

LC77700B

(PowerPC 405IAP Embedded Processor)

User's Manual

CPM

(Clock & Power Management)

Date:6/14/2001
Revision:1.01

Contents

1. CPM Overview	3
2. CPM Class	3
3. CPM Programing	4
4. CPM Registers	4
4.1 CPM Enable Register (CPMER)	4
4.2 CPM Force Register (CPMFR)	4
4.3 CPM Status Register (CPMSR)	4
5. Signal Description	5
5.1 System Interface	5
5.2 DCR Interface	5
5.3 CPM Interface	5
5.4 Interrupt Interface	6

1. CPM Overview

CPM : Clock & Power Management

The CPM controller supports three different types of sleep interfaces to the functional units:

-CLASS1 Interface:

The CPM_Sleep_N signal is asserted by the CPM controller when a register bit is set by software. The functional unit is unconditionally put to sleep.

-CLASS2 Interface:

If sleeping is permissible, the functional unit asserts the Sleep_Req signal to the CPM controller which responds by asserting CPM_Sleep_N if the Enable for that unit is set. The CPM_Sleep_N signal to a class2 unit is deasserted when the CPM controller's Enable bit for that unit is reset, or when the unit deasserts its Sleep_Req signal. In order for the wake-up to succeed, the unit logic that recognizes deassertion of the CPM_Sleep_N signal must be in a different clock domain that is not put to sleep.

-CLASS3 Interface:

The CPM class3 interface has a CPM_SleepInit signal that is asserted by the CPM controller to request that a functional unit go to sleep. If it can be put to sleep, the unit asserts the Sleep_Req signal to the CPM controller. The CPM_Sleep_N signal is then asserted by the controller to shut off the class3 clocks in the functional unit. Either the functional unit or the CPM controller can end the sleep state. If the CPM controller's Enable bit for that unit is reset it immediately deasserts CPM_SleepInit and CPM_Sleep_N. As with the class2 unit, the unit logic that recognizes deassertion of the CPM_Sleep_N signal must be in a different clock domain that is not put to sleep.

2. CPM Class

Each functional unit has one bit in each of the three CPM registers assigned to it. The bit assigned is the same in all three registers. Table 1 shows the bit assignment and CPM class for each of the functional units.

Table 1 CPM Class

Bit Assignment	IP	Class
0	IrDA	Class3
1	IEEE 1284	Class3
2	Touch Panel Controller	Class3
3	PowerPC405 Processor	Class2
4	DMA Controller	Class2
5	PLB to OPB Bridge	Class2
6	OPB to PLB Bridge	Class2
7	PLB Bus Arbiter	Class2
-	-	-
9	SDRAM Memory Controller	Class2
10	EBC(External Bus Controller)	Class2
11	Smart Card	Class2
12	PCMCIA	Class1
13	UIC	Class1
14	GPIO0	Class1
15	GPIO1	Class1
16	GPIO2	Class1
17	GPIO3	Class1
18	GPIO4	Class1
19	UART0	Class1

20	UART1	Class1
21	UART2	Class1
22	HDLC0	Class1
23	HDLC	Class1
24	IOM2	Class1
25	CPU Timers	Class1
26	GPT	Class1
27	PS2	Class1
28	KeyScan(16*10)	Class1
29	KeyScan(6*4)	Class1
30	USB Host	Class1
31	USB Device	Class1
	Serial Comm Port	No Support
	LCD Controller	No Support
	OPB Bus Arbiter	No Support

3. CPM Programing

The CPM controller is programmable via the DCR bus .

4. CPM Registers

Table2 lists the registers used to program the CPM controller.

Table2 CPM Registers

Register Name	DCR Address	Access	Default(0:31)
CPMER[0:31]	0x044	Read/Write	0x0000_0000
CPMFR[0:31]	0x045	Read/Write	0x0000_0000
CPMSR[0:31]	0x046	Read Only	0x0000_0000

4.1 CPM Enable Register (CPMER)

The bits in this register are used to enable the process of putting a functional unit to sleep.The way in which the interface signals to the functional units are controlled when the Enable bit is set depends upon the class of the unit.The interface signals are controlled as follows:

Class1:When the bit is set, the CPM_Sleep_N signal to the class1 unit is asserted. When this bit is reset the CPM_Sleep_N signal is deasserted.

Class2:When the bit is set, and the Sleep_Req signal from the class2 unit is asserted (the unit is requesting sleep state) the CPM_Sleep_N signal to the class2 unit is asserted.When this bit is reset the CPM_Sleep_N signal is deasserted.

Class3:When the bit is set,the CPM_SleepInit signal to the class3 unit is asserted (the CPM controller is requesting permission to put the unit to sleep).When the class3 unit responds by activating the Sleep_Req (the unit is giving permission to be put to sleep), the CPM_Sleep_N signal is asserted. When this bit is reset the CPM_SleepInit and the CPM_Sleep_N signals are deasserted.

