AIT8427

Data Sheet
(Preliminary)

Issued Date	
Released Edition	0.1
Document number	TD8427 12030001
Applicant	
To Customer	

Notice

(1) This preliminary specification contains confidential material and information. No part of this document is to be reproduced in any form or by any means without prior agreement from Alpha Imaging Technology Corp.
(2) The preliminary specification is provided “As Is”, and might be subject to change without prior notice.
(3) Alpha Imaging Technology Corp. disclaims all warranties with regard to this preliminary specification.

Revision History

Release	Date	Modification
R0.1	2011/03/30	Preliminary Release

Contents

Figures

Tables

1 Overview

The AIT8427 is a highly integrated multi-media processor for portable devices focused on rich multimedia functions.

The AIT8427 includes an embedded 32-bit ARM9 processor, AIT's advanced ISP engine, Full High Definition (1080P) H264 video CODEC, hardware 2D-Graphic Engine, SD/MMC/MS interface, USB 2.0 High Speed and LCD Display Interface.

The powerful on-chip ISP (image signal processor) supports Anti-Shaking mechanism and implements the most advanced algorithm to deliver high-quality image and precision control for AF(Auto Focus), AE(Auto Exposure) and AWB(Auto White Balance). The maximum resolution supported by AIT8427 is 12 mega–pixel.

The H.264 video CODEC supports up to 30 frames per second with 1080P resolution. Pure hardwired architecture achieves low power operation and extends battery time.

Audio functions such as MP3, WMA, AAC, EAAC+, MIDI, SBC and AMR are also supported for features such as music streaming and playback, voice recording, audio/video synchronization, etc. All audios can be played with 3D surround sound.

The 2D graphic hardware acceleration enhances the GUI performance.

The LCD Display Interface of AIT8427 supports dual displays that can be TFT, TFD, LTPS, or Color-STN LCD panels. The AIT8427 could support a wide range of resolutions of LCD panels up to WVGA with 16 million colors.

A built-in HDMI transmitter enables the direct connection between AIT8427 to the high definition TV sets. TV out is also supported for both NTSC and PAL.

The AIT8427, powered by Alpha Imaging Technology, will provide complete development environment for customer. "Time-to-Market" is possible.

Applications

- DV
- HD Webcam

1.1 Function List

□ **CPU**

- ARM926 EJ-S 32 bit processor
- Clock frequency up to 400MHz

□ **Memory**

- 16KB I-cache and 16KB D-cache
- 8KB I-TCM and 8KB D-TCM
- Internal SRAM
- Stacked DDR SDRAM up to 512Mb

□ **DSC**

- Support up to 12M pixel sensor with Bayer, YUV format in parallel interface
- MIPI CSI interface support
- High performance scaling engine to support real-time image scale-up interpolation
- Support advanced image effort such as face detection, anti-shaking, panorama, and smile-detection

□ **ISP**

- Calibrated and Enhanced Automatic Defect Pixel Compensation
- Lens shading correction
- Enhanced Interpolation (Demosaic) algorithm
- Enhanced chroma shading
- High Performance Noise Reduction
- Space Color non-Uniformity Compensation
- Enhanced False Color Reduction
- Color space conversion
- Enhanced Gamma table
- Space-variant non-linear Wide Dynamic Range control
- Edge Enhancement
- High performance Anti-crosstalk
- 3D LUT color correction
- Histogram Equalization or Modification
- Blue edge reduction
- Anti-flare
- Brightness/Contrast enhancement
- Hue/Saturation enhancement
- Enhanced Auto-Exposure/ Auto-White-Balance/ Auto-focus
- Black Level Compensation
- 2D filtering to reduce jaggy
- Support multiple special effect like sepia, binary, emboss, negative, sketch, oil, crayon, blackboard
- Support Flash strobe function (IGBT and LED)
- Support Optical Zoom lens
- Support mechanical shutter
- Step-less linear zoom up to 50 steps
- Support digital zoom up to 16x
- Support still image stabilizer
- Support face detection function
- Support panorama function (with stacked SDRAM)

□ **LCD**

- Support LCD panel up to WVGA, with 16M colors
- Support 8/9/12/16 bit CPU and RGB interface
- 4 layers Graphics mode OSD
- Transparent and overlay PIP support
- Real-time scaling display engine
- Image rotation 90/180/270 degree and Mirror display
- 2 layers alpha blending support

☐ **JPEG**

- Real-time high performance JPEG engine
- Compliant with JPEG baseline standard (ISO/IEC 10918) with JFIF.
- Hardware JPEG engine supports up to 15 frame per second @ 8M resolution
- Supports YUV 422 encoder format
- Supports YUV 444/422/420/411 decoder format

☐ **Video**

- Pure hardware-based video engine for low power requirement
- H.264 Baseline, Main and High profiles encode at 1080P @ 30fps
- H.264 Baseline, Main and High profiles decode at 1080P @ 30fps
- Hardware supported post processing filter

☐ **Voice / Audio**

- On-chip audio ADC/DAC for voice / audio recording and playback
- High performance audio codec
- Optimized power performance for audio
- Optional I2S interface
- Synchronous digital audio interface
- AMR, MP3, MP3 HD, WMA, WMA Pro, AAC-LC, EAAC, EAAC+, 64-polyphonics stereo MIDI, Bluetooth SBC, Real Audio, OGG etc..
- α-EQ 10-band graphic equalizer
- α -3D 3-dimensinal surround sound effect

☐ **2D Graphic Engine**

- BitBLT
- Line draw
- Color expansion
- Raster Operation
- Pattern / Solid Fill
- Transparent overlay
- Hardware cursor
- Hardware image rotate, scaling and color format transform

☐ **TV out**

- HDMI Transmitter built-in
- 10-bit DAC for direct composite signal to TV
- Support NTSC / PAL

☐ **Storage Controller**

- Support SD / SDIO / mini-SD / T-Flash / MMC / RS-MMC / MS host controller
- Support 512 / 2K bytes page SLC / MLC NAND flash

☐ **USB**

- USB 2.0 High Speed device controller
- Support Mass Storage, PC cam
- Support MTP for Windows DRM
- Support PictBridge direct printer connection

☐ **Peripheral**

- Built-in HDMI Transmitter
- TV DAC
- SPI
- UART
- PWM
- GPIOs
- IGBT
- PCM
- Power-on Button

☐ **Clock**

- Support input range from 2~50MHz
- Multiple PLL for various application combination

☐ **Package**

- 13 x 13mm 276-pin STFBGA with 1.2mm thickness

1.2 Block Diagram

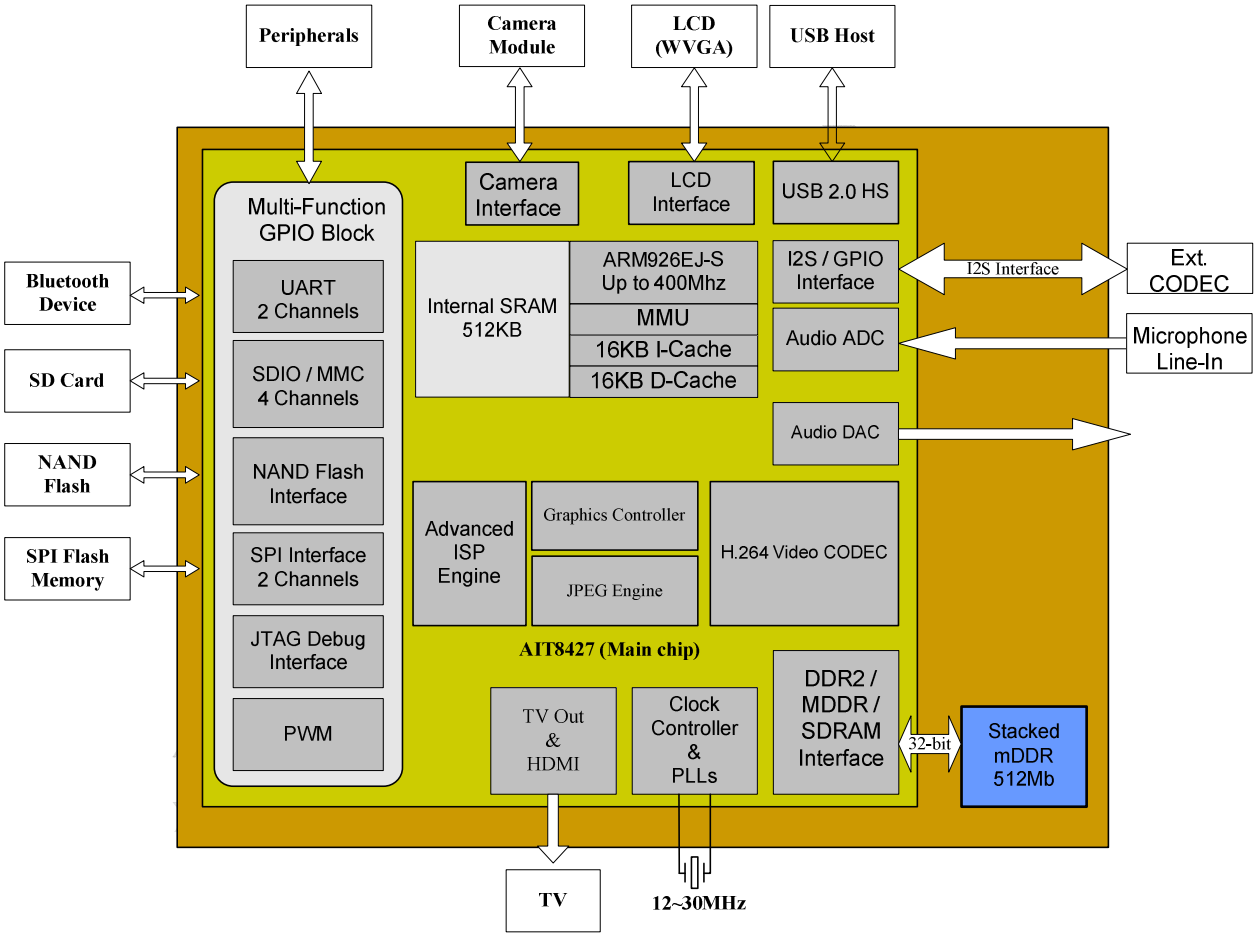


Figure 1. Block Diagram

2 Signal Description

2.1 Pin List

I : Input pin
IO: Bidirection pin
ID: Input pin with pull-down configuration
IU: Input pin with pull-up configuration
IOU: Bidirection pin with pull-up configuration
IOD: Bidirection pin with pull-down configuration
O: Output pin
Z: Tri-state at reset
HZ: Tri-state Hi-Z at reset
LZ: Tri-state Lo-Z at reset

Pin Num	Pin Name	Type	Loc.	Rst# state	Description
HOST_IF or AGPIO (18 pins) (VDD_HIF/VSS)					
1	PRST_	IU	R11	Z	Chip reset (low active)
2	PHCS_	IOU	U10	HZ	Host bus chip enable
3	PHRD_	IOD	U11	LZ	Host bus read enable
4	PHWE_	IOD	V11	LZ	Host bus write enable
5	PHD0	IOD	V12	LZ	Host EBI data bus [0]
6	PHD1	IOD	U12	LZ	Host EBI data bus [1]
7	PHD2	IOD	T12	LZ	Host EBI data bus [2]
8	PHD3	IOD	V10	LZ	Host EBI data bus [3]
9	PHD4	IOD	V13	LZ	Host EBI data bus [4]
10	PHD5	IOD	U13	LZ	Host EBI data bus [5]
11	PHD6	IOD	U14	LZ	Host EBI data bus [6]
12	PHD7	IOD	T11	LZ	Host EBI data bus [7]
13	PHLCD_BY	IOD	T13	LZ	LCD bypass mode enable
14	PHLCD_A0	IOD	R13	LZ	Command or data selection
15	PHINT	IOD	L10	LZ	Host bus interrupt
16	PHWAIT_	IOU	M11	HZ	Host bus wait signal
17	PHI2C_SDA	IOU	M9	HZ	Host I2C data
18	PHI2C_SCL	IOU	M10	HZ	Host I2C clock
Main Clock interface (2 pins) (VDD_CLK/VSS_CLK)					
19	XSCI	I	A3	Z	Crystal oscillator and main clock input
20	XSCO	O	A4	Z	Crystal oscillator output
I2S Serial Interface (6 pins) (VDD_I2S/VSS)					
21	PI2S_SCK	IOD	T14	LZ	I2S_SCK
22	PI2S_WS	IOD	U15	LZ	I2S_WS
23	PI2S_SDO	IOD	U16	LZ	I2S_SDO
24	PI2S_SDI	IOD	V16	LZ	I2S_SDI
25	PI2S_MCLK	IOD	V15	LZ	I2S_MCLK
26	PHI2S_SDO	IOD	T15	LZ	Host bypass to I2S SDO output
Sensor Interface (19 pins) (VDD_SEN/VSS)					
27	PHSYNC	ID	G16	LZ	Sensor HD
28	PVSYNC	ID	H15	LZ	Sensor VD
29	PD0	ID	E17	LZ	Sensor raw data [0]

Pin Num	Pin Name	Type	Loc.	Rst# state	Description
30	PD1	ID	G15	LZ	Sensor raw data [1]
31	PD2	ID	D16	LZ	Sensor raw data [2]
32	PD3	ID	F15	LZ	Sensor raw data [3]
33	PD4	ID	E16	LZ	Sensor raw data [4]
34	PD5	ID	D17	LZ	Sensor raw data [5]
35	PD6	ID	C17	LZ	Sensor raw data [6]
36	PD7	ID	F16	LZ	Sensor raw data [7]
37	PD8	ID	C18	LZ	Sensor raw data [8]
38	PD9	ID	C16	LZ	Sensor raw data [9]
39	PSEN	IOD	B17	LZ	Sensor enable
40	PSDA	IOD	B18	LZ	Sensor serial interface data
41	PSCK	IOD	A17	LZ	Sensor serial interface clock
42	PS_RST_	IOD	C15	LZ	Sensor reset
43	PDCLK	O	D18	L	Sensor clock
44	PPXL_CLK	ID	A18	LZ	Sensor pixel clock
45	PS_GPIO	IOD	A16	LZ	Sensor GPIO
MIPI-RX I/F (10 pins) (VDD_MIPI_RX/VSS_MIPI_RX)					
46	MIPI_RX_CKP	AI	H18	Z	MIPI RX D-PHY positive clock lane input
47	MIPI_RX_CKN	AI	H17	Z	MIPI RX D-PHY negative clock lane input
48	MIPI_RX_DA0P	AI	K18	Z	MIPI RX D-PHY positive data lane1 input
49	MIPI_RX_DA0N	AI	K17	Z	MIPI RX D-PHY negative data lane1 input
50	MIPI_RX_DA1P	AI	J18	Z	MIPI RX D-PHY positive data lane2 input
51	MIPI_RX_DA1N	AI	J17	Z	MIPI RX D-PHY negative data lane2 input
52	MIPI_RX_DA2P	AI	G18	Z	MIPI RX D-PHY positive data lane3 input
53	MIPI_RX_DA2N	AI	G17	Z	MIPI RX D-PHY negative data lane3 input
54	MIPI_RX_DA3P	AI	F18	Z	MIPI RX D-PHY positive data lane4 input
55	MIPI_RX_DA3N	AI	F17	Z	MIPI RX D-PHY negative data lane4 input
LCD Interface (31 pins) (VDD_LCD/VSS)					
56	PLCD0	IOD	P4	LZ	LCD data [0] LGPIO0 | RGB_D0 | CCIR_D0
57	PLCD1	IOD	R2	LZ	LCD data [1] LGPIO1 | RGB_D1 | CCIR_D1
58	PLCD2	IOD	R3	LZ	LCD data [2] LGPIO2 | RGB_D2 | CCIR_D2
59	PLCD3	IOD	T1	LZ	LCD data [3] LGPIO3 | RGB_D3 | CCIR_D3
60	PLCD4	IOD	T2	LZ	LCD data [4] LGPIO4 | RGB_D4 | CCIR_D4
61	PLCD5	IOD	T3	LZ	LCD data [5] LGPIO5 | RGB_D5 | CCIR_D5
62	PLCD6	IOD	T4	LZ	LCD data [6] LGPIO6 | RGB_D6 | CCIR_D6
63	PLCD7	IOD	U1	LZ	LCD data [7] LGPIO7 | RGB_D7 | CCIR_D7
64	PLCD8	IOD	T5	LZ	LCD data [8] LGPIO8 | RGB_D8 | CCIR_D8
65	PLCD9	IOD	V1	LZ	LCD data [9] LGPIO9 | RGB_D9 | CCIR_D9
66	PLCD10	IOD	V2	LZ	LCD data [10] LGPIO10 | RGB_D10 | CCIR_D10
67	PLCD11	IOD	U2	LZ	LCD data [11] LGPIO11 | RGB_D11 | CCIR_D11
68	PLCD12	IOD	R5	LZ	LCD data [12] LGPIO12 | RGB_D12 | CCIR_D12

Pin Num	Pin Name	Type	Loc.	Rst# state	Description
69	PLCD13	IOD	U3	LZ	LCD data [13] LGPIO13 | RGB_D13 | CCIR_D13
70	PLCD14	IOD	V3	LZ	LCD data [14] LGPIO14 | RGB_D14 | CCIR_D14
71	PLCD15	IOD	U4	LZ	LCD data [15] LGPIO15 | RGB_D15 | CCIR_D15
72	PLCD16	IOD	V4	LZ	LCD data [16] LGPIO16 | RGB_D16 | HGPO5 (2) | IGBT (2)
73	PLCD17	IOD	U5	LZ	LCD data [17] LGPIO17 | RGB_D17 | HGPO6 (2) | PWM1 (2)
74	PLCD18	IOD	R6	LZ	LCD data [18] RGB_D18
75	PLCD19	IOD	V5	LZ	LCD data [19] RGB_D19
76	PLCD20	IOD	T6	LZ	LCD data [20] RGB_D20
77	PLCD21	IOD	U7	LZ	LCD data [21] RGB_D21
78	PLCD22	IOD	U6	LZ	LCD data [22] RGB_D22
79	PLCD23	IOD	T7	LZ	LCD data [23] RGB_D23
80	PLCD_WE_	IO	V7	L	LCD write signal
81	PLCD_A0	IO	V6	H	LCD command or data selection
82	PLCD_RD_	IO	R7	L	LCD read signal
83	PLCD1_CS_	IO	U8	H	LCD1 enable
84	PLCD2_CS_	IO	T8	H	LCD2 enable
85	PLCD_FLM	IOD	T9	LZ	LCD_FLM
86	PLCD_GPIO	IOD	N7	LZ	LCD GPIO
PWRC Interface (4 pins) (PSAVDD/PSAGND)					
87	POR_N	AO	K7	L	Power-on reset output
88	PWRC_ON	AI	L6	Z	Power-on/off trigger signal, active-high level signal
89	PWR_EN	AO	M4	L	External DC-DC turn on control, active-high level signal
90	PWSENSE	AI	L7	Z	Battery low level input signal
Digital HDMI Interface (2 pins) (VDD_HDMI/VSS)					
91	PHDMI_SDA	IOU	N3	HZ	Digital HDMI I2C Slave Data
92	PHDMI_SCL	IOU	P3	HZ	Digital HDMI I2C Slave Clock
PMIC Interface (4 pins) (VDD_I2S/VSS) (Note1)					
93	PPMIC_SCL	IOU	L12	HZ	PMIC I2C Interface SCK
94	PPMIC_SDA	IOU	K12	HZ	PMIC I2C Interface SDA
95	PPMIC_INT	IOD	K11	LZ	PMIC Interrupt
96	PPMIC_CLK	IOD	J12	LZ	PMIC RTC Clock
BGPIO interface (22 pins) (VDD_BGPIO/VSS)					
97	BGPIO0	IOD	D1	LZ	BGPIO0 SDMMC CLK (1) | MS_SCLK | HGPO0 (1) | IGBT (3)
98	BGPIO1	IOD	C2	LZ	BGPIO1 SDMMC CMD (1) | MS_BS | HGPO1 (1) | PWM0 (2)
99	BGPIO2	IOD	C1	LZ	BGPIO2 SDMMC DATA0 (1) | HGPO2 (1) | PWM1 (3)
100	BGPIO3	IOD	F4	LZ	BGPIO3 SDMMC DATA1 (1) | MS_DATA1 | HGPO3 (1)
101	BGPIO4	IOD	E1	LZ	BGPIO4 SDMMC DATA2 (1) | MS_DATA2 | HGPO4 (1)
102	BGPIO5	IOD	F2	LZ	BGPIO5

