EIE/ENE 334 Microprocessors

Lecture 12:

NuMicro™ NUC140

Week #12 : Dejwoot KHAWPARISUTH

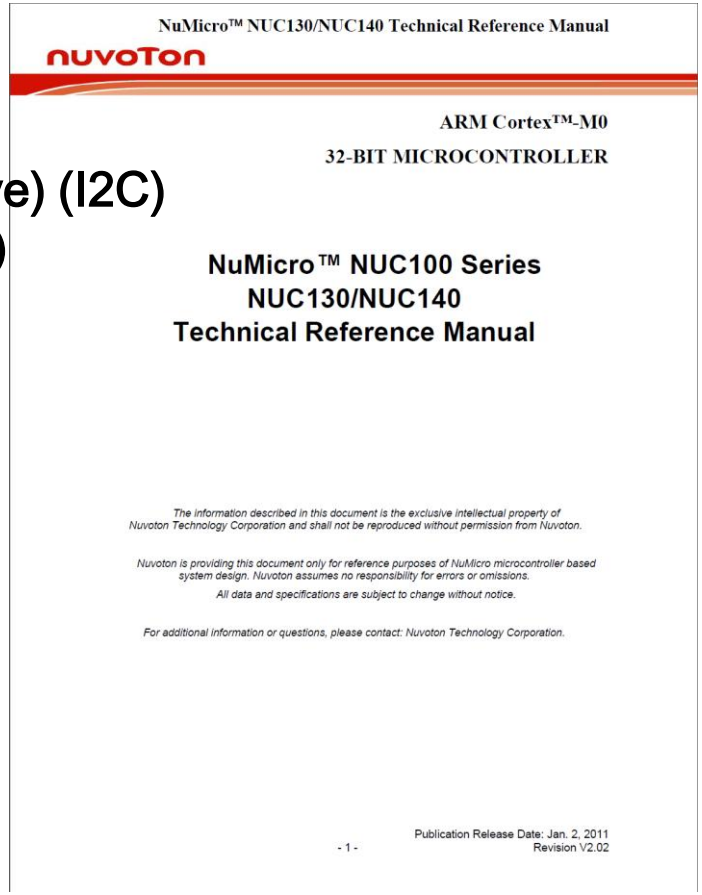
Adapted from

http://webstaff.kmutt.ac.th/~dejwoot.kha/

NuMicro™ NUC140: Technical Ref.

NuMicro™ NUC130/NUC140 Technical Reference Manual

5 FUNCTIONAL DESCRIPTION
5.1 ARM® Cortex™-M0 Core
5.2 System Manager
5.3 Clock Controller
5.5 General Purpose I/O (GPIO)
5.6 I2C Serial Interface Controller (Master/Slave) (I2C)
5.7 PWM Generator and Capture Timer (PWM)
5.10 Timer Controller (TMR)
5.11 Watchdog Timer (WDT)


NuMicro™ NUC130/NUC140 Technical Reference Manual

nuvoTon

ARM Cortex™-M0

32-BIT MICROCONTROLLER

NuMicro™ NUC100 Series
NUC130/NUC140
Technical Reference Manual

The information described in this document is the exclusive intellectual property of Nuvoton Technology Corporation and shall not be reproduced without permission from Nuvoton.

Nuvoton is providing this document only for reference purposes of NuMicro microcontroller based system design. Nuvoton assumes no responsibility for errors or omissions.

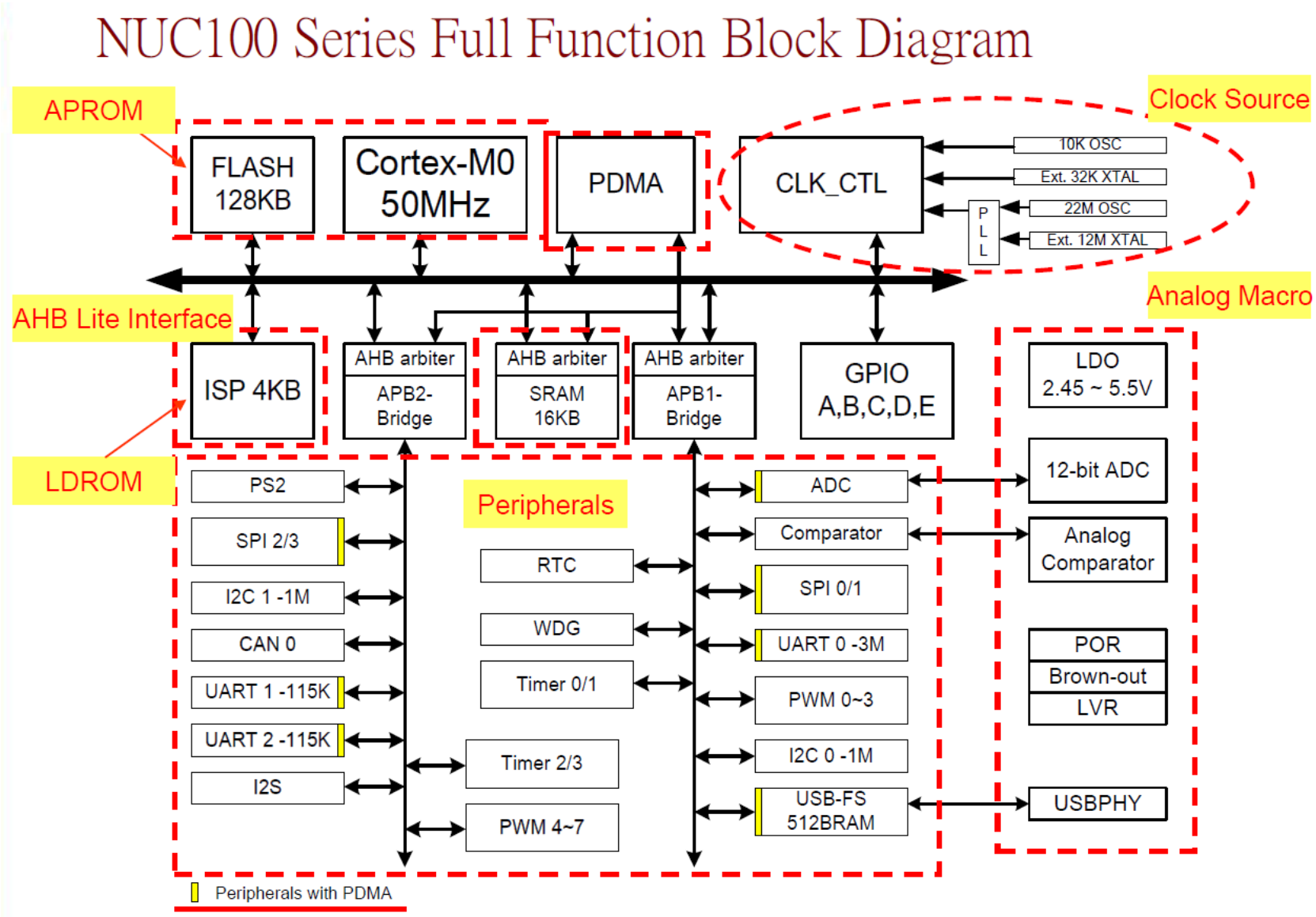
All data and specifications are subject to change without notice.

For additional information or questions, please contact: Nuvoton Technology Corporation.

