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 2](#_Toc121236419)

[Introduction 2](#_Toc121236420)

[Register Definition 2](#_Toc121236421)

[Register Map 2](#_Toc121236422)

[MON\_CONF 2](#_Toc121236423)

[ADC\_CTRL 3](#_Toc121236424)

[Function Details 3](#_Toc121236425)

[Block Diagram 3](#_Toc121236426)

[CYC\_WAKE IO Descriptions 4](#_Toc121236427)

[CYC\_WAKE Key Signal Descriptions 5](#_Toc121236428)

[CYC\_WAKE Function Descriptions 5](#_Toc121236429)

# 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 1 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\_REG | 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.

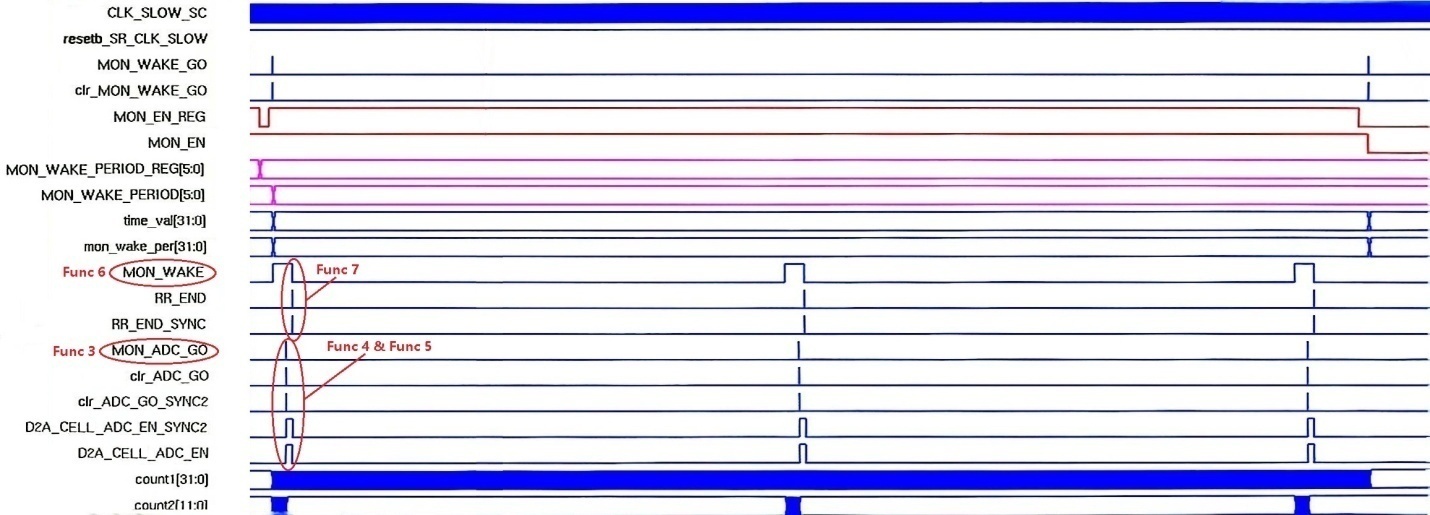


Fig 2. CYC\_WAKE Timing Diagram 1

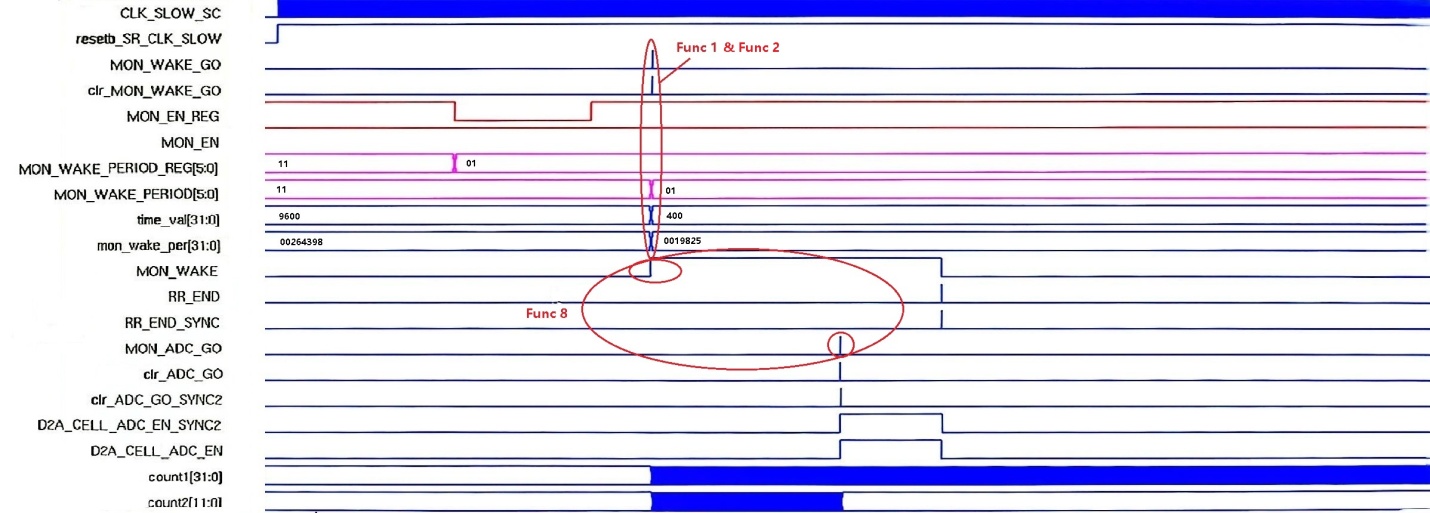


Fig 3. CYC\_WAKE Timing Diagram 2

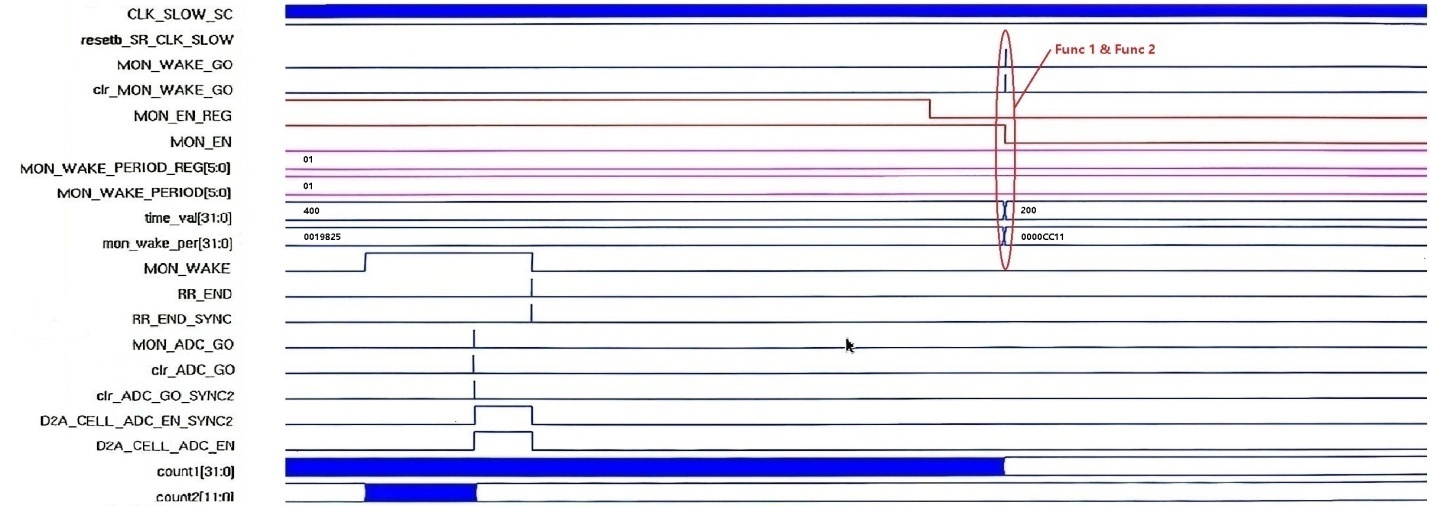


Fig 4. CYC\_WAKE Timing Diagram 3