Pin Num	Pin Name	Type	Loc.	Rst# state	Description
					SDMMC DATA3 (1) | MS_DATA3 | HGPO5 (1)
103	BGPIO6	IOD	E2	LZ	BGPIO6 SDMMC DATA4 (1) | SDMMC DATA0 (3) | SPI_CK (2) | PCM_CK (2) | HGPO6 (1)
104	BGPIO7	IOD	F3	LZ	BGPIO7 SDMMC DATA5 (1) | SDMMC_DATA1 (3) | SPI_CS_ (2) | PCM_SYNC (2) | HGPO7 (1) | UART_RX (2) | IGBT (4)
105	BGPIO8	IOD	D2	HZ	BGPIO8 SDMMC DATA6 (1) | SDMMC_DATA2 (3) | SPI_DO (2) | PCM_DO (2) | HGPO8 (1) | UART_TX (2) | PWM0 (3)
106	BGPIO9	IOD	F1	HZ	BGPIO9 SDMMC DATA7 (1) | SDMMC_DATA3 (3) | SPI_DI (2) | PCM_DI (2) | HGPO9 (1) | PWM1 (4)
107	BGPIO10	IOD	G1	HZ	BGPIO10 ARM_TCK | HGPO10 (1) | I2CM_SCK (1) | UART_RX (3)
108	BGPIO11	IOD	C3	HZ	BGPIO11 ARM_TMS | HGPO11 (1) | I2CM_SDA (1) | UART_TX (3)
109	BGPIO12	IOD	G2	HZ	BGPIO12 ARM_TDI | HGPO12 (1) | SPI_CK (3) | PCM_CK (3)
110	BGPIO13	IOD	C4	HZ	BGPIO13 ARM_TRST | HGPO13 (1) | SPI_CS_ (3) | PCM_SYNC (3)
111	BGPIO14	IOD	D3	HZ	BGPIO14 ARM_TDO | SDMMC_CLK (3) | HGPO14 (1) | PCM_DO (3) | SPI_DO (3)
112	BGPIO15	IOD	H1	HZ	BGPIO15 ARM_RTCK | SDMMC_CMD (3) | HGPO15 (1) | PCM_DI(3) | SPI_DI (3)
113	BGPIO16	IOD	G3	HZ	BGPIO16 I2CM_SCK (2)
114	BGPIO17	IOD	G4	HZ	BGPIO17 I2CM_SDA (2)
115	BGPIO18	IOD	H3	HZ	BGPIO18 SPI_CLK
116	BGPIO19	IOD	H2	LZ	BGPIO19 SPI_CS_N
117	BGPIO20	IOD	H7	LZ	BGPIO20 SPI_DO
118	BGPIO21	IOD	J7	LZ	BGPIO21 SPI_DI
CGPIO interface (32 pins) (VDD_CGPIO/VSS)					
119	CGPIO0	IOU	C14	HZ	CGPIO0 SPI_CK | PNF_RB#
120	CGPIO1	IOU	A15	LZ	CGPIO1 SPI_CS_ | PNF_RE#
121	CGPIO2	IOD	C13	HZ	CGPIO2 SPI_DO | PNF_CE#
122	CGPIO3	IOD	B14	LZ	CGPIO3 SPI_DI | PNF_CLE
123	CGPIO4	IOD	B13	LZ	CGPIO4 I2CM_SCK (1) | PNF_ALE
124	CGPIO5	IOD	B15	LZ	CGPIO5 I2CM_SDA (1) | PNF_WE#
125	CGPIO6	IOD	D14	LZ	CGPIO6 I2CM_SCK (2) | PNF_WP#
126	CGPIO7	IOD	A14	LZ	CGPIO7

Pin Num	Pin Name	Type	Loc.	Rst# state	Description
					I2CM_SDA (2) | PNF_D0
127	CGPIO8	IOD	C8	LZ	CGPIO8 PWM | PNF_D1
128	CGPIO9	IOD	A13	LZ	CGPIO9 SDMMC CLK (2) | PNF_D2 | SPI_CK (4) | PCM_CK (4)
129	CGPIO10	IOD	D6	LZ	CGPIO10 SDMMC CMD (2) | PNF_D3 | SPI_CS_ (4) | PCM_SYNC (4)
130	CGPIO11	IOD	A10	LZ	CGPIO11 SDMMC DATA0 (2) | PNF_D4 | SPI_DO (4) | PCM_DO (4)
131	CGPIO12	IOD	C7	LZ	CGPIO12 SDMMC DATA1 (2) | PNF_D5 | SPI_DI (4) | PCM_DI (4)
132	CGPIO13	IOD	A9	LZ	CGPIO13 SDMMC DATA2 (2) | PNF_D6 | I2CM_SCK (2) | UART_RX (4)
133	CGPIO14	IOD	A8	LZ	CGPIO14 SDMMC DATA3 (2) | PNF_D7 | I2CM_SDA (2) | UART_TX (4)
134	CGPIO15	IOD	A7	LZ	CGPIO15 UART_TX (1)
135	CGPIO16	IOD	C11	LZ	CGPIO16 UART_RX (1)
136	CGPIO17	IOD	A6	LZ	CGPIO17 UART_CTS (1)
137	CGPIO18	IOD	C12	LZ	CGPIO18 UART_RTS (1)
138	CGPIO19	IOD	B9	LZ	CGPIO19 UART_TX (2)
139	CGPIO20	IOD	C6	LZ	CGPIO20 UART_RX (2)
140	CGPIO21	IOD	B8	LZ	CGPIO21 SDMMC CLK (3)
141	CGPIO22	IOD	D9	LZ	CGPIO22 SDMMC CMD (3)
142	CGPIO23	IOD	B7	LZ	CGPIO23 SDMMC DATA0 (3)
143	CGPIO24	IOD	B6	LZ	CGPIO24 SDMMC DATA1 (3)
144	CGPIO25	IOD	C9	LZ	CGPIO25 SDMMC DATA2 (3)
145	CGPIO26	IOD	C10	LZ	CGPIO26 SDMMC DATA3 (3)
146	CGPIO27	IOD	A11	LZ	CGPIO27 SIF_CK
147	CGPIO28	IOD	D11	LZ	CGPIO28 SIF_CS_
148	CGPIO29	IOD	B11	LZ	CGPIO29 SIF_DO
149	CGPIO30	IOD	D12	LZ	CGPIO30 SIF_DI
150	CGPIO31	IOD	B10	LZ	CGPIO31 Chip boot mode selection (0: MoviNand, 1: USB update)
HDMI (10 pins)					**(VDD12_ANA, VDD33_ANA, VDD12_TMDS/VSS_TMDS)**
151	HDMI_CH0_P	O	L2	Z	HDMI differential CH0 positive output
152	HDMI_CH0_N	O	L1	Z	HDMI differential CH0 negative output
153	HDMI_CH1_P	O	M2	Z	HDMI differential CH1 positive output
154	HDMI_CH1_N	O	M1	Z	HDMI differential CH1 negative output
155	HDMI_CH2_P	O	N2	Z	HDMI differential CH2 positive output

Pin Num	Pin Name	Type	Loc.	Rst# state	Description
156	HDMI_CH2_N	O	N1	Z	HDMI differential CH2 negative output
157	HDMI_CLK_P	O	K2	Z	HDMI differential CLK positive output
158	HDMI_CLK_N	O	K1	Z	HDMI differential CLK negative output
159	HDMI_REXT	O	J4	Z	Connect to external 12Kohm resistor
160	HDMI_HPD	I/O	J3	Z	HDMI hot plug detect
Audio & Voice (14 pins) (AVDD_AUDIO, AVDD_MIC/AVSS_AUDIO)					
161	PAU_LIP	AI	R17	Z	**Left channel ADC positive microphone input**
162	PAU_LIN	AI	R18	Z	**Left channel ADC negative microphone input**
163	PAU_RIP	AI	P17	Z	Right channel ADC positive microphone input
164	PAU_RIN	AI	P18	Z	Right channel ADC negative microphone input
165	PAUXL	AI	P16	Z	Left channel single-ended auxiliary input
166	PAUXR	AI	R15	Z	Right channel single-ended auxiliary input
167	MICBIAS	AO	N16	Z	Microphone bias output
168	PAU_ROUT2N	AO	V18	Z	Right channel negitive line output
169	PAU_ROUT2P	AO	V17	Z	Right channel positive line output
170	PAU_LOUT2N	AO	T18	Z	Left channel negitive line output
171	PAU_LOUT2P	AO	T17	Z	Left channel positive line output
172	PAU_LOUT	AO	U18	Z	Left channel headphone output (6mW output power into 16Ω//200pF load) (also used for external AMP usage)
173	PAU_ROUT	AO	U17	Z	Right channel headphone output (6mW output power into 16Ω//200pF load) (also used for external AMP usage)
174	PVREF	AO	R16	Z	Band-gap reference voltage (A 10uF capacitor is recommended to connect to this pin)
TV out (2 pins) (AVDD_TV/AVSS_TV)					
175	TVOUTC	AO	R8	Z	TV composite current out
176	TV_FSRES	AI	R9	Z	TV_FSRES out
USB (5 pins) (VDD_USB_3V3, VDD_USB_1V2/VSS_USB)					
177	PUSBDP	IO	M17	Z	USB D+
178	PUSBDN	IO	M18	Z	USB D-
179	RREF	AI	M16	Z	USB2.0 external reference resistor connection
180	USB_ID	I	M15	Z	USB bus ID pin
181	VBUS	IO	M13	Z	USB bus power
Power and Ground (94 pins)					
182-199	VDD_CORE	P	D5/F8 F9/G8 G9/G11 G13/H6 J6/J13 K6/K13 K15/L13 M8/N8 N11/N12		Core power (16 pins)
200	VDD_SEN	P	E15		IO PAD power for SENSOR
201	VDD_LCD	P	M6		IO PAD power for LCD
202	VDD_I2S	P	R14		IO PAD power for I2S
203	VDD_HIF	P	R12		IO PAD power for HOST_IF (or GPIO GroupA)
204	VDD_BGPIO	P	H4		IO PAD power for GPIO group B
205	VDD_CGPIO	P	D13		IO PAD power for GPIO group C
206-211	VDD_DRAM_1	P	A5/A12 B5/F7 F11/F12		IO PAD power for DRAM (6 pins)
212-214	VDD_DRAM_2	P	T10/U9		IO PAD power for DRAM (6 pins)

Pin Num	Pin Name	Type	Loc.	Rst# state	Description
			V9		
215	VDD_DDR_1V2	P	D8		DDR DLL power
216	VSS_DDR_1V2	G	D7		DDR DLL ground
217	VDD_CLK	P	B3		IO PAD power for PMCLK
218	VSS_CLK	G	B4		IO PAD ground for crystal pad
219-220	VDD_MIPI_RX	P	J15/J16		MIPI RX power
221	VSS_MIPI_RX	G	K16		MIPI RX Ground
222	VDD_USB_1V2	P	L15		Core power for USB
223	VDD_USB_3V3	P	N18		IO PAD power for USB
224	VSS_USB	G	L17		IO PAD ground for USB
225	AVDD_AUDIO	P	P15		Audio Codec power
226	AVSS_AUDIO	G	T16		Audio Codec ground
227	AVDD_MIC	P	N15		Audio Codec power
228	AVDD_PLL	P	B1		DPLL power
229	AVSS_PLL	G	A1		DPLL ground
230	AVDD_PLL_SS	P	B2		Spread Spectrum PLL Power
231	AVSS_PLL_SS	G	A2		Spread Spectrum PLL Ground
232	AVDD_TV	P	N10		Analog VDD for video DAC
233	AVSS_TV	G	N9		Analog VSS for video DAC
234	VDD12_ANA	P	K3		HDMI PLL power
235	VDD12_TMDS	P	L3		HDMI transceiver power
237-238	VSS_TMDS	G	J2/L4		HDMI transceiver ground
239	VDD_HDMI	P	R1		Digital HDMI Interface Power
240	PSAVDD	P	N4		PWRC I/O power
241	PSAGND	G	P2		PWRC I/O ground
242-274	VSS	G	B12/B16 C5/D4 D10/D15 E3/E4 E18/G6 H9/H10 H12/H13 H16/J1 J8/J9 J10/J11 K8/K9 K10/L9 L16/L18 M3/N17 P1/R4 R10/V8 V14		Common ground (33 pins)
Test mode (2 pins) (VDD_CGPIO/VSS)					
275	PTEST_EN	ID	G10	LZ	For test mode only
276	PSCAN_EN	ID	F10	LZ	For test mode only

To better utilize the IO pins, most pins are multi-function pins. *The actual function and usage of IO pins are configured by software.* For details of multi-function pin description, please contact AIT sales for chip application note.

2.2 Pin Diagrams

AIT8427
STFBGA276 pin diagram (DV)
(top view)
Body = 13x13mm
Ball array = 18 x 18mm
Ball pitch = 0.65 mm (ball size=0.3 mm)

A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15	A16	A17	A18
AVSS_PLL	AVSS_PLL_SS	XSCI_0	XSCO_0	VDD_DRAM_1	CGPIO17	CGPIO15	CGPIO14	CGPIO13	CGPIO11	CGPIO27	VDD_DRAM_1	CGPIO9	CGPIO7	CGPIO1	PS_GPIO	PSCK	PPXL_CLK
B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	B11	B12	B13	B14	B15	B16	B17	B18
AVDD_PLL	AVDD_PLL_SS	VDD_CLK_0	VSS_CLK_0	VDD_DRAM_1	CGPIO24	CGPIO23	CGPIO21	CGPIO19	CGPIO31	CGPIO29	VSS	CGPIO4	CGPIO3	CGPIO5	VSS	PSEN	PSDA
C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16	C17	C18
BGPIO2	BGPIO1	BGPIO11	BGPIO13	VSS	CGPIO20	CGPIO12	CGPIO8	CGPIO25	CGPIO26	CGPIO16	CGPIO18	CGPIO2	CGPIO0	PS_RST_	PD9	PD6	PD8
D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13	D14	D15	D16	D17	D18
BGPIO0	BGPIO8	BGPIO14	VSS	VDD_CORE	CGPIO10	VSS_DDR_1V2	VDD_DDR_1V2	CGPIO22	VSS	CGPIO28	CGPIO30	VDD_GPIO_C	CGPIO6	VSS	PD2	PD5	PDCLK
E1	E2	E3	E4	E5	E6	E7	E8	E9	E10	E11	E12	E13	E14	E15	E16	E17	E18
BGPIO4	BGPIO6	VSS	VSS											VDD_SEN	PD4	PD0	VSS
F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15	F16	F17	F18
BGPIO9	BGPIO5	BGPIO7	BGPIO3			VDD_DRAM_1	VDD_CORE	VDD_CORE	PSCAN_EN	VDD_DRAM_1	VDD_DRAM_1			PD3	PD7	RXDA3N	RXDA3P
G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	G16	G17	G18
BGPIO10	BGPIO12	BGPIO16	BGPIO17		VSS		VDD_CORE	VDD_CORE	PTEST_EN	VDD_CORE		VDD_CORE		PD1	PHSYNC	RXDA2N	RXDA2P
H1	H2	H3	H4	H5	H6	H7	H8	H9	H10	H11	H12	H13	H14	H15	H16	H17	H18
BGPIO15	BGPIO19	BGPIO18	VDD_GPIO_B		VDD_CORE	BGPIO20		VSS	VSS		VSS	VSS		PVSYNC	VSS	RXCKN	RXCKP
J1	J2	J3	J4	J5	J6	J7	J8	J9	J10	J11	J12	J13	J14	J15	J16	J17	J18
VSS	VSS_TMDS	HPD	REXT		VDD_CORE	BGPIO21	VSS	VSS	VSS	VSS	PPMIC_CLK	VDD_CORE		VDD_MIPI_RX	VDD_MIPI_RX	RXDA1N	RXDA1P
K1	K2	K3	K4	K5	K6	K7	K8	K9	K10	K11	K12	K13	K14	K15	K16	K17	K18
OUTN_CK	OUTP_CK	VDD12_ANA	VDD33_ANA		VDD_CORE	POR_N	VSS	VSS	VSS	PPMIC_INT	PPMIC_SDA	VDD_CORE		VDD_CORE	VSS_MIPI_RX	RXDA0N	RXDA0P
L1	L2	L3	L4	L5	L6	L7	L8	L9	L10	L11	L12	L13	L14	L15	L16	L17	L18
OUTN_CH0	OUTP_CH0	VDD12_TMDS	VSS_TMDS		PWRC_ON	PW SENSE		VSS	PHINT		PPMIC_SCL	VDD_CORE		VDD_USB_1V2	VSS	VSS_USB	VSS
M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12	M13	M14	M15	M16	M17	M18
OUTN_CH1	OUTP_CH1	VSS	PWR_EN		VDD_LCD		VDD_CORE	PHI2C_SDA	PHI2C_SCL	PHWAIT_		VBUS		USB_ID	RREF	PUSBDP	PUSBDN
N1	N2	N3	N4	N5	N6	N7	N8	N9	N10	N11	N12	N13	N14	N15	N16	N17	N18
OUTN_CH2	OUTP_CH2	PHDMI_SDA	PSAVDD			PLCD_GPIO	VDD_CORE	AVSS_TV	AVDD_TV	VDD_CORE	VDD_CORE			AVDD_MIC	MICBIAS	VSS	VDD_USB_3V3
P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	P14	P15	P16	P17	P18
VSS	PSAGND	PHDMI_SCL	PLCD0											AVDD_AUDIO	PAUXL	PAU_RIP	PAU_RIN
R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12	R13	R14	R15	R16	R17	R18
VDD_HDMI	PLCD1	PLCD2	VSS	PLCD12	PLCD18	PLCD_RD	TVOUTC	TV_FSRES	VSS	PRST_	VDD_HIF	PHLCD_A0	VDD_I2S	PAUXR	PVREF	PAU_LIP	PAU_LIN
T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13	T14	T15	T16	T17	T18
PLCD3	PLCD4	PLCD5	PLCD6	PLCD8	PLCD20	PLCD23	PLCD2_CS	PLCD_FLM	VDD_DRAM_2	PHD7	PHD2	PHLCD_BY	PI2S_SCK	PHI2S_SDO	AVSS_AUDIO	PAU_LOUT2P	PAU_LOUT2N
U1	U2	U3	U4	U5	U6	U7	U8	U9	U10	U11	U12	U13	U14	U15	U16	U17	U18
PLCD7	PLCD11	PLCD13	PLCD15	PLCD17	PLCD22	PLCD21	PLCD1_CS	VDD_DRAM_2	PHCS_	PHRD_	PHD1	PHD5	PHD6	PI2S_WS	PI2S_SDO	PAU_ROUT	PAU_LOUT
V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13	V14	V15	V16	V17	V18
PLCD9	PLCD10	PLCD14	PLCD16	PLCD19	PLCD_A0	PLCD_WE	VSS	VDD_DRAM_2	PHD3	PHWE_	PHD0	PHD4	VSS	PI2S_MCLK	PI2S_SDI	PAU_ROUT2P	PAU_ROUT2N

A1	A2	A3	A4	A5	A6	A7	A8	A9
AVSS_PLL	AVSS_PLL_SS	XSCI_0	XSCO_0	VDD_DRAM_1	CGPIO17	CGPIO15	CGPIO14	CGPIO13
B1	B2	B3	B4	B5	B6	B7	B8	B9
AVDD_PLL	AVDD_PLL_SS	VDD_CLK_0	VSS_CLK_0	VDD_DRAM_1	CGPIO24	CGPIO23	CGPIO21	CGPIO19
C1	C2	C3	C4	C5	C6	C7	C8	C9
BGPIO2	BGPIO1	BGPIO11	BGPIO13	VSS	CGPIO20	CGPIO12	CGPIO8	CGPIO25
D1	D2	D3	D4	D5	D6	D7	D8	D9
BGPIO0	BGPIO8	BGPIO14	VSS	VDD_CORE	CGPIO10	VSS_DDR_1V2	VDD_DDR_1V2	CGPIO22
E1	E2	E3	E4	E5	E6	E7	E8	E9
BGPIO4	BGPIO6	VSS	VSS					
F1	F2	F3	F4	F5	F6	F7	F8	F9
BGPIO9	BGPIO5	BGPIO7	BGPIO3			VDD_DRAM_1	VDD_CORE	VDD_CORE
G1	G2	G3	G4	G5	G6	G7	G8	G9
BGPIO10	BGPIO12	BGPIO16	BGPIO17		VSS		VDD_CORE	VDD_CORE
H1	H2	H3	H4	H5	H6	H7	H8	H9
BGPIO15	BGPIO19	BGPIO18	VDD_GPIO_B		VDD_CORE	BGPIO20		VSS
J1	J2	J3	J4	J5	J6	J7	J8	J9
VSS	VSS_TMDS	HPD	REXT		VDD_CORE	BGPIO21	VSS	VSS