- 1 -

Publication Release Date: Jan. 2, 2011
Revision V2.02

NuMicro™ NUC140: Block Diagram



NuMicro™ NUC140: Memory Map

Address Space	Token	Controllers
Flash and SRAM Memory Space		
0x0000_0000 – 0x0001_FFFF	FLASH_BA	FLASH Memory Space (128KB)
0x2000_0000 – 0x2000_3FFF	SRAM_BA	SRAM Memory Space (16KB)
0x6000_0000 – 0x6001_FFFF	EXTMEM_BA	External Memory Space (128KB)
AHB Controllers Space (0x5000_0000 – 0x501F_FFFF)		
0x5000_0000 – 0x5000_01FF	GCR_BA	System Global Control Registers
0x5000_0200 – 0x5000_02FF	CLK_BA	Clock Control Registers
0x5000_0300 – 0x5000_03FF	INT_BA	Interrupt Multiplexer Control Registers
0x5000_4000 – 0x5000_7FFF	GPIO_BA	GPIO Control Registers
0x5000_8000 – 0x5000_BFFF	PDMA_BA	Peripheral DMA Control Registers
0x5000_C000 – 0x5000_FFFF	FMC_BA	Flash Memory Control Registers
0x5001_0000 – 0x5001_03FF	EBI_BA	External Bus Interface Control Registers

NuMicro™ NUC140: Memory Map

APB1 Controllers Space (0x4000_0000 ~ 0x400F_FFFF)		
0x4000_4000 – 0x4000_7FFF	WDT_BA	Watchdog Timer Control Registers
0x4000_8000 – 0x4000_BFFF	RTC_BA	Real Time Clock (RTC) Control Register
0x4001_0000 – 0x4001_3FFF	TMR01_BA	Timer0/Timer1 Control Registers
0x4002_0000 – 0x4002_3FFF	I2C0_BA	I^2C0 Interface Control Registers
0x4003_0000 – 0x4003_3FFF	SPI0_BA	SPI0 with master/slave function Control Registers
0x4003_4000 – 0x4003_7FFF	SPI1_BA	SPI1 with master/slave function Control Registers
0x4004_0000 – 0x4004_3FFF	PWMA_BA	PWM0/1/2/3 Control Registers
0x4005_0000 – 0x4005_3FFF	UART0_BA	UART0 Control Registers
0x4006_0000 – 0x4006_3FFF	USBD_BA	USB 2.0 FS device Controller Registers
0x400D_0000 – 0x400D_3FFF	ACMP_BA	Analog Comparator Control Registers
0x400E_0000 – 0x400E_FFFF	ADC_BA	Analog-Digital-Converter (ADC) Control Registers

NuMicro™ NUC140: Memory Map

APB2 Controllers Space (0x4010_0000 ~ 0x401F_FFFF)		
0x4010_0000 – 0x4010_3FFF	PS2_BA	PS/2 Interface Control Registers
0x4011_0000 – 0x4011_3FFF	TMR23_BA	Timer2/Timer3 Control Registers
0x4012_0000 – 0x4012_3FFF	I2C1_BA	I^2C1 Interface Control Registers
0x4013_0000 – 0x4013_3FFF	SPI2_BA	SPI2 with master/slave function Control Registers
0x4013_4000 – 0x4013_7FFF	SPI3_BA	SPI3 with master/slave function Control Registers
0x4014_0000 – 0x4014_3FFF	PWMB_BA	PWM4/5/6/7 Control Registers
0x4015_0000 – 0x4015_3FFF	UART1_BA	UART1 Control Registers
0x4015_4000 – 0x4015_7FFF	UART2_BA	UART2 Control Registers
0x4018_0000 – 0x4018_3FFF	CAN0_BA	CAN0 Bus Control Registers
0x401A_0000 – 0x401A_3FFF	I2S_BA	I^2S Interface Control Registers
System Controllers Space (0xE000_E000 ~ 0xE000_EFFF)		
0xE000_E010 – 0xE000_E0FF	SCS_BA	System Timer Control Registers
0xE000_E100 – 0xE000_ECFF	SCS_BA	External Interrupt Controller Control Registers
0xE000_ED00 – 0xE000_ED8F	SCS_BA	System Control Registers

NuMicro™ NUC140: System Manager Control register

Register	Offset	R/W	Description	Reset Value
GCR_BA = 0x5000_0000				
PDID	GCR_BA+0x00	R	Part Device Identification Number Register	0x0014_0018[1]
RSTSRC	GCR_BA+0x04	R/W	System Reset Source Register	0x0000_00XX
IPRSTC1	GCR_BA+0x08	R/W	IP Reset Control Register1	0x0000_0000
IPRSTC2	GCR_BA+0x0C	R/W	IP Reset Control Register2	0x0000_0000
BODCR	GCR_BA+0x18	R/W	Brown-Out Detector Control Register	0x0000_008X
TEMPCR	GCR_BA+0x1C	R/W	Temperature Sensor Control Register	0x0000_0000
PORCR	GCR_BA+0x24	R/W	Power-On-Reset Controller Register	0x0000_00XX
GPA_MFP	GCR_BA+0x30	R/W	GPIOA Multiple Function and Input Type Control Register	0x0000_0000
GPB_MFP	GCR_BA+0x34	R/W	GPIOB Multiple Function and Input Type Control Register	0x0000_0000
GPC_MFP	GCR_BA+0x38	R/W	GPIOC Multiple Function and Input Type Control Register	0x0000_0000
GPD_MFP	GCR_BA+0x3C	R/W	GPIOD Multiple Function and Input Type Control Register	0x0000_0000
GPE_MFP	GCR_BA+0x40	R/W	GPIOE Multiple Function and Input Type Control Register	0x0000_0000
ALT_MFP	GCR_BA+0x50	R/W	Alternative Multiple Function Pin Control Register	0x0000_0000
REGWRPROT	GCR_BA+0x100	R/W	Register Write Protect register	0x0000_0000

Note: [1] Dependents on part number.

Part Device ID Code Register (PDID)

Register	Offset	R/W	Description	Reset Value
PDID	GCR_BA+0x00	R	Part Device Identification Number Register	0x0014_0018[1]

[1] Each part number has a unique default reset value.

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

Bits	Descriptions	
[31:0]	PDID	**Part Device Identification Number** This register reflects device part number code. S/W can read this register to identify which device is used.

Clock Controller:

