Amlogic S905X S905L GPIO User Guide

Revision 0.1

Distribute to Wesion!

Amlogic, Ltd.
2518 Mission College Blvd, Suite 120
Santa Clara, CA 95054
U.S.A.
www.amlogic.com

Legal Notices

© 2016 Amlogic, Ltd. All rights reserved. Amlogic ® is registered trademarks of Amlogic, Ltd. All other registered trademarks, trademarks and service marks are property of their respective owners.

This document is Amlogic Company confidential and is not intended for any external distribution without authorization.

Distribute to Wesion!

Table of Content

Distribute to Wesion!

Revision History

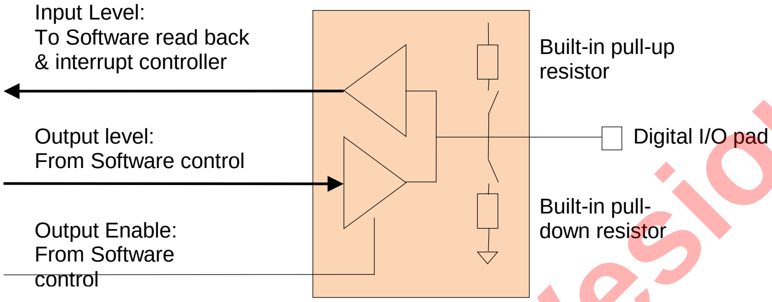
Revision	Revised Date	By	Changes
0.1	15th,Apr., 2016	Yan Han	Initial release draft
0.2	04th, May,2016	Yan Han	Add GPIOZ information

Distribute to Wesion!

1. Overview

The SOC has a number of multi-function digital I/O pads that can be multiplexed to a number of internal resources (e.g. PWM generators, SDIO controllers …). When a digital I/O is not being used for any specific purpose, it is converted to a general purpose GPIO pin. A GPIO pin can be statically set to high/low logical levels. The structure of a GPIO is given below.

Figure 1. GPIO Structure



Each GPIO pin has an individual control bitthat can be used to set the output level, output enable of the I/O pad (0 = output, 1 = input). Additionally all digital I/O pads have built-in pull-up and pull-down resistors. The input from the digital I/O pad can be read back in software and is also connected to an interrupt controller. The interrupt controller allows up to 8 GPIO pins to be used as active high, active low, rising edge or falling edge interrupts.

Finally, the GPIO's are grouped into voltage domains. Some GPIOs (depending on the PCB design) can be configured to operate between 0 and 1.8v while others may operate between 0.0v and 3.3v.

Distribute to wesion!

2. GPIO Multiplex Function

The GPIO multiplex functions are shown in the sections below, where the RegNN[MM] corresponds to CBUS registers defined in Table 1

Table 1 Pin Mux Registers

Pin Mux Registers	Abbreviation	Offset	Default Value after power-on/Reset
PERIPHS_PIN_MUX_0	REG0	0xc88344b0	0x0000
PERIPHS_PIN_MUX_1	REG1	0xc88344b4	0x0000
PERIPHS_PIN_MUX_2	REG2	0xc88344b8	0x0000
PERIPHS_PIN_MUX_3	REG3	0xc88344bc	0x0000
PERIPHS_PIN_MUX_4	REG4	0xc88344c0	0x0000
PERIPHS_PIN_MUX_5	REG5	0xc88344c4	0x0000
PERIPHS_PIN_MUX_6	REG6	0xc88344c8	0x0000

Pin Mux Registers	Abbreviation	Offset	Default Value after power-on/Reset
PERIPHS_PIN_MUX_7	REG7	0xc88344cc	0x0000
PERIPHS_PIN_MUX_8	REG8	0xc88344d0	0x0000
PERIPHS_PIN_MUX_9	REG9	0xc88344d4	0x0000
AO_RTI_PIN_MUX_REG	AO_REG	0xc8100014	0x0000
AO_RTI_PIN_MUX_REG2	AO_REG2	0xc8100018	0x0000