Upper Left Corner (A1 ~ A9, J1 ~ J9)

A10	A11	A12	A13	A14	A15	A16	A17	A18
CGPIO11	CGPIO27	VDD_ DRAM_1	CGPIO9	CGPIO7	CGPIO1	PS_GPIO	PSCK	PPXL_ CLK
B10	**B11**	**B12**	**B13**	**B14**	**B15**	**B16**	**B17**	**B18**
CGPIO31	CGPIO29	VSS	CGPIO4	CGPIO3	CGPIO5	VSS	PSEN	PSDA
C10	**C11**	**C12**	**C13**	**C14**	**C15**	**C16**	**C17**	**C18**
CGPIO26	CGPIO16	CGPIO18	CGPIO2	CGPIO0	PS_RST_	PD9	PD6	PD8
D10	**D11**	**D12**	**D13**	**D14**	**D15**	**D16**	**D17**	**D18**
VSS	CGPIO28	CGPIO30	VDD_ GPIO_C	CGPIO6	VSS	PD2	PD5	PDCLK
E10	**E11**	**E12**	**E13**	**E14**	**E15**	**E16**	**E17**	**E18**
					VDD_SE N	PD4	PD0	VSS
F10	**F11**	**F12**	**F13**	**F14**	**F15**	**F16**	**F17**	**F18**
PSCAN_ EN	VDD_ DRAM_1	VDD_ DRAM_1			PD3	PD7	RXDA3N	RXDA3P
G10	**G11**	**G12**	**G13**	**G14**	**G15**	**G16**	**G17**	**G18**
PTEST_ EN	VDD_ CORE		VDD_ CORE		PD1	PHSYNC	RXDA2N	RXDA2P
H10	**H11**	**H12**	**H13**	**H14**	**H15**	**H16**	**H17**	**H18**
VSS		VSS	VSS		PVSYNC	VSS	RXCKN	RXCKP
J10	**J11**	**J12**	**J13**	**J14**	**J15**	**J16**	**J17**	**J18**
VSS	VSS	PPMIC_ CLK	VDD_ CORE		VDD_ MIPI_RX	VDD_ MIPI_RX	RXDA1N	RXDA1P

Upper Right Corner (A10 ~ A18, J10 ~ L18)

K1	K2	K3	K4	K5	K6	K7	K8	K9
OUTN_CK	OUTP_CK	VDD12_ANA	VDD33_ANA		VDD_CORE	POR_N	VSS	VSS
L1	**L2**	**L3**	**L4**	**L5**	**L6**	**L7**	**L8**	**L9**
OUTN_CH0	OUTP_CH0	VDD12_TMDS	VSS_TMDS		PWRC_ON	PW SENSE		VSS
M1	**M2**	**M3**	**M4**	**M5**	**M6**	**M7**	**M8**	**M9**
OUTN_CH1	OUTP_CH1	VSS	PWR_EN		VDD_LCD		VDD_CORE	PHI2C_SDA
N1	**N2**	**N3**	**N4**	**N5**	**N6**	**N7**	**N8**	**N9**
OUTN_CH2	OUTP_CH2	PHDMI_SDA	PSAVDD			PLCD_GPIO	VDD_CORE	AVSS_TV
P1	**P2**	**P3**	**P4**	**P5**	**P6**	**P7**	**P8**	**P9**
VSS	PSAGND	PHDMI_SCL	PLCD0					
R1	**R2**	**R3**	**R4**	**R5**	**R6**	**R7**	**R8**	**R9**
VDD_HDMI	PLCD1	PLCD2	VSS	PLCD12	PLCD18	PLCD_RD	TVOUTC	TV_FSRES
T1	**T2**	**T3**	**T4**	**T5**	**T6**	**T7**	**T8**	**T9**
PLCD3	PLCD4	PLCD5	PLCD6	PLCD8	PLCD20	PLCD23	PLCD2_CS	PLCD_FLM
U1	**U2**	**U3**	**U4**	**U5**	**U6**	**U7**	**U8**	**U9**
PLCD7	PLCD11	PLCD13	PLCD15	PLCD17	PLCD22	PLCD21	PLCD1_CS	VDD_DRAM_2
V1	**V2**	**V3**	**V4**	**V5**	**V6**	**V7**	**V8**	**V9**
PLCD9	PLCD10	PLCD14	PLCD16	PLCD19	PLCD_A0	PLCD_WE	VSS	VDD_DRAM_2

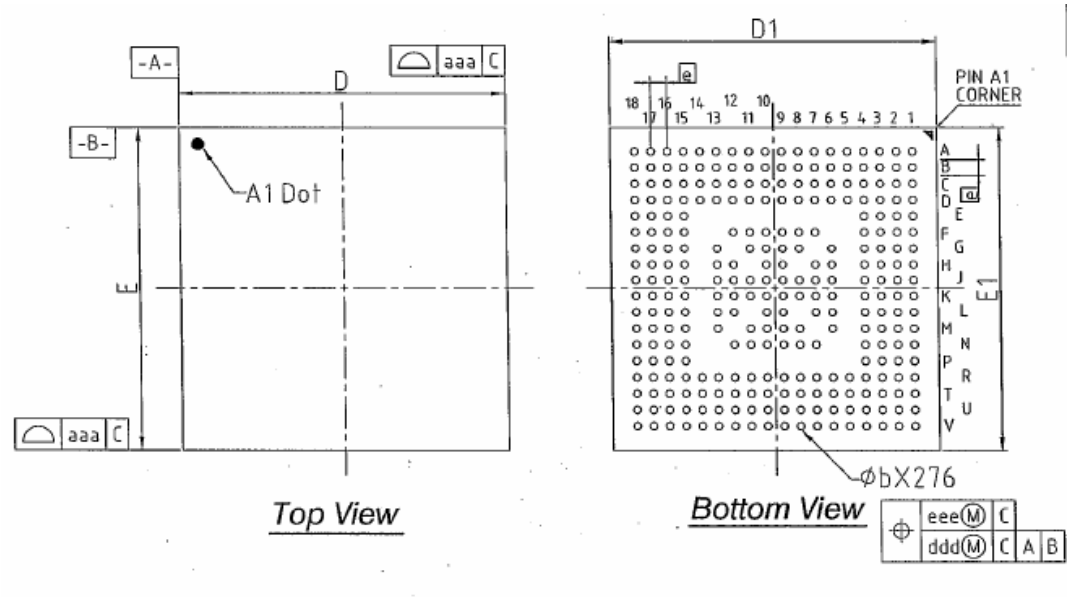
Lower Left Corner (K1 ~ K9, V1 ~ V9)

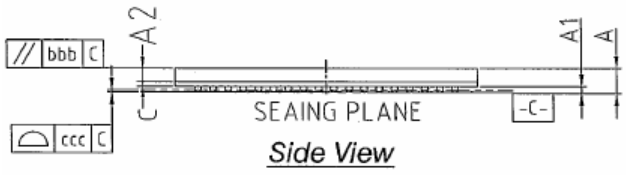
K10	K11	K12	K13	K14	K15	K16	K17	K18
VSS	PPMIC_ INT	PPMIC_ SDA	VDD_ CORE		VDD_ CORE	VSS_ MIPI_RX	RXDA0N	RXDA0P
L10	L11	L12	L13	L14	L15	L16	L17	L18
PHINT		PPMIC_ SCL	VDD_ CORE		VDD_US B_1V2	VSS	VSS _USB	VSS
M10	M11	M12	M13	M14	M15	M16	M17	M18
PHI2C _SCL	PHWAIT_		VBUS		USB_ID	RREF	PUSBDP	PUSBDN
N10	N11	N12	N13	N14	N15	N16	N17	N18
AVDD _TV	VDD_ CORE	VDD_ CORE			AVDD_ MIC	MICBIAS	VSS	VDD_US B_3V3
P10	P11	P12	P13	P14	P15	P16	P17	P18
					AVDD_ AUDIO	PAUXL	PAU_RIP	PAU_RIN

R10	R11	R12	R13	R14	R15	R16	R17	R18
VSS	PRST_	VDD_HIF	PHLCD _A0	VDD_I2S	PAUXR	PVREF	PAU_LIP	PAU_LIN
T10	T11	T12	T13	T14	T15	T16	T17	T18
VDD_ DRAM_2	PHD7	PHD2	PHLCD _BY	PI2S_ SCK	PHI2S_ SDO	AVSS_ AUDIO	PAU_ LOUT2P	PAU_ LOUT2N
U10	U11	U12	U13	U14	U15	U16	U17	U18
PHCS_	PHRD_	PHD1	PHD5	PHD6	PI2S_ WS	PI2S_ SDO	PAU_ ROUT	PAU_ LOUT
V10	V11	V12	V13	V14	V15	V16	V17	V18
PHD3	PHWE_	PHD0	PHD4	VSS	PI2S_ MCLK	PI2S_ SDI	PAU_ ROUT2P	PAU_ ROUT2N

Lower Right Corner (K10 ~ K18, V10 ~ V18)

2.3 Package Dimension





Symbol	Dimension in mm			Dimension in inch		
	Min	Nom	Max	Min	Nom	Max
A	---	---	1.20	---	---	0.047
A1	0.16	0.21	0.26	0.006	0.008	0.010
A2	0.68	0.70	0.71	0.027	0.028	0.028
c	0.16	0.20	0.23	0.006	0.008	0.009
Φb	0.25	0.30	0.35	0.010	0.012	0.014
D	12.90	13.00	13.10	0.508	0.512	0.516
D1	---	11.05	---	---	0.435	---
E	12.90	13.00	13.10	0.508	0.512	0.516
E1	---	11.05	---	---	0.435	---
e	0.65 BSC			0.026 BSC		
aaa	0.15			0.006		
bbb	0.20			0.008		
ccc	0.20			0.008		
ddd	0.15			0.006		
eee	0.08			0.003		
MD/ME	18/18			18/18		

NOTE: 1.CONTROLLING DIMENSION: MILLIMETER.
2.REFERANCE DOCUMENT:JEDEC MO-207 and MO-216

NOTE: CONTROLLING DIMENSION : MILLIMETER

3 Function Description

3.1 Boot-up

AIT8427 provides different options for boot-up. The boot code can be loaded from internal ROM or from the external serial flash device. The LCD data pins PLCD[15:0] are used for boot-strapping configuration pins during reset, and have internal pull-down resistor enabled by default.
The internal ROM stores program code to enable AIT8427 to boot up, activate USB controller and SDIO controller. After reset is complete, the ROM code is first loaded into the TCM and starts execution. The firmware detects if valid program code is available at the external memory device attached on the SDIO interface first. If detected, it continues to load code from the external memory device. In the case when the external memory device is not initialized, the USB interface can be used and download the complete firmware code into the external memory device.

<table>
<tr><td>PLCD15
PLCD14
PLCD13</td><td>BOOT Device selection
0 means pull-down
1 means pull-up
{PLCD15, PLCD14, PLCD13}
000 – Internal ROM
001 – Serial flash
010 –Reserved
011 – NAND flash
others – Reserved</td></tr>
<tr><td>PLCD12
PLCD11
PLCD10
PLCD9
PLCD8</td><td>BOOT Device mode
0 means pull-down
1 means pull-up
These bits have different meaning according to BOOT Device selection.
<table>
<tr><th>BOOT Device selection</th><th>PLCD12</th><th>PLCD11</th><th>PLCD10</th><th>PLCD9</th><th>PLCD8</th></tr>
<tr><td>000</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>001</td><td>byte count 1</td><td>byte count 0</td><td>Address 16</td><td>Pad set bit 1</td><td>Pad set bit 0</td></tr>
<tr><td>011</td><td>Boot mode 1</td><td>Boot mode 0</td><td>-</td><td>-</td><td>-</td></tr>
</table>
1. When serial flash is used,

{PLCD12, PLCD11}
00 – Load 4K bytes boot data
01 – Load 8K bytes boot data
10 – Load 16K bytes boot data
11 – Load 32K bytes boot data

{PLCD10}
0 – 24 bit address
1 – 16 bit address

{PLCD9, PLCD8}
<table>
<tr><th>Pad set</th><th>CLK</th><th>CS_N</th><th>DO</th><th>DI</th></tr>
<tr><td>00</td><td>PSIF_CLK</td><td>PSIF_CS#</td><td>PSIF_DO</td><td>PSIF_DI</td></tr>
<tr><td>01</td><td>PCGPIO0</td><td>PCGPIO1</td><td>PCGPIO2</td><td>PCGPIO3</td></tr>
<tr><td>10</td><td>PBGPIO6</td><td>PBGPIO7</td><td>PBGPIO8</td><td>PBGPIO9</td></tr>
</table>
2. When NAND flash is used,
{PLCD12, PLCD11}
00 – mode 1
01 – mode 2
10 – mode 3
11 – mode 4</td></tr>
</table>

PLCD7	**Reserved**
PLCD6	**Reserved**
PLCD5	**Reserved**
PLCD4	**PLL auto power-on wait time** Pull down (Default): Hardware will wait 2^16 external clock then switch all internal clock to DPLL output clock Pull up: Hardware will wait 2^17 external clock then switch all internal clock to DPLL output clock
PLCD3	**PLL auto power-on mode** Pull down (Default): After power on reset, PLL will automatically power on, and wait for some time (See PLCD4), hardware will automatically switch all clock to PLL clock. Pull up: No automatic operation
PLCD2	**JTAG debug mode** Pull down (Default): GPIO pins will be configured as normal GPIO pin. Pull up: Some GPIO pins will be configured as JTAG debug pin.
PLCD1	**Reserved**
PLCD0	**Reserved**

3.2 ARM926EJ-S

AIT8427 uses ARM926EJ-S as embedded CPU core. ARM9 core adopt “Harvard architecture”, it means the ARM9 core has 2 separated memory interface for instruction and data, that is, can access instruction and data simultaneously. This is different from ARM7 “Von Neumann architecture” which has only 1 memory interface for instruction and instruction.

The other major difference from ARM7 is that ARM9 has two TCM interface: I-TCM and D-TCM. TCM (tightly couple memory) is separated from AHB interface to general frame buffer pool. TCM is connected to a dedicated physical memory (called TCM memory) only used by ARM core. I-TCM is for ARM to access instruction while D-TCM is for data access. If the TCM memory works at the same frequency as ARM core, ARM core can execute 1 instruction 1 cycle from I-TCM memory and get 1 data 1 cycle from D-TCM memory. It can avoid the share frame buffer pool access overhead and get much better performance.

AIT’s ARM9 platform architecture is shown as following diagram:

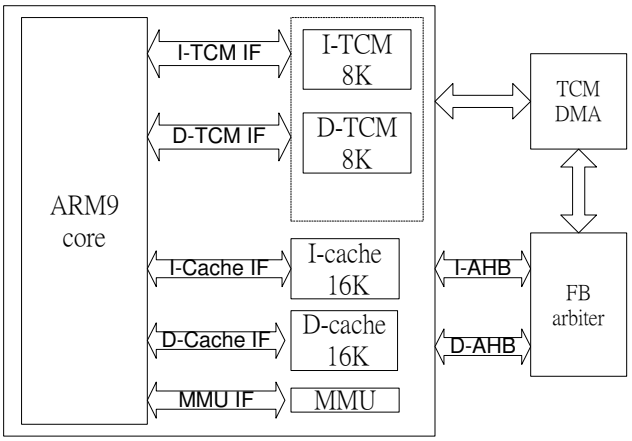


Figure 2. ARM926 Memory Configuration

ARM926EJS also embeds cache and MMU function. AIT8427's cache size: 16K for instruction, 16K for data. All data or code access is through 2 AHB interface. I-AHB is for ARM to access instruction while D-AHB is for data access.

For power saving issue, AIT8427 will turn off unnecessary clock domain including ARM9 during idle routine, CPU will resume after receiving interrupt from others module or timers.

3.3 CPU Peripheral

AIT8427 integrates 3 peripherals: AIC, watchdog and timers. As the following figure shows, all on-chip peripherals are accessible by ARM9 instructions. The peripheral register set is composed of control, mode, data, status and enable/disable registers. Each peripheral has its own mapping addresses.

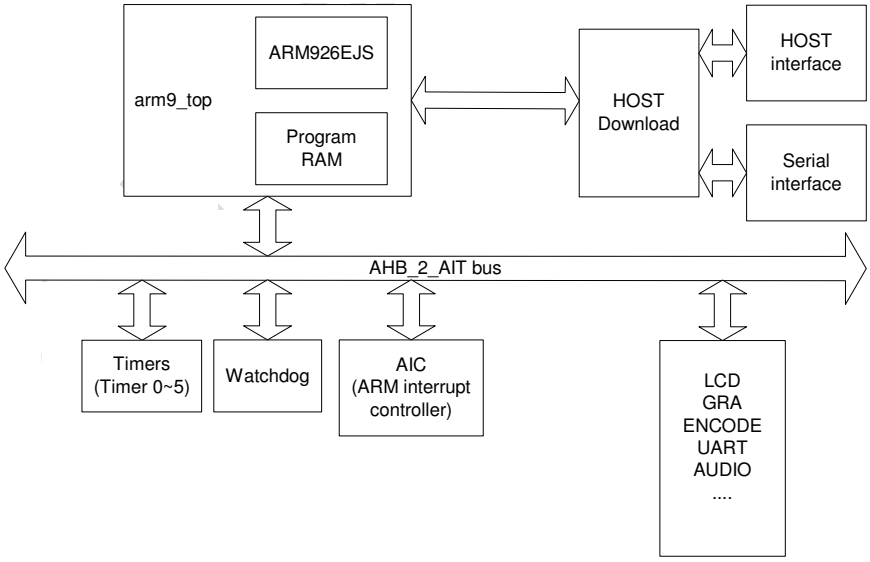


Figure 3. ARM9 and peripheral block diagram

The ARM Interrupt Controller (AIC) controls the internal sources from the other modules (LCD/GRA/JPEG, etc.) to provide an interrupt request(IRQ) and/or fast interrupt request(FIQ) to the ARM926EJS. The programmer can mask the interrupt source inside AIC. The programmer also can assign the 8-level priority for each interrupt source, and AIC provides the priority arbitration among the unmasked interrupt sources. AIC also provides the auto-vectoring feature to reduce the interrupt latency time. AIC is a well-functional ARM interrupt controller. Detail features are list as following:

- 32 interrupt sources
- Programmable interrupt source attribute
  - 4 types of interrupt trigger: Low-level/high-level/positive-edge/negative-edge
  - 0~7 priority level (low to high)
  - FIQ/IRQ request

- Programmable interrupt handler address
- masked/unmasked
- Priority arbitration & preemption for IRQ mode
- Auto Vectoring
- Automatically load the interrupt handler address into IRQ/FIQ vector
- Software trigger interrupt enable for debugging (but only edge-trigger type)

Six identical 32-bit timers are integrated in AIT8427. Each channel can be independently programmed to perform a wide range of functions, including frequency measurement, interval measurement. All six counters can be started simultaneously. Detail features are list as following:

- 6 sets of timers
- 32 bit counter for each timer
- 32 bit compare value can be programmed
- 8 kinds of frequency
  - CPU clock divided by 1, 2, 4, 8, 16, 32, 128, 1024
- 2 types of interrupt
  - Compare equivalence
  - Counter overflow
- Auto restart enable / disable
  - When compare equivalence occurs, counter rolls back to 0 or continue to count
- Enable simultaneously restart all 6 timers

Watchdog is a 16-bit countdown counter and is used to prevent system lock-up if the software becomes trapped in a deadlock. All programmable registers of Watchdog are password protected to prevent the unintentional programming. If software didn't reset watchdog timer before it reach zero, 2 signals may be generated if they are enabled:
--- Reset: If "Watchdog reset enable" is set
--- Interrupt: If "Watchdog interrupt enable" is set
Detail features are list as following:

- Access key protected
- 19 bit countdown counter
  - 5 bits programmable, 14 bits fixed to 1
- 4 sets of clock frequency
  - CPU clock divided by 8, 32, 128, 1024
- types of exception
  - Interrupt
  - Watchdog reset CPU / whole chip

Host download is used to download bootstrap code to TCM. It translates the signals between the host/serial flash interface and CPU interface. For the host interface, SW should enable the host download mode. Then, write the

code to appropriate TCM location. When the download code has been finished, disable the mode and release CPU. For the serial flash interface, the code will be automatically downloaded to TCM. There are 8K byte I-TCM and 8K byte D-TCM.

