/W	Default	Description
31	R/W	0	CNTL_JIC: There is internal logic to dynamically enable the gated clocks. If this gated clock logic doesn’t work, you can set This bit to always enable the clock. Setting This bit wastes power.
30	R	0	Unused
29-28	R/W	0	QTR_CLK_EXT: These two Bits extend the clock divider to 12 Bits: QTR_CLK = {[29:28],[21:12]}
27	R	0	unused
26	R	0	Read back level of the SDA line
25	R	0	Read back level of the SCL line
24	R/W	0	Sets the level of the SDA line if manual mode is enabled. If This bit is ‘0’, then the SDA line is pulled low. If This bit is ‘1’ then the SDA line is tri-stated.
23	R/W	0	Sets the level of the SCL line if manual mode is enabled. If This bit is ‘0’, then the SCL line is pulled low. If This bit is ‘1’ then the SCL line is tri-stated.
22	R/W	0	This bit is used to enable manual mode. Manual I2C mode is controlled by Bits 12,13,14 and 15 above.

Bit(s)	R/W	Default	Description
21:12	R/W	0x142	QTR_CLK_DLY. This value corresponds to period of the SCL clock divided by 4 Quarter Clock Delay = * System Clock Frequency For example, if the system clock is 133Mhz, and the I2C clock period is 10uS (100khz), then Quarter Clock Delay = * 133 Mhz = 332
11:8	R	-	READ_DATA_COUNT: This value corresponds to the number of bytes READ over the I2C bus. If this value is zero, then no data has been read. If this value is 1, then Bits [7:0] in TOKEN_RDATA_REG0 contains valid data. The software can read this register after an I2C transaction to get the number of bytes to read from the I2C device.
7:4	R	-	CURRENT_TOKEN: This value reflects the current token being processed. In the event of an error, the software can use this value to determine the error location.
3	R	-	ERROR: This read only Bit is set if the I2C device generates a NACK during writing. This bit is cleared at on the clock cycle after the START Bit is set to 1 indicating the start of list processing. Errors can be ignored by setting the ACK_IGNORE Bit(s) below. Errors will be generated on Writes to devices that return NACK instead of ACK. A NACK is returned by a device if it is unable to accept any more data (for example because it is processing some other real-time function). In the event of an ERROR, the I2C module will automatically generate a STOP condition on the bus.
2	R	-	STATUS: This bit reflects the status of the List processor: 0: IDLE 1: Running. The list processor will enter this state on the clock cycle after the START Bit is set. The software can poll the status register to determine when processing is complete.
1	R/W	0	ACK_IGNORE: Set to 1 to disable I2C ACK detection. The I2C bus uses an ACK signal after every byte transfer to detect problems during the transfer. Current Software implementations of the I2C bus ignore this ACK. This bit is for compatibility with the current Amlogic software. This bit should be set to 0 to allow NACK operations to abort I2C bus transactions. If a NACK occurs, the ERROR bit above will be set.
0	R/W	0	START: Set to 1 to start list processing. Setting This bit to 0 while the list processor is operating causes the list processor to abort the current I2C operation and generate an I2C STOP command on the I2C bus. Normally This bit is set to 1 and left high until processing is complete. To re-start the list processor with a new list (after a previous list has been exhausted), simply set This bit to zero then to one.

I2C_M_0_SLAVE ADDRESS 0x7c01

Bit(s)	R/W	Default	Description
31:29	R	0	Reserved
28	R/W	0	USE_CNTL_SCL_LOW: If This bit is set to 1, then Bits[27:16] control the SCL low time.
27:16	R/W	0	SCL Low delay.
15:14	R	0	Unused
13-11	R/W	0	SCL_FILTER: A filter was added in the SCL input path to allow for filtering of slow rise times. 0 = no filtering, 7 = max filtering

Bit(s)	R/W	Default	Description
10:8	R/W	0	SDA FILTER: A filter was added in the SDA input path to allow for filtering of slow rise times. 0 = no filtering, 7 = max filtering
7:0	R/W	0x00	SLAVE_ADDRESS. This is a 7-bit value for a 7-bit I2C device, or (0xF0 | {A9,A8}) for a 10-bit I2C device. By convention, the slave address is typically stored in by first left shifting it so that it's MSB is D7 (The I2C bus assumes the 7-bit address is left shifted one). Additionally, since the SLAVE address is always an 7-bit value, D0 is always 0. NOTE: The I2C always transfers 8-bits even for address. The I2C hardware will use D0 to dictate the direction of the bus. Therefore, D0 should always be '0' when this register is set.

I2C_M_0_TOKEN_LIST_REG0 0x7c02

The register below describes the first 8 tokens in the token list.

Bit(s)	R/W	Default	Description
31:28	R/W	0x00	8th token in the list to process
27:24	R/W	0x00	7th token in the list to process
23:20	R/W	0x00	6th token in the list to process
19:16	R/W	0x00	5th token in the list to process
15:12	R/W	0x00	4th token in the list to process
11:8	R/W	0x00	3rd token in the list to process
7:4	R/W	0x00	2nd token in the list to process
3:0	R/W	0x00	1st token in the list to process (See the table below for token definitions)

Table 12-1 Token Definitions

Command Token	Value	Data	Description
END	0x0	N/A	Used to tell the I2C module that this is the end of the Token list. This token is not associated with the I2C bus, but rather with the state-machine that drives the token list processor.
START	0x1	N/A	The START Token is used to tell an I2C device that this is the beginning of an I2C transfer
SLAVE_ADDR-WRITE	0x2	7-bits	This bit-sequence is used to address a device and tell the device it is being WRITTEN
SLAVE_ADDR-READ	0x3	7-bits	This bit sequence is used to address a device and tell the device it is being READ.

Command Token	Value	Data	Description
DATA	0x4	8-bits	This 8-bit byte sequence is a byte transfer (READ or WRITE). The DATA token corresponds to a WRITE if it follows a SLAVE_ADDR-WRITE token. The DATA token corresponds to a READ if it follows a SLAVE_ADDR-READ token.
DATA-LAST	0x5	8-bits	Used to indicate the last 8-bit byte transfer is a byte transfer of a READ.
STOP	0x6	N/A	This tells the I2C device it is no longer being addressed

Write data associated with the DATA token should be placed into the I2C_TOKEN_WDATA_REG0 or I2C_TOKEN_WDATA_REG1 registers. Read data associated with the DATA or DATA-LAST token can be read from the I2C_TOKEN_RDATA_REG0 or I2C_TOKEN_RDATA_REG1 registers.

I2C_M_0_TOKEN_LIST_REG1 0x7c03

Bit(s)	R/W	Default	Description
31:28	R/W	0x00	16th token in the list to process
27:24	R/W	0x00	15th token in the list to process
23:20	R/W	0x00	14th token in the list to process
19:16	R/W	0x00	13th token in the list to process
15:12	R/W	0x00	12th token in the list to process
11:8	R/W	0x00	11th token in the list to process
7:4	R/W	0x00	10th token in the list to process
3:0	R/W	0x00	9th token in the list to process

I2C_M_0_TOKEN_WDATA_REG0 0x7c04

Bit(s)	R/W	Default	Description
31:24	R/W	0x00	4th data byte written for a DATA (write) token.
23:16	R/W	0x00	3rd data byte written for a DATA (write) token.
15:8	R/W	0x00	2nd data byte written for a DATA (write) token.
7:0	R/W	0x00	1st data byte written for a DATA (write) token.

I2C_M_0_TOKEN_WDATA_REG1 0x7c05

Bit(s)	R/W	Default	Description
31:24	R/W	0x00	8th data byte written for a DATA (write) token.

23:16	R/W	0x00	7th data byte written for a DATA (write) token.
15:8	R/W	0x00	6th data byte written for a DATA (write) token.
7:0	R/W	0x00	5th data byte written for a DATA (write) token.

I2C_M_0_TOKEN_RDATA_REG0 0x7c06

Bit(s)	R/W	Default	Description
31:24	R/W	0x00	4th data byte read for a DATA or DATA-LAST (READ) token.
23:16	R/W	0x00	3rd data byte read for a DATA or DATA-LAST (READ) token.
15:8	R/W	0x00	2nd data byte read for a DATA or DATA-LAST (READ) token.
7:0	R/W	0x00	1st data byte read for a DATA or DATA-LAST (READ) token.

I2C_M_0_TOKEN_RDATA_REG1 0x7c07

Bit(s)	R/W	Default	Description
31:24	R/W	0x00	8th data byte read for a DATA or DATA-LAST (READ) token.
23:16	R/W	0x00	7th data byte read for a DATA or DATA-LAST (READ) token.
15:8	R/W	0x00	6th data byte read for a DATA or DATA-LAST (READ) token.
7:0	R/W	0x00	5th data byte read for a DATA or DATA-LAST (READ) token.

I2C_M_1_ 0x7800~0x7807

See I2C_M_0

I2C_M_2_ 0x7400~0x7407

See I2C_M_0

I2C_M_3_ 0x7000~0x7007

See I2C_M_0

For I2C module in AO domain, each register final address = 0x FF805000 + offset * 4 for master mode, final address = 0x FF806000 + offset * 4 for slave mode.

AO_I2C_M_0_CONTROL_REG 0x0

Bit(s)	R/W	Default	Description
31	R/W	0	CNTL_JIC: There is internal logic to dynamically enable the gated clocks. If this gated clock logic doesn't work, you can set this bit to always enable the clock. Setting this bit wastes power.
30	R	0	Unused

Bit(s)	R/W	Default	Description
29-28	R/W	0	QTR_CLK_EXT: These two bits extend the clock divider to 12 bits: QTR_CLK = {[29:28],[21:12]}
27	R	0	unused
26	R	0	Read back level of the SDA line
25	R	0	Read back level of the SCL line
24	R/W	0	Sets the level of the SDA line if manual mode is enabled. If this bit is '0', then the SDA line is pulled low. If this bit is '1' then the SDA line is tri-stated.
23	R/W	0	Sets the level of the SCL line if manual mode is enabled. If this bit is '0', then the SCL line is pulled low. If this bit is '1' then the SCL line is tri-stated.
22	R/W	0	This bit is used to enable manual mode. Manual I2C mode is controlled by bits 12,13,14 and 15 above.
21:12	R/W	0x142	QTR_CLK_DLY. This value corresponds to period of the SCL clock divided by 4 Quarter Clock Delay = * System Clock Frequency For example, if the system clock is 133Mhz, and the I2C clock period is 10uS (100khz), then Quarter Clock Delay = * 133 Mhz = 332
11:8	R	-	READ_DATA_COUNT: This value corresponds to the number of bytes READ over the I2C bus. If this value is zero, then no data has been read. If this value is 1, then bits [7:0] in TOKEN_RDATA_REG0 contains valid data. The software can read this register after an I2C transaction to get the number of bytes to read from the I2C device.
7:4	R	-	CURRENT_TOKEN: This value reflects the current token being processed. In the event of an error, the software can use this value to determine the error location.
3	R	-	ERROR: This read only bit is set if the I2C device generates a NACK during writing. This bit is cleared at on the clock cycle after the START bit is set to 1 indicating the start of list processing. Errors can be ignored by setting the ACK_IGNORE bit below. Errors will be generated on Writes to devices that return NACK instead of ACK. A NACK is returned by a device if it is unable to accept any more data (for example because it is processing some other real-time function). In the event of an ERROR, the I2C module will automatically generate a STOP condition on the bus.
2	R	-	STATUS: This bit reflects the status of the List processor: 0: IDLE 1: Running. The list processor will enter this state on the clock cycle after the START bit is set. The software can poll the status register to determine when processing is complete.
1	R/W	0	ACK_IGNORE: Set to 1 to disable I2C ACK detection. The I2C bus uses an ACK signal after every byte transfer to detect problems during the transfer. Current Software implementations of the I2C bus ignore this ACK. This bit is for compatibility with the current Amlogic software. This bit should be set to 0 to allow NACK operations to abort I2C bus transactions. If a NACK occurs, the ERROR bit above will be set.

Bit(s)	R/W	Default	Description
0	R/W	0	START: Set to 1 to start list processing. Setting this bit to 0 while the list processor is operating causes the list processor to abort the current I2C operation and generate an I2C STOP command on the I2C bus. Normally this bit is set to 1 and left high until processing is complete. To re-start the list processor with a new list (after a previous list has been exhausted), simply set this bit to zero then to one.