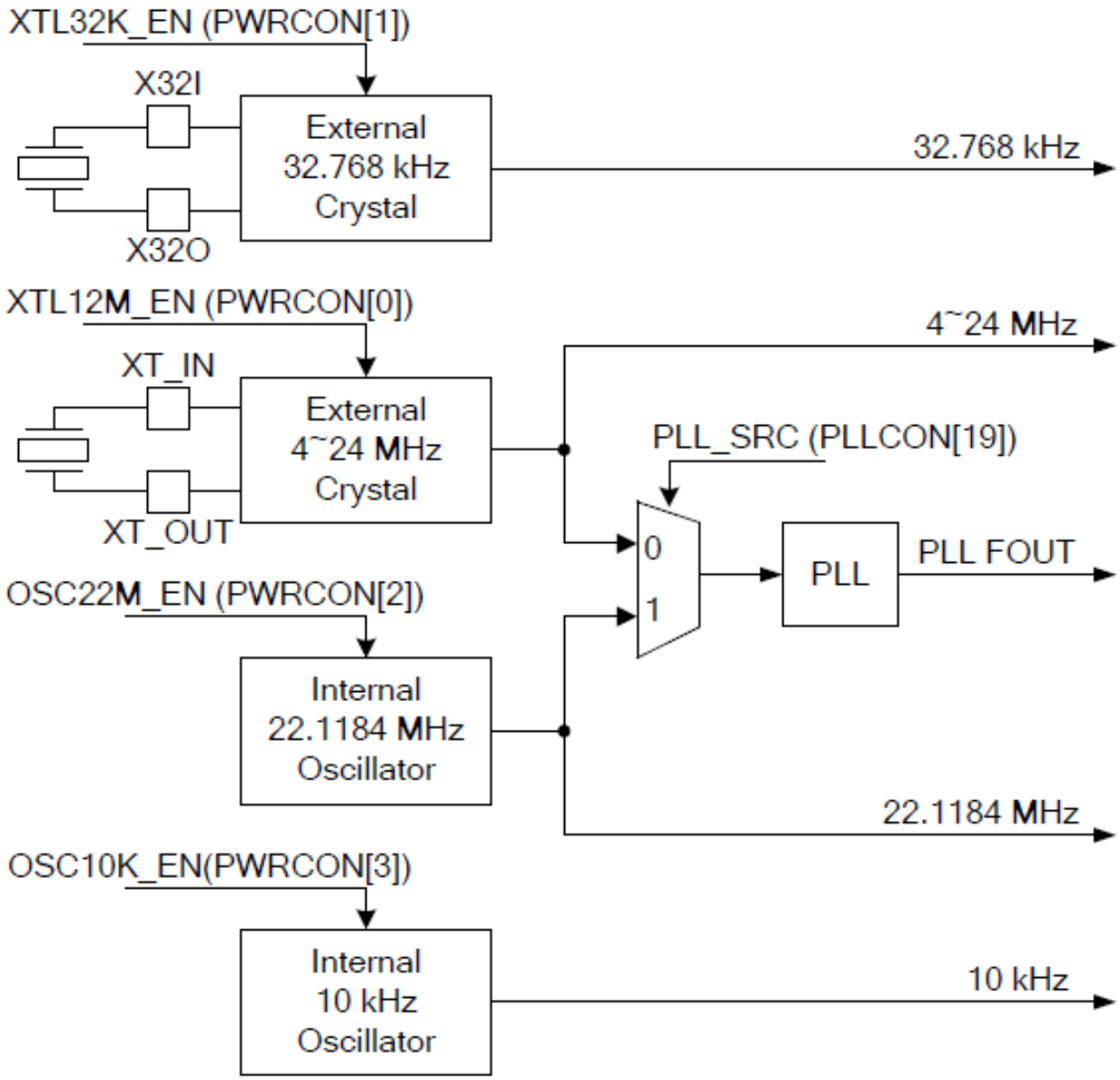
- generates the clocks for the whole chip
- implements the power control function with the individually clock ON/OFF control, clock source selection and a clock divider
- To enter power down mode
  - sets the power down enable bit (PWR_DOWN_EN)
  - Cortex-M0 core executes the WFI instruction

  (clock controller turns off the external 4~24 MHz high speed crystal and internal 22.1184 MHz high speed oscillator to reduce the overall system power consumption)

Clock Controller: Register Map

Register	Offset	R/W	Description	Reset Value
CLK_BA = 0x5000_0200				
PWRCON	CLK_BA+0x00	R/W	System Power Down Control Register	0x0000_001X
AHBCLK	CLK_BA+0x04	R/W	AHB Devices Clock Enable Control Register	0x0000_000D
APBCLK	CLK_BA+0x08	R/W	APB Devices Clock Enable Control Register	0x0000_000X
CLKSTATUS	CLK_BA+0x0C	R/W	Clock status monitor Register	0x0000_00XX
CLKSEL0	CLK_BA+0x10	R/W	Clock Source Select Control Register 0	0x0000_003X
CLKSEL1	CLK_BA+0x14	R/W	Clock Source Select Control Register 1	0xFFFF_FFFF
CLKSEL2	CLK_BA+0x1C	R/W	Clock Source Select Control Register 2	0x0000_00FF
CLKDIV	CLK_BA+0x18	R/W	Clock Divider Number Register	0x0000_0000
PLLCON	CLK_BA+0x20	R/W	PLL Control Register	0x0005_C22E
FRQDIV	CLK_BA+0x24	R/W	Frequency Divider Control Register	0x0000_0000

Clock Controller: Clock Generator



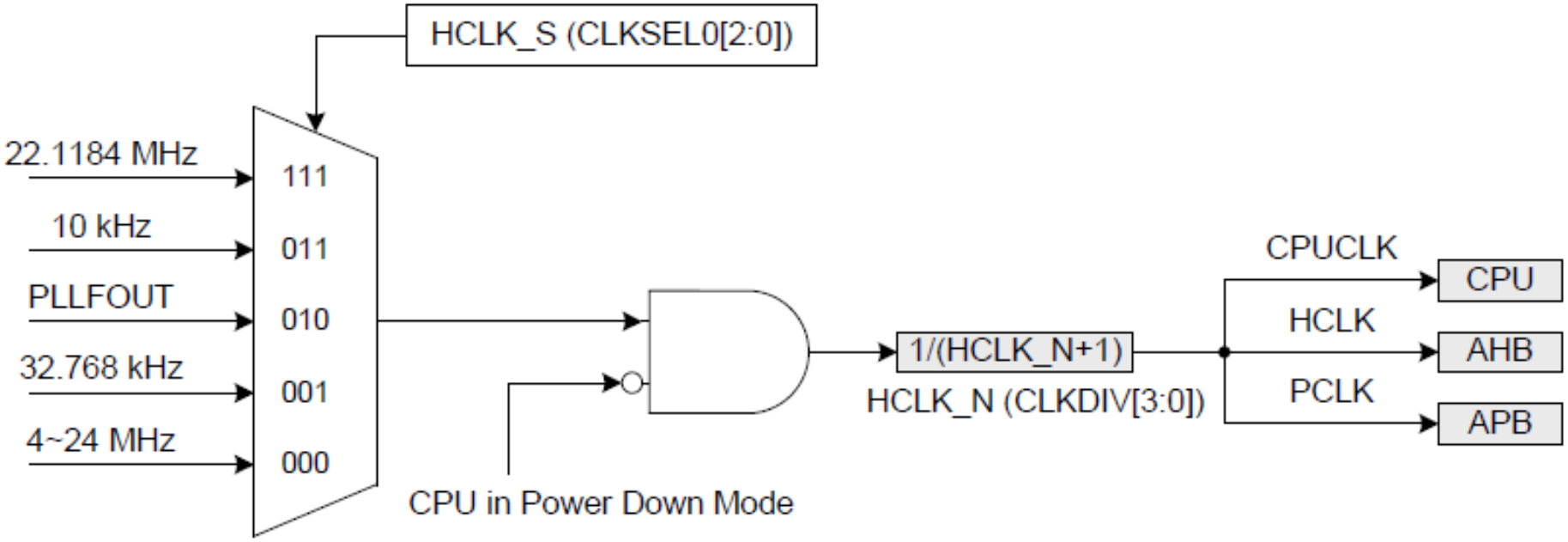
Clock Controller: Clock Generator

Power Down Control Register (PWRCON)

Except the BIT[6], all the other bits are protected, program these bits need to write “59h”, “16h”, “88h” to address 0x5000_0100 to disable register protection. Reference the register REGWRPROT at address GCR_BA+0x100

[3]	**OSC10K_EN**	**Internal 10 kHz Low Speed Oscillator Enable** (write-protection bit) 1 = Enable internal 10 kHz low speed oscillator 0 = Disable internal 10 kHz low speed oscillator
[2]	**OSC22M_EN**	**Internal 22.1184 MHz High Speed Oscillator Enable** (write-protection bit) 1 = Enable internal 22.1184 MHz high speed oscillator 0 = Disable internal 22.1184 MHz high speed oscillator
[1]	**XTL32K_EN**	**External 32.768 kHz Low Speed Crystal Enable** (write-protection bit) 1 = Enable external 32.768 kHz low speed crystal (Normal operation) 0 = Disable external 32.768 kHz low speed crystal
[0]	**XTL12M_EN**	**External 4~24 MHz High Speed Crystal Enable** (write-protection bit) The bit default value is set by flash controller user configuration register config0 [26:24]. When the default clock source is from external 4~24 MHz high speed crystal, this bit is set to 1 automatically 1 = Enable external 4~24 MHz high speed crystal 0 = Disable external 4~24 MHz high speed crystal

