System controller

Sébastien Bourdeauducq

January 2010

1 Overview

The system controller includes basic functionality that is found on most SoC designs :

- a **GPIO controller**, which can be used for software-driven low-speed communication with peripherals and for simple user interaction like controlling LEDs and detecting keypresses.
- two **timers** with a precision of one clock cycle.
- a 32-bit **system identification** value.

2 GPIO controller

The GPIO controller can support a maximum of 32 inputs and 32 outputs. Bidirectional signals are not supported. The `ninputs` and `noutputs` control the actual number of input and outputs.

It is possible to generate an interrupt when an input changes. The interrupt will be generated on both rising and falling edges of the input.

Offset	Read/Write	Default	Description
0x00	R	N/A	Inputs.
0x04	RW	0	Outputs.
0x08	RW	0	Interrupt enable. Lists the input pins whose level changes (bit set in the 0x08 register) generate an interrupt.

3 Dual timer

The system controller provides two independent timers. Timer 0 uses registers 0x10, 0x14 and 0x18, while timer 1 uses registers 0x20, 0x24 and 0x28.

3.1 Timer control register, offset 0x10/0x20

Bits	Access	Default	Description
0	RW	0	(Enable bit). If this bit is set, the counter register counts upwards until it reaches the value stored in the compare register.
1	RW	0	If this bit is set, the counter will automatically restart from 1 when the compare value is reached, otherwise the counter will be disabled.
31 – 2	—	0	Reserved.

3.2 Compare register, offset 0x14/0x24

This register holds the value to which the counter is compared to, in order to stop/restart the timer and generate an interrupt.

3.3 Counter register, offset 0x18/0x28

This register holds the current value of the timer counter. It can be read or written at any time. Writing it does not clear the trigger bit (bit 0 of the timer control register). The trigger bit should always be manually reset.

4 Capabilities

The system controller provides a 32-bit value intended to be used as user-defined bit mask that defines the presence of certain peripherals or features in the bitstream. It is readable from register 0x38. It is defined using the `capabilities` input.

5 System identification

The system controller provides a 32-bit value defined at synthesis time that can be used to identify bitstreams or boards. The value is set by the `systemid` Verilog parameter and read using the register 0x3c.

Writing any value to this register sends a hard system reset by asserting the `hard_reset` output.

Copyright notice