AO_I2C_M_0_SLAVE_ADDR 0x1

Bit(s)	R/W	Default	Description
31:29	R	0	Unused
28	R/W	0	USE_CNTL_SCL_LOW: If this bit is set to 1, then bits[27:16] control the SCL low time.
27:16	R/W	0	SCL Low delay. This is a new feature in M8baby. In the previous M8baby design, the SCL low time was controlled by bits[21:12] of the register above. In this design, the SCL delay is controlled independently by these bits.
15:14	R	0	Unused
13:11	R/W	0	SCL_FILTER: A filter was added in the SCL input path to allow for filtering of slow rise times. 0 = no filtering, 7 = max filtering
10:8	R/W	0	SDA FILTER: A filter was added in the SDA input path to allow for filtering of slow rise times. 0 = no filtering, 7 = max filtering
7:0	R/W	0x00	SLAVE_ADDRESS. This is a 7-bit value for a 7-bit I2C device, or (0xF0 | {A9,A8}) for a 10-bit I2C device. By convention, the slave address is typically stored in by first left shifting it so that it's MSB is D7 (The I2C bus assumes the 7-bit address is left shifted one). Additionally, since the SLAVE address is always an 7-bit value, D0 is always 0. NOTE: The I2C always transfers 8-bits even for address. The I2C hardware will use D0 to dictate the direction of the bus. Therefore, D0 should always be '0' when this register is set.

AO_I2C_M_0_TOKEN_LIST0 0x2

The register below describes the first 8 tokens in the token list.

Bit(s)	R/W	Default	Description
31:28	R/W	0x00	8th token in the list to process
27:24	R/W	0x00	7th token in the list to process
23:20	R/W	0x00	6th token in the list to process
19:16	R/W	0x00	5th token in the list to process
15:12	R/W	0x00	4th token in the list to process
11:8	R/W	0x00	3rd token in the list to process
7:4	R/W	0x00	2nd token in the list to process

3:0	R/W	0x00	1st token in the list to process (See the table below for token definitions)

Table 12-2 Token Definitions

Command Token	Value	Data	Description
END	0x0	N/A	Used to tell the I2C module that this is the end of the Token list. This token is not associated with the I2C bus, but rather with the state-machine that drives the token list processor.
START	0x1	N/A	The START Token is used to tell an I2C device that this is the beginning of an I2C transfer
SLAVE_ADDR-WRITE	0x2	7-bits	This bit-sequence is used to address a device and tell the device it is being WRITTEN
SLAVE_ADDR-READ	0x3	7-bits	This bit sequence is used to address a device and tell the device it is being READ.
DATA	0x4	8-bits	This 8-bit byte sequence is a byte transfer (READ or WRITE). The DATA token corresponds to a WRITE if it follows a SLAVE_ADDR-WRITE token. The DATA token corresponds to a READ if it follows a SLAVE_ADDR-READ token.
DATA-LAST	0x5	8-bits	Used to indicate the last 8-bit byte transfer is a byte transfer of a READ.
STOP	0x6	N/A	This tells the I2C device it is no longer being addressed

Write data associated with the DATA token should be placed into the I2C_TOKEN_WDATA_REG0 or I2C_TOKEN_WDATA_REG1 registers. Read data associated with the DATA or DATA-LAST token can be read from the I2C_TOKEN_RDATA_REG0 or I2C_TOKEN_RDATA_REG1 registers.

AO_I2C_M_0_TOKEN_LIST1 0x3

Bit(s)	R/W	Default	Description
31:28	R/W	0x00	16th token in the list to process
27:24	R/W	0x00	15th token in the list to process
23:20	R/W	0x00	14th token in the list to process
19:16	R/W	0x00	13th token in the list to process
15:12	R/W	0x00	12th token in the list to process
11:8	R/W	0x00	11th token in the list to process
7:4	R/W	0x00	10th token in the list to process
3:0	R/W	0x00	9th token in the list to process

AO_I2C_M_0_WDATA_REG0 0x4

Bit(s)	R/W	Default	Description
31:24	R/W	0x00	4th data byte written for a DATA (write) token.
23:16	R/W	0x00	3rd data byte written for a DATA (write) token.
15:8	R/W	0x00	2nd data byte written for a DATA (write) token.
7:0	R/W	0x00	1st data byte written for a DATA (write) token.

AO_I2C_M_0_WDATA_REG1 0x5

Bit(s)	R/W	Default	Description
31:24	R/W	0x00	8th data byte written for a DATA (write) token.
23:16	R/W	0x00	7th data byte written for a DATA (write) token.
15:8	R/W	0x00	6th data byte written for a DATA (write) token.
7:0	R/W	0x00	5th data byte written for a DATA (write) token.

AO_I2C_M_0_RDATA_REG0 0x6

Bit(s)	R/W	Default	Description
31:24	R/W	0x00	4th data byte read for a DATA or DATA-LAST (READ) token.
23:16	R/W	0x00	3rd data byte read for a DATA or DATA-LAST (READ) token.
15:8	R/W	0x00	2nd data byte read for a DATA or DATA-LAST (READ) token.
7:0	R/W	0x00	1st data byte read for a DATA or DATA-LAST (READ) token.

AO_I2C_M_0_RDATA_REG1 0x7

Bit(s)	R/W	Default	Description
31:24	R/W	0x00	8th data byte read for a DATA or DATA-LAST (READ) token.
23:16	R/W	0x00	7th data byte read for a DATA or DATA-LAST (READ) token.
15:8	R/W	0x00	6th data byte read for a DATA or DATA-LAST (READ) token.
7:0	R/W	0x00	5th data byte read for a DATA or DATA-LAST (READ) token.

AO_I2C_M_0_TIMEOUT_TH 0x8

Bit(s)	R/W	Default	Description
31:24	R/W	0x00	8th data byte read for a DATA or DATA-LAST (READ) token.

23:16	R/W	0x00	7th data byte read for a DATA or DATA-LAST (READ) token.
15:8	R/W	0x00	6th data byte read for a DATA or DATA-LAST (READ) token.
7:0	R/W	0x00	5th data byte read for a DATA or DATA-LAST (READ) token.

AO_I2C_S_CONTROL_REG 0x0

Bit(s)	R/W	Default	Description
31-29	R	0	REG POINTER: There are 5 internal registers inside the I2C slave module. The I2C Master sets this value using the byte that follows the address byte in the I2C data stream. Register 4 (numbered 0,1,…4) is the status register.
28	R/W	0	SEND READY: This bit is set to '1' by the ARC to indicate to the slave machine that the I2C slave module is ready to send data. This bit is cleared by the I2C module when it has sent 4 bytes to the I2C master. This bit is also available in the status register that can be read by the I2C master. The I2C master can read the status register to see when the I2C slave module has data to send.
27	R/W	0	RECEIVE READY: This bit is set to '1' by the ARC to indicate to the slave machine that the I2C slave module is ready to receive data. This bit is cleared by the I2C module when it has received 4 bytes from the I2C master. This bit is also available in the status register that can be read by the I2C master. The I2C master can read the status register to see when the I2C slave module is ready to receive data.
26	R	0	BUSY: Read only status bit. '1' indicates that the I2C slave module is sending or receiving data.
25	R/W	0	IRQ_EN: If this bit is set, then an interrupt will be sent to the ARC whenever 4 bytes have been read or 4 bytes have been written to the I2C slave module.
24	R/W	0	ACK Always: Typically the ACK of a slave I2C device is dependent upon the availability of data (if reading) and room to store data (when we are being written). Our I2C module has a status register that can be read continuously. This bit can be set if the I2C master wants to continually read the status register.
23-16	R/W	0	Slave Address: Bits [7:1] are used to identify the device. Bit [0] is ignored since this corresponds to the R/W bit.
15-8	R/W	0x27	HOLD TIME: Data hold time after the falling edge of SCL. This hold time is computed as Hold time = (MPEG system clock period) * (value + 1).
7	R/W	0	Enable: A '1' enables the I2C slave state machine.
6-0	R/W	0x06	Sampling rate. Defined as MPEG system clock / (value + 1). The SDA and SCL inputs into the slave module are sampled as a way of filtering the inputs. A rising or falling edge is determined by when 3 successive samples are either high or low respectively.

AO_I2C_S_SEND_REG: Send Data 0x1

Bit(s)	R/W	Default	Description
31-0	R/W	0	The I2C slave module can send up to 4 bytes of data starting with the byte located at bits [7:0]

AO_I2C_S_RECV_REG: Received Data 0x2

Bit(s)	R/W	Default	Description
31-0	R	0	This read only register corresponds to the 4 bytes of data written to the I2C slave module by an external I2C master. Bits [7:0] correspond to the first byte written by the I2C master.

AO_I2C_S_CNTL1_REG 0x3

Bit(s)	R/W	Default	Description
31-6	R	0	Unused
5-3	R/W	0	SCL_FILTER: A filter was added in the SCL input path to allow for filtering of slow rise times. 0 = no filtering, 7 = max filtering
2-0	R/W	0x00	SDA FILTER: A filter was added in the SDA input path to allow for filtering of slow rise times. 0 = no filtering, 7 = max filtering

12.4 Universal Asynchronous Receiver And Transmitter

12.4.1 Overview

There are a number of UART's in the chip that offer 2-wire (RX/TX) and 4-wire (RX/TX, CTS/RTS) connections at the digital I/O pins. Each UART contains one transmit FIFO and a receive FIFO (see depths below). The FIFO's are filled by the CPU and read by the CPU. In some cases, the receive FIFO can be configured to be pushed directly to DDR memory without CPU intervention.

Table 12-3 UART List

UART	RX/TX FIFO depths	RX FIFO DMA to DDR	Comment
UART0	128 bytes	Yes	Located in the EE domain
UART1	64 bytes	Yes	Located in the EE domain
UART2	64 bytes	Yes	Located in the EE domain
UART0-AO	64 bytes	No	Located in the Always On domain
UART2-AO	64 bytes	No	Located in the Always On domain

12.4.2 Features

Input filters: The CTS (clear to send) and RX (receive) input paths have input filters to deal with slow rise times. The filters are configurable to use a 125nS or 1uS sampling mechanism. There is an implied 3 system clock cycle delay (15nS for a typical system clock of 200Mhz) that is used to synchronize and detect the rising/falling edge of the RXD signal. The RXD signal may be passed through an optional filter to deglitch the external signal in noisy conditions. The deglitch filter has two settings which add to the ¡§detection delay¡¨ of the RXD signal by the internal logic:

- Filter setting 1 (125nS strobe): 375nS ~ 2.6uS
- Filter setting 2 (1uS strobe): 3uS ~ 21uS

The filter is described in the register specification. If the filter is disabled, the shortest RXD low time and high time is 12 system clock cycles (60nS for a system clock of 200Mhz).

Clear to Send: CTS is a signal sent from the receiver UART back to the transmitting UART to tell the transmitting UART to stop sending data. The CTS signal must be received before the next START symbol is sent. The transmitting UART is allowed to send one more byte after the CTS signal is recognized. The CTS signal coming into the chip goes through some synchronization and detection which adds an additional 5 system clocks (typically 25nS for a 200Mhz system clock). This setup time for CTS detection is called CTSstop. The CTS input also has an optional filter can be used to deglitch the incoming CTS signal. If the filter is disabled, the CTS signal must be de-asserted 5 system clock cycles before the start of the next BYTE transfer. If the CTS filter is enabled, then additional time must be added to the 25nS requirement. There are two programmable filter settings that effectively delay CTS being seen by the internal logic:

- Filter setting 1 (125nS strobe): 375nS ~ 2.6uS
- Filter setting 2 (1uS strobe): 3uS ~ 21uS

Interrupts: The UARTs can generate interrupts if the receive FIFO exceeds a pre-programmed threshold. An interrupt can also be generated if there is a frame or parity error.

Clock independent operation: Because the system clock can be altered to accommodate dynamic frequency scaling, the UARTs have an option in which they use the 24Mhz crystal clock as the source for the UART.

12.4.3 Functional Description

The UART requires that a Baud Rate be established. The UART supports rates as slow as 1Hz up to rates as high as 8 MBits/Sec. Once the baud rate has been established, bytes are transmitted as they are written to the transmit-FIFO by the CPU. A large transmit-FIFO exists to allow the CPU to pre-load a transmit package because the CPU can often write faster than the UART can transmit the data.

Data this automatically received by the UART is placed into the receive FIFO one byte at a time. The receive-FIFO decouples the UART from the CPU allowing the CPU to read the UART byte data at a rate not dictated by the UART.

Figure 12-1 UART Timing Diagram

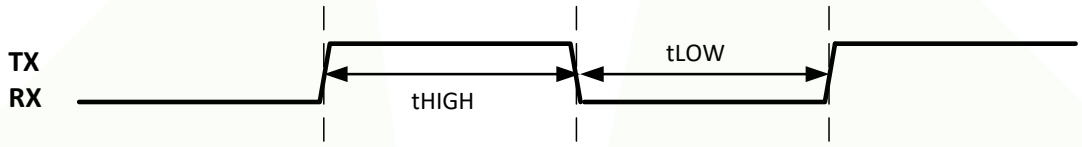


Table 12-4 UART Timing Specification

Symbol	Description	Min.	Max.	Unit	Notes
tHIGH	TX/RX high time	60ns	167ms		
tLOW	TX/RX low time	60ns	167ms		

12.4.4 Register Description

The following register description is uniformly applied to all UART instantiations in the chip.