3.4 LCD Interface

LCD controller is the interface between the display data buffer and the LCD panel. It can support following display devices, parallel panel, serial panel, RGB panel, and TV.

For the parallel and serial panel, LCD controller sends index and command to the LCD panel and refreshes frame data to the panel. It supports CPU interface. The frame data bit width can be 8-bit/9-bit/12-bit/16-bit/18-bit. About the sending index and command, it is used to read/write the LCD panel internal register or the embedded SRAM. For the RGB interface panel, it generates timing information periodically. The pixel data are sent during active region with pixel clock. Some panels also contains serial interface for accessing LCD panel internal registers.

About dealing with the frame data before displaying in the LCD panel, four windows basic setting should be programmed. Besides, the window priority is also important. If further window overlay function is used , such as transparent or semi-transparent function, window overlay operation registers should be set.

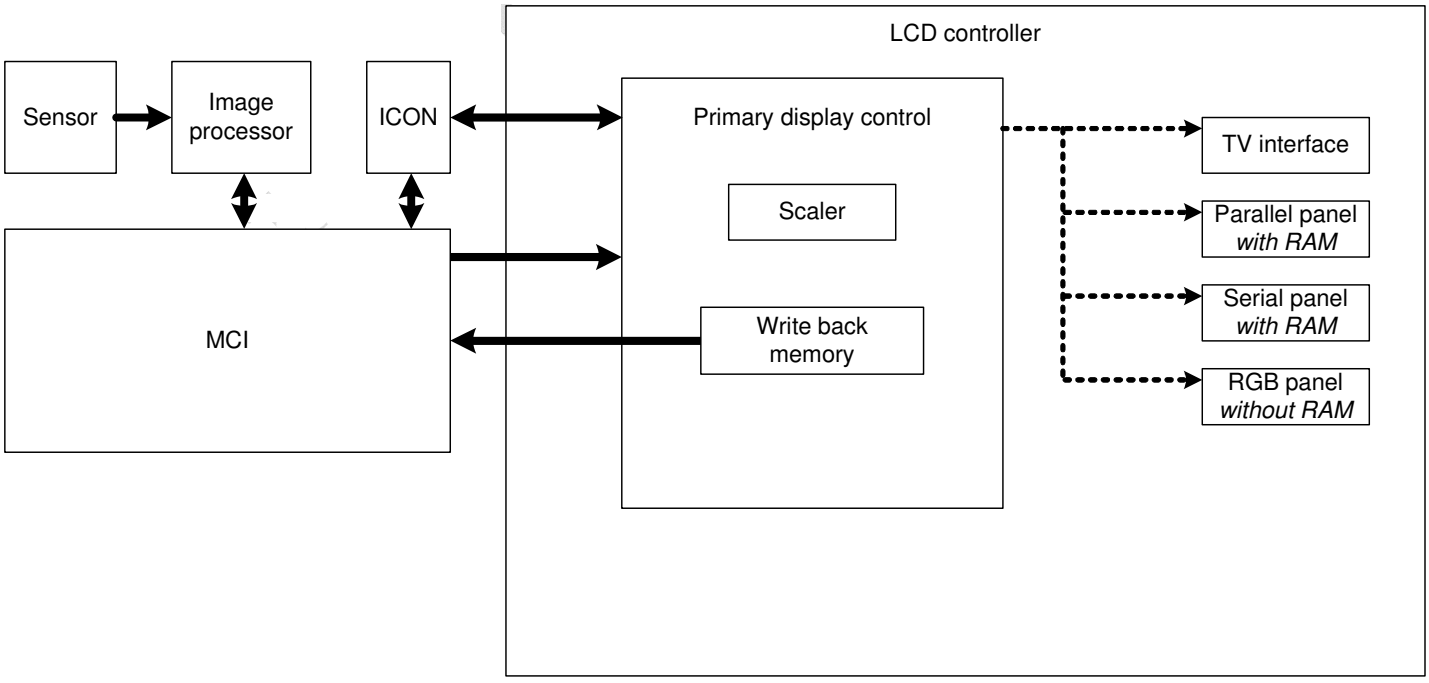


Figure 4. LCD block diagram

Features of LCD interface controller:

1. 4 window layers (main/PIP/overlay/ICON) and one background color are supported

2. Main window supports
   - 4-bit/8-bit color index (256x24 LUT)
   - 16-bit RGB565/ARGB3454/ARGB4444 data format
   - 24-bit RGB888 data format
   - 32-bit ARGB8888 data format
   - YUV422 (CrY1CbY0) data format
   - YUV420 data format
   - YUV420 UV interleaved data format
   - RGB gain and offset tuning for YUV420 data
   - Dither function for YUV420 data
   - Max window size: 8192x8192
   - Window rotate 0/90/180/270 degree and mirror function
3. Overlay window supports
   - 4-bit/8-bit color index (256x24 LUT)
   - 16-bit RGB565/ARGB3454/ARGB4444 data format
   - 24-bit RGB888 data format
   - 32-bit ARGB8888 data format
   - YUV422 (CrY1CbY0) data format
   - YUV420 data format
   - YUV420 UV interleaved data format
   - RGB gain and offset tuning for YUV420 data
   - Dither function for YUV420 data
   - Max window size: 8192x8192
   - Window rotate 0/90/180/270 degree and mirror function
4. PIP window supports
   - 16-bit RGB565/ARGB3454/ARGB4444 data format
   - 24-bit RGB888 data format
   - 32-bit ARGB8888 data format
   - YUV422 (CrY1CbY0) data format
   - YUV420 data format
   - YUV420 UV interleaved data format
   - Scaling function for YUV422/YUV420 (LPF function is only for non-rotate display)
   - Max window size: 8192x8192
   - RGB gain and offset tuning
   - Dither function
5. ICON window supports
   - 16-bit RGB565/ARGB3454/ARGB4444 data format
   - 32-bit ARGB8888 data format
6. Scaler functions:
   - Scaling up with different horizontal and vertical scaling ratios. One of horizontal and vertical scaling ratios can be 1.
   - Scaling down with different horizontal and vertical scaling ratios. One of horizontal and vertical scaling ratios can be 1.
   - Scaled frame can be grabbed from any horizontal or vertical coordinates.
7. 4 window layers (main/PIP/overlay/ICON) transparent and alpha-blending function
8. Support parallel panel with embedded RAM up to 24-bit RGB888 data format
9. Support TV encoder interface
10. Support serial panel with embedded RAM
    - SPI clock (SCL) is generated by dividing 1 to 256 of system clock 260MHZ, and the range is 520KHz ~ 260MHz
    - SPI chip select (CS) polarity is programmable.
    - SPI CS setup time (CS falling edge to SCL falling edge) is programmable.
    - SPI CS hold time (SCL rising edge to CS rising edge) is programmable.

11. Support RGB panel without embedded RAM
- RGB DOT clock is generated by dividing 1 to 256 of system clock 260MHZ, and the range is 520KHz ~ 260MHz.
- Partially display by pixel or by line.
- Programmable vertical and horizontal porches, vertical and horizontal SYNC active width
- Provide HSYNC, VYSNC INT.
- Provide line matched INT, user can set line number in register, then it will INT when the line number matched
- Support serial (delta) RGB panel

12. Support writ back frame data(RGB888/RGB565) to memory
- Write back frame data of primary display to memory while primary display is refreshing.
- Write back frame data to memory alone (do not refresh panel)
- Write back frame data format can be RGB565 or RGB888.
- Write back frame data R , G , B order can be RGB(R in MSB) or BGR(B in MSB).
- Width and height of write back frame are specified by control register.
- Start X and Y position of write back frame are specified by control register.
- Provide write back done interrupt

3.4.1 Panel control

AIT8427 supports parallel panel with internal SRAM and the interface meets CPU interface. 8-bit/9-bit/16-bit/18-bit CPU interface are used for LCD panel index/command mode. For the frame data transfer mode, 8-bit/9-bit/12-bit/16-bit/18-bit are supported. If the frame data is 8-bit format, it can be sent to the LCD panel via RGB332 mode 8-bit data bus. If the frame data is 12-bit format, RGB444 of 8-bit data bus or RGB444 of 12-bit data bus or RGB444 of 16-bit data bus are supported . For the 16-bit frame data, there are two kinds of transfer format can be used, such as RGB565 of 8-bit or 16-bit data bus. For the 18-bit frame data, AIT8427 supports RGB666 of 8-bit data bus or RGB99 of 9-bit data bus or RGB666/RGB2-16/RGB16-2 of 16-bit data bus or RGB666 of 18-bit data bus. For 24-bit data format, 8-bit 888, 18-bit 18-6 and 6-18 mode are supported. There also exist some options to locate the valid data bus in different position of 16 or 18-bit data bus, such as 8-bit data bus can locate in low byte or high byte of 16-bit data bus…etc. The following diagrams show some typical LCD operation timing waveforms.

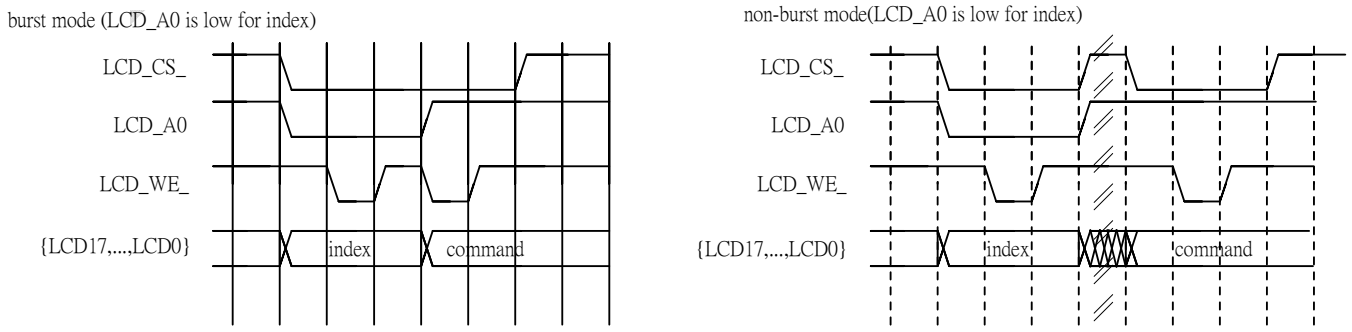


Figure 5. burst and non-burst mode for 80-system

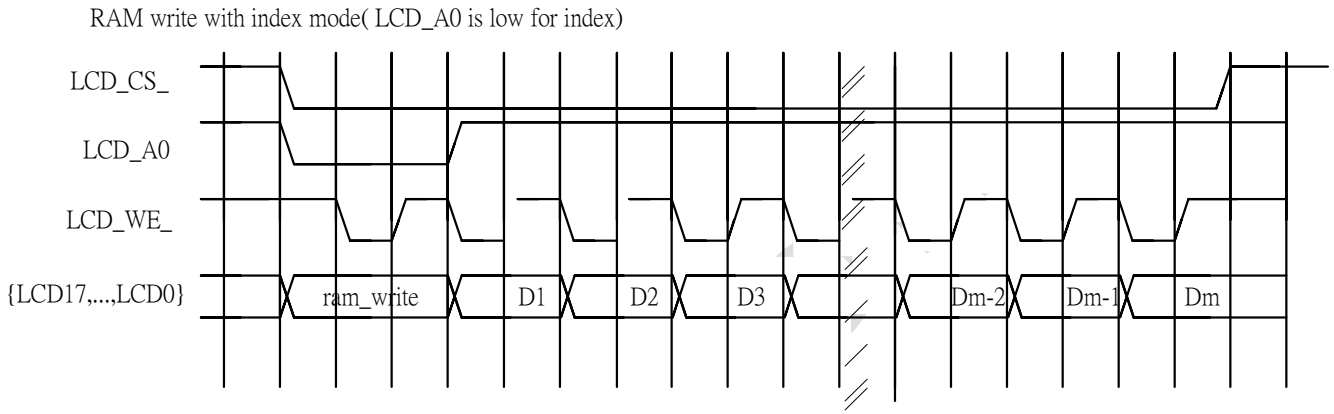


Figure 6. RAM_WR command is sent before frame data for burst mode

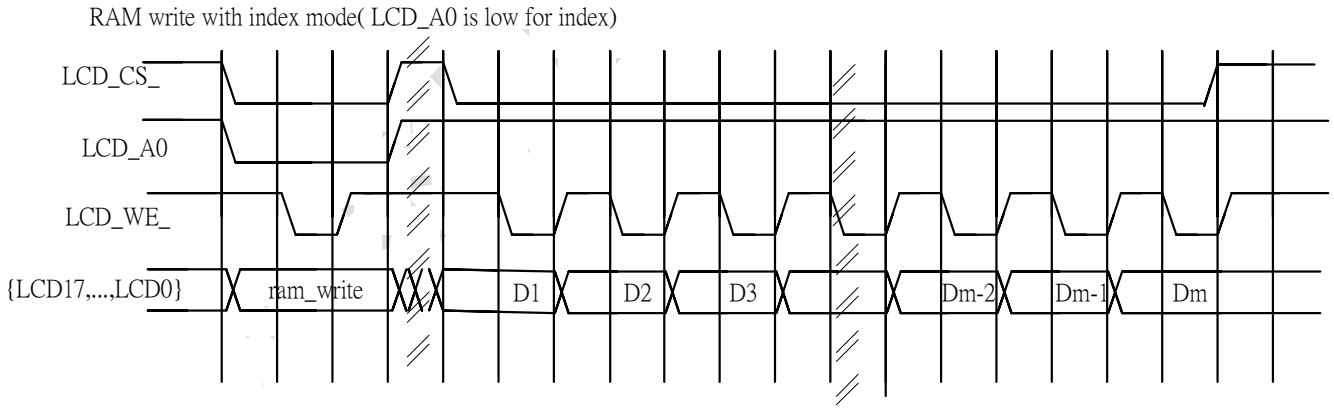


Figure 7. RAM_WR command is sent before frame data for non-burst mode

3.4.2 Window setting

Before enable the LCD controller to send the display data to LCD panel, the window setting should be programmed unless only background color is desired to be shown. Main and overlay window supports 4-bit/8-bit/16-bit data format. Besides, overlay window also supports YUV420/YUV422 data format. PIP window is usually used as the image data layer and it supports RGB 565, RGB888, YUV420 (for video encode/decode) and YUV422 (for preview or jpeg decode) data format. ICON window supports 24-bit data format and source data of ICON engine maybe 8-bit index or RGB565. 0/90/180/270 degree rotation and mirror function are supported for main/pip/overlay windows via pixel offset and row offset of each window registers. ICON window doesn't support rotate or mirror function and if these function is needed, please deal with the source data in the memory before you display.

3.4.3 Grab and scaling function

LCD controller can grab source image when it reading data from memory. User needs to set start offset, row offset, and grab out width, height. Start offset is calculated from grab X-Y position and source image width.

LCD scaler can scale up or scale down input source image in horizontal or vertical direction. But there is one restriction, it can't scale up in one direction and scale down in the other direction. Besides scaling function, LCD scaler can also grab output

image by the user programmed range. The source image can be rotated before to LCD scaler.

3.4.4 Frame data generation

These four windows support transparent and semi-transparent function except game engine. The related registers for main/pip/overlay window are located in LCD controller. For ICON window, you should program the ICON register for the transparent/semi-transparent settings and then ICON engine will output transparent and semi-transparent signals to indicate the current data status. After collecting all the enabled windows transparent and semi-transparent setting, the final data will output from window overlay operation and send to LCD panel. Transparent and semi-transparent function can't be enabled at the same time otherwise the result is not the same as expected!

3.4.5 Frame data refresh

There are two methods for refreshing one frame data, one is firmware refresh and the other is triggered by the sensor data of a frame is stored in the buffer. For the firmware refresh mode, it starts to send the frame data to LCD panel when 0x80007000[1] is set and this bit will be cleared when the refresh is done. For the pre view mode (0x80007000[0] is '1'), the frame data will be sent to LCD panel when one frame is received in the frame buffer. Frame done interrupt status is set for both cases when frame display is done.

3.4.6 FLM and VSYNC function

In order to prevent the flicker of the moving pictures (such as video mode), AIT8427 LCD controller support detect the input signal FLM from LCD panel or output VSYNC signal to LCD panel. For FLM, it is output by LCD panel to indicate that driver IC will update GRAM data to panel. In order to prevent flicker of the frame in the LCD panel, LCD controller will send the frame data to LCD panel by waiting for a while after detecting FLM signal. If FLM mode is enabled, LCD controller will refresh the frame data to LCD panel after detecting every FLM and waiting for the cycle count programmed in the LCD registers. This will be improved in the next project to check if there exists any new frame data needs to be sent to LCD panel after detecting FLM instead of refreshing data to LCD panel all the time. VSYNC is the output signal asserting before we start to send data to LCD panel, in other words, before LCD panel CS_N asserts. VSYNC period is programmed by the cycle count in LCD register.

3.4.7 LCD write-back function

"LCD write-back" can write frame data to memory while LCD panel refreshing. This function is useful when you do some complex image operations (like scaling, image overlaying, transparent function…). You can enable this function to store this image into memory. In next time, if you want to show this image again, then you can read it from memory without doing complex image operations again.

It can write whole frame or partial frame to memory. For the partial frame write back, it needs specifying start X and Y position and writes back width and height. The write out frame format has two options: RGB565 and RGB888.

This function is enabled and disabled before refres hing a frame. Turning on or off at frame refreshing will occur error

3.4.8 HDMI Transmitter

HDMI (High-Definition Multimedia Interface) is provided for transmitting digital television audiovisual signals from DVD players, set-top boxes and other audiovisual sources to television sets, projectors and other video displays. It can carry high quality multi-channel audio data and can carry all standard and high-definition consumer electronics video formats.

HDMI device uses a TMDS transmitter which encodes and serially transmits an input data stream over a TMDS link to a TMDS receiver. The TMDS link is used to send audiovisual data to the monitor. These audiovisual data items are processed in a variety of ways and are presented to the TMDS encoder.

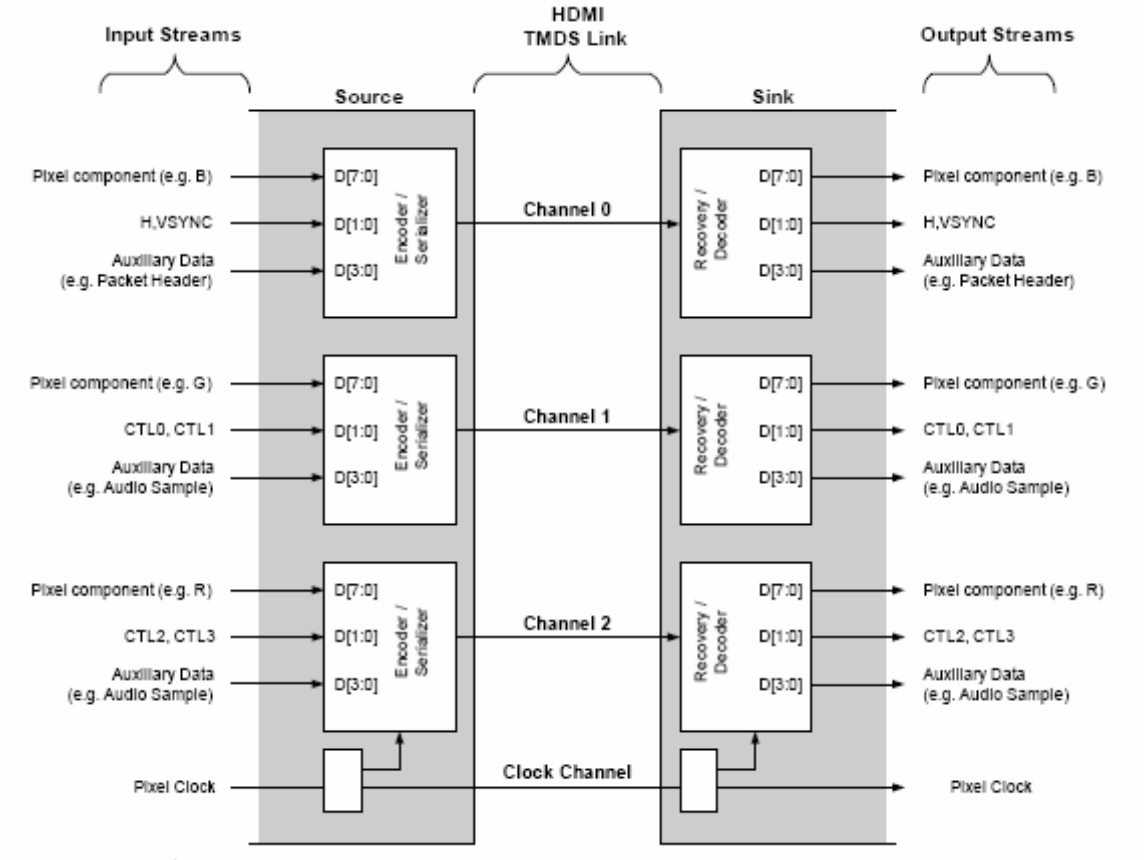


Figure 13. HDMI overview

Features of the HDMI transmitter in AIT8427:

1. Pixel clock rate from 25MHz to 165MHz.
2. Support 1280 X 720 @ 60P, or 1920 X 1080 @ 60i
3. Support pixel encoding RGB 4:4:4, YCbCr 4:4:4 and YCbCr 4:2:2.
4. Only support progressive video format, don't support interlace format.
5. Only support 24-bit color depth, don't support deep color mode.
6. Support L-PCM, AC3/IEC61937 compress audio.
7. Support multi-channel L-PCM.
8. Support 32kHz – 192kHz audio sample rate / frame rate.
9. Auto ACR extraction and transmission.
10. Auto AVI & ADO(audio) InfoFrame transmission.
11. 2 programmable packets for other usage.
12. Support for different HSYNC, VSYNC polarity.

