CLK Driver
User Guide
V1.00.01

CLK Driver

1.1. CLK introduction

The clock controller generates the clock sources for the whole device, including all AMBA interface modules and all peripheral clocks. Clock gating is provided on all peripheral clocks to minimize power consumption.

1.2. Type Definition

E_SYS_IP_DIV

Enumeration identifier	Value	Description
E_SYS_ADC_DIV	0	ADC source clock divider setting
E_SYS_UART_DIV	1	UART source clock divider setting
E_SYS_HCLK_DIV	2	HCLK source clock divider setting

E_SYS_IP_CLK

Enumeration identifier	Value	Description
E_SYS_WDG_CLK	4	Watch Dog Timer clock enable control
E_SYS_RTC_CLK	5	RTC clock enable control
E_SYS_TMR0_CLK	6	Timer0 clock enable control
E_SYS_TMR1_CLK	7	Timer1 clock enable control
E_SYS_I2C0_CLK	8	I2C0 clock enable control
E_SYS_SPI0_CLK	12	SPI0 clock enable control
E_SYS_DPWM_CLK	13	DPWM clock enable control
E_SYS_UART0_CLK	16	UART0 clock enable control
E_SYS_BIQ_CLK	18	BIQ clock enable control
E_SYS_CRC_CLK	19	CRC clock enable control
E_SYS_PWM01_CLK	20	PWM01 clock enable control
E_SYS_ACMP_CLK	22	ACMP clock enable control
E_SYS_SBRAM_CLK	26	SBRAM clock enable control

E_SYS_ADC_CLK	28	ADC clock enable control
E_SYS_I2S_CLK	29	I2S clock enable control
E_SYS_ANA_CLK	30	ANA clock enable control
E_SYS_PDMA_CLK	33	PDMA clock enable control
E_SYS_ISP_CLK	34	ISP clock enable control

E_SYS_IP_CLKSRC

Enumeration identifier	Value	Description
E_SYS_I2S_CLKSRC	0	I2S clock source setting
E_SYS_PWM01_CLKSRC	1	PWM01 clock source setting
E_SYS_UART_CLKSRC	2	UART clock source setting
E_SYS_TMR1_CLKSRC	3	TMR1 clock source setting
E_SYS_TMR0_CLKSRC	4	TMR0 clock source setting
E_SYS_ADC_CLKSRC	5	ADC clock source setting
E_SYS_WDG_CLKSRC	6	Watch Dog Timer clock source setting
E_SYS_DPWM_CLKSRC	7	DPWM clock source setting

E_SYS_OSC_CTRL

Enumeration identifier	Value	Description
E_SYS_NONE	0	---
E_SYS_XTL32K	1	External 32.768 KHz clock enable control
E_SYS_OSC49M	2	OSC49M clock enable control
E_SYS_OSC10K	3	OSC10K clock enable control

1.3. Functions

DrvSYS_GetExtClock

Prototype

uint32_t DrvSYS_GetEXTClock(void)

Description

To get external clock frequency. The clock UNIT is in kHz.

Parameter

None

Include

Driver/ DrvSYS.h

Return Value

The external clock frequency.

Example

```
uint32_t u32clock;
/* Get external crystal clock frequency */
u32clock = DrvSYS_GetExtClock ( );
```

DrvSYS_GetHCLK

Prototype

uint32_t DrvSYS_GetHCLK(void)

Description

To get HCLK clock and the UNIT is in kHz.

Parameter

None

Include

Driver/ DrvSYS.h

Return Value

HCLK clock frequency in kHz.

Example

```
uint32_t u32clock;
/* Get current HCLK clock */
u32clock = DrvSYS_GetHCLK ( );
```

DrvSYS_ SetClockDivider

Prototype

int32_t DrvSYS_SetClockDivider(E_SYS_IP_DIV eIpDiv , int32_t i32value)

Description

To set IP divider number from the corresponding clock source.

Parameter

eIpDiv [in]

It could be E_SYS_ADC_DIV / E_SYS_CAN_DIV / E_SYS_UART_DIV / E_SYS_USB_DIV /E_SYS_HCLK_DIV

i32Data [in]

0 ~ 255 for ADC, 0 ~ 15 for other IP or HCLK.

Include

Driver/ DrvSYS.h

Return Value

E_SUCCESS: Operation successful.

E_DRVSYS_ERR_IPDIV: Incorrect arguments

Example

```
DrvSYS_SetClockDivider(E_SYS_ADC_DIV, 0x55);
/* Set ADC clock divide number 0x01; ADC clock = ADC source clock / (1+1) */
DrvSYS_SetClockDivider (E_SYS_ADC_DIV, 0x01);
/* Set UART clock divide number 0x02; UART clock = UART source clock / (2+1) */
DrvSYS_SetClockDivider (E_SYS_UART_DIV, 0x02);
/* Set HCLK clock divide number 0x03; HCLK clock = HCLK source clock / (3+1) */
DrvSYS_SetIPClockSource (E_SYS_HCLK_DIV, 0x03);
```

DrvSYS_ SetHCLKSource

Prototype

int32_t DrvSYS_SetHCLKSource(uint8_t u8ClkSrcSel)

Description

To select HCLK clock source.

