CYC_WAKE

REVISION HISTORY

Revision Number	Date	Description of Change	Author
V0.0	9/22/2022	Draft version	Su Aixue

Table of Contents

CYC_WAKE

Introduction

The CYC_WAKE is a simple, periodic pulse generator. The period of the output wake can be programmed. In other words, the CYC_WAKE can generate square wave with various frequency.

The CYC_WAKE module has the following features:
• Latch MON_EN_REG, MON_WAKE_PERIOD_REG to MON_EN, MON_WAKE_PERIOD once MON_WAKE_GO is high; (HWR001_CYC_WAKE)
• Output clr_MON_WAKE_GO high (lasting 1 CLK_SLOW) after register latched done; (HWR002_CYC_WAKE)
• Output MON_ADC_GO according to MON_WAKE_PERIOD; (HWR003_CYC_WAKE)
• When D2A_CELL_ADC_EN is high, MON_ADC_GO will be low; (HWR003_CYC_WAKE)
• MON_ADC_GO can be cleared by clr_ADC_GO; (HWR003_CYC_WAKE)
• Output MON_WAKE according to MON_WAKE_PERIOD; (200ms-3.2s, step 200ms; 3.2s-156.8s, step 3.2s) (HWR004_CYC_WAKE)
• MON_WAKE can be cleared by RR_END; (HWR004_CYC_WAKE)
• MON_WAKE shall be high earlier 10ms than MON_ADC_GO. (HWR004_CYC_WAKE)

Register Definition

Register Map

Table 11 CYC_WAKE Register Map

ADDRESS	NAME	DESCRIPTION	RESET VALUE
CYC_WAKE			
0x0000	**MON_CONF**	CYC_WAKE configuration register	0x45
0x1FF7	**ADC_CTRL**	ADC control register	0x00

MON_CONF

Register 1. MON_CONF (CYC_WAKE configuration register, offset 0x0000)

BIT	NAME	ACCESS	RESET	DESCRIPTION
7:2	**MON_WAKE_PERIOD**	R/W	6'h11	Period Configuration 00 0000: 0.2s 00 0001: 0.4s 00 0010: 0.6s … 00 1111: 3.2s 01 0000: 6.4s 01 0001: 9.6s 01 0010: 12.8s … 11 1111: 156.8s

1	**REV**	R	1'b0	Reserved
0	**MON_WAKE_EN**	R/W	1'b1	CYC_WAKE Enable Bit 0: Disable 1: Enable

ADC_CTRL

Register 11. ADC_CTRL (ADC control register, offset 0x1FF6)

BIT	NAME	ACCESS	RESET	DESCRIPTION
7	**MON_WAKE_GO**	R/W	1'b0	Mon-wake Starting Bit 0: Ready 1: Execute
6:3	**REV**	R	4'h0	Reserved
2:0	**--**	--	--	--

Function Details

Block Diagram

The main elements of CYC_WAKE and their interactions are shown in Fig 1.