13. Flexible routing option for different footprint and PCB trace

3.5 Sensor Interface

The Video-In-Interface (abbr. as VIF) is used to communicate with CMOS (or CCD with CMOS interface) sensor, receive and process sensor frame data, then send out for further processing. In most case, the CMOS sensor is in Master mode while VIF is in Slave mode. That is, the VIF just receives frame synchronization and data signals from sensor. The master clock for sensor will be (13*N/M) Mhz (depend on sensor type) to achieve 30 fps VGA or SVGA or other types of preview mode. When the light is dark and the exposure time needs to be long, the VIF can be programmed to enter “night-shot-mode” for lower frame rate without affecting system performance.

To meet post image flow, VIF should receive sensor data in Bayer 10-bit or YUV422 8-bit data bus. Besides, it can also receive generic data (like JPEG bit-stream) and store to memory for flexibility.

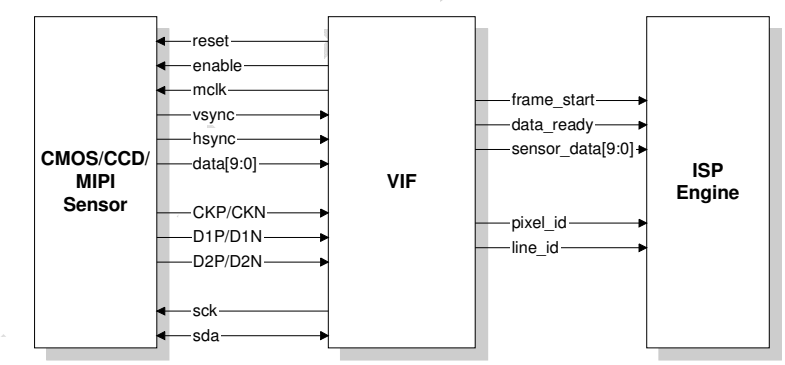


Figure 14. Basic sensor Interface

Sensor interface features:

- Support sensor master mode (VIF in slave mode)
- Support CCD sensor with CMOS like interface
- Support max. 4 GPIOs for sensor control and serial interface usage
- Support various sensor clock speed (13*N/M Mhz)
- Support various pixel data output flow: YCbCr422 to YCbCr444 without CDSP, Raw data and JPEG
- Support frame skip mechanism from 1/16 to 15/16 frames in view finder mode
- Flexible VSYNC, HSYNC, and MCLK control
- Most register settings are frame synchronization
- Support hardware calibration
- Support black level compensation
- Provide interrupts for HOST and CPU about frame start, frame end, exposure end, and generic line number interrupt
- Support one IGBT output for flash light control
- Full synchronous and synthesizable design

3.6 ISP

AIT8427 implements a hardware ISP engine that handles camera related functions such as AE, AWB, AF, etc. It also supports sensor improvement features such as defect pixel compensation, anti-flare, anti-crosstalk, lens shading correction, etc. Many special effects, including sepia, emboss, sketch, etc, are also done by this engine.

3.6.1 Interpolation (De-mosaic)

The incoming pixels from sensor are Bayer-pattern raw RGB data. With this kind of raw data, a pixel contains only one of the RGB components. It needs to go through a process called interpolation to generate all RGB components for each pixel. The general way of generating the missing two color components is to take the neighboring pixels with the missing color components and do the averaging. But this approach makes pictures look un-sharp.

AIT8427 implements a special algorithm called Edge-Adaptive interpolation. The approach is to detect edges of color block in the picture and if a pixel falls into the proximity of an edge, then a special way of averaging is conducted to generate the RGB components for the pixel. In this way, the sharp edges in the picture are maintained thus making the pictures much sharper than regular approaches.

One of the artifacts caused by unsophisticated interpolation is mosaic effect. Mosaic is a phenomenon caused by the high frequency in the picture (for example, high concentration of very fine black and white strips). It happens when the picture frequency is over the resolution of the lens and sensor, or can be caused by interpolation. What appear are some visual artifacts. For example, the highly concentrated vertical black stripes against a white background may show up as color stripes.

3.6.2 Defect Pixel Compensation

Sensors contain defect pixels due to contamination in the manufacturing process or other causes. There are two types of defect pixels: dead pixels and hot pixels. Dead pixels show up in the image constantly while hot pixels are image dependant. AIT8427 can correct both dead pixels and hot pixels. By doing defect pixel compensation, the bad pixels can be corrected using neighboring good pixels, thus making the sensor usable even if it contains defect pixels.

3.6.3 Auto-Exposure

Auto-Exposure is done by dividing frame into windows and analyzing the computed luminance data for the windows to decide for the right exposure. Programmable Histogram Analysis is also conducted to facilitate accurate exposure.
According to the set brightness, the Auto-Exposure engine controls the sensor exposure time, analog gain, digital gain, etc. to achieve the desired exposure. Temporal filter with fin gain control is also implemented to achieve smooth exposure change.

3.6.4 Flicker Reduction

For CCD sensor, the exposure is done to the whole frame, just like film camera. But for CMOS sensor, the exposure is done line by line. Under light source with flicker (fluorescent light, for example, release bursts of light at 50 or 60Hz), the amount of exposure for each line may not be the same. Under this situation, white rolling stripes show up in the image. The ISP needs to detect the flicker and eliminate it by adjusting the line exposure time in sync with the flicker frequency so the amount of exposure for every line is the same. That means the exposure time adjustment must be in increments of a flicker period, which is a very big unit and can cause un-smooth exposure transition. AIT8427 implements fine gain control to augment the flicker control to facilitate

smooth exposure transition.

3.6.5 Auto White Balance

For Under light sources of different color temperature, the images may look tinted. Blue tint under cool light, and yellow tint under warm light. While human eyes can dynamically adjust for the light source and see white object as white, sensors can not. White balance is designed to compensate for different kinds of light sources and reveal the true color of the objects and scenes.

AIT8427's AWB engine is based on statistical light temperature detection algorithm for accurate white balance control. It supports both automatic and manual white balance.

3.6.6 Auto Focus

AIT8427 supports auto focus by implementing algorithms that detect the sharpness of the incoming image and adjusts the focus by controlling the lens movement. The performance of the auto focus algorithm is defined by two criteria: first, how soon can it move the lens into focus; second, will it oscillate after being in focus. AIT8427 implements a proprietary fast convergent auto focus algorithm that is able to drive the lens to settle into focus in just couple of frames.

The Auto Focus lens control is done by hardware/firmware. The embedded CPU programs the GPIOs (PWM/Serial bus) to control the AF lens movement.

AIT8427 supports PWM output waveform with frequency from 13Mhz to 100Hz (real speed depends on input clock rate) and duty (0% < duty < 100%).

Serial flash is supported to store information such as lens calibration data.

3.6.7 Lens Shading Compensation

Lens shading causes images to darken away from the optical axis. This phenomenon is especially serious for small lens used in mobile phones. The ISP needs to build in mechanism to compensate for the loss of light.

The complexity of Lens Shading Compensation lies in that the shading does not always show up as perfect coaxial circles at the center of the image. AIT8427 supports programmable lens center and proprietary Lens Shading algorithm that allows very flexible shading compensation.

3.6.8 Chroma Shading Compensation

Chroma shading is similar to lens shading. While lens shading causes luminance variations, chroma shading causes color variations. More complicated than lens shading, the chroma shading sometimes can not be modeled by a linear equation, which means the shading is not a simple coaxial circle that gradually shades deeper. There can exist a non-continuous color intensity change as illustrated in the picture below. AIT8427 implements programmable 3 arrays to fit "any" chroma shading model to allow accurately modeling this condition and therefore be able to compensate for it.

3.6.9 Noise Reduction

Noise happens easily in low light situation. The pictures look grainy and un-smooth. Noise Reduction reduces the noise and recovers the smooth and colorful pictures.

The feature not only improves the picture quality but also improves the bit rate of MPEG4/H263 video compression. It's because noise contributes to the high frequency part of the picture which is very inefficient for compression. By reducing the noise, the compressed bit rate improves.

Noise Reduction is usually implemented with filtering functions. The drawback of simple filtering is the image loses the sharpness due to the averaging effect caused by filtering. AIT8427 implements a special Edge Adaptive Noise Reduction algorithm that maintains the sharp edges while filtering out the noise. Besides, it also considers cross-talk and signal level and dot noise.

3.6.10 Wide Dynamic Range

Dynamic range is the difference in brightness between the brightest part and the darkest part of an image. In the image signal processing, it is often necessary to perform dynamic range compression in order to show the captured image on the final display so that no image information would be lost. A common technique for performing the dynamic range compression is based on gamma correction, which applies a non-linear transform curve to each pixel in the image, and boosts the brightness of the pixels in a dark region.

AIT8427 adapted an industry-leading solution from a -party to improve the dynamic range result. The WDR hardware automatically generates a unique transform curve for each image, based on a statistical analysis. And each pixel can have a different curve, based on the analysis. The can produce a best result under all conditions, which mimics the human retina behavior.

3.6.11 False Color Reduction

False color is caused by pixels interfering with each other at high frequency area. It is especially noticeable in the high contrast area, for example, at the boundary of dark and light. What appears is color blur of different colors that do not exist in the scene. AIT8427 implements Edge Intensity Detection and Chroma Filtering to solve this problem.

3.6.12 Anti-Crosstalk

The crosstalk is when the photons falling on one pixel are sensed by other pixels around it. It usually causes the discrepancy of Gb and Gr because of the pixel interference. As pixel size of sensors become smaller and smaller, this problem is getting more serious, especially for sensors of 3M pixels and up. The problem with crosstalk is it can not be statically modeled. AIT8427 implements 2D high stop filter to solve this problem.

3.6.13 Anti-Flare

For lens of less quality or in-appropriate design, the dark area of image will pick up a cast of light that looks like flare. AIT8427 is able to compensate for this effect and restores to vivid and dark black.

3.6.14 Black-level Compensation

Pixels of sensor may also pick up a slight cast of light due to dark current. The effect looks like the image is slightly over-exposed. This over-estimated increment needs to be deducted. Most sensors nowadays already have this feature built in. AIT8427 still implements this function just in case.

3.6.15 Histogram Analysis

By analyzing histogram distribution, we can do luminance enhancement to increase the dynamic range of the picture.

3.6.16 2D Filtering

AIT8427 supports real digital zoom. That is, a portion of the preview window can be cropped and scale up to desired picture size. Up to 16 x digital zoom is supported.

As a common problem to image that is scaled up or down, the generated image usually looks jaggy. AIT8427 eliminates this problem by applying a 2-dimentional filtering to reduce the jaggy outline and produces smooth pictures.

3.6.17 Edge Enhancement

AIT8427 implements an edge enhancement function that produces sharper images and with less noise enhancement. The level of sharpness is programmable

3.6.18 3D LUT color correction

Color management usually plays a decisive role in ISP. Color management is highly improving in AIT8427 by 3D LUT color correction. We can modify colors more free and more heavy, now.

3.6.19 Histogram Equalization or Modification

AIT8427 supports histogram equalization or modification by using programmable curve and good algorithm.

3.6.20 Color Space Conversion

AIT8427 Color Space Conversion converts pixels from RGB color space to YCbCr color space bases on ITU.BT-601. Also, it implements a programmable color space conversion function that can be configured to do conversion to different types of color spaces.

3.6.21 Gamma Correction

Most LCD displays produce an intensity that is proportional to the power of the signal amplitude, therefore is non-linear. But the intensity representation of pixels (0 to 255) output from sensor is linear. By gamma correcting the signal before been displayed, the intensity output of the display becomes linear. The result is a more real reproduction of the image.

3.6.22 Brightness/Contrast/Hue/Saturation Control

Brightness enhances the light intensity across all regions of the image. Contrast is achieved by stretching the light intensity of the mid-tone regions of the image and pushing both light and dark regions into saturation. Saturation is achieved by stretching the color intensity of the image.

3.6.23 Special Effects

Many special effects are also supported. These effects are both for preview and for capture. Because the special effects are handled by hardware, there is no overhead to the software. And because of that, not just still picture but also video can have these special effects in real time. Therefore, many interesting features can be

implemented. For example, a sepia video, video with emboss effect, video with oil paint touch, etc. all can be produced.

Figure 8. ISP Special Effect Picture Example

3.6.24 Face Detection

Face detection function in the preview mode is supported by software algorithm and hardware acceleration circuits. The position of the detected face can be used as the target of AE and AF, and the color setting can be set for better human skin presentation. The minimum size for face detection is 20x20 pixels. AIT8427 can detect the multiple faces in a picture. H/W face ICON engine is embedded to draw face frame quickly and in flexible way.

3.7 JPEG Codec

JPEG block provides hardware solution for baseline JPEG encode and decode. In jpeg encode, we provide two data formats YUV 422 for JPEG to compress. In JPEG decode, YUV 444/422/420/411 are decodable. Besides, JPEG provide two data path modes for user to fetch data. User can choice FIFO mode to get compress data from or put compressed data into JPEG FIFO. This is used to process large size picture that memory can't store it and it can also save the bandwidth of frame buffer. If we want to store the compressed into external storage device, we need to enable this mode, too. Otherwise, User can choice another mode called memory mode to store compressed data into frame buffer and waiting for CPU/HOST have free time to get it.
The JPEG block also provides multiple data source input to frame buffer or HOST/CPU. User can specify the data format that user wants to get. And JPEG will get the data from corresponding module and store it into Frame buffer or JPEG FIFO through JPEG data path for user to use controller is the interface to control sending data to display device, such as LCD panel or TV out interface.

JPEG Codec module features:

- Real-time JPEG encode up to 12M pixel resolution
- Max decode resolution up to 8192 (H) * 8192 (V)
- Compliant with JPEG baseline standard (ISO/IEC 10918)
- Supports YUV 422 encoder format
- Supports YUV 444/422/420/411 decoder format
- Support memory & FIFO path
- Generate marker data automatically
- Programmable scaling factor to adjust Q-value
- Support sticker overlay encode
- Image stabilizer function
- Motion JPEG function support
- Share FIFO / MEM path for various data format (RGB565, RGB888, YCbCr422, Raw data)

3.8 Video CODEC

AIT8427 has dedicated hardware to support H.264 encoding, and a multiple-format video decode engine. The formats that the decoder supports include H.264, MPEG-1 / MPEG-2 / MPEG-4 / H.263, DivX, VC1, and RV. The following table summarizes the video encoding and decoding capability for each format.

H.264 encoding	Up to High Profile supported
H.264 decoding	Up to High Profile supported Baseline Profile, levels 1 - 4.1 Main Profile, levels 1 – 4.1 High Profile, levels 1 – 4.1

- Encode Image data from sensor or any other source through CCIR601 interface. Hardware encoder will process and store encoded bit stream into frame buffer. Image data can be processed by AIT professional high quality image signal processor to achieve special effect, image fine-tuning and scaling, etc..
- Image data to be encoded can be combined with data from icon engine to achieve sticker effect before encoded by codec. User can choose different sticker shape for their various purpose.
- For VGA resolution video encode, large internal frame buffer can be used as PIP frame for LCD display to save DRAM bandwidth and power consumption. Current frame of encoding content should still be allocated in external SDRAM.
- Image data after decoding can be processed by post-processing like de-blocking engine to reduce MPEG4 artificial block boundary. High bit-rate may not be processed to get sharper image to display.
- After decode and post processing, image can be re-processed by ISP to achieve special effect, image scaling or rotation. User can choose to output the image to LCD or TV display with video and audio synchronization.

3.7.1 H.264 encode

Features	Decoder support
Standard	H.264 (05/2003) Advanced video coding for generic audiovisual services
Profiles	Baseline Profile, levels 1 - 4.1 Main Profile, levels 1 – 4.1 High Profile, levels 1 – 4.1
Input data format	H.264 byte or NAL unit stream
Slices	I, P and B slices
Entropy decoding	CAVLC, CABAC
Weighted prediction	Supported
Sequence type	Progressive only
Encoding scheme	rate control with CBR
Maximum frame size	1280x720
Maximum frame rate	30 fps

3.7.2 H.264 Decoder

Features	Decoder support
Standard	H.264 (05/2003) Advanced video coding for generic audiovisual services
Profiles	Baseline Profile, levels 1 - 4.1 Main Profile, levels 1 – 4.1 High Profile, levels 1 – 4.1
Input data format	H.264 byte or NAL unit stream
Slices	I, P and B slices
Entropy decoding	CAVLC, CABAC
Maximum number of slice groups	8 (if more than 1, S decoding)
Arbitrary slice order	Supported (SW performs entropy decoding)
Redundant slices	Supported, but not utilized
Image cropping	Not performed by the decoder, cropping parameters are returned to the application
Weighted prediction	Supported
Sequence type	Progressive, interlaced
Decoding scheme	Frame by frame Slice by slice
Maximum frame size	1280x720
Maximum frame rate	30 fps
Error detection and concealment	Supported

3.9 Audio Function

The audio block comprises two operations mode, i.e., the CPU mode and auto mode for user selection.
The CPU mode operation is used for the user who wants to debug or trace the incoming data form voice SDM or I2S interface and the data outgoing to I2S interface. This means that the user can easy to follow the interrupt step to trace the data flow from one side to another. Usually user should use the auto mode to do the normal operation, it takes advantage of the data pool/first-in-first-out buffer to store the temporary frame samples.
When the auto mode is activated all the data is pointing to the FIFO, that is said, it is a frame buffer for burst data in or out for waiting and be processing to CPU or to I2S interface or voice SDM and so on..., for instance, user wants to play a MP3 audio has to fill up the compressed data into I2S data FIFO, waiting for the interrupt signal from I2S.
The audio module is designed to provide a standard audio interface that can be connect to any audio/voice melody chip and provide a path that can record voice simultaneously.

Audio interface features:

- Wide range for I2S sampling rate and data formats
- I2S time counter for AV sync use
- Bi-directional LRCK and BCK
- Provide I2S_MCLK out
- Internal voice ADC support

- CPU/Auto mode support
- Speech sigma-delta modulation support (input)
- I2S compatible with Audio serial port (ASPORT)

3.9.1 Audio Datapath

The whole big picture is shown as following. The data flow via the path from one block to another is shown. It is also show the relationship for the blocks.