Table 2 GPIO Bank X Pin Multiplexing Table

Package Name	Func1	Func2	Func3	Func4
GPIOX_0	SDIO_D0 reg5[31]			
GPIOX_1	SDIO_D1 reg5[30]			
GPIOX_2	SDIO_D2 reg5[29]			
GPIOX_3	SDIO_D3 reg5[28]			
GPIOX_4	SDIO_CLK reg5[27]			
GPIOX_5	SDIO_CMD reg5[26]			
GPIOX_6	PWM_A reg5[25]			
GPIOX_7	SDIO_IRQ reg5[24]	PWM_F reg5[14]		
GPIOX_8	PCM_OUT_A reg5[23]	UART_TX_C reg5[13]		SPI_MOSI reg5[3]
GPIOX_9	PCM_IN_A reg5[22]	UART_RX_C reg5[12]		SPI_MISO reg5[2]
GPIOX_10	PCM_FS_A reg5[21]	UART_CTS_C reg5[11]	I2C_SDA_D reg5[5]	SPI_SS0 reg5[1]
GPIOX_11	PCM_CLK_A reg5[20]	UART_RTS_C reg5[10]	I2C_SCK_D reg5[4]	SPI_SCLK reg5[0]
GPIOX_12	UART_TX_A (long fifo) reg5[19]	SLIP_UART_TX reg5[9]		
GPIOX_13	UART_RX_A reg5[18]	SLIP_UART_RX reg5[8]		
GPIOX_14	UART_CTS_A reg5[17]	SLIP_UART_CTS reg5[7]		
GPIOX_15	UART_RTS_A reg5[16]	SLIP_UART_RTS reg5[6]		
GPIOX_16	PWM_E reg5[15]			

Distribute to Wesion!

Package Name	Func1	Func2	Func3	Func4
GPIOX_17	GPIO (BT_EN)			
GPIOX_18	GPIO (BT_WAKE_CPU)			

Table 3 GPIO Bank DV Pin Multiplexing Table

Package Name	Func1	Func2	Func3
GPIODV_24	UART_TX_B reg2[16]	DMIC_IN reg2[7]	I2C_SDA_A reg1[15]
GPIODV_25	UART_RX_B reg2[15]	DMIC_CLK_OUT reg2[6]	I2C_SCK_A reg1[14]
GPIODV_26	UART_CTS_B reg2[14]		I2C_SDA_B reg1[13]
GPIODV_27	UART_RTS_B reg2[13]		I2C_SCK_B reg1[12]
GPIODV_28	PWM_D reg2[12]	REMOTE_INPUT reg1[9]	I2C_SDA_C reg1[11]
GPIODV_29	PWM_B reg2[11]	PWM_VS reg2[5]	I2C_SCK_C reg1[10]

Table 4 GPIO Bank H Pin Multiplexing Table

Package Name	Func1	Func2	Func3	Func4
GPIOH_0	HDMI_HPD reg6[31]			
GPIOH_1	HDMI_SDA reg6[30]			
GPIOH_2	HDMI_SCL reg6[29]			
GPIOH_3	GPIO(5V_EN)			
GPIOH_4	SPDIF_OUT reg6[28]	SPDIF_IN reg6[27]		
GPIOH_5				
GPIOH_6		JTAG_TCK	I2S_AM_CLK reg6[26]	
GPIOH_7		JTAG_TMS	I2S_AO_CLK_OUT reg6[25]	I2S_AO_CLK_IN reg6[22]
GPIOH_8		JTAG_TDI	I2S_LR_CLK_OUT reg6[24]	I2S_LR_CLK_IN reg6[21]
GPIOH_9		JTAG_TDO	I2SOUT_CH01 reg6[23]	

Table 5 GPIO Bank CLK Pin Multiplexing Table

Package Name	Func1	Func2
GPIOCLK_0	CLK24	CLK12(wifi)
GPIOCLK_1	CLK25	pwm_F

Distribute to Wesion!