Base_adr:0xffd00000

Final_adr = base_adr + offset *4

UART0 = 128x8 FIFOs

UART1 = 64x8 FIFOs

UART2 = 64x8 FIFOs

UARTx_WFIFO: Write data 0x9000

Bit(s)	R/W	Default	Description
31-8	R	0	unused
7-0	R/W	-	Write FIFO data. The Write FIFO holds 128 or 64 bytes. The Write FIFO can be written as long as it is not full.

UARTx_RFIFO: Read Data 0x9001

Bit(s)	R/W	Default	Description
31-8	R	0	unused
7-0	R/W	-	Read FIFO data. The Read FIFO holds 128 or 64 bytes. The empty flag can be used to determine if data is available

UARTx_CONTROL: UART Mode 0x9002

Bit(s)	R/W	Default	Description
31	R/W	0	Invert the RTS signal
30	R/W	0	Mask Error: Set to 1 to mask errors
29	R/W	0	Invert the CTS signal
28	R/W	0	Transmit byte Interrupt: Set to 1 to enable the generation an interrupt whenever a byte is read from the transmit FIFO
27	R/W	0	Receive byte Interrupt: Set to 1 to enable the generation an interrupt whenever a byte is written to the receive FIFO
26	R/W	0	Set to 1 to invert the TX pin
25	R/W	0	Set to 1 to invert the RX pin
24	R/W	0	Clear Error
23	R/W	0	Reset the receive state machine
22	R/W	0	Reset the transmit state machine
21-20	R/W	0	Character length: 00 = 8 Bits, 01 = 7 Bits, 10 = 6 Bits, 11 = 5 Bits
19	R/W	1	Parity Enable: Set to 1 to enable parity
18	R/W	0	Parity type: 0 = even, 1 = odd
17-16	R/W	0	Stop bit length: 00 = 1 bit, 01 = 2 Bits
15	R/W	0	Two Wire mode:
14	R/W	0	Unused
13	R/W	0	Receive Enable. Set to 1 to enable the UART receive function
12	R/W	0	Transmit Enable. Set to 1 to enable the UART transmit function
11-0	R/W	0x120	Old Baud rate

UARTx_STATUS: UART Status 0x9003

Bit(s)	R/W	Default	Description
31-27	R	0	Unused
26	R	0	UART_RECV_BUSY: This bit will be 1 if the uart receive state machine is busy
25	R	0	UART_XMIT_BUSY: This bit will be 1 if the uart transmit state machine is busy
24	R	0	RECV_FIFO_OVERFLOW:
23	R	0	CTS Level
22	R	0	Transmit FIFO Empty
21	R	0	Transmit FIFO Full
20	R	0	Receive FIFO empty
19	R	0	Receive FIFO full
18	R	0	This bit is set if the FIFO is written when it is full. To clear This bit, wriTE bit 24 of register 0x2132
17	R	0	Frame error. To clear This bit, wriTE bit 24 of register UART0_CONTROL
16	R	0	Parity error. To clear This bit, wriTE bit 24 of register UART0_CONTROL
15	R	0	Unused
14-8	R	0	Transmit FIFO count. Number of bytes in the transmit FIFO
7	R	0	Unused
6-0	R	0	Receive FIFO count. Number of bytes in the receive FIFO

UARTx_MISC: UART IRQ CONTROL 0x9004

Bit(s)	R/W	Default	Description
31	R/W	0	Added a “just in case” bit that can be set to 1 to enable clocks always. The default is 0 meaning the auto-clock gating logic is enabled.
30	R/W	0	USE old Rx Baud: There was a bug in the RX baud rate generator. The Rx baud rate generator was re-designed to compute a baud rate correctly. If you want to use the old (stupid) logic, you can set This bit to 1.
29	R/W	0	ASYNC_FIFO_PURGE: This bit can be set to 1 after all UART bytes have been received in order to purge the data into the Async FIFO. This bit is needed because the UART receives 8-bit data, but the ASYNC FIFO can only be written with 16-bit data. In this case there might be a residual byte if the UART is not receiving an even number of bytes.
28	R/W	0	ASYNC_FIFO_EN: If This bit is set to 1, then the UART received data is automatically sent to the Async FIFO module which will in turn automatically send the data to DDR memory

Bit(s)	R/W	Default	Description
27	R/W	0	CTS: Filter Timebase select: 1 = 1uS, 0 = 111nS timebase. A filter was added to the CTS input to allow for a little digital filtering. The amount of filtering is controlled by this timebase (longer = more filtering) and the value in Bits FILTER_SEL below.
26-24	R/W	0	CTS: FILTER_SEL: 0 = no filter, 7 = max filtering
23-20	R/W	0	old BAUD_RATE_EXT: These 4 Bits extend the baud rate divider to 16-bits: Baud Rate = {Reg4[23:20],Reg2[11:0]}
19	R/W	0	RX: Filter Timebase select: 1 = 1uS, 0 = 111nS timebase. A filter was added to the RX input to allow for a little digital filtering. The amount of filtering is controlled by this timebase (longer = more filtering) and the value in Bits FILTER_SEL below.
18-16	R/W	0	RX: FILTER_SEL: 0 = no filter, 7 = max filtering
15-8	R/W	32	XMIT_IRQ_CNT: The UART can be configured to generate an interrupt if the number of bytes in the transmit FIFO drops below this value.
7:0	R/W	15	RECV_IRQ_CNT: The UART can be configured to generate an interrupt after a certain number of bytes have been received by the UART.

UARTx_REG5 0x9005

Bit(s)	R/W	Default	Description
31-28	R/W	0	unused
27	R/W	0	Xtal2_clk_sel: 0: see Xtal_clk_sel 1: xtal_div2(12M)
26	R/W	0	Xtal_clk_sel: 0: xtal_div3(8M); 1: xtal(24M);(need set xtal_tick_en =1 first);
24	R/W	0	USE_XTAL_CLK: If this bit is set, then the clock for generating the UART Baud rate comes from the crystal pad. This allows the UART to operate independent of clk81.
23	R/W	0	USE New Baud rate. Over the years, the baud rate has been extended by concatenating bits from different registers. To take advantage of the full 23-bit baud rate generate (extended to 23 bits to accommodate very low baud rates), you must set this bit. If this bit is set, then the baud rate is configured using bits [22:0] below
22:0	R/W	15	NEW_BAUD_RATE: If bit[23] = 1 above, then the baud rate for the UART is computed using these bits. This was added in M6 to accommodate lower baud rates.

UART1_WFIFO: Write data 0x8c00

See UART0 bit descriptions

UART1_RFIFO: Read Data 0x8c01

See UART0 bit descriptions

UART1_CONTROL: UART Mode 0x8c02

See UART0 bit descriptions

UART1_STATUS: UART Status 0x8c03

See UART0 bit descriptions

UART1_MISC: UART IRQ CONTROL 0x8c04

See UART0 bit descriptions

UART1_REG5 0x8c05

See UART0 bit descriptions

UART2_YFIFO: write data 0x8800

See UART0 bit descriptions

UART2_RFIFO: Read Data 0x8801

See UART0 bit descriptions

UART2_CONTROL: UART Mode 0x8802

See UART0 bit descriptions

UART2_STATUS: UART Status 0x8803

See UART0 bit descriptions

UART2_MISC: UART IRQ CONTROL 0x8804

See UART0 bit descriptions

UART2_REG5 0x8805

See UART0 bit descriptions

UART3_DF_REG_A73 0x8400

See UART3_SLIP bit descriptions

UART3_DF_REG_A74 0x8401

See UART3_SLIP bit descriptions

UART3_DF_REG_A75 0x8402

See UART3_SLIP bit descriptions

UART3_DF_REG_A76 0x8403

See UART3_SLIP bit descriptions

UART3_DF_REG_A77 0x8404

See UART3_SLIP bit descriptions

UART3_DF_REG_A78 0x8405

See UART3_SLIP bit descriptions

UART3_DF_REG_A79 0x8406

See UART3_SLIP bit descriptions

UART3_DF_REG_A80 0x8407

See UART3_SLIP bit descriptions

UART3_DF_REG_A81 0x8408

See UART3_SLIP bit descriptions

UART3_DF_REG_A82 0x8409

See UART3_SLIP bit descriptions

UART3_DF_REG_A83 0x8410

See UART3_SLIP bit descriptions

UART3_DF_REG_A84 0x8411

See UART3_SLIP bit descriptions

UART3_DF_REG_A85 0x8412

See UART3_SLIP bit descriptions

UART3_DF_REG_A86 0x8413

See UART3_SLIP bit descriptions

UART3_DF_REG_A87 0x8414

See UART3_SLIP bit descriptions

UART3_DF_REG_A88 0x8415

See UART3_SLIP bit descriptions

UART3_DF_REG_A89 0x8416

See UART3_SLIP bit descriptions

UART3_DF_REG_A96 0x8417

See UART3_SLIP bit descriptions

UART3_DF_REG_A97 0x8418

See UART3_SLIP bit descriptions

UART3_DF_REG_A128 0x8420

See UART3_SLIP bit descriptions

UART3_DF_REG_A129 0x8421

See UART3_SLIP bit descriptions

UART3_DF_REG_A130 0x8422

See UART3_SLIP bit descriptions

UART3_DF_REG_A131 0x8423

See UART3_SLIP bit descriptions

UART3_DF_REG_A132 0x8424

See UART3_SLIP bit descriptions

UART3_DF_REG_A133 0x8425

See UART3_SLIP bit descriptions

UART3_DF_REG_A134 0x8426

See UART3_SLIP bit descriptions

UART3_DF_REG_A135 0x8427

See UART3_SLIP bit descriptions

UART3_DF_REG_A136 0x8428

See UART3_SLIP bit descriptions

12.5 Infrared Remote

12.5.1 Overview

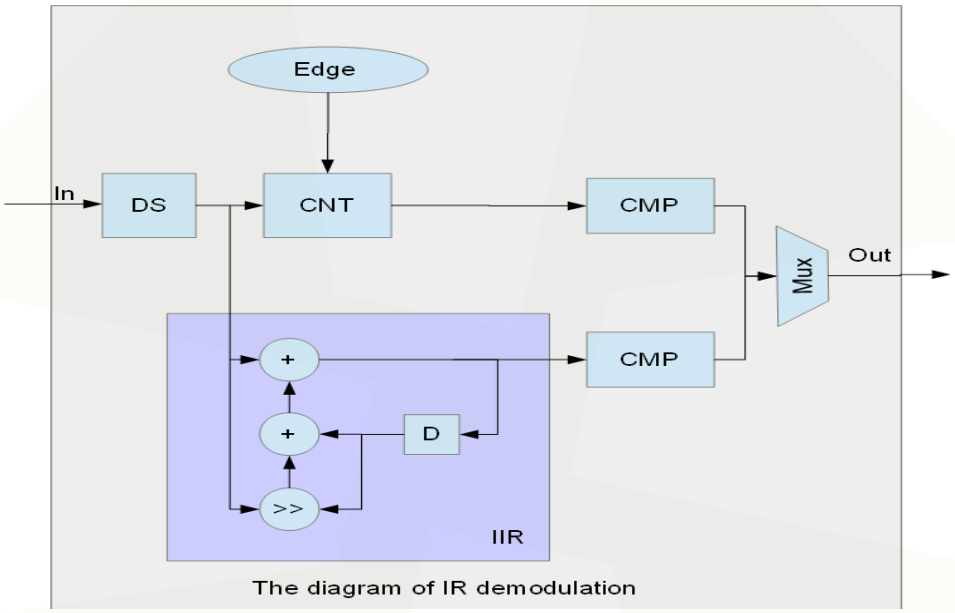
IR module includes 3 sub modules: IR demodulation, Legacy IR remote control and Multi-format IR remote control.

12.5.2 IR Demodulation

IR demolation module demodulate the modulate signal to get the envelop signal.

IR demulation diagram is shown below.

Figure 12-2 IR Demulation Diagram



It implements two methods for demodulation. The input data is an one-bit stream.

The input data are down-sampled firstly. The rate of downsample are configured by register, ranges from 0 to 255.

One method is implemented in the upper path.

The CNT is used to counte the time for ‘0’ and ‘1’. It is cleared when input data changes(from 1 to 0 or from 0 to 1).

When (CNT>REG_IR_PROCT1 && in==1) , it outputs ‘1’. REG_IR_PROCT1 is used to canceled the glitch. Actually, the operation of downsample also have some utility to cancel the glitch.

When (CNT>REG_IR_DECT0 && in==0) , it outputs ‘0’. REG_IR_DECT0 is used to confirm the ‘0’. The value should be bigger than one carrier period.

The other method is implemented in the lower path.