Clock Controller: System Clock



[2:0]	**HCLK_S**	**HCLK clock source select** (write-protection bits) 1. Before clock switching, the related clock sources (both pre-select and new-select) must be turn on 2. The 3-bit default value is reloaded from the value of CFOSC (Config0[26:24]) in user configuration register of Flash controller by any reset. Therefore the default value is either 000b or 111b. 3. These bits are protected bit, It means programming this bit needs to write "59h", "16h", "88h" to address 0x5000_0100 to disable register protection. Reference the register REGWRPROT at address GCR_BA+0x100. 000 = Clock source from external 4~24 MHz high speed crystal clock 001 = Clock source from external 32.768 kHz low speed crystal clock 010 = Clock source from PLL clock 011 = Clock source from internal 10 kHz low speed oscillator clock 111 = Clock source from internal 22.1184 MHz high speed oscillator clock

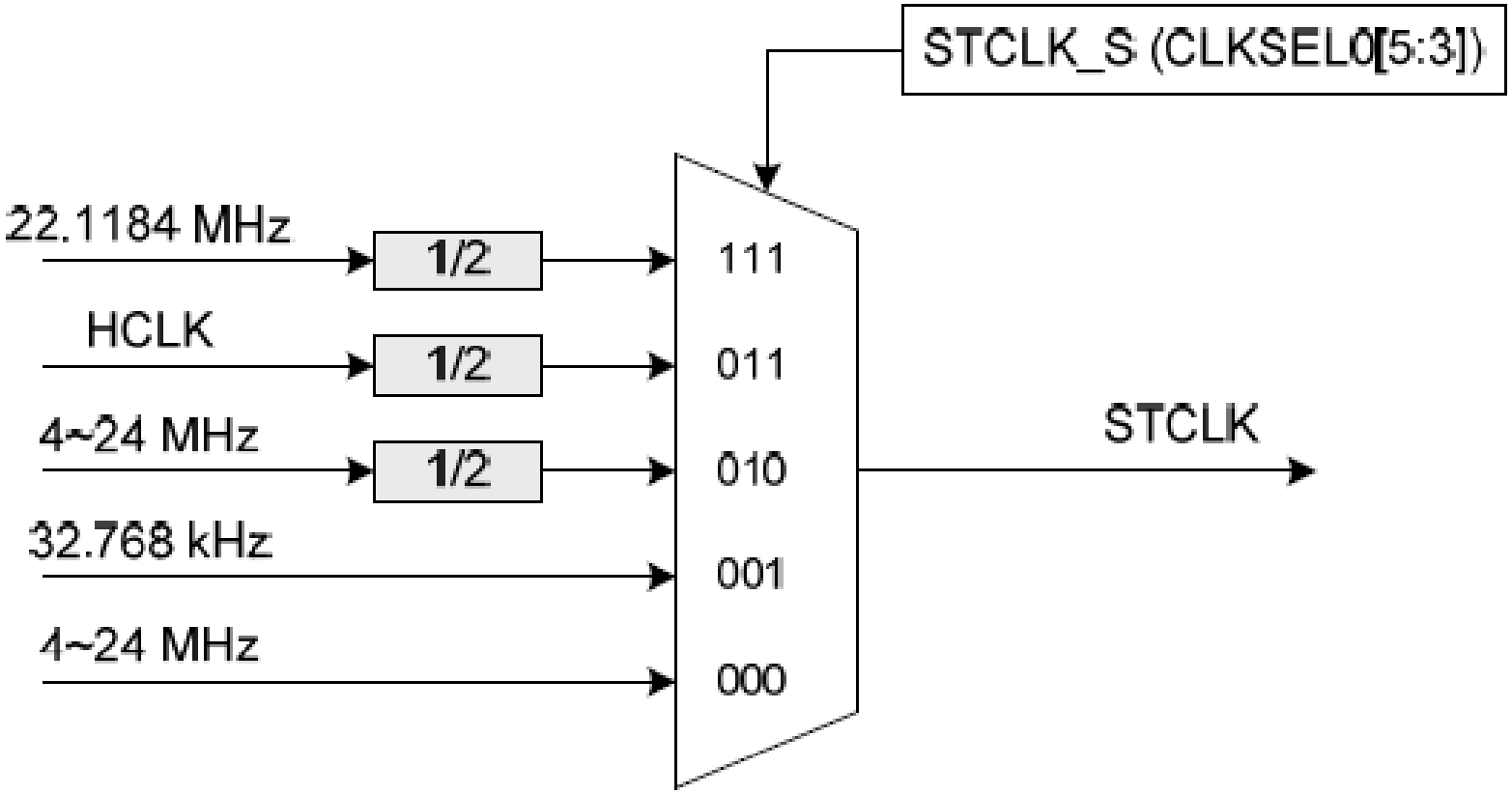
Clock Controller: System Clock

Clock Divider Register (CLKDIV)

Register	Offset	R/W	Description	Reset Value
CLKDIV	CLK_BA+0x18	R/W	Clock Divider Number Register	0x0000_0000

Bits	Descriptions	
[31:24]	**Reserved**	Reserved
[23:16]	**ADC_N**	**ADC clock divide number from ADC clock source** The ADC clock frequency = (ADC clock source frequency) / (ADC_N + 1)
[15:12]	**Reserved**	Reserved
[11:8]	**UART_N**	**UART clock divide number from UART clock source** The UART clock frequency = (UART clock source frequency) / (UART_N + 1)
[7:4]	**USB_N**	**USB clock divide number from PLL clock** The USB clock frequency = (PLL frequency) / (USB_N + 1)
[3:0]	**HCLK_N**	**HCLK clock divide number from HCLK clock source** The HCLK clock frequency = (HCLK clock source frequency) / (HCLK_N + 1)

Clock Controller: SysTick Clock



Clock Controller: PLL Control Register (PLLCON)

Register	Offset	R/W	Description	Reset Value
PLLCON	CLK_BA+0x20	R/W	PLL Control Register	0x0005_C22E

Bits	Descriptions	
[31:20]	**Reserved**	Reserved
[19]	**PLL_SRC**	**PLL Source Clock Select** 1 = PLL source clock from internal 22.1184 MHz high speed oscillator 0 = PLL source clock from external 4~24 MHz high speed crystal
[18]	**OE**	**PLL OE (FOUT enable) pin Control** 0 = PLL FOUT enable 1 = PLL FOUT is fixed low
[17]	**BP**	**PLL Bypass Control** 0 = PLL is in normal mode (default) 1 = PLL clock output is same as clock input (XTALin)
[16]	**PD**	**Power Down Mode** If set the PWR_DOWN_EN bit to 1 in PWRCON register, the PLL will enter power down mode too. 0 = PLL is in normal mode 1 = PLL is in power down mode (default)
[15:14]	**OUT_DV**	**PLL Output Divider Control Pins** Refer to the formulas below the table.
[13:9]	**IN_DV**	**PLL Input Divider Control Pins** Refer to the formulas below the table.

Clock Controller: PLL Control Register (PLLCON)

Register	Offset	R/W	Description	Reset Value
PLLCON	CLK_BA+0x20	R/W	PLL Control Register	0x0005_C22E

Bits	Descriptions	
[8:0]	FB_DV	**PLL Feedback Divider Control Pins** Refer to the formulas below the table.