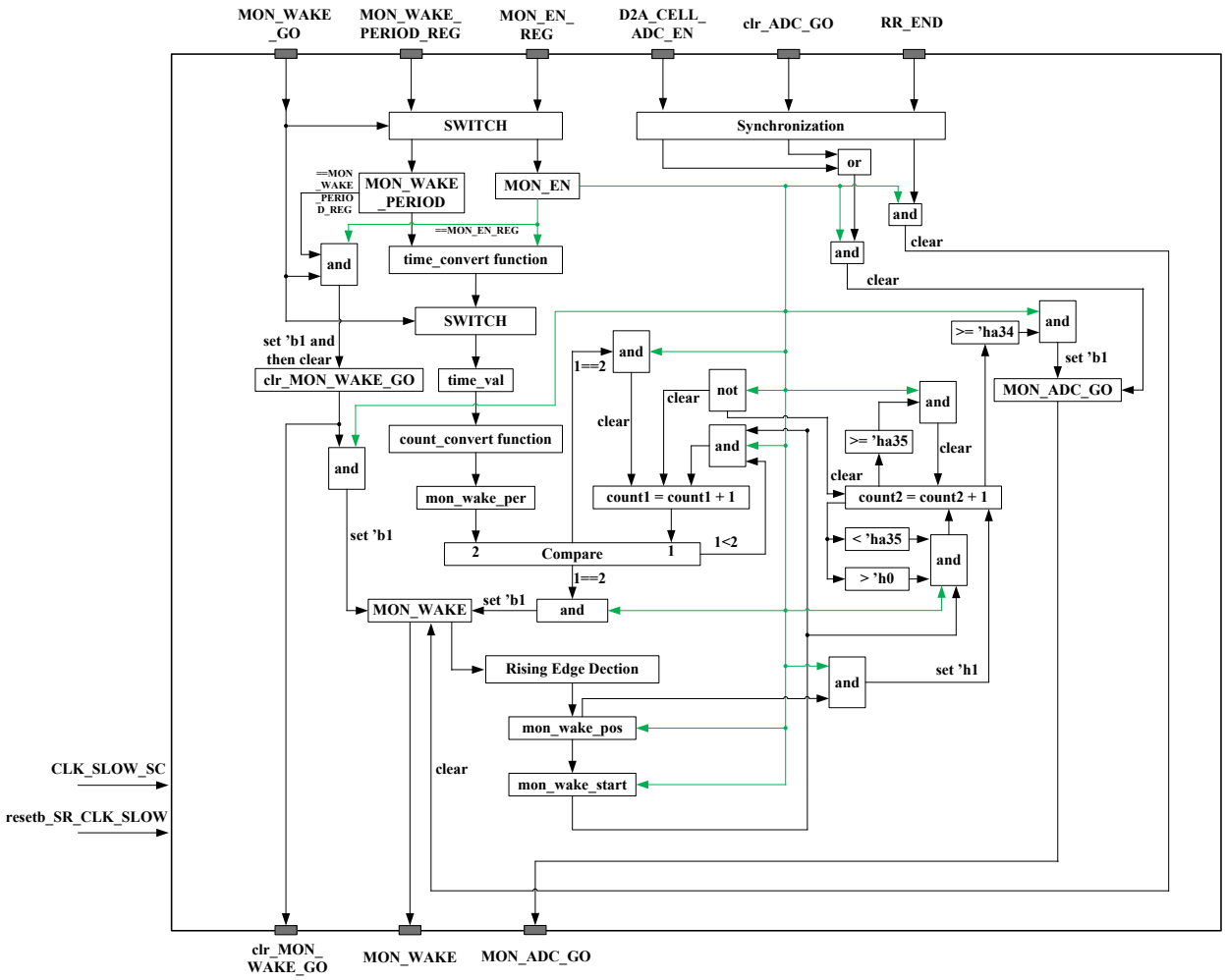

Fig 1. CYC_WAKE Block Diagram

CYC_WAKE IO Descriptions

This section provides the CYC_WAKE IO descriptions.

Table 2 CYC_WAKE IO descriptions

Signal	Width	Duration	I/O	Default Value	Register
CLK_SLOW_SC	1	--	I	--	--
resetb_SR_CLK_SLOW	1	--	I	--	--
MON_WAKE_GO	1	1~2 CLK_SLOW_SC	I	--	MON_WAKE_GO
MON_EN_REG	1	--	I	--	MON_WAKE_EN
MON_WAKE_PERIOD_R EG	1	--	I	--	MON_WAKE_PERIOD
D2A_CELL_ADC_EN	1	43874 CLK_REG_SC	I	--	--
clr_ADC_GO	1	16 CLK_ADC_SC	I	--	--
RR_END	1	16 CLK_ADC_SC	I	--	--
clr_MON_WAKE_GO	1	1 CLK_SLOW_SC	O	1'b0	--

MON_WAKE	1	4017 CLK_SLOW_SC	O	1'b0	--
MON_ADC_GO	1	4 CLK_SLOW_SC	O	1'b0	--

CYC_WAKE Key Signal Descriptions

Table 3 CYC_WAKE key signal descriptions

Signal	Width	Duration	Default Value	Description
time_val	32	--	32'h2580	When MON_EN = 1, the value of this signal is the duration corresponding to the MON_WAKE_PERIOD_REG. This signal is used to store the period of the MON_WAKE output signal.
mon_wake_per	32	--	32'h264398	The value of this signal is the clock periods' number corresponding the time_val.
clr_MON_WAKE_GO	1	1 CLK_SLOW_SC	1'b0	The clr_MON_WAKE_GO is a pulse lasting for a clock period. When "MON_WAKE_GO = 1" is detected, the clr_MON_WAKE_GO pulse will generate at the rising edge of the next clock. This signal is an output signal, and is used to clear the MON_WAKE_GO.
MON_WAKE	1	4017 CLK_SLOW_SC	1'b0	When MON_EN = 1 and clr_MON_WAKE_GO = 1, the first rising edge of MON_WAKE will be generated, and MON_WAKE is cleared by the next RR_END_SYNC (RR_END after synchronization). The second and the subsequent rising edge of MON_WAKE is generated at the moment when the following signal count1 = mon_wake_per and MON_EN = 1. It is an output signal and it is a periodic signal.
mon_wake_pos	1	1 CLK_SLOW_SC	1'b0	When the rising edge of MON_WAKE appear and MON_EN = 1, the pulse of mon_wake_pos (width = 1 clock period) is generated.
mon_wake_start	1	--	1'b0	When mon_wake_pos = 1, mon_wake_start = 1.
count1	32	--	32'b0	When MON_EN = 1 and mon_wake_start = 1, the count1 starts to count, and count back to 0 when it is up to mon_wake_per or MON_EN = 0.
count2	12	--	12'b0	When MON_EN = 1 and mon_wake_pos = 1, the count2 starts to count, and count back to 0 when it is up to 12'h35 (corresponding to 10ms) or MON_EN = 0. This signal is used to count the time that MON_WAKE earlier than MON_ADC_GO.