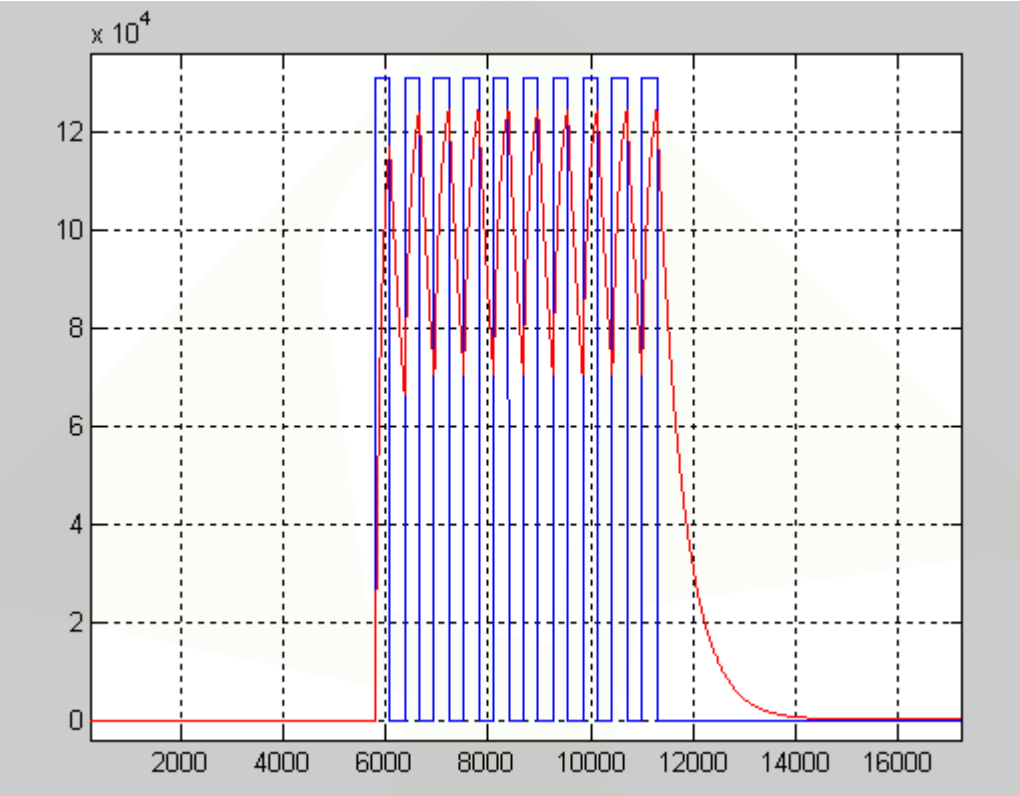
It is a one tap IIR filter. The input data control the integrating factor and the factor can be configured from registers. When 1 inputs, the accumulator can grow quickly. When 0 input, the accumulator decays slowly, and when the number of 0 is bigger enough, it decays soon.

When (accum> REG_IR_IIR_THD1), it outputs ‘1’.

When (accum< REG_IR_IIR_THD0), it outputs ‘0’.

The diagram shows the simple simulation result. The blue is the input data, the red is the filter out.

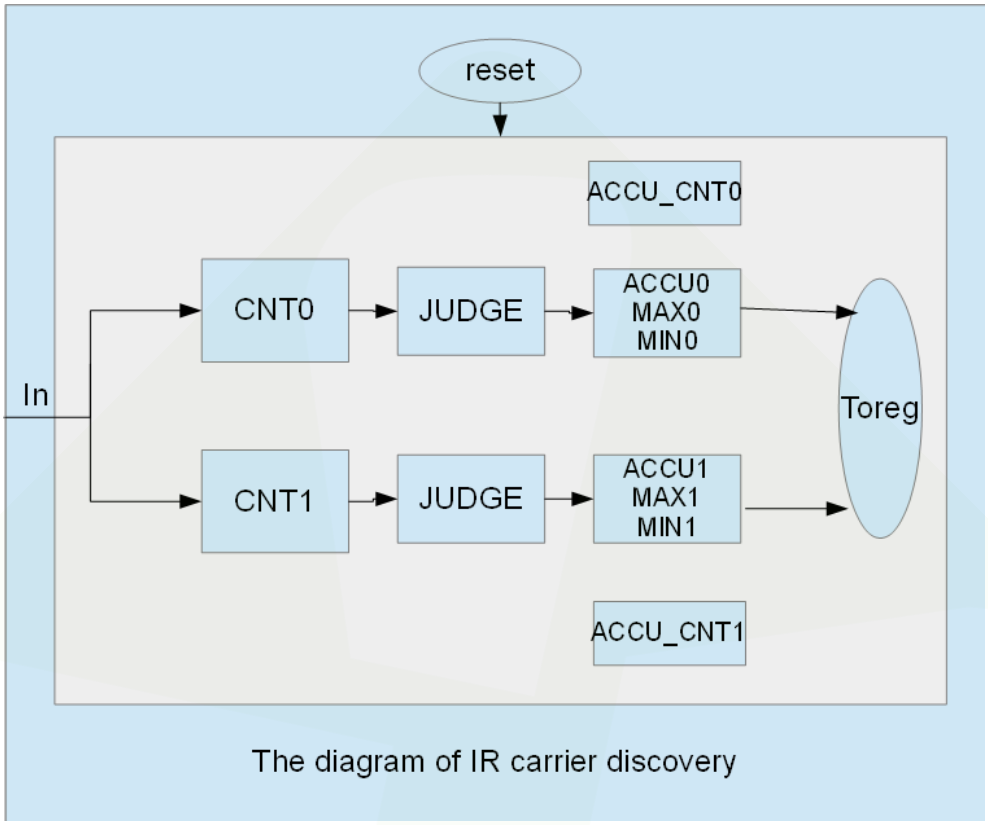
Figure 12-3 Simulation Result of IR Demulation



Because the input carrier signal is a rectangular wave signal, the carrier can be estimated easily by count the wave period. For estimating precisely, the counter for the long “0” and start period and and glitch should be removed.

The simple diagram is shown below:

Figure 12-4 IR Carrier Discovery



CNT is the counter for time of “0” and “1”.

JUDGE is used to remove the long “0” and glitch.

ACC_CNT is the counter for number of accumulating. If ACC_CNT== REG_IR_ST_CNT_THD, then write the value to registers.

For example:

Set REG_IR_ST_CNT_THD to 256, then accumulates 256 times.

The carrier can be calculated:

Fc = Fsys/(RO_IR_SUM_CNT0 +RO_IR_SUM_CNT1)*256;

DUT = RO_IR_SUM_CNT1/(RO_IR_SUM_CNT0 +RO_IR_SUM_CNT1);

Where Fc is the carrier frequency, Fsys is the system clock frequency, DUT is the carrier duty ratio.

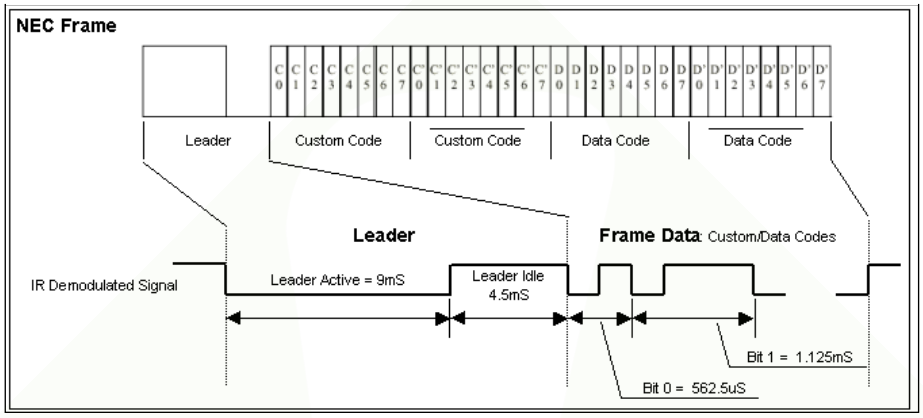
12.5.3 Lagacy IR Control

The Lagacy IR Remote control module has two modes of operation:

- NEC Frame decoder mode
- General Time Measurement mode

NEC Frame Decoder: The NEC Frame Decoder mode operates by analyzing the waveform of a TV remote.

Figure 12-5 NEC Frame Decoder



The waveform has a number of components, each of which must fit within a time window to be considered valid. If the entire waveform meets the specifications described by the registers below, then the TV remote codes are captured and an interrupt is generated.

General Time Measurement: Some remotes don't follow the standard NEC format, so additional registers provide the ability to measure the time between rising and/or falling edges of the IR signal. Since the time measurement is done in hardware, the software only needs to read a "width" measurement from a register for every rising and/or falling edge event.

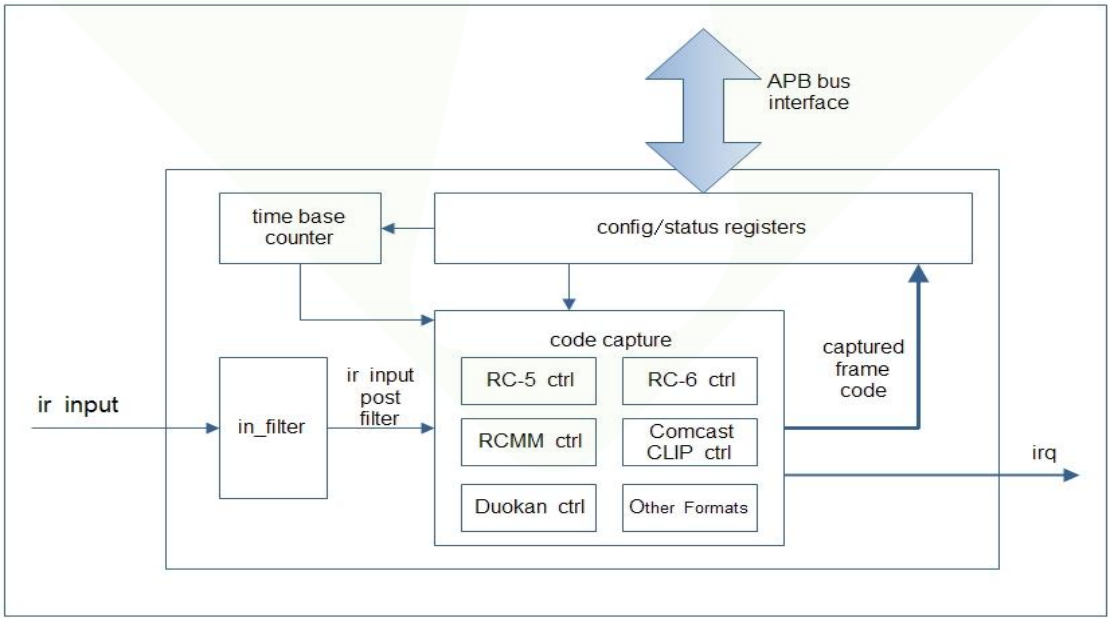
12.5.4 Multi Format IR Control

The decoder mainly consisted of two blocks:

- Decoder with input filter
- A set of registers including control & clock, data and tuning

The function diagram of IR decoder is illustrated in the figure below.

Figure 12-6 IR Decoder Function Block



IR Decoder decodes the IR remote control input signal. 13 operation modes are supported:

- Hardware Decode IR transmission protocol compatible frame decoder mode (NEC MITSUBISHI Thomson Toshiba Sony SIRC RC5 RC6 RCMM Duokan Comcast Sanyo Modes)
- General programmable time measurement frame decoder mode (General Mode)

In Hardware Decode Mode, the Decoder uses signal pattern search mechanism to decode data frame. It can detect logical ‘0’ ,‘1’ ,“00”,”01”,”10” and ”11”, as well as data frame start and end. Whenever Decoder detects and decodes the data frame, the data are kept in data register.

In General Mode, the Decoder uses edge detection mechanism to decode data frame. It can detect each input signal edge and record the time between two edges. The time measurement result is kept in control register.

The user should set proper operation mode corresponding to the selection of remote controller.

There is a simple time-based signal Filter between the signal input and the Decoder. The Filter is programmable and helps to improve signal integrity.

12.5.5 Register Description

The following register description is uniformly applied to all UART instantiations in the chip.