4.2 CPM Force Register (CPMFR)

Setting a bit in this register forces assertion of the CPM_Sleep_N signal to the functional unit.For a class1 unit this is the same as setting the Enable bit for that unit. For a class2 or class3 unit the CPM_Sleep_N is asserted regardless of the state of the Sleep_Req signal coming from that unit.

4.3 CPM Status Register (CPMSR)

The CPM Status Register is a read-only register that shows the current state of all the CPM_Sleep_N signals.

5. Signal Description

5.1 System Interface

Table3 System Interface

Signal Name	I/O	Function
sys_clk	IN	System clock
reset	IN	reset

5.2 DCR Interface

Table4 DCR Interface

Signal Name	I/O	Function
dcrwrite	IN	PPC405 -> CPM.DCR bus write.
dcrread	IN	PPC405 -> CPM.DCR bus read.
dcrabus[0:9]	IN	PPC405 -> CPM.DCR address bus
dcrack	OUT	CPM -> OR_logic.Acknowledge to PPC405
dcrbus_from[0:31]	IN	DCR data input.
dcrbus_to[0:31]	OUT	DCR data output.

5.3 CPM Interface

Table5 CPM Interface

Unit	Class	Signal Name	I/O	Function
IrDA	3	cpm_sleep_init_irda	OUT	Sleep request to IrDA
		irda_sleep_req	IN	Sleep enable from IrDA
		cpm_sleep_irda_n	OUT	Shut off the clock of IrDA
IEEE1284	3	cpm_sleep_init_ppu	OUT	Sleep request to IEEE1284
		ppu_sleep_req	IN	Sleep enable from IEEE1284
		cpm_sleep_ppu_n	OUT	Shut off the clock of IEEE1284
TPC	3	cpm_sleep_init_tpc	OUT	Sleep request to TPC
		tpc_sleep_req	IN	Sleep enable from TPC
		cpm_sleep_tpc_n	OUT	Shut off the clock of TPC
PPC405	2	ppc_sleep_req	IN	Sleep enable from PPC405
		cpm_sleep_ppc_n	OUT	Shut off the clock of PPC405
DMA controller	2	dma_sleep_req	IN	Sleep enable from DMA controller
		cpm_sleep_dma_n	OUT	Shut off the clock of DMA controller
PLB_OPB bridge	2	plbopb_sleep_req	IN	Sleep enable from PLB_OPB bridge
		cpm_sleep_plbopb_n	OUT	Shut off the clock of PLB_OPB bridge
OPB_PLB bridge	2	opbplb_sleep_req	IN	Sleep enable from OPB_PLB bridge
		cpm_sleep_opbplb_n	OUT	Shut off the clock of OPB_PLB bridge
PLB arbiter	2	plbabt_sleep_req	IN	Sleep enable from PLB arbiter
		cpm_sleep_plbabt_n	OUT	Shut off the clock of PLB arbiter
SDRAM controller (HSPLB)	2	hsplb_sleep_req	IN	Sleep enable from HSPLB
		cpm_sleep_hsplb_n	OUT	Shut off the clock of HSPLB
External Bus Controller	2	ebc_sleep_req	IN	Sleep enable from EBC
		cpm_sleep_ebc_n	OUT	Shut off the clock of EBC
SCI(SmartCard interface)	2	sci_sleep_req	IN	Sleep enable from SCI
		cpm_sleep_sci_n	OUT	Shut off the clock of SCI
PCMCIA	1	cpm_sleep_pcmcia_n	OUT	Shut off the clock of PCMCIA
UIC	1	cpm_sleep_uic_n	OUT	Shut off the clock of UIC
GPIO0	1	cpm_sleep_gpio0_n	OUT	Shut off the clock of GPIO0
GPIO1	1	cpm_sleep_gpio1_n	OUT	Shut off the clock of GPIO1
GPIO2	1	cpm_sleep_gpio2_n	OUT	Shut off the clock of GPIO2
GPIO3	1	cpm_sleep_gpio3_n	OUT	Shut off the clock of GPIO3
GPIO4	1	cpm_sleep_gpio4_n	OUT	Shut off the clock of GPIO4