Output Clock Frequency Setting

$$FOUT = FIN \times \frac{NF}{NR} \times \frac{1}{NO}$$

Constraint:

1. $3.2MHz < FIN < 150MHz$
2. $800\,KHz < \frac{FIN}{2 * NR} < 8MHz$

   $100\,MHz < FCO = FIN \times \frac{NF}{NR} < 200\,MHz$
3. $120\,MHz < FCO$ is preferred

Symbol	Description
FOUT	Output Clock Frequency
FIN	Input (Reference) Clock Frequency
NR	Input Divider (IN_DV + 2)
NF	Feedback Divider (FB_DV + 2)
NO	OUT_DV = "00" : NO = 1 OUT_DV = "01" : NO = 2 OUT_DV = "10" : NO = 2 OUT_DV = "11" : NO = 4

Default Frequency Setting

The default value : 0xC22E
FIN = 12 MHz
NR = (1+2) = 3
NF = (46+2) = 48
NO = 4
FOUT = 12/4 x 48 x 1/3 = 48 MHz

General Purpose I/O: GPIO (A,B,C,D and E)

- 80 General Purpose I/O pins
- arranged in 5 ports named with GPIOA, GPIOB, GPIOC, GPIOD and GPIOE
- can be configured by software individually as input, output, open-drain or quasi-bidirectional mode.
- After reset, the I/O type of all pins stay in quasi-bidirectional mode and port data register GPIOx_DOUT[15:0] resets to 0x0000_FFFF.

General Purpose I/O: GPIO (A,B,C,D and E)

Features

- **Four I/O modes:**
  - Quasi bi-direction
  - Push-Pull output
  - Open-Drain output
  - Input only with high impendence

- **TTL/Schmitt trigger input selectable**

- **I/O pin can be configured as interrupt source with edge/level setting**

- **High driver and high sink IO mode support**

General Purpose I/O: Register Map

Register	Offset	R/W	Description	Reset Value
GP_BA = 0x5000_4000				
GPIOA_PMD	GP_BA+0x000	R/W	GPIO Port A Pin I/O Mode Control	0xFFFF_FFFF
GPIOA_OFFD	GP_BA+0x004	R/W	GPIO Port A Pin Digital Input Path Disable Control	0x0000_0000
GPIOA_DOUT	GP_BA+0x008	R/W	GPIO Port A Data Output Value	0x0000_FFFF
GPIOA_DMASK	GP_BA+0x00C	R/W	GPIO Port A Data Output Write Mask	0x0000_0000
GPIOA_PIN	GP_BA+0x010	R	GPIO Port A Pin Value	0x0000_XXXX
GPIOA_DBEN	GP_BA+0x014	R/W	GPIO Port A De-bounce Enable	0x0000_0000
GPIOA_IMD	GP_BA+0x018	R/W	GPIO Port A Interrupt Mode Control	0x0000_0000
GPIOA_IEN	GP_BA+0x01C	R/W	GPIO Port A Interrupt Enable	0x0000_0000
GPIOA_ISRC	GP_BA+0x020	R/W	GPIO Port A Interrupt Source Flag	0xXXXX_XXXX
GPIOB_PMD	GP_BA+0x040	R/W	GPIO Port B Pin I/O Mode Control	0xFFFF_FFFF

General Purpose I/O: GPIOx_PMD

GPIO Port [A/B/C/D/E] I/O Mode Control (GPIOx_PMD)

Register	Offset	R/W	Description	Reset Value
GPIOA_PMD	GP_BA+0x000	R/W	GPIO Port A Pin I/O Mode Control	0xFFFF_FFFF
GPIOB_PMD	GP_BA+0x040	R/W	GPIO Port B Pin I/O Mode Control	0xFFFF_FFFF
GPIOC_PMD	GP_BA+0x080	R/W	GPIO Port C Pin I/O Mode Control	0xFFFF_FFFF
GPIOD_PMD	GP_BA+0x0C0	R/W	GPIO Port D Pin I/O Mode Control	0xFFFF_FFFF
GPIOE_PMD	GP_BA+0x100	R/W	GPIO Port E Pin I/O Mode Control	0xFFFF_FFFF

31	30	29	28	27	26	25	24
PMD15		PMD14		PMD13		PMD12	
23	22	21	20	19	18	17	16
PMD11		PMD10		PMD9		PMD8	
15	14	13	12	11	10	9	8
PMD7		PMD6		PMD5		PMD4	
7	6	5	4	3	2	1	0
PMD3		PMD2		PMD1		PMD0	

Bits	Descriptions	
[2n+1:2n]	PMDn	**GPIOx I/O Pin[n] Mode Control** Determine each I/O type of GPIOx pins. 00 = GPIO port [n] pin is in INPUT mode 01 = GPIO port [n] pin is in OUTPUT mode 10 = GPIO port [n] pin is in Open-Drain mode 11 = GPIO port [n] pin is in Quasi-bidirectional mode

General Purpose I/O: GPIOx_DOUT

GPIO Port [A/B/C/D/E] Data Output Value (GPIOx_DOUT)

Register	Offset	R/W	Description	Reset Value
GPIOA_DOUT	GP_BA+0x008	R/W	GPIO Port A Data Output Value	0x0000_FFFF
GPIOB_DOUT	GP_BA+0x048	R/W	GPIO Port B Data Output Value	0x0000_FFFF
GPIOC_DOUT	GP_BA+0x088	R/W	GPIO Port C Data Output Value	0x0000_FFFF
GPIOD_DOUT	GP_BA+0x0C8	R/W	GPIO Port D Data Output Value	0x0000_FFFF
GPIOE_DOUT	GP_BA+0x108	R/W	GPIO Port E Data Output Value	0x0000_FFFF

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

Bits	Descriptions	
[31:16]	Reserved	Reserved
[n]	DOUT[n]	**GPIOx Pin[n] Output Value** Each of these bits control the status of a GPIO pin when the GPIO pin is configures as output, open-drain and quasi-mode. 1 = GPIO port [A/B/C/D/E] Pin[n] will drive High if the GPIO pin is configures as output, open-drain and quasi-mode. 0 = GPIO port [A/B/C/D/E] Pin[n] will drive Low if the GPIO pin is configures as output, open-drain and quasi-mode.

General Purpose I/O: GPIOx_PIN

GPIO Port [A/B/C/D/E] Pin Value (GPIOx_PIN)

Register	Offset	R/W	Description	Reset Value
GPIOA_PIN	GP_BA+0x010	R	GPIO Port A Pin Value	0x0000_XXXX
GPIOB_PIN	GP_BA+0x050	R	GPIO Port B Pin Value	0x0000_XXXX
GPIOC_PIN	GP_BA+0x090	R	GPIO Port C Pin Value	0x0000_XXXX
GPIOD_PIN	GP_BA+0x0D0	R	GPIO Port D Pin Value	0x0000_XXXX
GPIOE_PIN	GP_BA+0x110	R	GPIO Port E Pin Value	0x0000_XXXX

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

Bits	Descriptions	
[31:16]	Reserved	Reserved
[n]	PIN[n]	**Port [A/B/C/D/E] Pin Values** Each bit of the register reflects the actual status of the respective GPIO pin If bit is 1, it indicates the corresponding pin status is high, else the pin status is low

General Purpose I/O: GPIO (A,B,C,D and E)

- **Four I/O modes:**
  - Quasi bi-direction
  - Push-Pull output
  - Open-Drain output
  - Input only with high impendence

- Set GPIOx_PMD (PMDn[1:0])
- Output : GPIOx_DOUT
- Input : GPIOx_PIN

General Purpose I/O: GPIO (A,B,C,D and E)

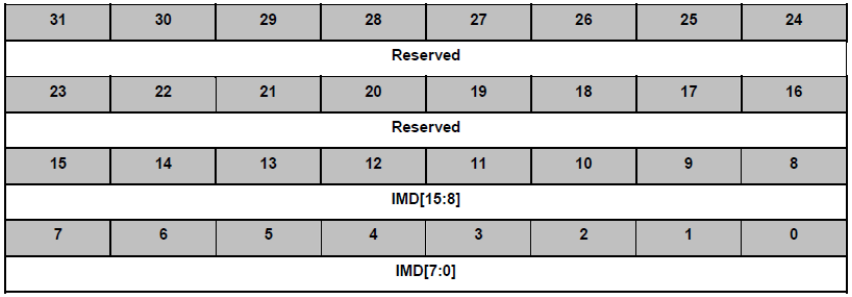
- GPIO Interrupt and wakeup function

Each GPIO pin can be set as chip interrupt source by setting correlative GPIOx_IEN bit and GPIOx_IMD. There are four types of interrupt condition can be selected: low level trigger, high level trigger, falling edge trigger and rising edge trigger. For edge trigger condition, user can enable input signal de-bounce function to prevent unexpected interrupt happened which caused by noise. The de-bounce clock source and sampling cycle can be set through DEBOUNCE register.
The GPIO can also be the chip wakeup source when chip enter idle mode or power down mode.