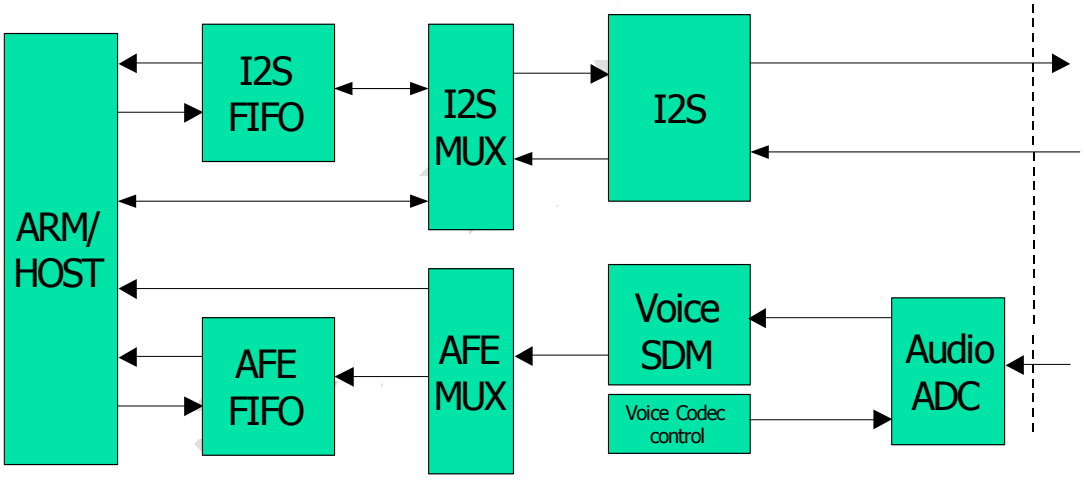


Figure 9. Audio Path and Flow

In figure 1.1, the data can be from host to FIFO or from CPU to FIFO or vice verse etc. The output path, we only provide data through I2S interface. You can configure I2S register to meet almost all the purpose you want. And the data will go through the path from I2S FIFO to I2S MUX and finally to the I2S in auto mode or from the CPU to I2S in cpu mode. I2S will convert the input data into a proper sampling rate, serial data and bit clock to audio/voice melody chips. For the input path, we can use our internal ADC to record voice data by microphone. After sigma-delta modulation and filtering, the PCM data will be store into AFE FIFO through AFE MUX in 8k or 16k sampling rate and waiting for CPU or Host to process it. Or we can use external audio chip to record the data and use I2S interface to receive it with proper sampling rate (8k, 11.025k, 12k, 16k, 22.05k, 24k, 32k, 44.1k, 48k …etc). The received data will pass to I2S FIFO through I2S MUX and waiting for CPU or Host to process it.

3.9.2 MP3 / AAC Decode Operation

This example is shown that the CPU processes and decompresses the mp3 data that is store into the I2S FIFO waiting for the I2S interrupt triggering the I2S FIFO to send the data out to the I2S data port. The PCM data is needed to convert to a standard format for a variety of audio/voice melody chips need, for instance, left adjusted or right adjusted in 64bits, 48bits or 32bits format is using. And sampling rate is also a point to be setting in 8kHz, 11.025kHz, 12kHz, 16kHz, 22.05kHz, 24kHz, 32kHz, 44.1kHz, 48kHz …etc of Left-Right channel clock frequency in polarity positive or not. In figure 3.1 shown is tell the host must be setting the audio CODEC chip to deal with the format between each other.

3.9.3 I2S Interface

The AIT8427 I2S interface can connect to two different types of audio codec with data transmit and data receive. One is with normal I2S interface and the other is with ADI audio interface. The following paragraph will illustrate

those two connections mode.

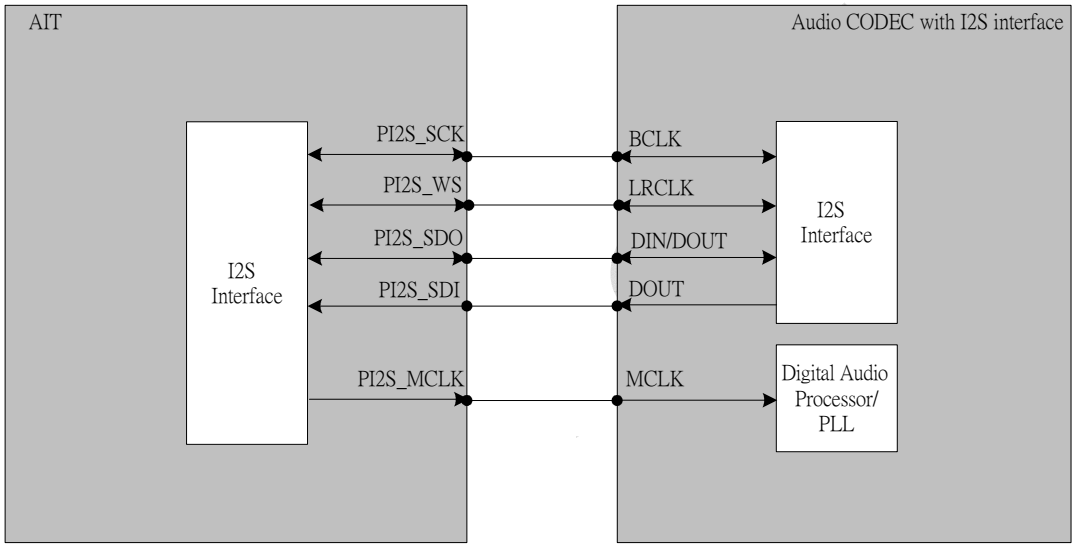


Figure 10. I2S Interface Connection

3.8.8.1 Normal I2S Bus Output Clock Format

The AIT8427 I2S interface can support 8k, 11.025k, 12k, 16k, 22.05k, 24k, 32k, 44.1k, 48k sample rate with 16/24/32-bits data format. However the output master clock format will be limited with output data size. The following table shows the supported output master clock format:

Master Clock Frequency	Output Data format		
	16-bits	24-bits	32-bits
128fs	All sample rate are available		All sample rate are available
192fs		All sample rate are available	
256fs	All sample rate are available		All sample rate are available
384fs		All sample rate are available	

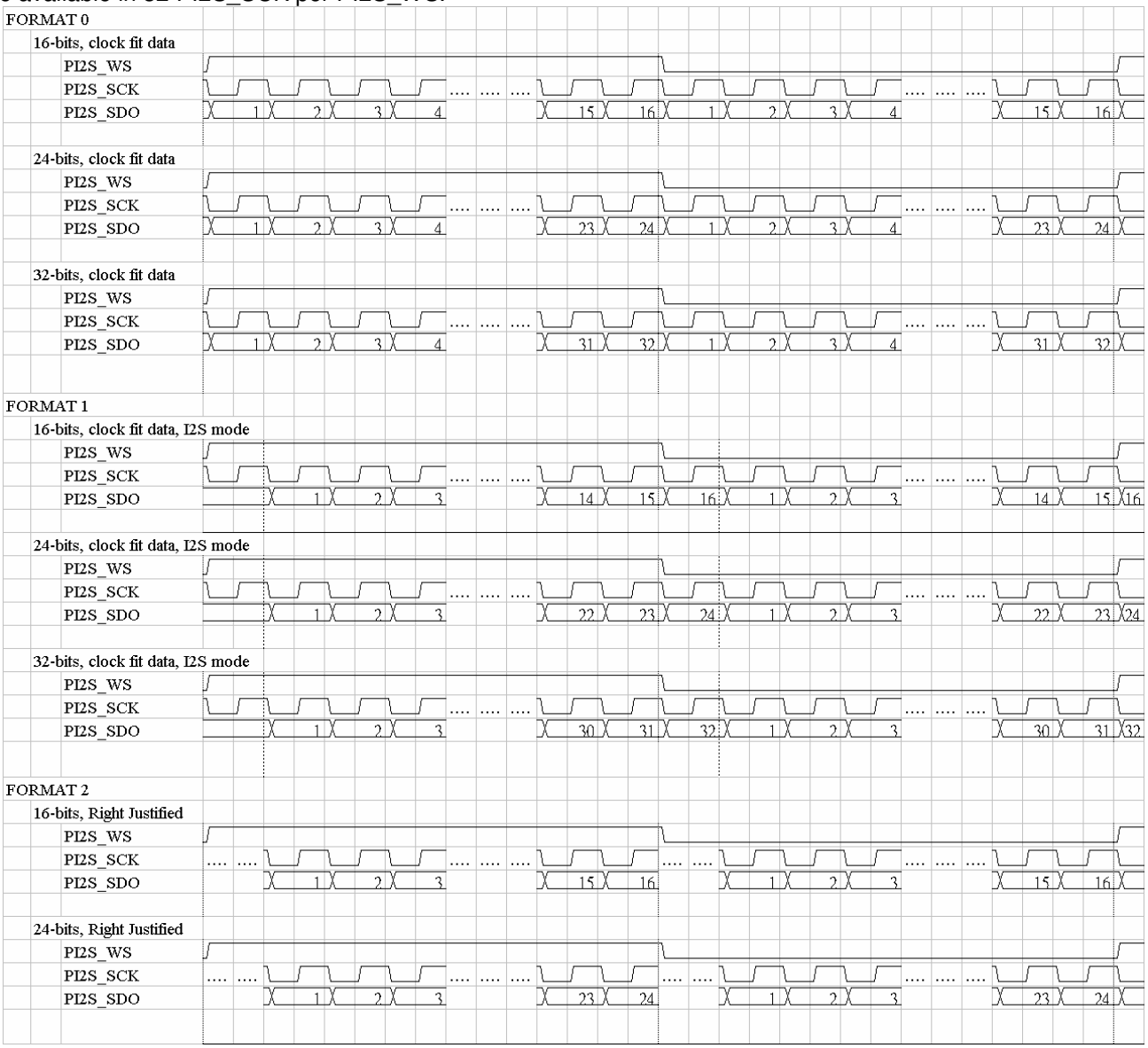
Table 1. I2S Supported Ouput Master Master Clock Format

Besides, the I2S bus can also operate in master mode or slave mode, meaning that the PI2S_SCK and PI2S_WS can be provided as input from audio codec (slave mode) or can provide as output to audio codec (master mode).

3.9.3.2 Normal I2S Bus Output Data Format

The following figure illustrates five types of output data format and timing that can be supported by AIT8427 I2S bus. For each output format, 16/24/32 data sizes are available and all data out are MSB first. The AIT8427 I2S bus can transmit 32/48/64 bit clocks (PI2S_SCK) in one left/right clock (PI2S_WS). But only 16 bits data format

are available in 32 PI2S_SCK per PI2S_WS.



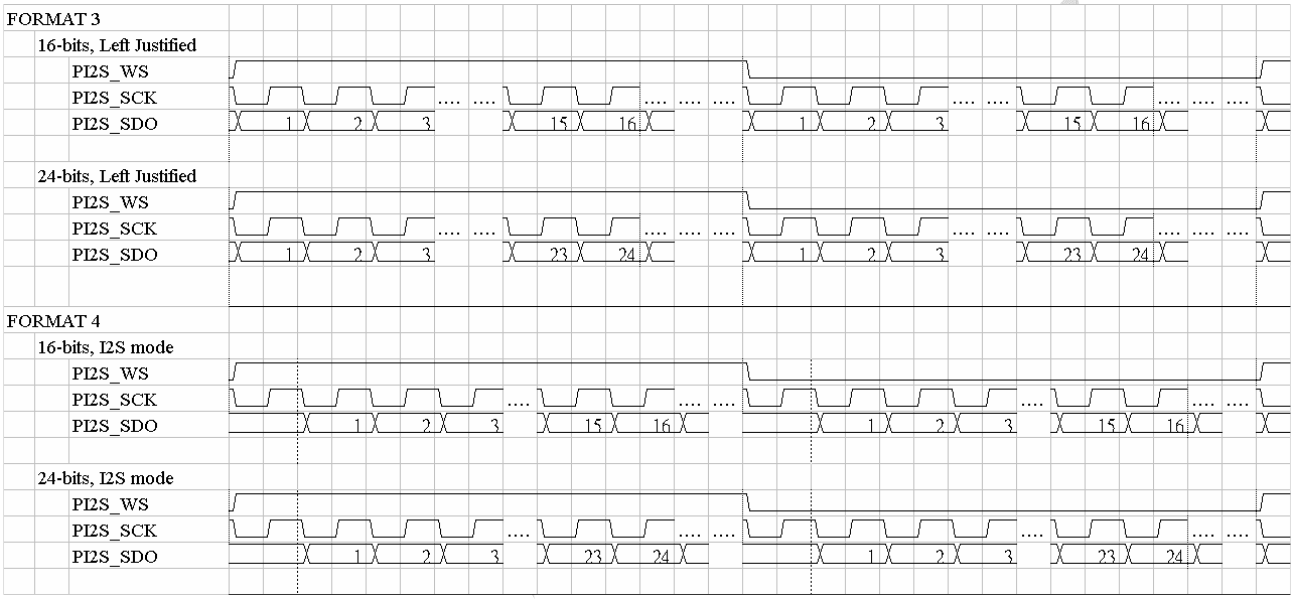


Figure 11. Audio Data Output Format

3.8.8.3 Normal I2S Bus Input Data Format

The following figure illustrates five types of input data format and timing that can be supported by AIT8427 I2S bus.

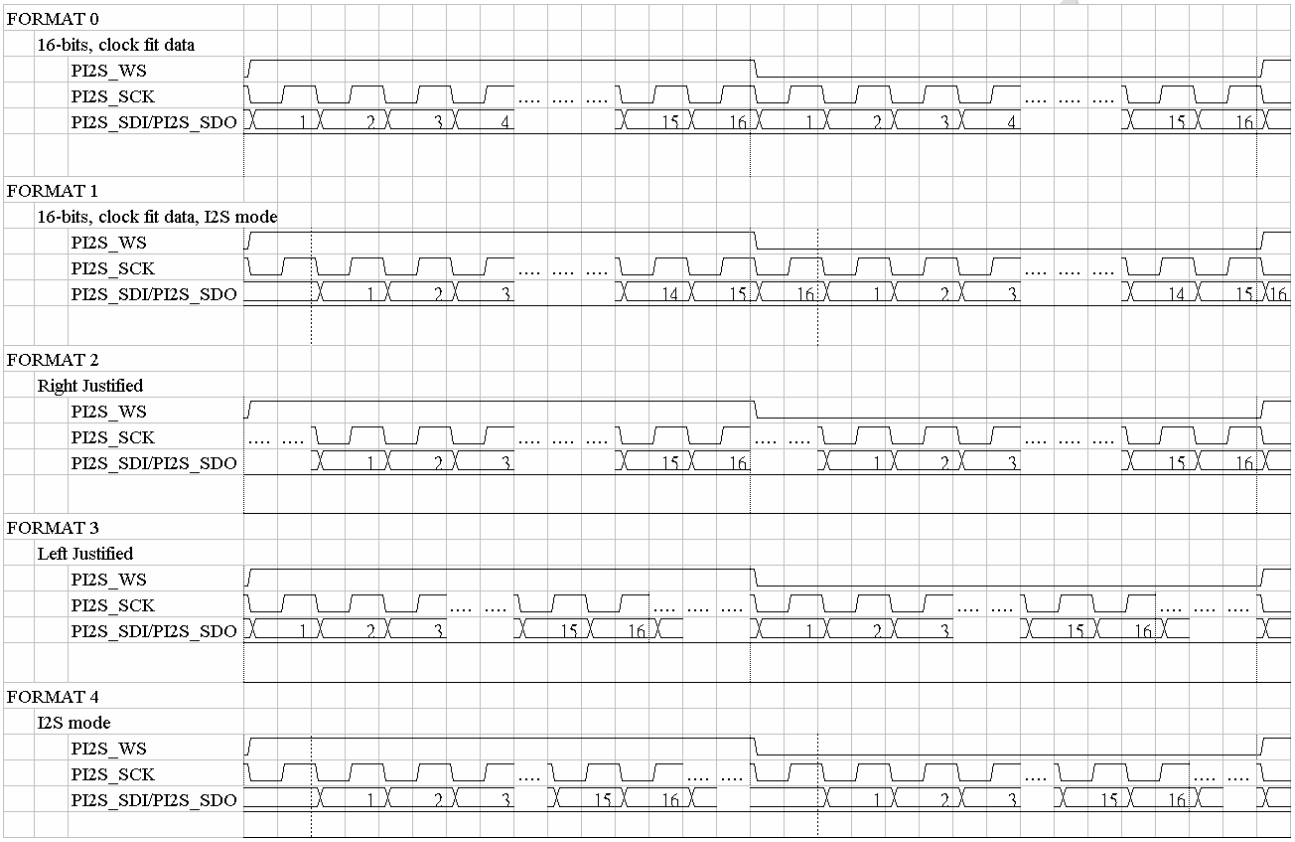


Figure 12. Audio Data Input Format

3.9.4 Internal Audio CODEC Function

The AIT8427 integrates an internal audio CODEC function which supports high quality audio playback and record functions.

- **Stereo 20-bit CODEC (at 2.5V supply):**
  - DAC SNR 95dB, THD -75dB@-6dBFS ('A' weighted)
  - ADC SNR 95dB, THD -72dB@-6dBFS ('A' weighted)
  - Mixer function
  - Sampling rate 8KHz~192kHz
  - System clock: 128Fs/256Fs (DAC/ADC) (2.048MHz~12.288MHz)
  - Support USB mode 12MHz clock
  - DAC supports mono-differential output (L+R)
  - Low power audio bypass mode
- **On-chip Headphone Amplifier:**
  - 6mW output power into 16Ω//200pF load
  - THD 65dB @16Ω load
  - Analog PGA: +6dB~-42dB with mute
- **Microphone Preamplifier:**
  - Programmable microphone bias output
  - Analog PGA: -10dB~+20dB, including 20/30/40dB gain-boost options

- Differential input

- **Digital Filters:**
  - Output path volume control: -47dB~0dB
  - Input path volume control: -47dB~0dB
  - Soft mute function
- **Digital Interface:**
  - Audio: Left-justified, Right-justified and I^2S
- **Other Features:**
  - Input automatic level control for ADC
  - Digital loopback mode
  - Power-on reset circuit
  - Microphone bias circuit

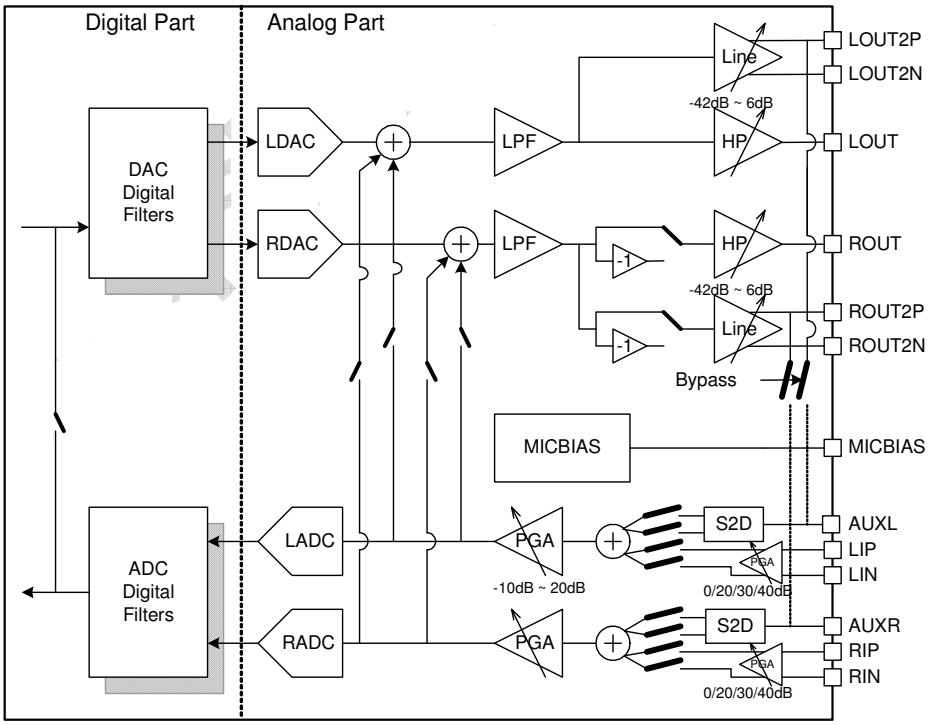


Figure 23. Audio CODEC Block Diagram

3.10 ICON Engine

AIT8427 Icon engine supports 2 separate icons path: LCD path and JPEG/Video encode path.
In LCD path, Icon can provide data for LCD controller's icon layer if "LCD's icon layer enable bit" is set. It can be used to put some icons (battery or message box or mail box or ….) on LCD panel. Total 8 LCD icons can be use at the same time.
In JPEG/Video path, Icon will overwrite YUV data from color/scaler to JPEG/Video engine. It can be used to put date information or sticker image on JPEG picture or video frame. Total 2 JPG/Video icons can be use at the same time.

3.10.1 Parameters

There are several parameters software needs to specify to use each icon. As the following figure shows:
1. Icon data start address, where the icon image is put.
2. Icon width/height
3. Icon start location (related to LCD icon window)
4. Icon end location (related to LCD icon window). Note: end location = start location + (Width or Height).
5. Icon format. Support 8-bit index, RGB565.