UART0	1	cpm_sleep_uart0_n	OUT	Shut off the clock of UART0
UART1	1	cpm_sleep_uart1_n	OUT	Shut off the clock of UART1
UART2	1	cpm_sleep_uart2_n	OUT	Shut off the clock of UART2
HDLC0	1	cpm_sleep_hdlc0_n	OUT	Shut off the clock of HDLC0
HDLC1	1	cpm_sleep_hdlc1_n	OUT	Shut off the clock of HDLC1
IOM2	1	cpm_sleep_iom2_n	OUT	Shut off the clock of IOM2
CPU timer	1	cpm_sleep_ppctimer_n	OUT	Shut off the clock of GPU timer
GPT	1	cpm_sleep_gpt_n	OUT	Shut off the clock of GPT
Keyscan(16*10)	1	cpm_sleep_kscn1610_n	OUT	Shut off the clock of Keyscan(16*10)
Keyscan(6*4)	1	cpm_sleep_kscn64_n	OUT	Shut off the clock of Keyscan(6*4)
PS2	1	cpm_sleep_ps2_n	OUT	Shut off the clock of PS2
USB Host	1	cpm_sleep_usbh_n	OUT	Shut off the clock of USB Host
USB Device	1	cpm_sleep_usbd_n	OUT	Shut off the clock of USB Device

5.4 Interrupt Interface

Table6 Interrupt Interface

Signal Name	I/O	Function
msr_ee	IN	PPC405 -> CPM. This signal, which implements the MSR External Interrupt Enable bit (MSR[EE]), is input from PPC405 unit. This signal enables external interrupts to wake the processor from asleep mode.
msr_ce	IN	PPC405 -> CPM. This signal, which implements the MSR Critical Interrupt Enable bit (MSR[CE]), is input from PPC405 unit. This signal enables critical interrupts to wake the processor from a sleep mode.
ppc_timer_ireq	IN	PPC405 -> CPM. This timer interrupt request signal is input from PPC405 unit as a method of waking the processor from a sleep mode. It is the logical OR of timer interrupt requests from the Programmable Interval Timer (PIT) and Fixed Interval Timer (FIT), which are enabled by MSR[EE], and the watchdog timer, which is enabled by MSR[CE]. The logical AND of MSR[CE, EE] with the appropriate timer interrupt is done within the PPC405.
ppc_timerreset_ireq	IN	PPC405 -> CPM. This timer reset request signal is input from PPC405 unit as a method of waking the processor from a sleep mode. It is the logical OR of the system, chip, and core reset request signals generated by the second expiration of the watchdog timer.
uic_crit_ireq	IN	UIC -> CPM. This interrupt request signal is input from UIC(Universal Interrupt Controller) unit as a method of waking the processor from a sleep mode. This signal is enabled by msr_ce.
uic_noncrit_ireq	IN	UIC -> CPM. This interrupt request signal is input from UIC(Universal Interrupt Controller) unit as a method of waking the processor from a sleep mode. This signal is enabled by msr_ee.

If CPMER[3] is "1", PPC405 is a sleep mode. Interrupt signals wake the PPC405 from a sleep mode.

The following statement is the condition that CPMER[3] changes to "0".

```
assign reset_cpmer3 = (uic_noncrit_ireq & msr_ee) | (uic_crit_ireq & msr_ce) | ppc_timerreset_ireq | ppc_timer_ireq;
```

- Any and all SANYO products described or contained herein do not have specifications that can handle applications that require extremely high levels of reliability, such as life-support systems, aircraft's control systems, or other applications whose failure can be reasonably expected to result in serious physical and/or material damage. Consult with your SANYO representative nearest you before using any SANYO products described or contained herein in such applications.
- SANYO assumes no responsibility for equipment failures that result from using products at values that exceed, even momentarily, rated values (such as maximum ratings, operating condition ranges, or other parameters) listed in products specifications of any and all SANYO products described or contained herein.
- Specifications of any and all SANYO products described or contained herein stipulate the performance, characteristics, and functions of the described products in the independent state, and are not guarantees of the performance, characteristics, and functions of the described products as mounted in the customer's products or equipment. To verify symptoms and states that cannot be evaluated in an independent device, the customer should always evaluate and test devices mounted in the customer's products or equipment.
- SANYO Electric Co., Ltd. strives to supply high-quality high-reliability products. However, any and all semiconductor products fail with some probability. It is possible that these probabilistic failures could give rise to accidents or events that could endanger human lives, that could give rise to smoke or fire, or that could cause damage to other property. When designing equipment, adopt safety measures so that these kinds of accidents or events cannot occur. Such measures include but are not limited to protective circuits and error prevention circuits for safe design, redundant design, and structural design.
- In the event that any or all SANYO products (including technical data, services) described or contained herein are controlled under any of applicable local export control laws and regulations, such products must not be exported without obtaining the export license from the authorities concerned in accordance with the above law.
- No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording, or any information storage or retrieval system, or otherwise, without the prior written permission of SANYO Electric Co., Ltd.
- Any and all information described or contained herein are subject to change without notice due to product/technology improvement, etc. When designing equipment, refer to the "Delivery Specification" for the SANYO product that you intend to use.
- Information (including circuit diagrams and circuit parameters) herein is for example only; it is not guaranteed for volume production. SANYO believes information herein is accurate and reliable, but no guarantees are made or implied regarding its use or any infringements of intellectual property rights or other rights of third parties.