General Purpose I/O: GPIOx_IMD

GPIO Port [A/B/C/D/E] Interrupt Mode Control (GPIOx_IMD)

Register	Offset	R/W	Description	Reset Value
GPIOA_IMD	GP_BA+0x018	R/W	GPIO Port A Interrupt Mode Control	0xXXXX_0000
GPIOB_IMD	GP_BA+0x058	R/W	GPIO Port B Interrupt Mode Control	0xXXXX_0000
GPIOC_IMD	GP_BA+0x098	R/W	GPIO Port C Interrupt Mode Control	0xXXXX_0000
GPIOD_IMD	GP_BA+0x0D8	R/W	GPIO Port D Interrupt Mode Control	0xXXXX_0000
GPIOE_IMD	GP_BA+0x118	R/W	GPIO Port E Interrupt Mode Control	0xXXXX_0000



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

Bits	Descriptions	
[31:16]	Reserved	Reserved
[n]	IMD[n]	**Port [A/B/C/D/E] Edge or Level Detection Interrupt Control** IMD[n] is used to control the interrupt is by level trigger or by edge trigger. If the interrupt is by edge trigger, the trigger source can be controlled by de-bounce. If the interrupt is by level trigger, the input source is sampled by one HCLK clock and generates the interrupt. 1 = Level trigger interrupt 0 = Edge trigger interrupt If set pin as the level trigger interrupt, then only one level can be set on the registers GPIOx_IEN. If set both the level to trigger interrupt, the setting is ignored and no interrupt will occur The de-bounce function is valid for edge triggered interrupt. If the interrupt mode is level triggered, the de-bounce enable bit is ignored.

General Purpose I/O: GPIOx_IEN

GPIO Port [A/B/C/D] Interrupt Enable Control (GPIOx_IEN)

Register	Offset	R/W	Description	Reset Value
GPIOA_IEN	GP_BA+0x01C	R/W	GPIO Port A Interrupt Enable	0x0000_0000
GPIOB_IEN	GP_BA+0x05C	R/W	GPIO Port B Interrupt Enable	0x0000_0000
GPIOC_IEN	GP_BA+0x09C	R/W	GPIO Port C Interrupt Enable	0x0000_0000
GPIOD_IEN	GP_BA+0x0DC	R/W	GPIO Port D Interrupt Enable	0x0000_0000
GPIOE_IEN	GP_BA+0x11C	R/W	GPIO Port E Interrupt Enable	0x0000_0000

Bits	Descriptions	
[n+16]	IR_EN[n]	**Port [A/B/C/D/E] Interrupt Enable by Input Rising Edge or Input Level High** IR_EN[n] used to enable the interrupt for each of the corresponding input GPIO_PIN[n]. Set bit to 1 also enable the pin wake-up function When set the IR_EN[n] bit to 1: If the interrupt is level trigger, the input PIN[n] state at level "high" will generate the interrupt. If the interrupt is edge trigger, the input PIN[n] state change from "low-to-high" will generate the interrupt. 1 = Enable the PIN[n] level-high or low-to-high interrupt 0 = Disable the PIN[n] level-high or low-to-high interrupt
[n]	IF_EN[n]	**Port [A/B/C/D/E] Interrupt Enable by Input Falling Edge or Input Level Low** IF_EN[n] used to enable the interrupt for each of the corresponding input GPIO_PIN[n]. Set bit to 1 also enable the pin wake-up function When set the IF_EN[n] bit to 1: If the interrupt is level trigger, the input PIN[n] state at level "low" will generate the interrupt. If the interrupt is edge trigger, the input PIN[n] state change from "high-to-low" will generate the interrupt. 1 = Enable the PIN[n] state low-level or high-to-low change interrupt 0 = Disable the PIN[n] state low-level or high-to-low change interrupt

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

General Purpose I/O: Pxn_PDIO

GPIO Px.n Pin Data Input/Output (Pxn_PDIO)

Register	Offset	R/W	Description	Reset Value
PAn_PDIO	GP_BA+0x200 - GP_BA+0x23C	R/W	GPIO PA.n Pin Data Input/Output	0x0000_0001
PBn_PDIO	GP_BA+0x240 - GP_BA+0x27C	R/W	GPIO PB.n Pin Data Input/Output	0x0000_0001
PCn_PDIO	GP_BA+0x280 - GP_BA+0x2BC	R/W	GPIO PC.n Pin Data Input/Output	0x0000_0001
PDn_PDIO	GP_BA+0x2C0 - GP_BA+0x2FC	R/W	GPIO PD.n Pin Data Input/Output	0x0000_0001
PEn_PDIO	GP_BA+0x300 - GP_BA+0x3FC	R/W	GPIO PE.n Pin Data Input/Output	0x0000_0001

General Purpose I/O: Pxn_PDIO

Note: x = A/B/C/D/E and n = 0~15

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

Bits	Descriptions	
[0]	**Pxn_PDIO**	**GPIO Px.n Pin Data Input/Output** Write this bit can control one GPIO pin output value 1 = Set corresponding GPIO pin to high 0 = Set corresponding GPIO pin to low Read this register to get GPIO pin status. For example: write PA0_PDIO will reflect the written value to bit GPIOA_DOUT[0], read PA0_PDIO will return the value of GPIOA_PIN[0] Note: The write operation will not be affected by register GPIOx_DMASK

Timer Controller (TMR):

- Four 32-bit timers, **TIMER0~TIMER3**
- Can be used for
  - frequency measurement
  - event counting
  - interval measurement
  - clock generation
  - delay timing
  - and so on

Timer Controller (TMR):

Features

- 4 sets of 32-bit timers with 24-bit up-timer and one 8-bit pre-scale counter
- Independent clock source for each timer
- Provides one-shot, periodic, toggle and continuous counting operation modes
- Support event counting function to count the event from external pin
- Support input capture function to capture or reset counter value
- 24-bit timer value is readable through TDR (Timer Data Register)

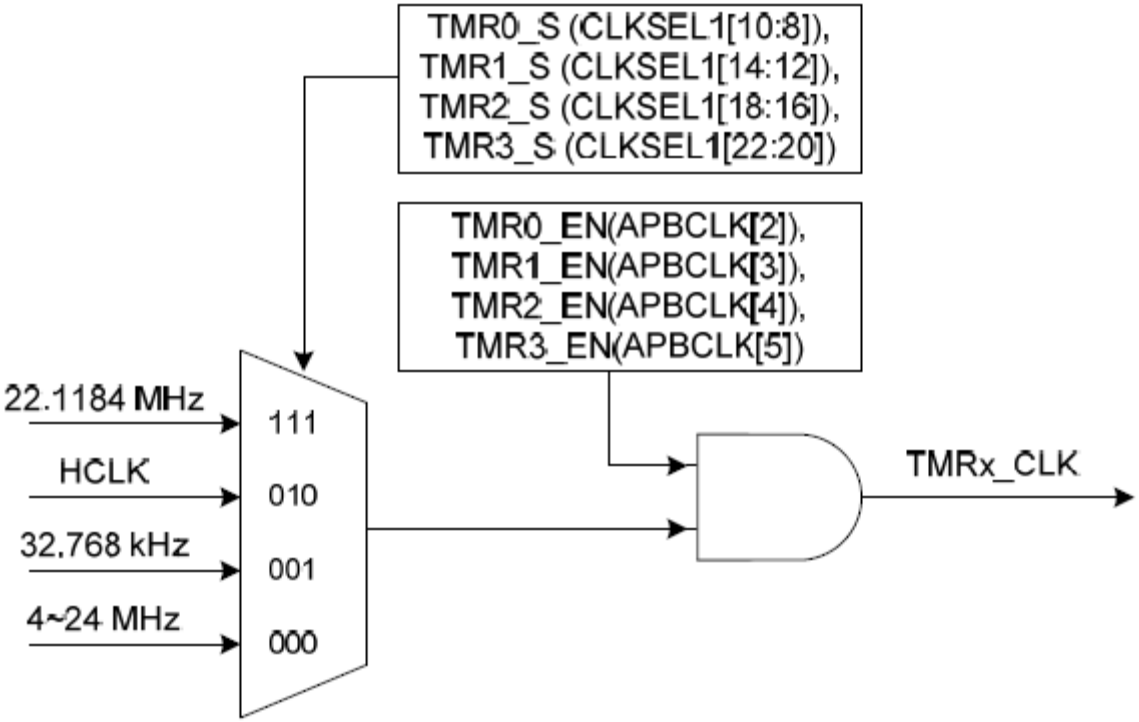
Timer Controller (TMR):