Table 6 GPIO Bank AO Pin Multiplexing Table

Pin Name	Func1	Func2	Func3	Func4
GPIOAO_0	UART_TX_AO_A ao_reg[12]	UART_TX_AO_B ao_reg[26]		
GPIOAO_1	UART_RX_AO_A ao_reg[11]	UART_RX_AO_B ao_reg[25]		
GPIOAO_2	UART_CTS_AO_A ao_reg[10]	UART_CTS_AO_B ao_reg[8]		
GPIOAO_3	UART_RTS_AO_A ao_reg[9]	UART_RTS_AO_B ao_reg[7]		PWM_AO_A ao_reg[22]
GPIOAO_4	UART_TX_AO_B ao_reg[24]	I2C_SCK_AO ao_reg[6]	I2C_SLAVE_SCK_AO ao_reg[2]	
GPIOAO_5	UART_RX_AO_B ao_reg[23]	I2C_SDA_AO ao_reg[5]	I2C_SLAVE_SDA_AO ao_reg[1]	
GPIOAO_6	CLK_32K_IN		SPDIF_OUT ao_reg[16]	PWM_AO_B ao_reg[18]
GPIOAO_7	REMOTE_INPUT ao_reg[0]	REMOTE_OUTPUT ao_reg[21]		
GPIOAO_8	AO_CEC ao_reg[15]	EE_CEC ao_reg[14]	I2SOUT_CH23 ao_reg2[0]	PWM_AO_A ao_reg[17]
GPIOAO_9	REMOTE_OUTPUT ao_reg[31]	SPDIF_OUT ao_reg[4]	I2SOUT_CH45 ao_reg2[1]	PWM_AO_B ao_reg[3]
TEST_N	TEST_N	wd_gpio ao_reg[13:20]	I2SOUT_CH67 ao_reg2[2]	
RESET_N	RESET_N			

Table 7 GPIO Bank BOOT Pin Multiplexing Table

Pin Name	Func1	Func2	Func3
BOOT_0	EMMC_NAND_D0 reg7[31]		
BOOT_1	EMMC_NAND_D1 reg7[31]		
BOOT_2	EMMC_NAND_D2 reg7[31]		
BOOT_3	EMMC_NAND_D3 reg7[31]		
BOOT_4	EMMC_NAND_D4 reg7[31]		
BOOT_5	EMMC_NAND_D5 reg7[31]		
BOOT_6	EMMC_NAND_D6 reg7[31]		
BOOT_7	EMMC_NAND_D7 reg7[31]		
BOOT_8	EMMC_CLK reg7[30]	NAND_CE0 reg7[7]	
BOOT_9		NAND_CE1 reg7[6]	
BOOT_10	EMMC_CMD reg7[29]	NAND_RB0 reg7[5]	

Distribute to Wesion!

Pin Name	Func1	Func2	Func3
BOOT_11		NAND_ALE reg7[4]	NOR_D reg7[13]
BOOT_12		NAND_CLE reg7[3]	NOR_Q reg7[12]
BOOT_13		NAND_WEN_CLK reg7[2]	NOR_C reg7[11]
BOOT_14		NAND_REN_WR reg7[1]	

Table 8 GPIO Bank CARD Pin Multiplexing Table

Pin Name	Func1	Func2	Func3	Func4
CARD_0	SDCARD_D1 reg6[5]			JTAG_TDI
CARD_1	SDCARD_D0 reg6[4]			JTAG_TDO
CARD_2	SDCARD_CLK reg6[3]			JTAG_CLK
CARD_3	SDCARD_CMD reg6[2]			JTAG_TMS
CARD_4	SDCARD_D3 reg6[1]	UART_TX_AO_A reg6[9]	UART_RX_AO_A reg6[11]	
CARD_5	SDCARD_D2 reg6[0]	UART_RX_AO_A reg6[8]	UART_TX_AO_A reg6[10]	
CARD_6	GPIO(Card_DET / EN)			

Table 9. GPIOZ_x Multi-Function Pin

Pin Name	Func1	Func2
GPIOZ_14	SPDIF_IN reg3[21]	ETH_LINK_LED reg4[25]
GPIOZ_15	PWM_C reg3[20]	ETH_ACT_LED reg4[24]

Distribute to Wesion!

3. GPIO Interrupt

There are 8 independent filtered GPIO interrupt modules that can be programmed to use any of the GPIOs in the chip as an interrupt source (listed in the table below). For example, to select boot_3 as the source for GPIO IRQ #0, then CBUS 0x2621 Bits[7:0] = 27 (according to the table below).