Base_adr: 0xff808000

Final_adr = base_adr + offset *4

AO_IR_DEC_DEMOD_CNTL0 0x30

Bits	R/W	Default	Description
31			Ir demodulor soft_reset:write 1 to reset ir_demodulor,will auto clr to 0
30			Reg_ir_fd_reset :It is used to reset the carrier detection module.write 1 to reset ,will auto clr to 0
29			Reg_ir_demod_mode:It is uset to set the demod mode. 0 = count mode; 1 = iir mode.default is 0
28			Ir demodulor clk gate bypass:write 1 will bypass clk gate,default is 0
27:16			Reg_ir_st_cnt_thd:It is used to set the statistics number for carrier detection.default is 0x40
15:8			Reg_ir_ds_rate:It is uset to set the downsample rate.default is 0x8
7:4			Reg_ir_fsft_1:It is used to set the shift value for input data “1”.default is 0x7
3:0			Reg_ir_fsft_0:It is used to set the shift value for input data “0”.default is 0x9

AO_IR_DEC_DEMOD_CNTL1 0x31

Bits	R/W	Default	Description
31			Reg_ir_demod_en:ir demod enable ,default is 0
30			Reg_ir_invt:ir input invert, invert input at 1.default is 0

29:16			Reg_ir_proctect1:It is used to set the protection threshold for 1. It is used to filter the glitch.default is 0xa
13:0			Reg_ir_detect0:It is used to set the detection threshold for signal “0”.default is 0x3e8

AO_IR_DEC_DEMOD_IIR_THD 0x32

Bits	R/W	Default	Description
31:16			Reg_ir_iir_thd1:It is used to set the detection threshold for “1” in iir mode.default is 0xdac
15:0			Reg_ir_iir_thd0:It is used to set the detection threshold for “0” in iir mode.default is 0xdac

AO_IR_DEC_DEMOD_THD0 0x33

Bits	R/W	Default	Description
29:16			Reg_ir_thd0_low:It is used to set the low threshold for “0” when statistics.default is 0x12c
13:0			Reg_ir_thd0_high:It is used to set the high threshold for “0” when statistics.default is 0x1770

AO_IR_DEC_DEMOD_THD1 0x34

Bits	R/W	Default	Description
29:16			Reg_ir_thd1_low:It is used to set the low threshold for “1” when statistics.default is 0x12c
13:0			Reg_ir_thd1_high:It is used to set the high threshold for “1” when statistics.default is 0x1770

AO_IR_DEC_DEMOD_SUM_CNT0 0x35

Bits	R/W	Default	Description
25:0			Ro_ir_sum_cnt0:It is used to report the sum value for the statistics data “0”.READ ONLY

AO_IR_DEC_DEMOD_SUM_CNT1 0x36

Bits	R/W	Default	Description
25:0			Ro_ir_sum_cnt1:It is used to report the sum value for the statistics data “1”.READ ONLY

AO_IR_DEC_DEMOD_CNT0 0x37

Bits	R/W	Default	Description
29:16			Ro_ir_max_cnt0:It is used to report the maximum value for the statistics data “0”.READ ONLY
13:0			Ro_ir_min_cnt0:It is used to report the minimum value for the statistics data “0”.READ ONLY

AO_IR_DEC_DEMOD_CNT1 0x38

Bits	R/W	Default	Description
29:16			Ro_ir_max_cnt1:It is used to report the maximum value for the statistics data “1”.READ ONLY

13:0			Ro_ir_min_cnt1:It is used to report the minimum value for the statistics data “1”.READ ONLY

AO_IR_DEC_LDR_ACTIVE: Leader Active Time 0x00

Bit(s)	R/W	Default	Description
31-29	R	0	unused
28-16	R/W	0x 1d8	Max Leader ACTIVE time: 9.44mS assuming base rate = 20uS
15-13	R	0	unused
12-0	R/W	0x1ac	Min Leader ACTIVE time: 8.56mS assuming base rate = 20uS

This register controls the min/max leader active time window. For most TV remote controls, the leader active time is about 9mS. The values in this register correspond to counts of the base rate programmed by register 0x2124

AO_IR_DEC_LDR_IDLE: Leader Idle Time 0x01

Bit(s)	R/W	Default	Description
31-26	R	0	unused
28-16	R/W	0xf8	Max Leader IDLE time: 4.96mS assuming base rate = 20uS
15-13	R	0	unused
12-0	R/W	0xca	Min Leader IDLE time: 4.04mS assuming base rate = 20uS

This register controls the min/max leader IDLE time window. For most TV remote controls, the leader idle time is about 4.5mS. The values in this register correspond to counts of the base rate programmed by register: 0x2124

AO_IR_DEC_LDR_REPEAT: Repeat Leader Idle Time 0x02

Bit(s)	R/W	Default	Description
31-26	R	0	Unused
25-16	R/W	0x7a	Max REPEAT Leader IDLE time: 2.44mS assuming base rate = 20uS
15-10	R	0	unused
9-0	R/W	0x66	Min REPEAT Leader IDLE time: 2.04mS assuming base rate = 20uS

This register controls the repeat leader IDLE time window. The repeat key uses the standard leader active time (9ms) but a shorter leader idle time.

AO_IR_DEC_BIT_0: BIT 0 Identification Time 0x03

Bit(s)	R/W	Default	Description
31-26	R	0	Unused

25-16	R/W	0x42	Max BIT 0 time: 1.32mS assuming base rate = 20uS
15-10	R	0	Unused
9-0	R/W	0x2e	Min BIT 0 time: 0.92mS assuming base rate = 20uS

This register controls the min/max BIT 0 time window. For most TV remote controls, the bit 0 time is about 1.125mS. The values in this register correspond to counts of the base rate programmed by register: 0x2124

AO_IR_DEC_REG0: Base Rate Generator 0x04

Bit(s)	R/W	Default	Description
31	R	0	Just in case bit. Normally this bit is set to 0 so that the auto-clock gating is enabled. If there is a problem related to the auto-clock gating, then this bit can be set to one to disable the auto-clock gating.
30-28	R/W	0	FILTER_COUNT: This is a new feature to Nike. The IR remote input now has a simple filter that accommodates slow rise times by providing a little hysteresis. The logic works as follows. If the input is low, then an input signal will only be considered HIGH if it remains high for (FILTER_COUNT * 111nS). Similarly, if the input is currently high, it will only be considered LOW if the input signal remains low for (FILTER_COUNT * 111nS).
27-25	R	0	Unused
24-12	R/w	0xFA0	Max Frame Time. 80mS assuming base rate = 20uS This value is used to determine if a code is a repeat code (e.g. leader followed by no data for this amount of time). This value can also be used to catch slow remote codes (i.e. code sequences that are longer than expected).
11-0	R/W	0x13	This value dictates the base rate time for all measurements associated with the IR decoder. In the past, the base rate was divided from the system clock. In the current design, the base rate is divided from a fixed 1uS timer. This 1uS timer is constant and doesn't change (even when the system clock does) Base rate = (count + 1) * 1uS

This register controls the master rate generator for all width measurements made by the IR decoder module.

AO_IR_DEC_FRAME: Frame Data 0x05

Bit(s)	R/W	Default	Description
31-0	R	0	Frame Data. Format: {custom code, ~custom code, data, ~data}

Note: New keys will be ignored until this register is read if the *hold first key* bit is set in the decode control register. Reading this register resets an internal hold first flag.

AO_IR_DEC_STATUS: Frame Status 0x06

Bit(s)	R/W	Default	Description
31	R/W	0	Sim faster: Reserved

Bit(s)	R/W	Default	Description
30	R/W	0	BIT_1_MATCH_EN: Set this bit to 1 to enable qualification of bit 1 times. In the previous IR decoder module, frame detection only looked at the BIT 0 time to identify a zero bit. If a zero bit time wasn't found, then it was assumed that the bit was a 1. In the updated IR decoder module, the module will look at the BIT 0 time to find zero bits, and the BIT 1 time to find 1 bits. If the width of a pulse doesn't match the zero or the one bit width time, then the frame is considered invalid.
29-20	R/W	0x89	Max BIT 1 time: 2.74mS assuming base rate = 20uS
19-10	R/W	0x57	Min BIT 1 time: 1.74mS assuming base rate = 20uS
9	R	-	IRQ Status. 1 if there is an interrupt
8	R	-	IR Decoder input. This is the level of the digital signal coming into the IR module for decoding. This is the same as reading the I/O pad level.
7	R	-	BUSY. 1 if the decoder isbusy
6-4	R	0	Decoder Status: For debug only 000: OK 001: last frame timed out 010: leader time error (invalid IR signal) 011: repeat error (repeat leader, but other IR transitions found). 100: Invalid bit
3-0	R	0	Frame Status Bit 3: Frame data valid Bit 2: data code error (data != ~data in IR bit stream) Bit 1: custom code error (custom_code != ~custom_code in IR bit stream) Bit 0: 1 = repeat key, 0 = standard key

AO_IR_DEC_REG1: Decode / Interrupt Control 0x07

Bit(s)	R/W	Default	Description
31	R	0	Unused
30	R/W	0	CNTL_1uS_EQ_CLK: This bit should be set to 1 if the clk81 (system clock) is less than 50 Mhz.
29	R/W	0	CNTL_111nS_EQ_CLK: This bit should be set to 1 if the clk81 (system clock) is less than 50 Mhz.
28-16	R	0	Time measurement since the last time the internal time counter was reset by the rising and/or falling edge of the IR signal. The selection of reset on rising and/or falling edge is determined by the selection of the IRQ (Bits 3-2 below)
15	R/W	1	ENABLE: If this bit is 1, then the state-machines are enabled. If this bit is zero, then the state-machines cleanup and immediately return to idle.

Bit(s)	R/W	Default	Description
14	R/W	0	USE SYSTEM CLOCK: This is a new feature. 1 = use the system clock, 0 = use the 1uS timebase tick. During normal operation, the module is setup to create a 20uS tick from the 1uS internal timebase of the chip. If the chip is configured to operate using the 32khz RTC oscillator, the 1uS timebase is invalid and therefore the 20uS timebase is invalid. In order to measure time correctly, the IR remote circuit can use the system clock (which in this case is the 32khz oscillator clock) as the master timebase.
13-9	R/W	1f	Number of bits in the IR frame (N-1)
8-7	R/W	1f	Decoder mode 00: NEC Frames: Decode Leader and 32 bits 01: Only accumulate bits (skip the leader) 10: Measure Mode: The internal width measuring counter is reset on the rising and/or falling edge of the IR remote signal based on the settings of IRQ Selection below. Just before being reset, the measured width is captured and stored so that it can be read in bits [28:16] of this register. 11: NEC Frames: Decode Leader and 32 bits
6	R/W	1	Hold First Key. If this bit is set true, then the frame data register (0x2125) will only be updated if hasn't already been updated. Once updated, the frame data register will not be updated again until it has been read. This bit can be used to guarantee the first TV remote code captured will not be overwritten by subsequent transmissions from a TV remote. NOTE: You must read the frame data register to clear an internal hold first flag if this bit is set.
5-4	R/W	11	Frame mask. These bits are used to qualify frames for capture. 00: Capture all frames good or bad 01: Capture only frames where data=~data. Ignore custom codes 10: Capture only frames where custom_code = ~custom_code. Ignore data codes 11: Capture only frames where (data=~data) and (custom_code = ~custom_code)
3-2	R/W	0	IRQ Selection and width measurement reset: 00: IR Decoder done 01: IR input rising or falling edge detected 10: IR input falling edge detected 11: IR rising edge detected
1	R/W	0	IR Polarity. Polarity of the input signal (VD[0])
0	R/W	0	Set to 1 to reset the IR decoder. This is useful because the IR remote state machine thinks in terms of milliseconds and may take tens of milliseconds to return to idle by itself.

AO_MF_IR_DEC_LDR_ACTIVE: Leader Active control 0x10

This register controls the min/max Leader Active time window. For example, for NEC format, the Leader Active time is about 9mS. To identify a Leader Active time between 8.60 mS and 9.40 mS (assuming base resolution = 20uS), user can set Max duration = 0x1d6 ('d470) to represent 9.40 mS, and set Min duration = 0x1ae ('d430) to represent 8.60 mS.

Bit(s)	R/W	Default	Description
31-29	R	0	Unused
28-16	R/W	0	Max duration of Leader's active part
15-13	R	0	Unused
12-0	R/W	0	Min duration of Leader's active part

AO_MF_IR_DEC_LDR_IDLE: Leader Idle control 0x11

Bit(s)	R/W	Default	Description
31-29	R	0	Unused
28-16	R/W	0	Max duration of Leader's idle part
15-13	R	0	Unused
12-0	R/W	0	Min duration of Leader's idle part

AO_MF_IR_DEC_LDR_REPEAT: Repeat Leader Idle Time 0x12

Bit(s)	R/W	Default	Description
31-26	R	0	Unused
25-16	R/W	0	Max duration of Repeat Code's Leader. In NECformat, it defines for the repeat leader's idle part. In Toshiba format, it defines for the repeat leader's second idle part (In Toshiba format, the repeat leader's first idle part has the same duration time as the normal leader idle part.)
15-10	R	0	Unused
9-0	R/W	0	Min duration of Repeat Code's Leader

AO_MF_IR_DEC_BIT_0 0x13

Bit(s)	R/W	Default	Description
31-26	R	0	Unused

25-16	R/W	0	Max duration of Duration Setting Regiser 0. It defines max tining duration for: Logic"0" for NEC/Toshiba/Sony/Thomas format or Half trailer bit for RC6 format (RC6's half trailer bit typically 888.89us) or time of Duokan/RCMM/4ppm format's Logic "00"
15-10	R	0	Unused
9-0	R/W	0	Min duration of Duration Setting Regiser 0.

AO_MF_IR_DEC_REG0 0x14

Bit(s)	R/W	Default	Description
31	R/W	0	Clock gating control just in case. Set 1 can force clock gating disabled.
30-28	R/W	0	Filter ctrl. Set the monitor timing for input filter, bigger value means longer monitor time. Value 0 = no filtering.
27-25	R	0	Unused
24-12	R/W	0	Max frame time. Max duration of one whole frame.
11-0	R/W	0	Base time parameter. Used to generate the timing resolution. Resolution = (base_time_paramter + 1) * (1/ Freq_sys_clk). For example, if Frequency of sys_clk is 1Mhz, and base_time_parameter=19, Then resolution = (19+1)*(1uS) = 20uS.