Parameter

u8ClkSrcSel [in]

It could be 0: Internal 48M clock;

1: External 32K clock;

2: Internal 10K clock.

Include

Driver/ DrvSYS.h

Return Value

E_SUCCESS: Operation successful.

E_DRVSYS_ERR_IPDIV: Incorrect arguments

Example

DrvSYS_SetHCLKSource(0);

DrvSYS_SetIPClock

Prototype

void DrvSYS_SetIPClock(E_SYS_IP_CLK eIpClk, int32_t i32Enable);

Description

To enable/disable coresponding IP clock.

Parameter

eIpClk [in]

It could be E_SYS_WDG_CLK,E_SYS_RTC_CLK, E_SYS_TMR0_CLK, E_SYS_TMR1_CLK, E_SYS_I2C0_CLK, E_SYS_SPI0_CLK, E_SYS_DPWM_CLK, E_SYS_UART0_CLK, E_SYS_BIQ_CLK, E_SYS_CRC_CLK, E_SYS_PWM01_CLK, E_SYS_ACMP_CLK, E_SYS_SBRAM_CLK, E_SYS_ADC_CLK, E_SYS_I2S_CLK, E_SYS_ANA_CLK, E_SYS_PDMA_CLK, E_SYS_ISP_CLK.

i32Enable [in]

1 : Enable / 0 : Disable.

Include

Driver/ DrvSYS.h

Return Value

None

Example

```
/* Enable I2C0 engine clock */
DrvSYS_SetIPClock (E_SYS_I2C0_CLK, 1);
/* Disable I2C0 engine clock */
DrvSYS_SetIPClock (E_SYS_I2C0_CLK, 0);
/* Enable SPI0 engine clock */
DrvSYS_SetIPClock (E_SYS_SPI0_CLK, 1);
/* Disable SPI0 engine clock */
DrvSYS_SetIPClock (E_SYS_SPI0_CLK, 0);
/* Enable TIMER0 engine clock */
DrvSYS_SetIPClock (E_SYS_TMR0_CLK, 1);
/* Disable TIMER0 engine clock */
DrvSYS_SetIPClock (E_SYS_TMR0_CLK, 0);
```

DrvSYS_ SetIPClockSource

Prototype

int32_t DrvSYS_SetIPClockSource(E_SYS_IP_CLKSRC eIpClkSrc, uint8_t u8ClkSrcSel)

Description

To select IP clock source.

Parameter

eIpClkSrc [in]

It could be E_SYS_I2S_CLKSRC, E_SYS_PWM01_CLKSRC, E_SYS_TMR1_CLKSRC, E_SYS_TMR0_CLKSRC, E_SYS_DPWM_CLKSRC, E_SYS_WDG_CLKSRC

u8ClkSrcSel [in]

Corresponding clock source.

Include

Driver/ DrvSYS.h

Return Value

E_SUCCESS: Operation successful.

E_DRVSYS_ERR_ IPSRC: Incorrect arguments

Example

DrvSYS_SetIPClockSource(E_SYS_I2S_CLKSRC, 1);

DrvSYS_Set SysTickSource

Prototype

int32_t DrvSYS_SetSysTickSource(uint8_t u8ClkSrcSel)

Description

To select sysyem tick clock source.

Parameter

u8ClkSrcSel [in]

It could be 0: Internal 10K clock;

1: External 32K clock;

2: Internal 10K clock / 2;

3: Internal 48M clock;

4~7: HCLK / 2

Include

Driver/ DrvSYS.h

Return Value

E_SUCCESS: Operation successful.

E_DRVSYS_ERR_ ARGUMENT: Incorrect arguments

Example

DrvSYS_SetSysTickSource(1);

DrvSYS_SetOscCtrl

Prototype

int32_t DrvSYS_SetOscCtrl(E_SYS_OSC_CTRL eOscCtrl, int32_t i32Enable)

Description

To enable or disable internal oscillator and external crystal include internal 10K and 49M oscillator, or external 32K crystal.

Parameter

eOscCtrl [in]

E_SYS_XTL32K / E_SYS_OSC49M / E_SYS_OSC10K.

i32Enable [in]

1: enable, 0: disable

Include

Driver/DrvSYS.h

Return Value

0	Succeed
< 0	incorrect parameter

Example

```
DrvSYS_SetOscCtrl (E_SYS_XTL32K, 1);    /* Enable external 32K */
```

DrvSYS _GetVersion

Prototype

uint32_t DrvSYS_GetVersion (void)

Description

This function is used to return the version number of DrvSYS.

Include

Driver/ DrvSYS.h

Return Value

The version number of SYS:

31:24	23:16	15:8	7:0

00000000	MAJOR_NUM	MINOR_NUM	BUILD_NUM

Example

```
/* Get the current version of SYS */
uint32_t u32SYSVer;
u32SYSVer = DrvSYS_GetVersion ();
```

2. Revision History

Version	Date	Description
1.00.01	Mar. 2011	Preliminary CLK Driver User Guide of ISD9160