Features

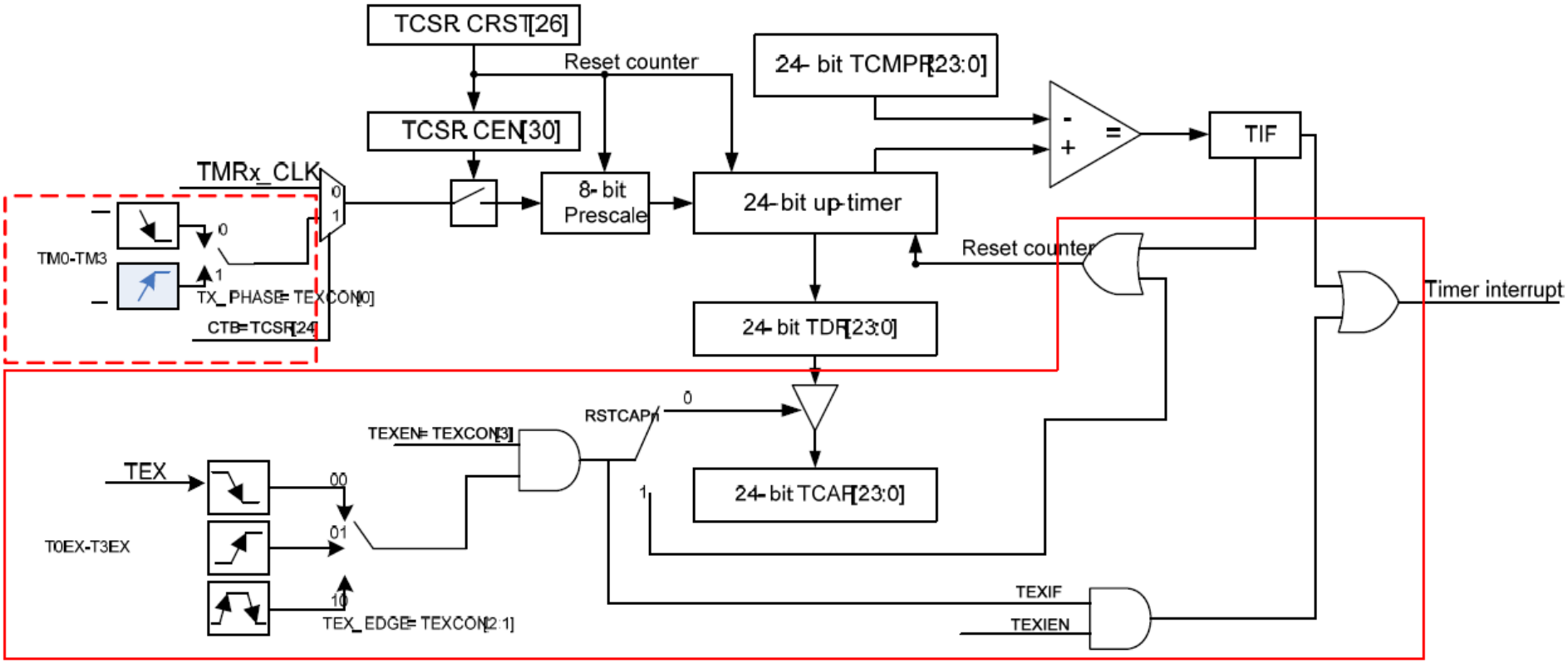
- Time out period = (Period of timer clock input) * (8-bit pre-scale counter + 1) * (24-bit **TCMP**)

- Maximum counting cycle time = (1 / T MHz) * (2^8) * (2^{24}), T is the period of timer clock

Timer Controller: Block Diagram



Timer Controller: Block Diagram



Timer Controller: Register Map

Register	Offset	R/W	Description	Reset Value
TMR_BA01 = 0x4001_0000 TMR_BA23 = 0x4011_0000				
TCSR0	TMR_BA01+0x00	R/W	Timer0 Control and Status Register	0x0000_0005
TCMPR0	TMR_BA01+0x04	R/W	Timer0 Compare Register	0x0000_0000
TISR0	TMR_BA01+0x08	R/W	Timer0 Interrupt Status Register	0x0000_0000
TDR0	TMR_BA01+0x0C	R	Timer0 Data Register	0x0000_0000
TCAP0	TMR_BA01+0x10	R	Timer0 Capture Data Register	0x0000_0000
TEXCON0	TMR_BA01+0x14	R/W	Timer0 External Control Register	0x0000_0000
TEXISR0	TMR_BA01+0x18	R/W	Timer0 External Interrupt Status Register	0x0000_0000

Timer Controller: One–Shot Mode

Once the timer counter value reaches timer compare register (TCMPR) value, if IE (TCSR[29] interrupt enable bit) is set to 1, then the timer interrupt flag is set and the interrupt signal is generated and sent to NVIC to inform CPU.
> It indicates that the timer counting overflow happens.

If IE (TCSR[29] interrupt enable bit) is set to 0, no interrupt signal is generated.

In this operating mode, once the timer counter value reaches timer compare register (TCMPR) value, the timer counter value goes back to counting initial value and CEN (timer enable bit) is cleared to 0 by timer controller.

Timer counting operation stops, once the timer counter value reaches timer compare register (TCMPR) value.

That is to say, timer operates timer counting and compares with TCMPR value function only one time after programming the timer compare register (TCMPR) value and CEN (timer enable bit) is set to 1.

Timer Controller: Periodic Mode

Once the timer counter value reaches timer compare register (TCMPR) value, if IE (TCSR[29] interrupt enable bit) is set to 1, then the timer interrupt flag is set and the interrupt signal is generated and sent to NVIC to inform CPU.
> It indicates that the timer counting overflow happens.

If IE (TCSR[29] interrupt enable bit) is set to 0, no interrupt signal is generated.

In this operating mode, once the timer counter value reaches timer compare register (TCMPR) value, the timer counter value goes back to counting initial value and CEN (timer enable bit) is kept at 1 (counting enable continuously). The timer counter operates up counting again.

The timer counting operation doesn't stop until the CEN is set to 0.
The interrupt signal is also generated periodically.

Timer Controller: Toggle Mode

Once the timer counter value reaches timer compare register (TCMPR) value,
if IE (TCSR[29] interrupt enable bit) is set to 1, then the timer interrupt flag is set
and the interrupt signal is generated and sent to NVIC to inform CPU.
> It indicates that the timer counting overflow happens.

The associated toggle output (tout) signal is set to 1.

If IE (TCSR[29] interrupt enable bit) is set to 0, no interrupt signal is generated.