AO_MF_IR_DEC_STATUS 0x16

Bit(s)	R/W	Default	Description
31	R/W	0	Frame data valid 1. (This bit is set to 1 when a captured frame is updated/stored into "FrameBody_1" register. A read of "FrameBody_1" register will clear This bit. "FrameBody_1" register is used to store the over 32bit MSBs of the formats whose length is more than 32 bit)
30	R/W	0	bit_1_match_en. Set to 1 to enable the check of whether logic"1" bit matches timing configure during the frame input process.

Bit(s)	R/W	Default	Description
29-20	R/W	0	Max Duration 1. Max duration of Duration Setting Regiser 1. It defines max duration for: Logic"1" for NEC/Toshiba/Sony format or Whole trailer bit for RC6 format (RC6's whole trailer bit typically 1777.78us) or time of Duokan/RCMM/4ppm format's Logic "01"
19-10	R/W	0	Min Duration 1. Min duration of Duration Setting Regiser 1.
9	R	0	irq_status. Appear as 1 if there is an interrupt.
8	R	0	ir_i_sync. IR remote serial input after synchronization. This is the level of the digital signal coming into the IR module for decoding. This is the same as reading the I/O pad level.
7	R	0	Busy. When =1, means state machine is active.
6-4	R	0	Decoder_status (for debug only). 000: OK 001: last frame timed out 010: leader time error (invalid IR signal) 011: repeat error (repeat leader, but other IR transitions found). 100: Invalid bit
3-0	R	0	Frame status. bit 3: Frame data valid (This bit is set to 1 when a captured frame is updated/stored into "FrameBody" register. A read of "FrameBody" register will clear This bit. If store and read occurs at the same time, This bit is set to 1 in common, But if "Hold first" is set to true and this valid Bit is already 1, a read clear takes precedance and This bit is clear to 0.) bit 2: data code error (data != ~data in IR bit stream) bit 1: custom code error (custom_code != ~custom_code in IR bit stream) bit 0: 1 = received frame is repeat key, 0 = received frame is normal key

AO_MF_IR_DEC_REG1 0x17

Bit(s)	R/W	Default	Description
31	R/W	0	Set to 1 to use faster timebase. .
30	R/W	0	cntl_1us_eq_clk. Just use sys_clk to relace 1uS tick.

Bit(s)	R/W	Default	Description
29	R/W	0	cntl_xtal3_eq_clk. Just use sys_clk to relace 111ns tick.
28-16	R	0	Pulse Width Counter. It stores the internal counter of pulse width duration. Commonly used as time measurement when decode_mode is set to measure width mode (software decode). Time measurement starts at the last time the internal time counter was reset by the rising and/or falling edge of the IR signal. The selection of reset on rising and/or falling edge is determined by the IRQ Selection field (Bits 3-2 below)
15	R/W	0	Enable. 1 = enable the state machine of IR decoder. 0 = disable the state machine of IR decoder.
14	R/W	0	cntl_use_sys_clk. Use sys_clk for the timebase. It's useful when sys_clk at low frequency (such as 32Khz) and cannot create 1uS timebase tick. 1 = use the system clock as timebase. 0 = use the 1uS timebase tick as timebase.
13-8	R/W	0	bit_length minus 1. (N-1). Used to set the value of frame body's bit length (frame body commonly includes address and data code part). If a format has 24 bit frame body, this value shall be set to 23.
7	R/W	0	Record_at_error. 1= record the frame body and status forcibly, even if data/custom code error check enabled by frame_mask and relative error occurs. 0 = if data/custom code error check enabled by frame_mask and relative error occurs, not record the frame body and status forcibly
6	R/W	0	Hold_first Used to hold the first captured frame data. If This bit is set to 1, then the "FrameBody/FrameBody_1" register will only be updated if hasn't already been updated. Once updated, the "FrameBody/FrameBody_1" register will not be updated again until it has been read. This bit can be used to guarantee the first TV remote code captured will not be overwritten by subsequent transmissions from a TV remote. NOTE: Read the"FrameBody" register can clear the internal "Frame data valid" flag, and read the "FrameBody_1" register can clear the "Frame data valid 1" flag.
5-4	R/W	0	Frame_mask. Some formats' body include bit-inversed data or custom/address code for error check. 00 = ignore error check from either data or custom/address code 01= check if data code matches its inverse values, ignore error check from custom/address code 10= check if custom/address code matches its inverse values, ignore error check from data code 11= check if data and custom codes match their inverse values

Bit(s)	R/W	Default	Description
3-2	R/W	0	Irq_sel. IRQ Selection and width measurement reset: 00: IR Decoder done 01: IR input rising or falling edge detected 10: IR input falling edge detected 11: IR rising edge detected
1	R/W	0	IR input polarity selection. Used to adjust/invert the polarity of IR input waveform.
0	R/W	0	Decoder Reset. Set to 1 to reset the IR decoder. This is useful because the IR remote state machine thinks in terms of milliseconds and may take tens of milliseconds to return to idle by itself.

AO_MF_IR_DEC_REG2 0x18

Bit(s)	R/W	Default	Description
31-27	R	0	Unused
26	R/W	0	Width_low_enable. Enable counter record of low pulse width duration. 0 = do not force enable of width low counter record 1 = force enable of width low counter record Some IR formats' decoding need to use internal width low counter record. By default, the width low counter record is enabled automatically for related formats. This bit is used for enable forcibly just in case. Besides, if "leader plus stop bit" method is enabled for repeat detection, This bit is also need to be enabled.
25	R/W	0	Width_high_enable. Enable counter record of high pulse width duration. 0 = do not force enable of width high counter record 1 = force enable of width high counter record Some IR formats' decoding need to use internal width high counter record. By default, the width high counter record is enabled automatically for related formats. This bit is used for enable forcibly just in case.
24	R/W	0	Enable "leader plus stop bit" method for repeat detection. 0 = "leader plus stop bit" method disabled 1 = "leader plus stop bit" method enabled Some IR formats use one normal frame's leader followed by a stop bit to represet repeat. There is no frame data in this kind repeat frame. To use this method, width_low_enable (Bit 26 of 0x20 offset register) shall be set to 1, and max_duration_3 and min_duration_3 in 0x28 offset register shall be set to appropriate value for stop bit's timing duration.
23-22	R	0	Unused

Bit(s)	R/W	Default	Description
21-16	R/W	0	Repeat_Bit_index. These Bits are used for compare bit method to set the index of the bit that is used as repeat flag. The index value can be 0 to 63. Compare bit method is one of the methods for repeat detection . Some IR formats use one bit in frame to represet whether the frame is repeat.
15	R/W	0	Running_count_tick_mode. This bit is only valid when use_clock_to_counter Bit is 0. 0 = use 100uS as increasing time unit of frame-to-frame counter 1 = use 10uS as increasing time unit of frame-to-frame counter
14	R/W	0	Use_clock_to_counter. If This bit is set to 1, the running_count_tick_mode Bit is ignored. 0 = do not use system clock as increasing time unit of frame-to-frame counter 1 = use system clock as increasing time unit of frame-to-frame counter
13	R/W	0	Enable frame-to-frame time counter (running-counter). 0 = frame-to-frame time counter disabled 1 = frame-to-frame time counter enabled If enabled, the frame-to-frame counter increases every 100uS or 10uS until it reaches its max value(all Bits are 1) or it is reset. When it reaches its max value, it keeps the value until it is reset. When it is reset, it becomes zero and then begin increasing again. The counter can be reset even when it has not reached its max value. The increasing time unit can be 100uS or 10uS or system clock frequency which is set by running_count_tick_mode and use_clock_to_counter settings. When a frame's data are capured and stored into FrameBody/FrameBody_1 register, frame-to-frame counter is reset to zero. After reset to zero, the frame-to-frame counter will begin increasing again, until it reaches its max value or it is reset. For repeat frame detection, users can use hardware detection by enabling compare frame or compare bit method, or users can read frame-to-frame counter to let software to make the decision.
12	R/W	0	Enable repeat time check for repeat detection. This bit is valid only when compare frame method or compare Bit method is enabled. 0 = repeat time check disabled 1 = repeat time check enabled When repeat frame detection is enabled by enabling compare frame or compare Bit method, the frame time interval may need to be checked in order to decide whether the frames are repeat (key pressed without release) or not. You can configure the repeat_time_max value by setting 0x38 offset register. If frame interval is smaller than the "repeat time max", it may considered as repeat. If frame interval is bigger than the "repeat time max", it is considered as not repeat.

Bit(s)	R/W	Default	Description
11	R/W	0	Enable compare frame method for repeat detection. 0 = compare frame method disabled 1 = compare frame method enabled Some IR formats transer the same data frame as repeat frame when the key is kept pressed without release. For repeat detection, compare frame method can be used. If a new frame and the old received frame are the same and the repeat time is under the limit(frame-to-frame time counter value is smaller than the repeat_time_max), the status register's frame_status0 is set to 1 automatically as repeat detected flag. You can configure the repeat_time_max value by setting 0x38 offset register.
10	R/W	0	Enable compare Bit method for repeat detection. 0 = compare Bit method disabled 1 = compare Bit method enabled Some IR formats use only one bit to represet whether the frame is repeat. You can compare only one bit instead of compare the whole frame for repeat detection. If compare frame method is enabled, then This bit is ignored.
9	R/W	0	Disable read-clear of FrameBody/FrameBody_1. 0 = read-clear enabled 1 = read-clear disabled FrameBody/FrameBody_1 registers are read-cleared in default. When these register are read, they are cleared to zero. This bit is used to disable this read-clear feature. (FrameBody/FrameBody_1 registers are used to store captured frame data).
8	R/W	0	input stream bit order. 0 = LSB first mode (first bit in input stream is considered as LSB) 1= MSB first mode (first bit in input stream is considered as MSB) Note: Commonly the following formats shall set 1 to enable MSB first mode (unless you insist on LSBfirst mode for your specified use): RC5, RC5 extend, RC6, RCMM, Duokan, Comcast
7:4	R	0	Unused

Bit(s)	R/W	Default	Description
3:0	R/W	0	Decode_mode.(format selection) 0x0 =NEC 0x1= skip leader (just Bits, without leader) 0x2=General time measurement (measure width, software decode) 0x3=MITSUBISHI 0x4=Thomson 0x5=Toshiba 0x6=Sony SIRC 0x7=RC5 0x8=Reserved 0x9=RC6 0xA=RCMM 0xB=Duokan 0xC=Reserved 0xD=Reserved 0xE=Comcast 0xF=Sanyo

AO_MF_IR_DEC_DURATN2 0x19

Bit(s)	R/W	Default	Description
31-26	R	0	Unused
25-16	R/W	0	Max duration of Duration Setting Regiser 2. It defines max duration for: Half bit for RC5/6 format (RC5 typically 888.89us for half bit, RC6 typically 444.44us) or time of Duokan/RCMM/4ppm format’s Logic “10” or time of Comcast/16ppm’s base duration
15-10	R	0	Unused
9-0	R/W	0	Min duration of Duration Setting Regiser 2.

AO_MF_IR_DEC_DURATN3 0x1a

Bit(s)	R/W	Default	Description
31-26	R	0	Unused

25-16	R/W	0	Max duration of Duration Setting Regiser 3. It defines max duration for: Whole bit for RC5/6 format (RC5 typically 1777.78us for whole bit, RC6 typically 888.89us) or time of Duokan/RCMM/4ppm format's Logic "11" or time of Comcast/16ppm's offset duration
15-10	R	0	Unused
9-0	R/W	0	Min duration of Duration Setting Regiser 3.

AO_MF_IR_DEC_FRAME: Frame Body (Frame Data, LSB 32Bit) 0x1b

Note: New keys will be ignored until ***FrameBody*** register is read if the *hold first key* Bit is set in the decode control register. Reading this register resets an internal frame data valid flag.

Bit(s)	R/W	Default	Description
31-0	R	0	32 bit Read-Only register stores frame body (LSB 32 bit) captured from IR remote data flow, commonly includes custom/address code and data code.

AO_MF_IR_DEC_FRAME: Frame Body 1 (Frame Data, MSB 32Bit) 0x1b

Note: New keys will be ignored until ***FrameBody*** register is read if the *hold first key* Bit is set in the decode control register. Reading this register resets an internal frame data valid flag.

Bit(s)	R/W	Default	Description
31-0	R	0	Stores frame body excess 32 bit range. (MSB 32 bit)

AO_MF_IR_DEC_STATUS_1 0x1c

Bit(s)	R/W	Default	Description
31-20	R	0	Unused
19-0	R	0	Stores the last frame-to-frame counter value before the last counter reset caused by the last frame data record/update.

AO_MF_IR_DEC_STATUS_2 0x1d

Bit(s)	R/W	Default	Description
31-20	R	0	Unused
19-0	R	0	Stores the value of the frame-to-frame counter which is running currently.