CYC_WAKE Function Descriptions

The CYC_WAKE module has the following functions:
• Latch MON_EN_REG, MON_WAKE_PERIOD_REG to MON_EN, MON_WAKE_PERIOD once MON_WAKE_GO is high; (Func 1) (HWR001_CYC_WAKE)
• Output clr_MON_WAKE_GO high (lasting 1 CLK_SLOW) after register latched done; (Func 2) (HWR002_CYC_WAKE)
• Output MON_ADC_GO according to MON_WAKE_PERIOD; (Func 3) (HWR003_CYC_WAKE)

• When D2A_CELL_ADC_EN is high, MON_ADC_GO will be low; (Func 4) (HWR003_CYC_WAKE)
• MON_ADC_GO can be cleared by clr_ADC_GO; (Func 5) (HWR003_CYC_WAKE)
• Output MON_WAKE according to MON_WAKE_PERIOD; (200ms-3.2s, step 200ms; 3.2s-156.8s, step 3.2s) (Func 6) (HWR004_CYC_WAKE)
• MON_WAKE can be cleared by RR_END; (Func 7) (HWR004_CYC_WAKE)
• MON_WAKE shall be high earlier 10ms than MON_ADC_GO. (Func 8) (HWR004_CYC_WAKE)

Above functions can be found in the following timing diagram. Fig 3 and Fig 4 are magnified version of Fig 2.
Func 1: Sample the value of MON_WAKE_GO using CLK_SLOW_SC. When the high level of MON_WAKE_GO is detected, the value of MON_EN and MON_WAKE_PERIOD will be set to the value of MON_EN_REG and MON_WAKE_PEEROD_REG.
Func 2: Sample the value of MON_WAKE_GO, MON_EN_REG, MON_EN, MON_WAKE_PERIOD_REG, MON_WAKE_PERIOD using CLK_SLOW_SC. When the high level of MON_WAKE_GO is detected, and the value of MON_EN/MON_WAKE_PERIOD being equal to MON_EN_REG/MON_WAKE_PERION_REG is detected at the same time, the clr_MON_WAKE_GO will be set to 1. Besides, sample the value of clr_MON_WAKE_GO using CLK_SLOW_SC. When the high level of CLK_SLOW_SC is detected, the value of clr_MON_WAKE_GO will be cleared.
Func 3: At the beginning, convert the value of MON_WAKE_PERIOD convert to timing value, and then covert the timing value convert the counts of clock cycles. Finally, generate a periodic wave (MON_ADC_GO) using counting method.
Func 6: Similar to Func 3.
Func 8: Implement it using counting method.

