TIMER_HW driver

Generated by Doxygen 1.9.1

1 File Documentation

1.1 altera_avalon_timer_regs.h File Reference

```
#include <io.h>
#include <system.h>
```

Macros

- #define TIMER_STOP IOWR_32DIRECT(TIMER_HW_IP_0_BASE,4,0x00000000)

  writing 0x0 to control register stops timer
- #define TIMER_RESET IOWR_32DIRECT(TIMER_HW_IP_0_BASE,4,0x40000000)

  writing 0x40000000 to control register resets timer
- #define TIMER_START IOWR_32DIRECT(TIMER_HW_IP_0_BASE,4,0x80000000)

  writing 0x80000000 to control register starts timer
- #define TIMER_READ IORD_32DIRECT(TIMER_HW_IP_0_BASE,0)

  reading time measurements

1.1.1 Detailed Description

This file defines the driver for a TIMER_HW component. The component uses a register interface to the data bus. There are defined two 32-bit registers:

- Data register. Size - 32 bits. Read-only register shows current time in 20ns units.
- Control register. Write-only register uses bits 30,31 as follows:
  - 0b00 -> bits30,31 resets the timer;
  - 0b01 -> bits30,31 stops the timer;
  - 0b10 -> bits30,31 starts the timer.

The component does not use hardware interrupts therefore polling the data register is the only way to get information about the timer as well as to read the time measurements.

1.1.2 Macro Definition Documentation

1.1.2.1 TIMER_READ `#define TIMER_READ IORD_32DIRECT(TIMER_HW_IP_0_BASE,0)`

this macro generates instruction to read the timer current value

1.1.2.2 TIMER_RESET `#define TIMER_RESET IOWR_32DIRECT(TIMER_HW_IP_0_BASE,4,0x40000000)`

This macro generates instruction to reset the timer and clears the timer current value

1.1.2.3 TIMER_START `#define TIMER_START IOWR_32DIRECT(TIMER_HW_IP_0_BASE,4,0x80000000)`

This macro generates instruction to start the timer

1.1.2.4 TIMER_STOP `#define TIMER_STOP IOWR_32DIRECT(TIMER_HW_IP_0_BASE,4,0x00000000)`

this macro generates instruction to stop the timer