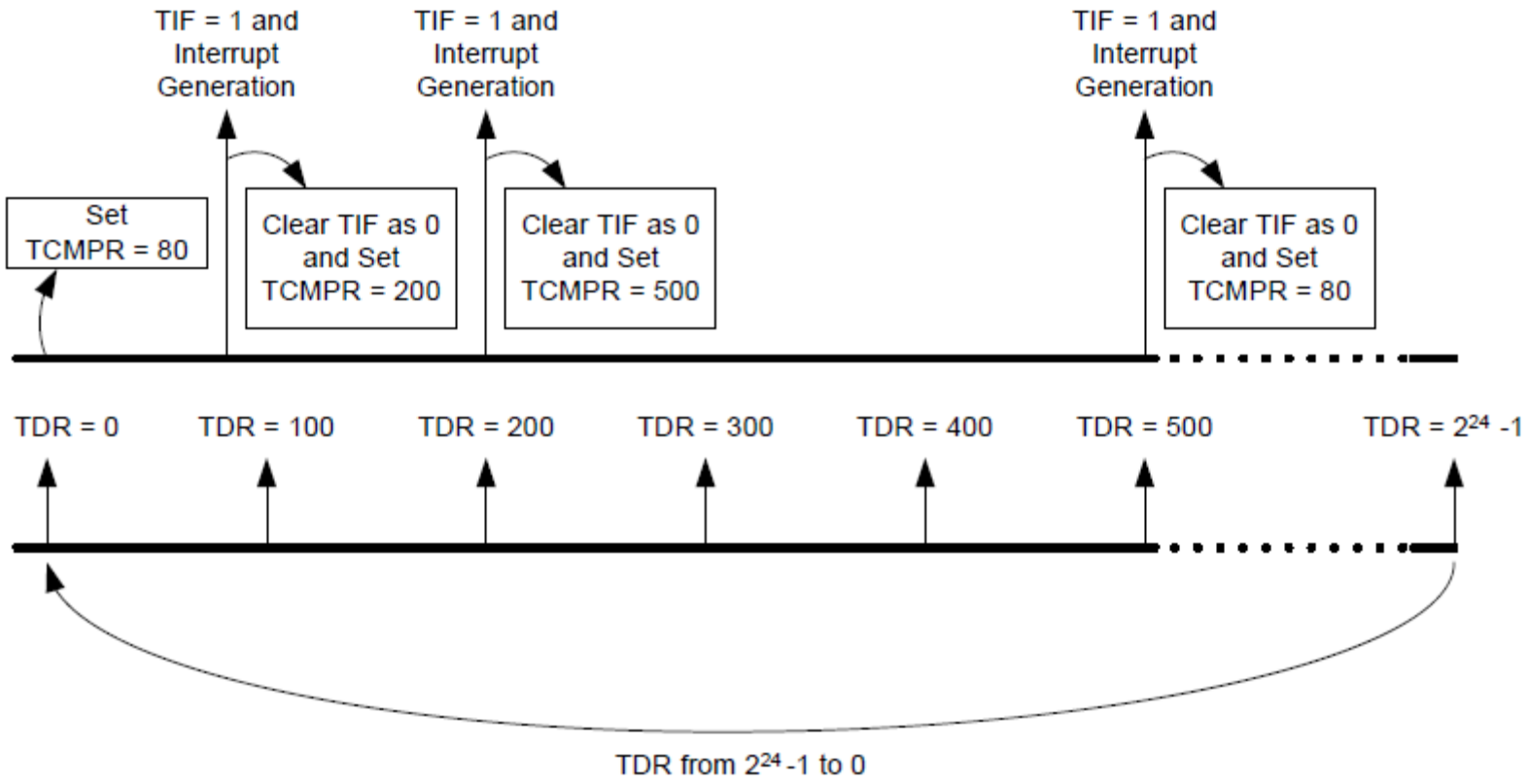
In this operating mode, once the timer counter value reaches timer compare
register (TCMPR) value, the timer counter value goes back to counting
initial value and CEN (timer enable bit) is kept at 1 (counting enable continuously).
The timer counter operates up counting again.

If the interrupt flag is cleared by software, once the timer counter value reaches timer compare register (TCMPR) value and IE (interrupt enable bit) is set to 1, then the timer interrupt flag is set and the interrupt signal is generated and sent to NVIC to inform CPU again. The associated toggle output (tout) signal is set to 0.

The timer counting operation doesn't stop until the CEN is set to 0.

Timer Controller: Continuous Counting Mode

User can change different TCMPR value immediately without disabling timer counting and restarting timer counting.



Timer Controller: Event Counting Function

from TM0~TM3 pins

In event counting function, the clock source of timer controller, TMRx_CLK should be set as HCLK.
It provides TM0~TM3 enabled or disabled de-bounce function by TEXCONx[7] and TM0~TM3 falling or rising phase counting setting by TEXCONx[0]. And, the event count source operating frequency should be less than 1/3 HCLK frequency if disable counting de-bounce or less than 1/8 HCLK frequency if enable counting de-bounce. Otherwise, the returned TDR value is incorrect.

Timer Controller: Input Capture Function

to capture or reset timer counter value

If TEXEN (Timer External Pin Enable) is set to 1 and RSTCAPSEL is set to 0, the timer counter value (TDR) will be captured into TCAP register when TEX (Timer External Pin) pin trigger condition occurred.
There are four TEX sources form specified pins, T0EX~T3EX pins.

If TEXEN is set to 1 and RSTCAPSEL is set to 1, the TDR will be reset to 0 when TEX pin trigger condition happened.
The TEX trigger edge can choose by TEX_EDGE.
When TEX trigger occurred, TEXIF (Timer External Interrupt Flag) is set to 1, and if enabled TEXIEN (Timer External Interrupt Enable Bit) to 1, the interrupt signal is generated then sent to NVIC to inform CPU.

It also provides T0EX~T3EX enabled or disabled capture de-bounce function by TEXCONx[6]. And, the TEX source operating frequency should be less than 1/3 HCLK frequency if disable TEX de-bounce or less than 1/8 HCLK frequency if enable TEX de-bounce.

Timer Controller: Example

```
void Timer_initial(void)
{
  /* Step 1. Enable and Select Timer clock source */
  SYSCLK->CLKSEL1.TMR0_S = 0; //Select 12Mhz for Timer0 clock source
  //DK: 0 = 12M, 1 = 32k, 2 = HCLK, 7 = 22M

  SYSCLK->APBCLK.TMR0_EN =1;  //Enable Timer0 clock source

  /* Step 2. Select Operation mode */
  TIMER0->TCSR.MODE=1;    //Select periodic mode for operation mode
  //DK: 0 = One shot, 1 = Periodic, 2 = Toggle, 3 = continuous counting mode

  /* Step 3. Select Time out period
    = (Period of timer clock input) * (8-bit Prescale + 1) * (24-bit TCMP)*/
  TIMER0->TCSR.PRESCALE=0;  // Set Prescale [0~255]
  TIMER0->TCMPR  = 1000000;   // Set TICR(TCMP) [0~16777215] : 24bits
                 // (1/12000000)*(0+1)*(1000000)= 83 msec or 12 Hz
                 // (1/22000000)*(0+1)*(1000000)= 45 msec (for 22MHz)

  /* Step 4. Enable interrupt */
  TIMER0->TCSR.IE = 1;
  TIMER0->TISR.TIF = 1;        // Write 1 to clear
  NVIC_EnableIRQ(TMR0_IRQn);  // Enable Timer0 Interrupt

  /* Step 5. Enable Timer module */
  TIMER0->TCSR.CRST = 1;     // Reset up counter
  TIMER0->TCSR.CEN = 1;      // Starts counting

  TIMER0->TCSR.TDR_EN=1;     // updated continuously with
  // the 24-bit up-timer value as the timer is counting.
}
```