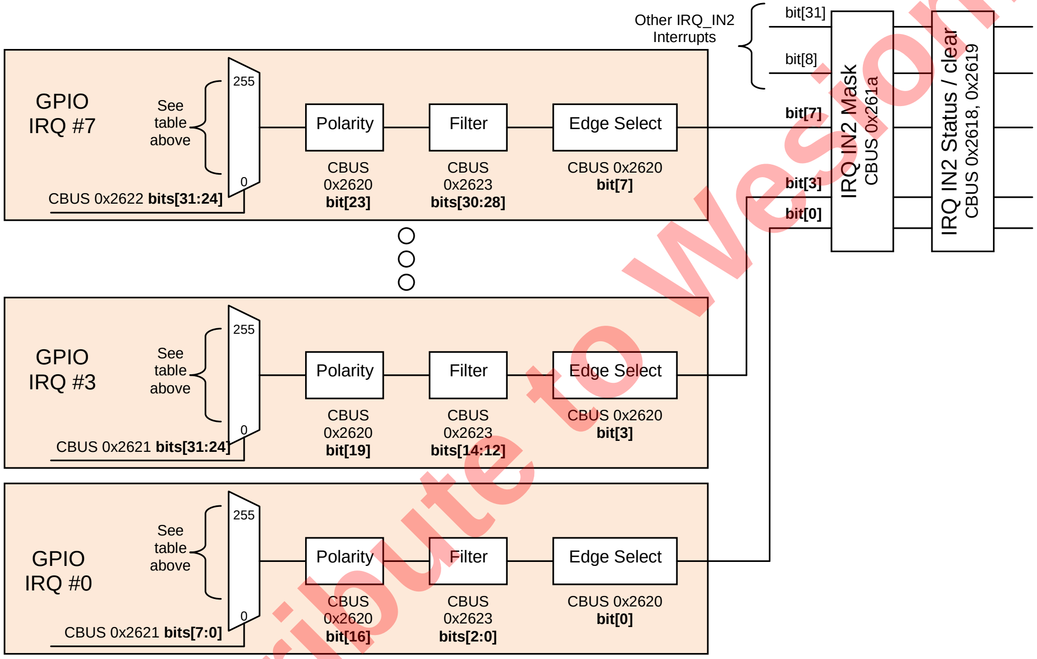
Table 10.GPIO Interrupt Sources

Input Mux Location CBUS registers 0x2621 and 0x2622	Description
255~133	Undefined (no interrupt)
132~129	gpioCLK[3:0]
128~106	gpioX[22:0]
88~59	gpioDV[29:0]

Input Mux Location CBUS registers 0x2621 and 0x2622	Description
58~52	Card[6:0]
51~34	Boot[17:0]
33~30	gpioH[3:0]
29~14	gpioZ[15:0]
13 ~ 0	gpioAO[13:0]

The diagram below illustrates the path a GPIO takes to become an interrupt. The eight GPIO interrupts respond to the MASK, STATUS and STATUS/CLEAR registers just like any other interrupt in the chip. The difference for the GPIO interrupts is that they can be filtered and conditioned. Filtering can either be bypassed or enabled so that a GPIO must be high or low for [3n] * 125nS, where n = 1, 2, …, 7.

Fig 2 GPIO Interrupt Path



Distribute to Weston!

NOTE: The input for the GPIO interrupt module (the input into the 256:1 mux) comes directly from the I/O pad of the chip. Therefore if a pad (say card_6) is configured as a UART TX pin, then in theory, the UART TX pin can be a GPIO interrupt since the TX pin will drive card_6 which in turn can drive the GPIO interrupt module.

4. Register Description

Table 11 and Table 12 shows the information of GPIO related registers.

Table 11 GPIO Register Infromation I
Finale address = 0xc8834400 + address * 4

Package Name	**OEN (Read Write)**	**OUT (Write Only)**	**IN (Read Only)**	**IO Type**
GPIOX_18~0	0x18 Bit[18:0]	0x19 Bit[18:0]	0x1a Bit[18:0]	Tri-State
GPIODV_29~0	0x0C Bit[29:0]	0x0D Bit[29:0]	0x0E Bit[29:0]	Tri-State
GPIOH_9~0	0x0F Bit[29:20]	0x10 Bit[29:20]	0x11 Bit[29:20]	Tri-State
BOOT_15~0	0x12 Bit[15:0]	0x13 Bit[15:0]	0x14 Bit[15:0]	Tri-State
CARD_6~0	0x12 Bit[26:20]	0x13 Bit[26:20]	0x14 Bit[26:20]	Tri-State
CLK_1~0	0x15 Bit[29:28]	0x16 Bit[29:28]	0x17 Bit[29:28]	Tri-State
GPIOZ_15~14	0x15 Bit[15:14]	0x16 Bit[15:14]	0x17 Bit[15:14]	Tri-State

Table 12 GPIO Register Information II
Finale address = 0xc8100000 + address * 4

Package Name	**OEN (Read Write)**	**OUT (Write Only)**	**IN (Read Only)**	**IO Type**
GPIOAO_9~0	0x09 Bit[9:0]	0x09 Bit[25:16]	0x0a Bit[9:0]	Tri-State

Distribute to Wesion!