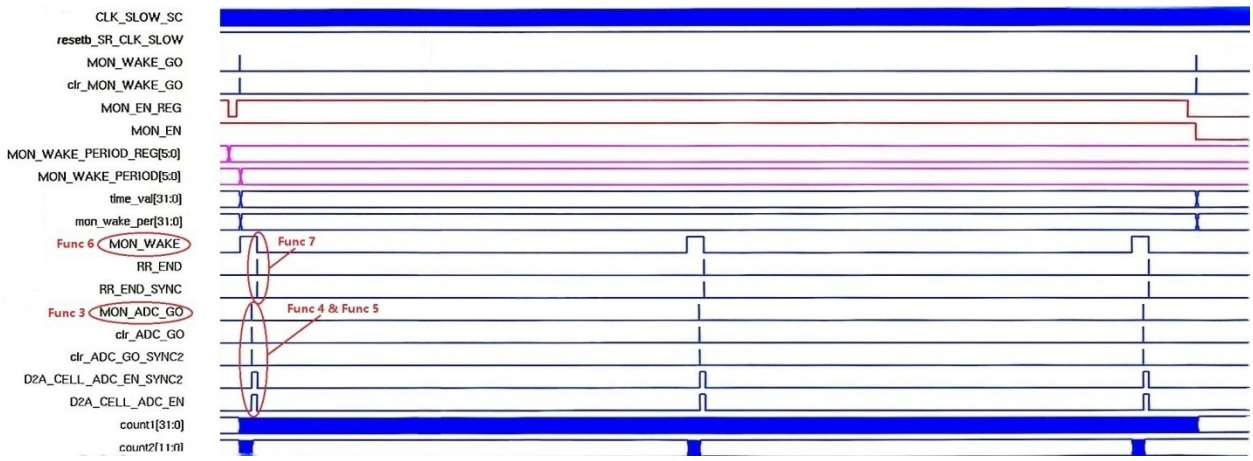

Fig 2. CYC_WAKE Timing Diagram 1

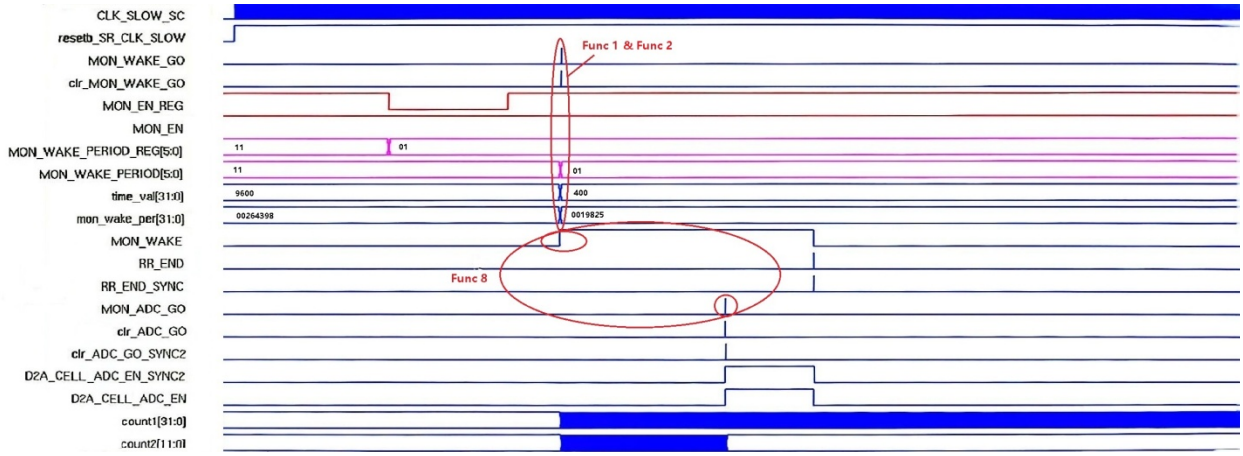

Fig 3. CYC_WAKE Timing Diagram 2

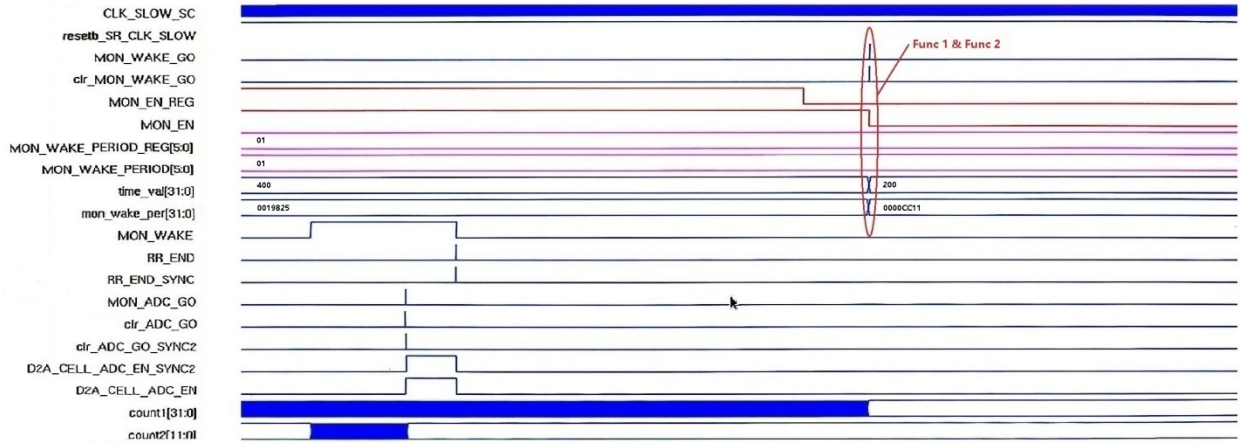

Fig 4. CYC_WAKE Timing Diagram 3