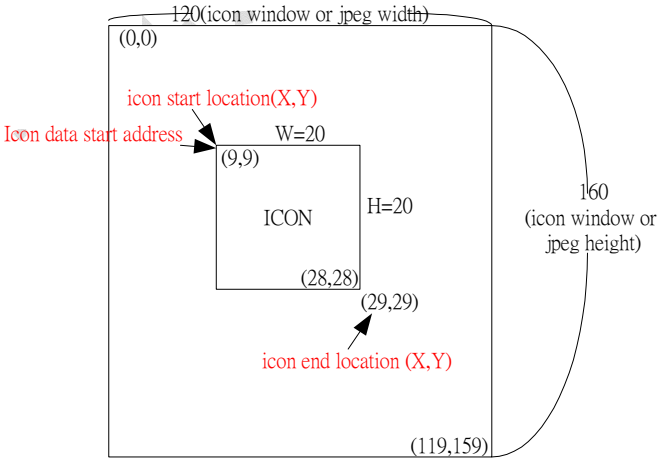


Figure 24. ICON Parameter

Icon engine also can support "transparent" and "semi- transparent" effect. "Background color" color and "weight" can be set for these two effects.

ICON engine features:

- LCD path up to 8 icons
- JPG/Video path up to 2 icons
- 2 path supported: LCD mode and JPEG/Video encode path can operate at the same time
- 2 icon format supported: 8-bit index, RGB565
- Support transparent/semi- transparent
- TV mode support

3.10.2 LCD path Operation

Icon0~7 serve LCD engine as icon layer window. If the “icon window”(LCD register) is enable, LCD engine will request RGB-888 data from Icon engine. Icon engine will get data from memory and convert to RGB888 then feed LCD engine. LCD engine will show icon image on LCD panel with other layers.

3.10.3 JPEG/Video operation

Icon8~9 engine can replace color/scaler output data stream with icon image data then send the data to JPEG engine or video engine for encode. So User can add the year/month/date information or cute sticker pattern on pictures.

3.10.4 Transparent and Semi- transparent

The following figure shows the effects of transparent and semi-transparent.

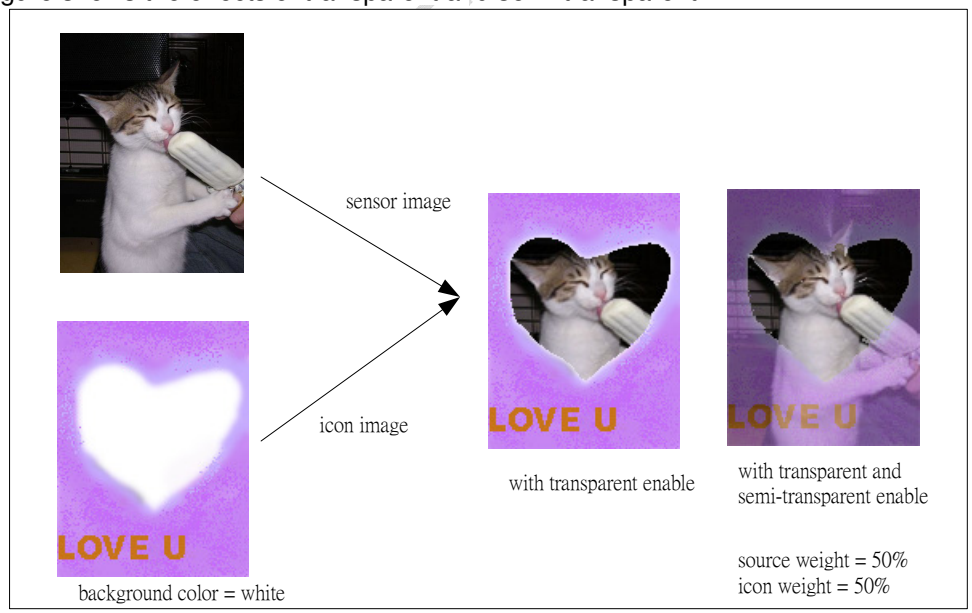


Figure 25. ICON Example

3.11 2D Graphics Engine

Graphic engine supports 2 main functions: “2D BitBLT Operation” and “Scaling operation”.
“2D BitBLT operation” is used to move a block of image data from a source location to a destination location and also can do some arithmetic or logic operation on both source and destination data. We called it “BitBLT”. In PC system, it is usually done by software. But in mobile phone system, we implement it by hardware to reduce the CPU work of moving pixel data in and out. This operation is usually used to do graphic effect on 2 images.
“Scaling operation” is some special data paths used to move data and scaling up data (from Frame buffer or from graphic FIFO) to JPEG engine (to generate JPEG file or decode JPEG file) or to frame buffer (for display). Those paths are very flexible for user to re-process existence data either in frame buffer or HOST.

2D Graphics Engine features:

- Host Write BitBLT(source data from Host)
- Host Read BitBLT(destination data to Host)

- Move BitBLT(source and destination are both in Frame buffer memory)
- Color expansion
- Pattern fill
- Solid fill
- 16 ROPs
- Transparent/semi-transparent support
- Rotate mode
- Line draw with ROP
- Flexible scaling paths enable you to re-process your existence image data

3.12 USB2.0 Client

The AIT8427 chip integrates a USB 2.0 device controller with embedded USB transceiver.

USB controller features:

- Compliance with USB2.0 specification
- Support high speed 480Mb/s and full speed 12Mb/s data rate
- Support control, bulk, isochronous and interrupt transfer
- Embedded USB transceiver and clock generator
- Simple packet data transfer control
- Internal DMA mode for efficient data transfer between USB controller and internal memory

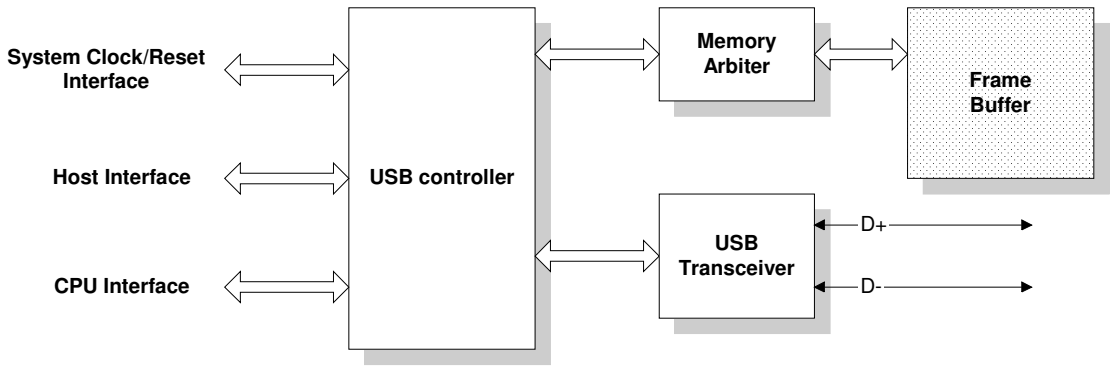


Figure 26. USB Architecture

Up to 4 endpoints can be configured as the USB specification defined:

1. Endpoint 0 (EP0): Control endpoint, 64-byte in length.
2. Endpoint 1 (EP1): could be configured as bulk/isochronous/interrupt endpoint. It can also support

high-bandwidth ISO transfers. (up to 3072-byte in length).

3. Endpoint 2 (EP2): could be configured as bulk/isochronous/interrupt endpoint, up to 512-byte in length.

4. Endpoint 3 (EP3): could be configured as bulk/isochronous/interrupt endpoint, up to 64-byte in length.

3.13 Connectivity

The AIT8427 provides a variety of interfaces for flexible system designs. Each of the interface can be configured by firmware, and can be used as GPIOs if the function is not needed.

3.13.1 I2S Interface

The I2S interface can be used to connect to an external audio CODEC or ADC.

3.13.2 SPI Interface

The SPI bus is a 4-wire serial communications interface used by many microprocessor peripheral chips. The SPI circuit is a synchronous data link that is standard across many Motorola microprocessors and other peripheral chips. It provides support for a low/medium bandwidth network connection amongst CPUs and other devices supporting the SPI. SPI bus is a master/slave interface. Whenever two devices communicate, one is referred to as the "master" and the other as the "slave" device. The master drives the serial clock. When using SPI, data is simultaneously transmitted and received, making it a full-duplex protocol.

SCLK (Serial Data Clock)

Only a master can drive the signal. Data is shifted and latched on the rising or falling edge of SCLK.

MOSI (Master Output/Slave Input)

Data is transmitted out of this pin if chip is a master and received into this pin if chip is a slave.

MISO (Master Input/Slave Output)

Data is received into this pin if chip is a master and transmitted out of this pin if chip is a slave.

SS (Slave Select)

It is used to indicate which slave is active.

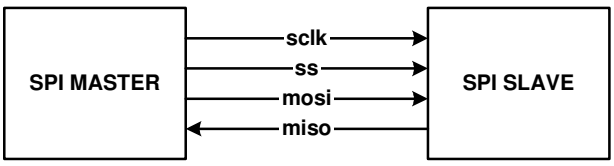


Figure 13. SPI System Connection Diagram

A typical waveform is shown below. A word is transferred between master and slave. The word length is 7-bit here.

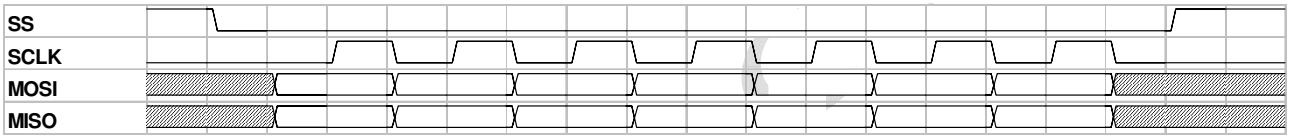


Figure 14. Typical SPI Waveform Diagram

Features of SPI Interface

- Polarity and phase of SPI clock (SCLK) is configurable
- Polarity of Chip select (SS) is configurable
- Word length is configurable, 3 ~ 32 bits
- Support two start modes, immediate mode and delay mode.
- Bit rates generated $\frac{1}{2(x+1)} \cdot clk_{system}$, where x is between 1 and 255 in integer.
- Allows Chip-Select (SS) signal remains assert between word transfers.
- Wait cycles between word transfers are programmable.
- Supports FIFO mode and DMA mode.
- 32 bytes receive FIFO and 32 bytes transmit FIFO

3.13.3 I2C Master Interface

AIT8427 supports one I2C master controller which can be used to control the CMOS sensor or other devices.
In Fig28, the data on SDA pin must be stable during the high period of the clock (SCL). Only the host may change the data while SCL is high. A high-to-low transition marks a START condition, and a low-to-high a STOP condition.

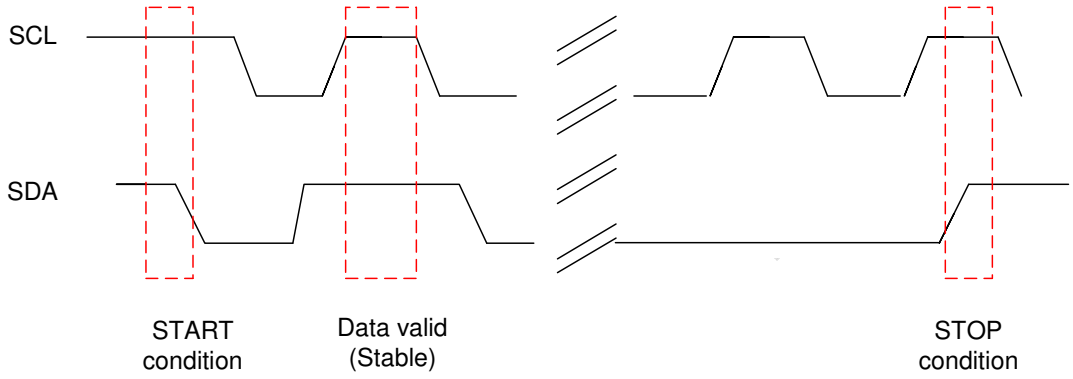


Figure 15: I2C Interface START and STOP Condition

The master device activates a START condition, and sends the first byte of data that contains the 7-bit address, and a direction bit(R/W#, 1 for read, 0 for write). The addressed device answers by pulling down the SDA line as acknowledged procedure.

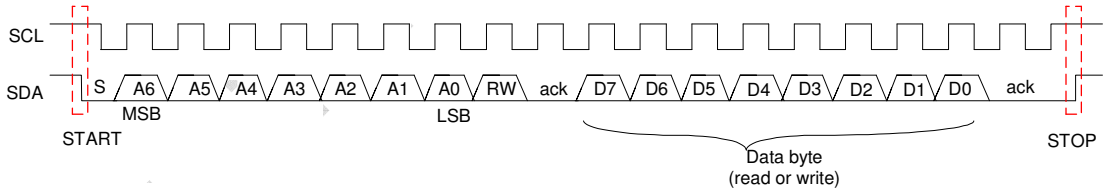


Figure 16: I2C Protocol

Features of the I2C master interface:

- 8-bits data transfer
- Support 8bit/16bit register address
- Support register address and write data FIFO to issue multiple I2C read/write access
- Support adjustable serial data line hold time
- Support internal port pulls up or external pull-up resistor/current source pulls up bus mode selection
- Support interrupt
- Support repeated start function (for CCI requirement)
- Support read from current location mode (for CCI requirement)
- Support 10-bit slave addressing (for CCI requirement)
- Support delay waiting cycle after each data (8bit+ 1 ack)
- Support clock stretching function

- No support slave wait state
- No support multi-master
- No arbitration, clock synchronization and master code
- No support START byte
- No support CBUS
- Support 100Kbps and 400Kbps transfers

The I2C master supports the clock stretching function, which the I2C slave device can slow down the transmission by holding the SCK signal to low. The slave device will hold down the SCK bus until it is ready to receive the next command. Therefore, under the protocol, the I2C master will not drive the SCK when it is under high level. The I2C master will check the real SCK level, and the transmission won't start until the SCK is really at high level which is driven by pull-up register.

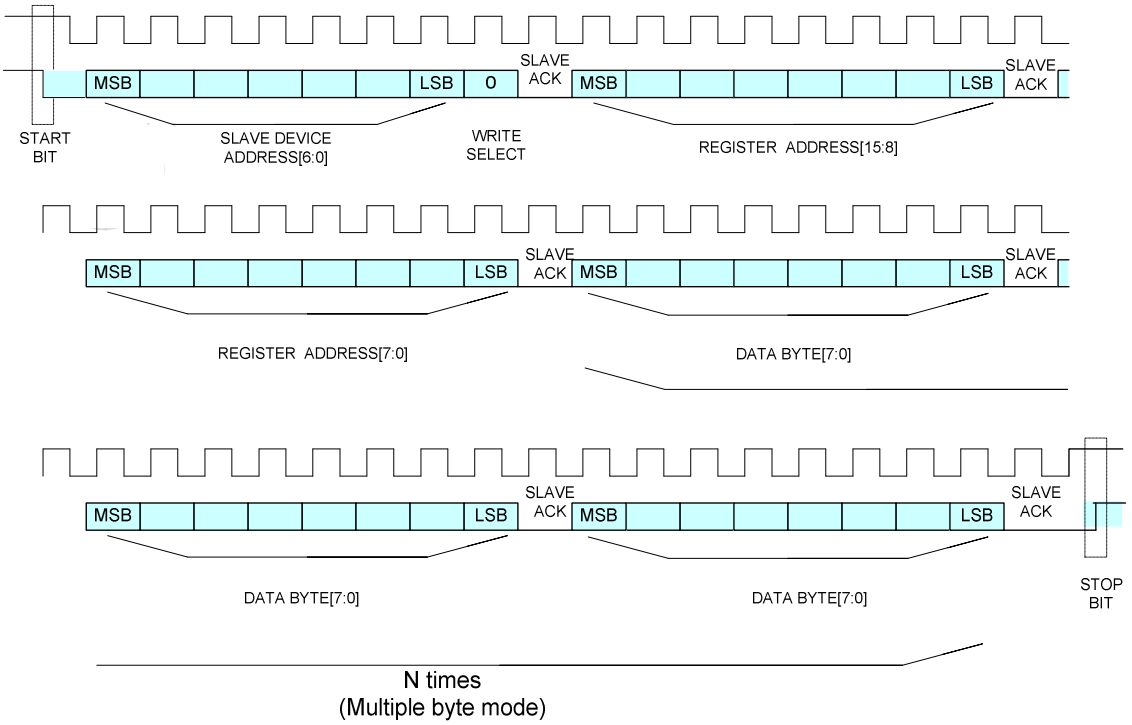


Figure 17: I2C Write Operations

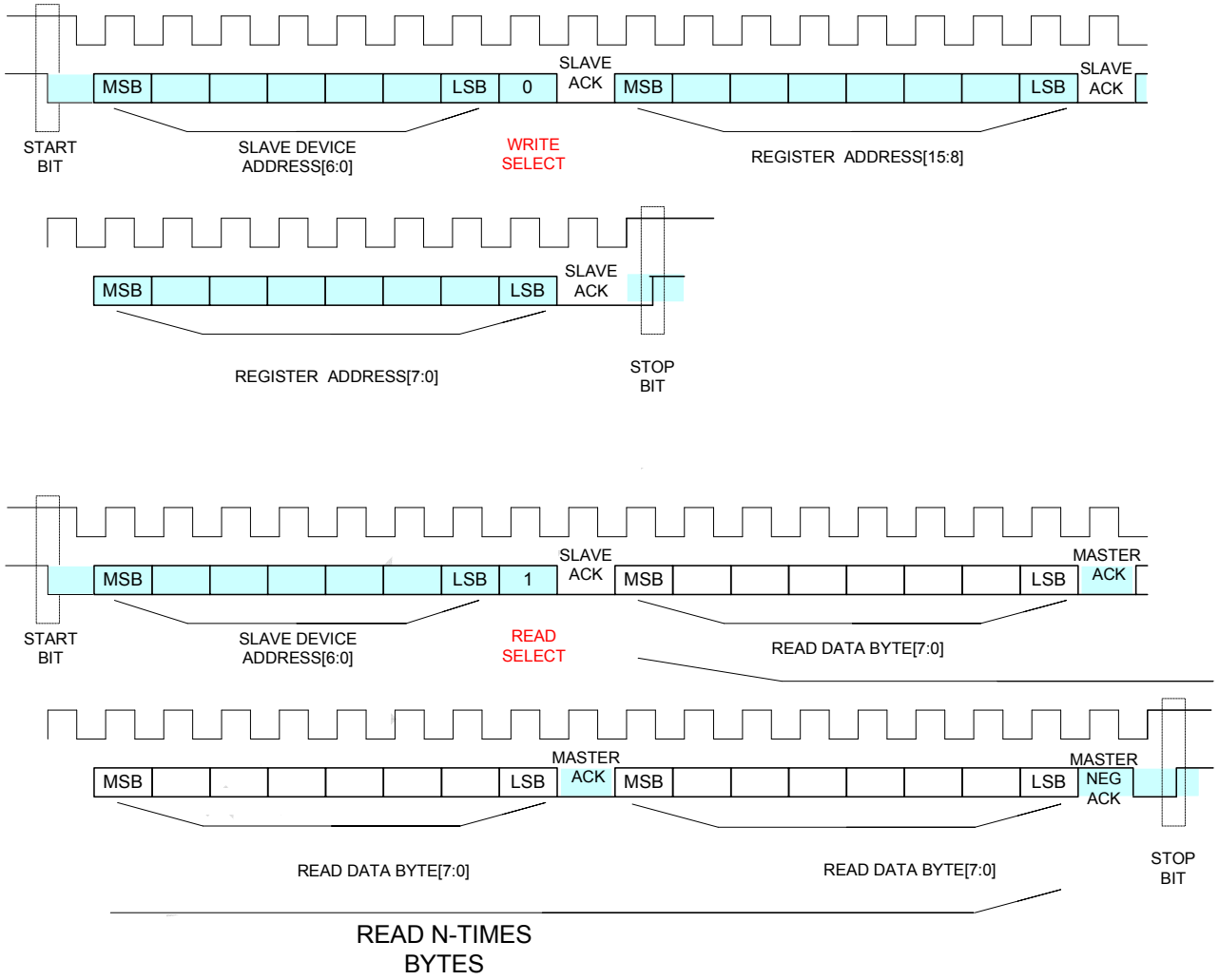


Figure 18: I2C Read Operations

3.13.4 I2C Slave Interface

AIT8427 supports an I2C slave interface which can be used to access internal registers for firmware development and debugging.

