SpartanMC

Timer Capture Module (timer-cap)

Table of Contents

List of Figures

List of Tables

Timer Capture Module (timer-cap)

The timer capture module is used to capture the value of a timer register after an external trigger signal.

Note: **The timer capture module always requires a basic timer module as input. Hence, it can not work autonomously.**

(Otherwise, a basic timer could be used as input for multiple capture moduls.)

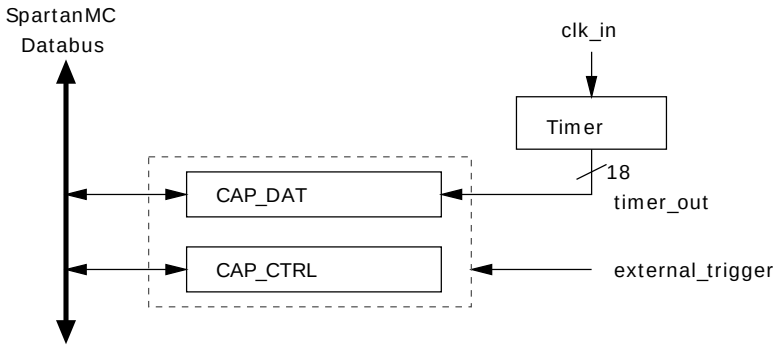

Figure 1: Capture module block diagram

1. Usage and Interrupts

If this module is triggerd by an external Signal, the current timer value is stored in the local capture register.

Optionally, an interrupt could be generated for each capture event. The interrupt flag is cleared with an access on the data register or control register.

The capture module provides two operation modes for interrupt generation: On the one hand, it could generate its interrupt on edges of a input signal. On the other hand, it could generate its interrupt during a specific level of the input signal. After completion of the capture procedure the enable bit in the control register is cleared. To start a new capture procedure this bit has to be set again.

2. Module parameters

Table 1: TIMER Capture module parameters

Parameter	Default Value	Descripton
BASE_ADR		Start address of the memory mapped peripheral registers. The value is taken as offset to the start address of the peripheral memory space. **This parameter is set by jConfig automatically.**

3. Peripheral Registers

3.1. Timer Capture Register Description

The timer capture module provides two 18 bit registers which are mapped to the SpartanMC address space e.g. 0x1A000 + BASE_ADR + Offset.

Table 2: Timer capture registers

Offset	Name	Access	Description
0	CAP_CTRL	read/ write	Configuration of the operation mode. (An acces on this register clears the interrupt flag)
1	CAP_DAT	read	Register for captured data. (An acces on this register clears the interrupt flag)

3.2. CAP_DAT Register

Table 3: CAP_DAT register layout

Bit	Name	Access	Default	Description
0-17	Capture Value	read	x	The captured data.

3.3. CAP_CTRL Register

Table 4: CAP_CTRL register layout

Bit	Name	Access	Default	Description
0	CAP_EN	read/ write	0	If set to one the capture logic is enabled. This bit is cleared after capture event completion.
1	CAP_EN_INT	read/ write	0	If set to one the interrupt is enabled.
2-4	CAP_MODE	read/ write	000	Sets the operation mode: 000 = capture disable 001 = not used 010 = capture on falling edge 011 = capture on raising edge 100 = capture on low input signal level 101 = capture on high input signal level 110 = capture on all edges 111 = capture on all edges
5-17	x	read	0	Not used.

Table 4: CAP_CTRL register layout

3.4. TIMER_CAP C-Header for Register Description

```
#ifndef __CAPTURE_H
#define __CAPTURE_H

#ifdef __cplusplus
extern "C" {
#endif

#define CAPTURE_EN            (1 << 0)
#define CAPTURE_EN_INT         (1 << 1)
#define CAPTURE_EDGE         (1 << 2)

#define CAPTURE_NON            (CAP_EDGE * 0)
#define CAPTURE_FALLING_EDGE   (CAP_EDGE * 2)
#define CAPTURE_RISING_EDGE      (CAP_EDGE * 3)
#define CAPTURE_LOW_LEVEL      (CAP_EDGE * 4)
#define CAPTURE_HIGH_LEVEL      (CAP_EDGE * 5)
#define CAPTURE_ANYTHING_EDGE   (CAP_EDGE * 7)

typedef struct cap {
   volatile unsigned int   CAP_CTRL;   // (r/w)
   volatile unsigned int   CAP_DAT;   // (r)
} capture_regs_t;

#ifdef __cplusplus
}
#endif

#endif
```