Pad pull-up/down Direction

The I/O pads contain both a pull-up and a pull-down. If a bit is set to 1 in the registers below, then the pull-up is enabled. If a bit is set to 0, then the pull-down is enabled.

NOTE: There are separate pull-up "enables" that must also be set to 1 in order for the pull-up/down direction to function. If an "enable" is set to 0, then the pull-up/down feature is disabled and the bits below are ignored (on a per pad basis).

PULL_UP_REG0 0x3a

Bit(s)	**R/W**	**Default**	**Description**
31~30	R/W		Unused
29~0	R/W		gpioDV[29:0] 1 = pull up. 0 = pull down

PULL_UP_REG1 0x3b

Bit(s)	R/W	Default	Description
31~30	R/W		Unused
29~20	R/W		gpioH[9:0] 1 = pull up. 0 = pull down
19~0	R/W		Reserved

PULL_UP_REG2 0x3c

Bit(s)	R/W	Default	Description
31~27	R/W		Reserved
26~20	R/W		card[6:0] 1 = pull up. 0 = pull down
19~16	R/W		Reserved
15~0	R/W		boot[15:0] 1 = pull up. 0 = pull down

PULL_UP_REG3 0x3d

Bit(s)	R/W	Default	Description
29~28	R/W		gpioCLK[1:0] 1= pull up. 0=pull down
27~16	R/W		reserved
15~0	R/W		gpioZ[15:0] 1 = pull up. 0 = pull down

PULL_UP_REG4 0x3e

Bit(s)	R/W	Default	Description
31~19	R/W		Unused
18~0	R/W		gpioX[18:0] 1 = pull up. 0 = pull down

Pad Pull-Up/Down Enables

Each I/O pad has a selectable pull-up or pull-down resistor. In order for the pull-up direction (up or down) to be operational, the appropriate bit below must be set in order to enable the pull-up/down function.

PULL_UP_EN_REG0 0x48

Bit(s)	R/W	Default	Description
31~30	R/W	0	Unused
29~0	R/W	0x0	gpioDV[29:0] pullup-enable: 1 = pullup or pull-down enabled. 0 = no-pull-up or pull-down

PULL_UP_EN_REG1 0x49

Bit(s)	R/W	Default	Description
31~30	R/W	0	Unused
29~20	R/W	0x0	gpioH[9:0] pullup-enable: 1 = pullup or pull-down enabled. 0 = no-pull-up or pull-down
19~0	R/W	0	Reserved

PULL_UP_EN_REG2 0x4a

Bit(s)	R/W	Default	Description
31~27	R/W	0	Reserved
26~20	R/W	0x1FF	card[6:0] pullup-enable: 1 = pullup or pull-down enabled. 0 = no-pull-up or pull-down
19~16	R/W	1	Reserved
15~0	R/W	0x3FFFF	boot[15:0] pullup-enable: 1 = pullup or pull-down enabled. 0 = no-pull-up or pull-down

PULL_UP_EN_REG3 0x4b

Distribute to Wesion!

Bit(s)	R/W	Default	Description
29~28	R/W	0xF	gpioCLK[1:0] pullup-enable: 1 = pullup or pull-down enabled. 0 = no-pull-up or pull-down
27~16	R/W	0	reserved
15~0	R/W	0x3FFF	gpioZ[15:0] pullup-enable: 1 = pullup or pull-down enabled. 0 = no-pull-up or pull-down

PULL_UP_EN_REG4 0x4c

Bit(s)	R/W	Default	Description
31~19	R/W	0	Unused
18~0	R/W	0x6FFFBF	gpioX[18:0] pullup-enable: 1 = pullup or pull-down enabled. 0 = no-pull-up or pull-down

Distribute to Wesion!