Features of the I2C slave interface:

- 8-bits data transfer
- Support 8bit/16bit/24bit/32bit register address
- Support address auto-increment data transfer
- Support adjustable serial data line hold time
- Support internal port pulls up or external pull-up resistor/current source pulls up bus mode

selection

- No support 10-bit slave addressing
- No support slave wait state
- Support auto error recovery
- Support error timer feature
- Fully synchronous and synthesizeble design
- More tolerance about wrong start timing
- Support 100Kbps and 400Kbps transfers

3.13.5 Serial Interface

The SIF (Serial Interface) is used to connect with serial interface slave devices, typically a serial flash memory in most systems. SIF supports boot auto-load function, which automatically downloads the program code from a serial flash to program memory upon power-up.

Features of the serial interface:

- Supports serial flash size up to 8M byte. Maximum 33MHz serial clock.
- SIF clock rate is programmable from system peripheral clock divided by any even number between 4 and 2^8.
- For serial flash, all common commands are supported.
- Supports Boot auto-load function.

3.13.6 SD / MMC Interface

AIT8427 supports a SD/MMC interface controller. The SD bus protocol is designed to be a super-set of the MMC protocol. The SD standard v1.0 is defined according to MMC specification v2.11 published by MMCA committee. SD v1.0 also has a configurable number of data bus signals. The number of data lines used for the data transfer can be either 1（DAT0）or 4（DAT0-DAT3）.

Features of the SD/MMC Interface:

- Support MMC 4.1 high-speed mode. Maximum data rate up to 416Mbits/s with 8-bit data bus
- Support SD2.0 spec with up to 32GB SD card capacity
- Support wide range of card clock frequency. The frequency can be clk_{system}, $1/3 \cdot clk_{system}$ or $\frac{1}{2(x+1)} \cdot clk_{system}$, where x is between 0 and 1023 in integer.
- Support FIFO mode and DMA mode
- Support clock control mechanism to avoid FIFO to be underflow or overflow.
- Support timer for response time, read access time and wait busy time.
- Automatic turn on/off the bus clock card used, and user can turn on/off clock manually.
- User can chose whether wait the last busy for write-card operation or not.
- Support block transfer. Block length can be 1 ~ 4096 bytes. One burst can be 1 ~ 65535 blocks.
- Support start bit error check.
- 5s timeout at 52MHz system clock
- Support 1-bit mode interrupt
- Support 4-bit mode interrupt with single/multiple data transaction

3.13.7 UART Interface

One UART interface is supported in AIT8427 for firmware development, debugging and controlling other external devices. The UART can support any baud rate if the internal system clock rate is high enough and the calculated baud rate variation met user's requirement. (PC standard baud rate is 2400/4800/9600/19.2k/38.4k/57.6k/115.2k/230.4k /460.8K/921.6K). For example, if the system clock is 100 MHz and the baud rate is 7 MHz, then the "Baud rate divider" must be set to 100M/7M= 14.28….= 14. The baud rate variation is about 0.5(0.28) cycle time=5ns, the require baud period is about 14.28x10 = 142.8ns. So baud rate variation = 5/143 = 3.49%(0.28/14=2%).

Features of the UART:

- 32-byte RX FIFO and 128-byte TX FIFO are employed to reduce the interrupt frequency.
- Programmable FIFO thresholds determined by firmware to adjust the frequency of interrupts.
- Interrupt enable/disable control can unmask/mask the trigger of interrupts. If all interrupts are disabled, CPU can still poll the interrupt status flags.
- The range of the TX/RX baud rates can be supported from 9600 bps to 7.3728 Mbps (Error rate = 2.8 %, based on 133 MHz system clock).
- No/Odd/Even/High/Low parity check schemes are provided to verify the correctness of data communications.
- Support CTS/RTS protocol for flow control.
- Support RX time out interrupt, if no data is received when UART RX is enabled after the timer expires.
- Support DMA mode to reduce the workload of CPU. The UART can move data to memory and read data from memory directly.
- Support programmable start address and total byte count for TX function to read data from memory in DMA mode.
- Support programmable start address and total byte threshold for RX to write data to memory in DMA mode.

The UART can also support CTS/RTS protocol, which can be used for flow control. The UART supports two modes of CTS/RTS handshaking.

Mode0:

Originally defined in RS-232, the DTE (host) asserts RTS to indicate a desire to transmit to the DCE (peripheral), and the DCE asserts CTS in response to grant permission. There is no way for the DTE to indicate that it is unable to accept data from the DCE.

Mode1:

A non-standard symmetric alternative, commonly called "RTS/CTS handshaking". CTS indicates permission from the DCE for the DTE to send data to the DCE (and is controlled by

the DCE independent of RTS), and RTS indicates permission from the DTE for the DCE to send data to the DTE.

The below table shows the meaning of RTS signal of UART in each mode.

		RTS active
Host	Mode 0	TX_FIFO_NOT_EMPTY
	Mode 1	RX_FIFO_NOT_FULL
Peripheral	Mode 0	RX_FIFO_NOT_FULL
	Mode 1	RX_FIFO_NOT_FULL

3.13.8 JTAG Interface

A standard 6-pin JTAG interface is supported for firmware development and debugging. Those pins are multiplexed with BGPIO[10:15], and also can be configured as optional I2C master and SPI interface.

Mutiplexed pins for JTAG Interface		
Function 1	Function 2	Function 3
BGPIO10	I2CM_SCL	ARM_TCK
BGPIO11	I2CM_SDA	ARM_TMS
BGPIO12	SPI_CLK	ARM_TDI
BGPIO13	SPI_CS_	ARM_TRST
BGPIO14	SPI_DO	ARM_TDO
BGPIO15	SPI_DI	ARM_RTCK

3.13.9 PWM Interface

AIT8427 supports very flexible Pulse Width Modulation (PWM) outputs which can generate waveforms with various frequency and duty cycle selections. The figure below shows a typical waveform from the PWM generator. Up to 4 PWM output pins can be supported.

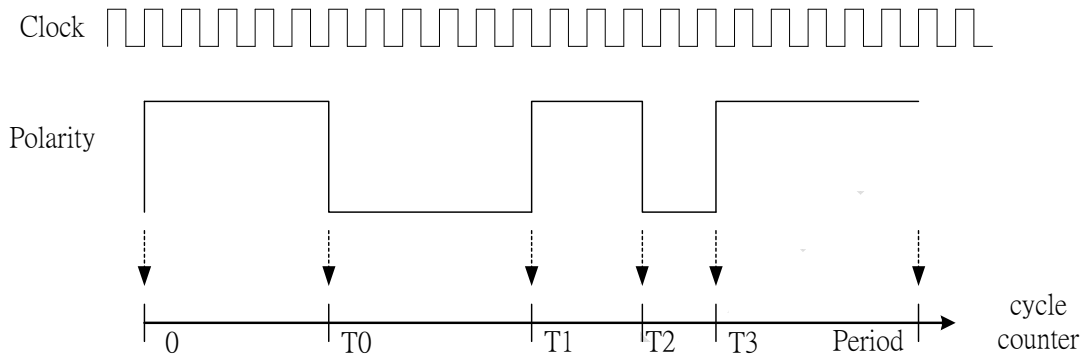


Figure 19 Typical PWM Output Waveform

In the PWM generator design, two sets of pulse (Pulse A and Pulse B) can be controlled by the firmware. This allows the user to generate the desired waveforms by controlling the combination of the two pulses. As the figure below shows, “Precedence” determines that pulse A or pulse B will be propagated first. “NA” is the number of pulse A, and “NB” is the number of pulse B. If NA≠NB, remaining pulse A or pulse B will be asymmetrically propagated. We call the combination of “NA” and “NB” as one round.

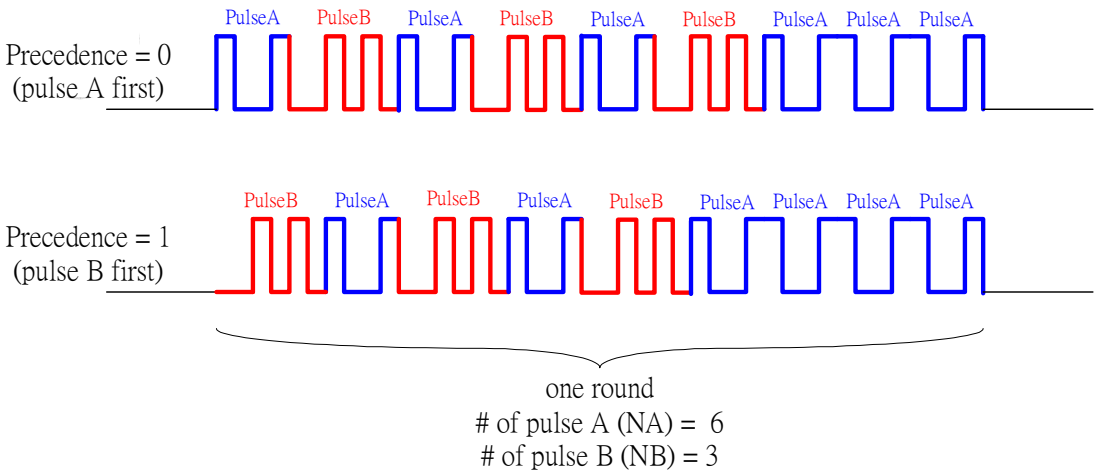


Figure 20 Pulse combination

Features of the PWM generator:

- Two sets of pulse configuration (Pulse A, Pulse B)
- Programmable Period / T0 / T1 / T2 / T3 (16 bits) for each pulse configuration.
- Positive / Negative polarity

- Various pulse combinations for each PWM output
- Pulse A and Pulse B are alternatively propagated.
- Programmable precedence of Pulse A and Pulse B
- Programmable number of pulse periods (NA, NB)
- One round mode : Set PWM Enable to trigger one round.
- Auto-cyclic mode: One round pulse can be repeatedly generated until auto-cyclic mode is disabled.
- Interrupts can be generated

4 Electrical Characteristics

4.1 Absolute Maximum Ratings

Parameter	Symbol	Min	Max	Units
Supply Core Voltage	VDDmax	-0.3	1.8	V
Supply Voltage (PLL)	$AVDD_{PLL}$	-0.3	1.8	V
Supply Voltage (DRAM 1.2V domain)	VDD_DDR_1V2	-0.3	1.8	V
Supply Voltage (USB 1.2V domain)	VDD_USB_1V2	-0.3	1.8	V
Supply Voltage (DRAM)	VDD_{DRAM}	-1.0	2.4	V
Supply IO Voltage	VDDIOmax	-0.5	3.75	V
Supply Voltage (Audio ADC/DAC)	$AVDD_{ADC}$	-0.5	3.75	V
Supply Voltage (USB)	VDD_{USB}	-0.5	3.75	V
IO Signal Voltage	VIOmax	-0.5	VDDIO[1]+0.3	V
ESD (human body mode)	ESD-HBM	<-2.0	>2.0	KV
ESD (machine mode)	ESD-MM	<-200	>200	V
Latch-Up		<-100	>100	mA
Storage Temperature	Tstorage	-40	125	℃

Table 2. Absolute Maximum Ratings

[1] The voltage depends on different power group

* Permanent device damage may occur if the absolute maximum ratings are exceeded

4.2 DC Recommended Operating Conditions

Parameter	Symbols	Min.	Typ.	Max.	Unit	Note
Supply Voltage (Core)	VDD_{core}	1.0	1.2	1.5	V	
Supply Voltage (DRAM 1.2V domain)	VDD_DDR_1V2	1.0	1.2	1.5	V	
Supply Voltage (USB 1.2V domain)	VDD_USB_1V2	1.0	1.2	1.5	V	

Parameter	Symbols	Min.	Typ.	Max.	Unit	Note
Supply Voltage (PLL)	$AVDD_{PLL}$	1.0	1.2	1.5	V	
Supply Voltage (PMCLK)	$AVDD_{CLK}$	1.6	1.8	2.0	V	
Supply Voltage (GPIOs)	VDD_{GPIO}	1.6		3.6	V	
Supply Voltage (Sensor)	VDD_{IO_SENOR}	1.6		3.6	V	
Supply Voltage (LCD)	VDD_{IO_LCD}	1.6		3.6	V	
Supply Voltage (I2S)	VDD_{IO_I2S}	1.6		3.6	V	
Supply Voltage (SIF)	VDD_{IO_SIF}	1.6		3.6	V	
Supply Voltage (DRAM I/O)	VDD_{DRAM}	1.6	1.8	2.0	V	
Supply Voltage (USB)	VDD_{USB}	3.0	3.3	3.6	V	
Supply Voltage (Audio ADC/DAC)	$AVDD_{AUDIO}$	1.7	2.8	3.3	V	
Supply Voltage (Audio MIC)	$AVDD_{MIC}$	1.7	2.8	3.3	V	
HDMI PLL Power VDD12_ANA	VDD_{12ANA}	1.08	1.2	1.32	V	
HDMI TX Power VDD12_TMDS	VDD_{12TMDS}	1.08	1.2	1.32	V	
HDMI Analog Power VDD33_ANA	VDD_{33ANA}	3.0	3.3	3.6	V	
Input Low Voltage	V_{IL}			0.6	V	
Input High Voltage	V_{IH}	$0.7VDD_{IO}$			V	
Output Low Voltage	V_{OL}			0.4	V	
Output High Voltage	V_{OH}	VDD_{IO}–0.4			V	
Operating Temperature	T_A	-40		80	℃	
Standby Current(core)	T_{STB}		TBD		uW	

Table 3. DC Recommended Operating Condition

Electrical Characteristics of Audio CODEC

PARAMETER	SYMBOL	TEST CONDITION	MIN	TYP	MAX	Unit
VREF pin decoupling capacitor				10		uF
DAC						
Full-Scale Output	0dBFS			AVDD/3.75		V_{RMS}
Input Clock Frequency	Fs*2		2.048		24.576	MHz
Signal-to-Noise Ratio	SNR	Full-scale input, A-weighted		95*		dB
Total Harmonic Distortion	THD	-6dBFS input		75		dB
Power supply rejection	PSR	1KHz, 30mVpp		60		dB
DAC PGA						
Gain Range			-42		6	dB

Gain Step			+6dB ~ -6dB: 2dB per step -6dB ~ -30dB: 4dB per step -30dB ~ -42dB: 6dB per step			dB
Headphone Amplifier						
Signal-to-Noise Ratio	SNR			95**		dB
Total Harmonic Distortion	THD	Load = 16Ω, P_O = 5mW		65		dB
Maximum output power	$P_{O,MAX}$	Load = 16Ω//200pF		6		mW
Quiescent Current	I_q			500		uA
MAX C_{load} when R_{load} = 16Ω				200		pF
MAX C_{load} when R_{load} = 10KΩ				50		pF
ADC						
Full-Scale Input	0dBFS	Differential input		AVDD/2		V_{RMS}
Input Clock Frequency	Fs*2		4.096		24.576	MHz
Signal-to-Noise Ratio	SNR	Full-scale input, A-weighted		95		dB
Total Harmonic Distortion	THD			72		dB
Power supply rejection	PSR	1KHz, 30mVpp		60		dB
ADC PGA						
Gain Range			-10		20	dB
Gain Step				2		dB
Extra Gain Boost		MIC_BOOST = 00 MIC_BOOST = 01 MIC_BOOST = 10 MIC_BOOST = 11		0 20 30 40		dB
Input Resistance (single-ended)	R_{in}	MIC_BOOST = 00		TBD		Ω
		MIC_BOOST = 11		TBD		Ω
Microphone Bias						
Maximum output current	I_{MAX}			3		mA
Output voltage		MICBIAS_VOL=00 MICBIAS_VOL=01 MICBIAS_VOL=10 MICBIAS_VOL=11		0.65AVDD 0.75AVDD 0.95AVDD 1.15AVDD		V
Output noise (20 ~ 20kHz)	N_{out}	MICBIAS_VOL=01		TBD		uV
Dropout voltage	$V_{drop,MIN}$	MICVDD – MICBIAS		400		mV

* For lineout path; ** For headphone- out path.

4.3 AC Characteristics

4.3.1 Sensor Interface Timing

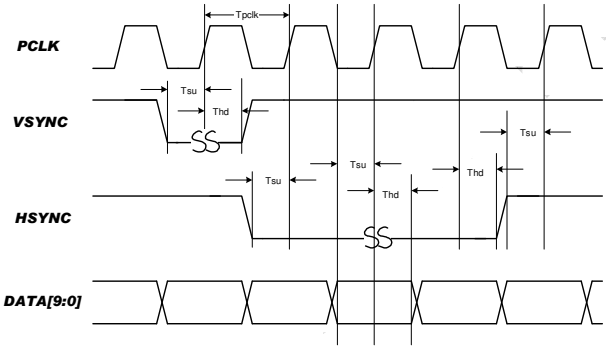


Figure 13. Sensor Pixel Timing Diagram

Symbol	Parameter	Min	Typ	Max	Unit
Tpclk	Pixel Clock period	8			ns
Tsu	VSYNC/HSYNC/DATA setup time	3			ns
Thd	VSYNC/HSYNC/DATA hold time	3			ns

Table 4. Sensor Pixel Timing Characteristics

4.3.2 LCD Interface Timing

4.3.2.1 68-system interface

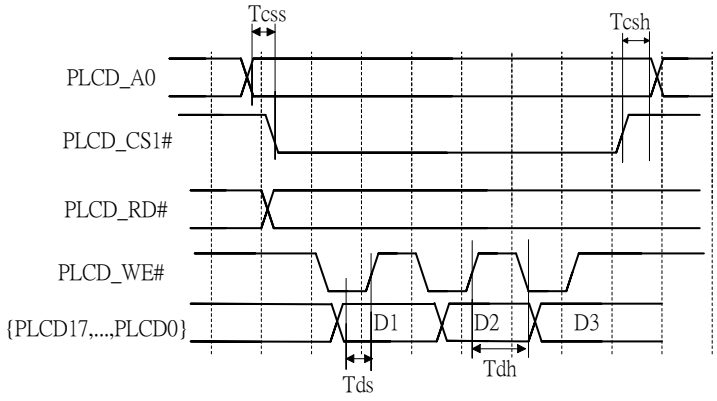


Figure 14. 68-system LCD Interface Timing Diagram

4.3.2.2 80-system interface

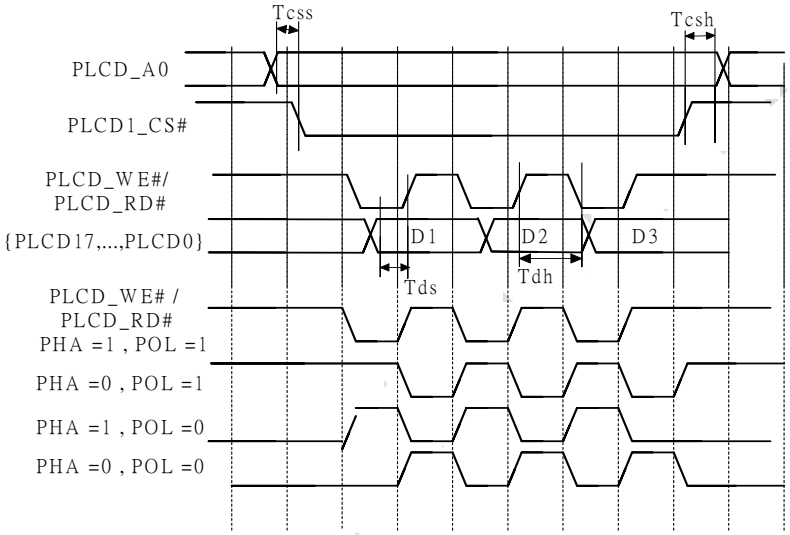


Figure 15. 80-system LCD Interface Timing Diagram

Symbol	Parameter	Min	Max	Unit
Tcss	Chip select setup time	10	-	ns
Tcsh	Chip select hold time	5	-	ns
Tds	Data bus setup time	20	-	ns
Tdh (write)	Write data bus hold time	15	-	ns
Tdh (read)	Read data bus hold time	5	-	ns

Table 5. LCD Timing Characteristics

4.4 Power Consumption

Parameter	Condition	TYP [VDDcore]	Unit	Note
Preview w/ ISP	QVGA @ 30fps	TBD	mW	
Capture w/ ISP	2M @ 15fps snap	TBD	mW	
Video encode w/ ISP	CIF @ 30fps with AAC encode	TBD	mW	
Video playback	CIF @ 30fps with AAC decode	TBD	mW	
Video encode w/ ISP	H.264 HD @ 30fps with AAC encode	TBD	mW	
Video playback	H.264 HD @ 30fps with AAC decode	TBD	mW	
Suspend	All clock including PMCLK is disabled	TBD	uW	