12.6 Pulse-Width Modulation

12.6.1 Overview

The chip has 5 PWM modules that can be connected to various digital I/O pins, among which 3 are in EE domain and 2 is in AO domain. Each PWM is driven by a programmable divider driven by a 4:1 clock selector. The PWM signal is generated using two 16-bit counters. One is the High and Low counter, which is individually programmable with values between 1 and 65535. Using a combination of the divided clock (divide by N) and the HIGH and LOW counters, a wide number of PWM configurations are possible. The other is delta-sigma couter, generate 18-bit sigma, the PWM-out is the highest sigma. The PWM outputs vs counters are also illustrate below.

Figure 12-7 PWM Block Diagram

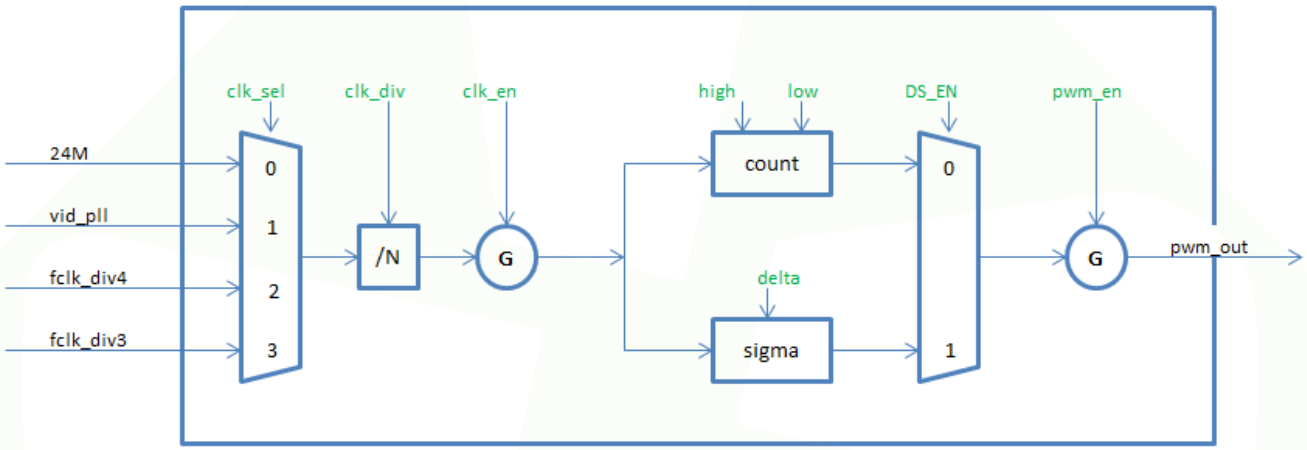


Figure 12-8 High/Low counter

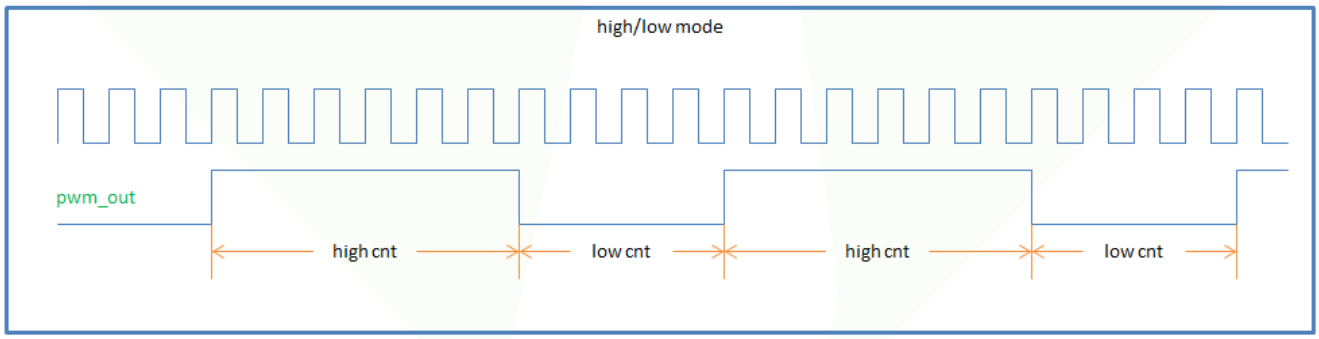
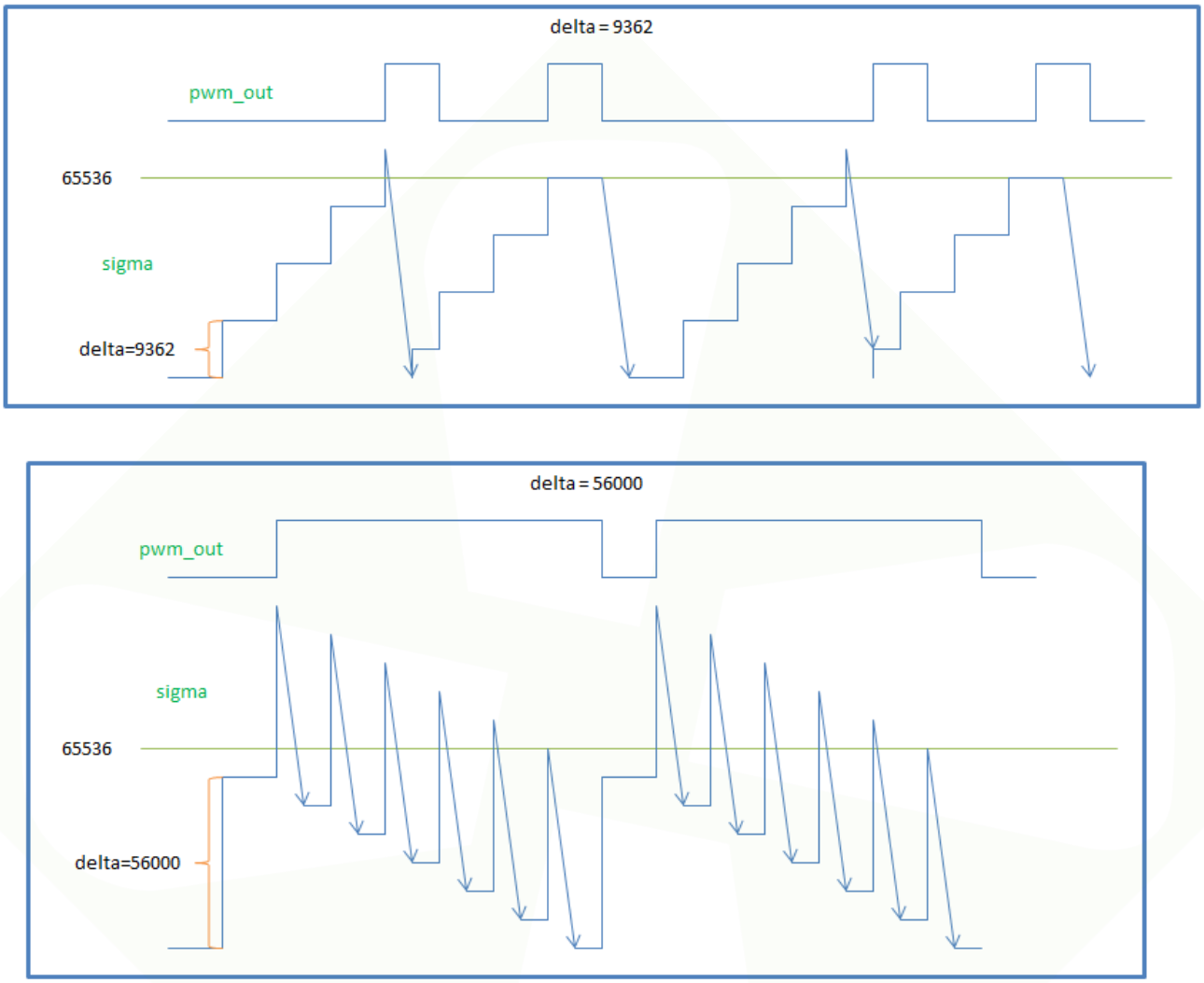
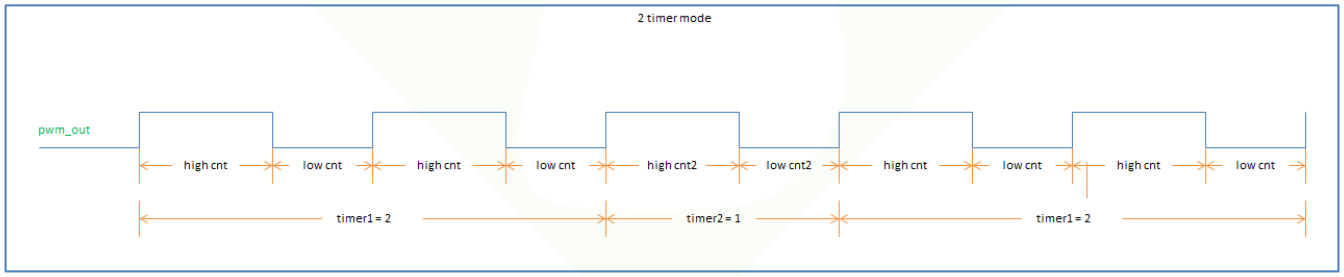


Figure 12-9 Delta-sigma conter, delta = 9362 & delta = 56000



PWM 2 timer mode is illustrated as following:

Figure 12-10 2 timer mode



12.6.2 Register Description

Each PWM module contains two PWM generators call A and B and controlled by the following registers.

For PWM modules in EE domain, the each register's final address =0xffd00000 + offset*4

PWM_PWM_A 0x6c00

Bit(s)	R/W	Default	Description
31-15	R/W	0	PWM_A_HIGH: This sets the high time (in clock counts) for the PWM_A generator output
15-0	R/W	0	PWM_A_LOW: This sets the high time (in clock counts) for the PWM_A generator output

PWM_PWM_B 0x6c01

Bit(s)	R/W	Default	Description
31-15	R/W	0	PWM_B_HIGH: This sets the high time (in clock counts) for the PWM_B generator output
15-0	R/W	0	PWM_B_LOW: This sets the high time (in clock counts) for the PWM_B generator output

PWM_MISC_REG_AB 0x6c02

Bit(s)	R/W	Default	Description
31	R/W	0	pwm_B_hiz when hiz mode, pwm_o will connect to pad_oe, then pad_o will equal this bit
30	R/W	0	pwm_A_hiz when hiz mode, pwm_o will connect to pad_oe, then pad_o will equal this bit
29	R/W	0	pwm_B_constant_en
28	R/W	0	pwm_A_constant_en
27	R/W	0	pwm_B_inv_en
26	R/W	0	pwm_A_inv_en
25	R/W	0	cntl_pwm_a2_en
24	R/W	0	cntl_pwm_b2_en
23	R/W	0	PWM_B_CLK_EN: Set this bit to 1 to enable PWM B clock
22-16	R/W	0	PWM_B_CLK_DIV: Selects the divider (N+1) for the PWM B clock. See the clock tress document
15	R/W	0	PWM_A_CLK_EN: Set this bit to 1 to enable PWM A clock
14-8	R/W	0	PWM_A_CLK_DIV: Selects the divider (N+1) for the PWM A clock. See the clock tress document
7-6	R/W	0	PWM_B_CLK_SEL: Select the clock for the PWM B. See the clock tress document
5-4	R/W	0	PWM_A_CLK_SEL: Select the clock for the PWM A. See the clock tress document

Bit(s)	R/W	Default	Description
3	R/W	0	DS_B_EN: This bit is only valid if PWM_B_EN is 0: if this bit is set to 1, then the PWM_B output is configured to generate a delta sigma output based on the settings in the register below. If this bit is set to 0, then the PWM_B output is set low.
2	R/W	0	DS_A_EN: This bit is only valid if PWM_A_EN is 0: if this bit is set to 1, then the PWM_A output is configured to generate a delta sigma output based on the settings in the register below. If this bit is set to 0, then the PWM_A output is set low.
1	R/W	0	PWM_B_EN: If this bit is set to 1, then the PWM_B output is configured to generate a PWM output based on the register above. If this bit is 0, then the PWM_B output is controlled by DS_B_EN above.
0	R/W	0	PWM_A_EN: If this bit is set to 1, then the PWM_A output is configured to generate a PWM output based on the register above. If this bit is 0, then the PWM_A output is controlled by DS_A_EN above.

