Use Vortex86 Watchdog Timer

2003-10-24

The watchdog timer work flow of Vortex86 is: If the watchdog timer expires the first time, the expired event will set SFTMR0_STS and timer will reload its initial value and count again. If the timer expire the second time, the expired event will set SFTMR1_STS.

Software Watchdog Timer Initial Value: Default Value: FFh			
I/O Address	Bit	Access	Description
84Ah	7:0	R/W	Software Watchdog Timer Initial Value Writing to this register will reload the software watchdog timer with the value specified in this register. If the software watchdog timer expires the first time, the expired event will set the SFTMR0_STS and the timer will reload its initial value and count again. If the timer expire the second time, the expired event will set the SFTMR1_STS. The timer value can't be read from this field.
Software Watchdog Timer Control Register: Default Value: 00h			
I/O Address	Bit	Access	Description
84Bh	7	R/W	Software Watchdog Timer Counting Enable The software watchdog timer will start to count when this bit is set to one.
	6	RO	Reserved
	5:4	R/W	Software Watchdog Timer Clock Select 00 : 4 ms 01 : 1 second 10 : 1 minute 11 : 1 hour
	3:2	R/W	Software Watchdog Timer Expiration Event 1 Routing Select When SFTMR1_STS is set to one, an SMI#/SFTIRQ/PCIRST# will be generated according to the following combination. 00 : No effect 01 : SMI# 10 : SFTIRQ 11 : PCIRST#
	1:0	R/W	Software Watchdog Timer Expiration Event 0 Routing Select When SFTMR0_STS is set to one, an SMI#/SFTIRQ/PCIRST# will be generated according to the following combination. 00 : No effect 01 : SMI# 10 : SFTIRQ

			11 : PCIRST#
Legacy Event Status Register: Default Value: 00h			
I/O Address	Bit	Access	Description
841h	7	R/WC	Software Watch Dog Timer Event 1 Status (SFTMR1_STS) This bit is set when the software watchdog timer expires the second time. This status bit does not have its corresponding enable bit and can survive under PCIRST#.
	6	R/WC	Software Watch Dog Timer Event 0 Status (SFTMR0_STS) This bit is set when the software watchdog timer expires the second time. This status bit does not have its corresponding enable bit and can survive under PCIRST#.

C Example

Those C codes for DOS will show you more:

```
#include <conio.h>
#include <stdio.h>
#include <time.h>

void main()
{
  clock_t clk;
  int    nTime = 5;

  /* set time out */
  outp(0x84a, nTime);

  /* set timer clock to 1 second and "Timer Expiration Event 0/1" to reset system. */
  outp(0x84b, 0x9c);

  printf("Press any key to stop clearing watchdog timer status...\n");
  while(!kbhit())
  {
    /* clear "Timer Expiration Event 0/1" bit */
    outp(0x841, 0xc0);
  }

  getch();

  printf("System will be reset after %d seconds.\n", nTime * 4);
```

```
  clk = clock();
  while(!kbhit())
    printf("%2.2f\r", (clock() - clk) / CLK_TCK);
}
```

Assembler Example code

```
  mov dx,84ah ; set timeout = 20 second
  mov al,5
  out dx,al
  mov dx,84bh ; set timer clock to 1 second and "Timer Expiration Event 0/1" to reset system.
  mov al,9ch
  out dx,al

  ; clearing watchdog timer status
  mov dx,841h
  mov al,0c0h
  out dx,al
```

Technical Support

For more technical support, please visit http://www.dmp.com.tw/tech or mail to tech@dmp.com.tw.