DS_A_B 0x6c03

Bit(s)	R/W	Default	Description
31-15	R/W	0	DS_B_VAL: This value represents the delta sigma setting for channel B (PWM_B)
15-0	R/W	0	DS_A_VAL: This value represents the delta sigma setting for channel A (PWM_A)

PWM_TIME_AB 0x6c04

Bit(s)	R/W	Default	Description
31-24	R/W	0	A_timer1
23:16	R/W	0	A_timer2
15:8	R/W	0	B_timer1
7:0	R/W	0	B_timer2

PWM_A2 0x6c05

Bit(s)	R/W	Default	Description
31-15	R/W	0	PWM_A2_HIGH: This sets the high time (in clock counts) for the PWM_A2 generator output
15-0	R/W	0	PWM_A2_LOW: This sets the high time (in clock counts) for the PWM_A2 generator output

PWM_B2 0x6c06

Bit(s)	R/W	Default	Description
31-15	R/W	0	PWM_B2_HIGH: This sets the high time (in clock counts) for the PWM_B2 generator output
15-0	R/W	0	PWM_B2_LOW: This sets the high time (in clock counts) for the PWM_B2 generator output

PWM_BLINK_AB 0x6c07

Bit(s)	R/W	Default	Description
31-10	R	0	Reserved
9	R/W	0	blink enable for pwm B
8	R/W	0	blink enable for pwm A
7-4	R/W	0	blink times for pwm B
3-0	R/W	0	blink times for pwm A

PWM_PWM_C_D: 0x6800~0x6807

See the registers for PWM A/B

PWM_PWM_E_F: 0x6400~0x6407

See the registers for PWM A/B

AO PWM' clock sources are xtal, clk81, fclk_div3, fclk_div4.

For the following register, each register's final address =0xFF807000 + offset*4

AO_PWM_PWM_A: PWM_A_DUTY_CYCLE 0x0

Bit(s)	R/W	Default	Description
31-15	R/W	0	PWM_A_HIGH: This sets the high time (in clock counts) for the PWM_A generator output
15-0	R/W	0	PWM_A_LOW: This sets the high time (in clock counts) for the PWM_A generator output

AO_PWM_PWM_B: PWM_B_DUTY_CYCLE 0x1

Bit(s)	R/W	Default	Description
31-15	R/W	0	PWM_B_HIGH: This sets the high time (in clock counts) for the PWM_B generator output
15-0	R/W	0	PWM_B_LOW: This sets the high time (in clock counts) for the PWM_B generator output

AO_PWM_MISC_REG_AB: 0x2

Bit(s)	R/W	Default	Description
31	R/W	0	pwm_B_hiz when hiz mode, pwm_o will connect to pad_oe, then pad_o will equal this bit
30	R/W	0	pwm_A_hiz when hiz mode, pwm_o will connect to pad_oe, then pad_o will equal this bit
29	R/W	0	pwm_B_constant_en set this bit to 1, then pwm can support 0%(100%) duty output

Bit(s)	R/W	Default	Description
28	R/W	0	pwm_A_constant_en set this bit to 1, then pwm can support 0%(100%) duty output
27	R/W	0	pwm_B_inv_en set this bit to 1, pwm output is inverted
26	R/W	0	pwm_A_inv_en set this bit to 1, pwm output is inverted
25	R/W	0	Pwm_a2_en
24	R/W	0	Pwm_b2_en
23	R/W	0	PWM_B_CLK_EN: Set this bit to 1 to enable PWM B clock
22-16	R/W	0	PWM_B_CLK_DIV: Selects the divider (N+1) for the PWM B clock. See the clock tress document
15	R/W	0	PWM_A_CLK_EN: Set this bit to 1 to enable PWM A clock
14-8	R/W	0	PWM_A_CLK_DIV: Selects the divider (N+1) for the PWM A clock. See the clock tress document
7-6	R/W	0	PWM_B_CLK_SEL: Select the clock for the PWM B. See the clock tress document
5-4	R/W	0	PWM_A_CLK_SEL: Select the clock for the PWM A. See the clock tress document
3	R/W	0	DS_B_EN: This bit is only valid if PWM_B_EN is 0: if this bit is set to 1, then the PWM_B output is configured to generate a delta sigma output based on the settings in the register below. If this bit is set to 0, then the PWM_B output is set low.
2	R/W	0	DS_A_EN: This bit is only valid if PWM_A_EN is 0: if this bit is set to 1, then the PWM_A output is configured to generate a delta sigma output based on the settings in the register below. If this bit is set to 0, then the PWM_A output is set low.
1	R/W	0	PWM_B_EN: If this bit is set to 1, then the PWM_B output is configured to generate a PWM output based on the register above. If this bit is 0, then the PWM_B output is controlled by DS_B_EN above.
0	R/W	0	PWM_A_EN: If this bit is set to 1, then the PWM_A output is configured to generate a PWM output based on the register above. If this bit is 0, then the PWM_A output is controlled by DS_A_EN above.

AO_PWM_DELTA_SIGMA_AB 0x3

Bit(s)	R/W	Default	Description
31-15	R/W	0	DS_B_VAL: This value represents the delta sigma setting for channel B (PWM_B)
15-0	R/W	0	DS_A_VAL: This value represents the delta sigma setting for channel A (PWM_A)

AO_PWM_TIME_AB 0x4

Bit(s)	R/W	Default	Description
31-24	R/W	0	A1_timer
23:16	R/W	0	A2_timer
15:8	R/W	0	B1_timer
7:0	R/W	0	B2_timer

AO_PWM_A2 0x5

Bit(s)	R/W	Default	Description
31-15	R/W	0	PWM_A2_HIGH: This sets the high time (in clock counts) for the PWM_A2 generator output
15-0	R/W	0	PWM_A2_LOW: This sets the high time (in clock counts) for the PWM_A2 generator output

AO_PWM_B2 0x6

Bit(s)	R/W	Default	Description
31-15	R/W	0	PWM_B2_HIGH: This sets the high time (in clock counts) for the PWM_B2 generator output
15-0	R/W	0	PWM_B2_LOW: This sets the high time (in clock counts) for the PWM_B2 generator output

AO_PWM_BLINK_AB 0x7

Bit(s)	R/W	Default	Description
31-10	R	0	Reserved
9	R/W	0	blink enable for pwm B
8	R/W	0	blink enable for pwm A
7-4	R/W	0	blink times for pwm B
3-0	R/W	0	blink times for pwm A

For the following register, each register's final address= 0xFF802000 + offset * 4

AO_PWM_PWM_C: PWM_C_DUTY_CYCLE 0x0

This is a new module to Nike. It replaces the older delta sigma (PWM like) generator in the HIU. This module allows the software to select either a PWM or delta-sigma output using the same module. There are two outputs: PWM_C and PWM_D. Either of these can be programmed to be PWM outputs or delta sigma outputs.

Bit(s)	R/W	Default	Description
31-15	R/W	0	PWM_C_HIGH: This sets the high time (in clock counts) for the PWM_C generator output

Bit(s)	R/W	Default	Description
15-0	R/W	0	PWM_C_LOW: This sets the high time (in clock counts) for the PWM_C generator output

AO_PWM_PWM_D: PWM_D_DUTY_CYCLE 0x1

Bit(s)	R/W	Default	Description
31-15	R/W	0	PWM_D_HIGH: This sets the high time (in clock counts) for the PWM_D generator output
15-0	R/W	0	PWM_D_LOW: This sets the high time (in clock counts) for the PWM_Dgenerator output

AO_PWM_MISC_REG_CD: 0x2

Bit(s)	R/W	Default	Description
31	R/W	0	pwm_D_hiz when hiz mode, pwm_o will connect to pad_oe, then pad_o will equal this bit
30	R/W	0	pwm_C_hiz when hiz mode, pwm_o will connect to pad_oe, then pad_o will equal this bit
29	R/W	0	pwm_D_constant_en set this bit to 1, then pwm can support 0%(100%) duty output
28	R/W	0	pwm_C_constant_en set this bit to 1, then pwm can support 0%(100%) duty output
27	R/W	0	pwm_D_inv_en set this bit to 1, pwm output is inverted
26	R/W	0	pwm_C_inv_en set this bit to 1, pwm output is inverted
25	R/W	0	Pwm_C2_en
24	R/W	0	Pwm_D2_en
23	R/W	0	PWM_D_CLK_EN: Set this bit to 1 to enable PWM Dclock
22-16	R/W	0	PWM_D_CLK_DIV: Selects the divider (N+1) for the PWM Dclock. See the clock tress document
15	R/W	0	PWM_C_CLK_EN: Set this bit to 1 to enable PWM C clock
14-8	R/W	0	PWM_C_CLK_DIV: Selects the divider (N+1) for the PWM C clock. See the clock tress document
7-6	R/W	0	PWM_D_CLK_SEL: Select the clock for the PWM D. See the clock tress document
5-4	R/W	0	PWM_C_CLK_SEL: Select the clock for the PWM C. See the clock tress document

Bit(s)	R/W	Default	Description
3	R/W	0	DS_D_EN: This bit is only valid if PWM_D_EN is 0: if this bit is set to 1, then the PWM_D output is configured to generate a delta sigma output based on the settings in the register below. If this bit is set to 0, then the PWM_D output is set low.
2	R/W	0	DS_C_EN: This bit is only valid if PWM_C_EN is 0: if this bit is set to 1, then the PWM_C output is configured to generate a delta sigma output based on the settings in the register below. If this bit is set to 0, then the PWM_C output is set low.
1	R/W	0	PWM_D_EN: If this bit is set to 1, then the PWM_D output is configured to generate a PWM output based on the register above. If this bit is 0, then the PWM_D output is controlled by DS_D_EN above.
0	R/W	0	PWM_C_EN: If this bit is set to 1, then the PWM_C output is configured to generate a PWM output based on the register above. If this bit is 0, then the PWM_C output is controlled by DS_C_EN above.

AO_PWM_DELTA_SIGMA_CD 0x3

Bit(s)	R/W	Default	Description
31-15	R/W	0	DS_D_VAL: This value represents the delta sigma setting for channel D (PWM_D)
15-0	R/W	0	DS_C_VAL: This value represents the delta sigma setting for channel C (PWM_C)

AO_PWM_TIME_CD 0x4

Bit(s)	R/W	Default	Description
31-24	R/W	0	C1_timer
23:16	R/W	0	C2_timer
15:8	R/W	0	D1_timer
7:0	R/W	0	D2_timer

AO_PWM_C2 0x5

Bit(s)	R/W	Default	Description
31-15	R/W	0	PWM_C2_HIGH: This sets the high time (in clock counts) for the PWM_C generator output
15-0	R/W	0	PWM_C2_LOW: This sets the high time (in clock counts) for the PWM_C generator output

AO_PWM_D2 0x6

Bit(s)	R/W	Default	Description
31-15	R/W	0	PWM_D2_HIGH: This sets the high time (in clock counts) for the PWM_D generator output
15-0	R/W	0	PWM_D2_LOW: This sets the high time (in clock counts) for the PWM_D generator output

AO_PWM_BLINK_CD 0x7

Bit(s)	R/W	Default	Description
31-10	R	0	Reserved
9	R/W	0	blink enable for pwm D
8	R/W	0	blink enable for pwm C
7-4	R/W	0	blink times for pwm D
3-0	R/W	0	blink times for pwm C

12.7 SAR ADC

12.7.1 Overview

This SAR ADC is a general purpose ADC for measuring analog signals. The module can make RAW ADC measurements or average a number of measurements to introduce filtering. The SAR ADC is a single block so an analog mux is placed in front to allow multiple different measurements to be made sequentially. Timing of the samples, and delays between muxing are all programmable as is the averaging to be applied to the SAR ADC.

12.7.2 Register Description

Each register final address = 0xff809000 + offset * 4

SAR_ADC_REG0: Control Register #0 0x80

Bit(s)	R/W	Default	Description
31	R	0	PANEL DETECT level.
30	R/W	0	DELTA_BUSY: If This bit is 1, then it indicates the delta processing engine is busy
29	R/W	0	AVG_BUSY: If This bit is 1, then it indicates the averaging engine is busy
28	R/W	0	SAMPLE_BUSY: If This bit is 1, then it indicates the sampling engine is busy
27	R/W	0	FIFO_FULL:
26	R/W	0	FIFO_EMPTY:
25-21	R/W	4	FIFO_COUNT: Current count of samples in the acquisition FIFO
20-19	R/W	0	ADC_BIAS_CTRL
18-16	R/W	0	CURR_CHAN_ID: These Bits represent the current channel (0..7) that is being sampled.
15	R/W	0	ADC_TEMP_SEN_SEL
14	R/W	0	SAMPLING_STOP: This bit can be used to cleanly stop the sampling process in the event that continuous sampling is enabled. To stop sampling, simply set This bit and wait for all processing modules to no longer indicate that they are busy.
13-12	R/W	0	CHAN_DELTA_EN: There are two Bits corresponding to Channels 0 and 1. Channel 0 and channel 1 can be individually enabled to take advantage of the delta processing module.
11	R/W	0	Unused
10	R/W	0	DETECT_IRQ_POL: This bit sets the polarity of the detect signal. The detect signal is used during X/Y panel applications to detect if the panel is touched
9	R/W	0	DETECT_IRQ_EN: If This bit is set to 1, then an interrupt will be generated if the DETECT signal is low/high. The polarity is set in the bit above.
8-4	R/W	0	FIFO_CNT_IRQ: When the FIFO contains N samples, then generate an interrupt (if bit 3 is set below).