Computer Operating Properly (COP) Specification

Author: Robert Hayes
rehayes@opencores.org

Rev. 0.1

June 16, 2009

This page has been intentionally left blank.

Revision History

Rev.	Date	Author	Description
0.1	03/06/09	Robert Hayes	First draft release

Contents

1 Introduction

The Computer Operating Properly Module, COP, is a watchdog timer module that triggers a system reset if it is not regularly serviced by writing two specific words to its control registers. The intention of the module is to bring an embedded system back to a "good" state after the software program has lost control of the system.

FEATURES

- **Programmable Watchdog period.**
- **Optional programmable interrupt before system reset**
- **Flexible inputs to control wait, stop and debug mode**
- **Static synchronous design**
- **Fully synthesizable**

2 Architecture

WISHBONE COMPATIBLE

The COP core is built around three primary blocks; the WISHBONE Interface, the Control Registers, and the Watchdog Counter.

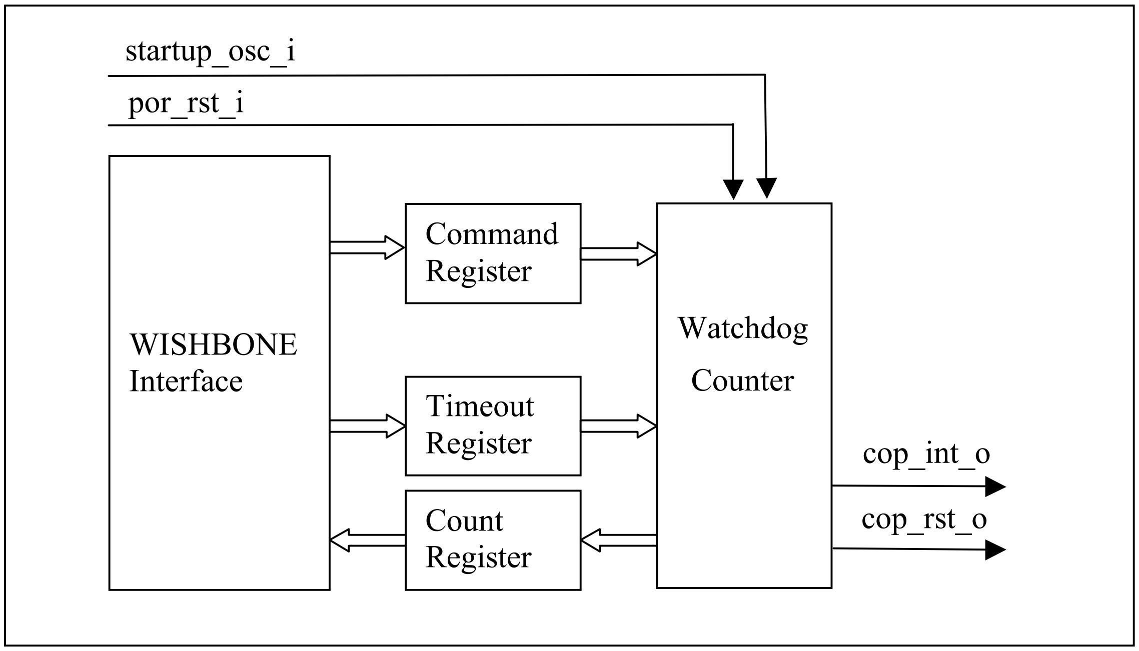

Fig. 2.1 Internal structure COP Core

2.1 WISHBONE Interface

The WISHBONE Interface isolates the COP functionality from the WISHBONE bus. This interface takes the bus specific signals and generates a generic set of control signals to drive the COP control registers. Isolating the WISHBONE bus should help promote COP module reusability by localizing the scope of changes needed to retarget the COP module to another bus environment.

2.2 Control Registers

The Command Register, Timeout Register and Count Register are combined into a single module that controls the programmable functions of the COP. Various bits in the Command Register define the basic operating mode and function enables of the COP.

2.3 Watchdog Counter

The Watchdog Counter is the key element of the COP module. This counter is preloaded with a starting value programmed into the Timeout Register and then decrements with each positive edge of the startup_osc_clk_i clock signal. When the Watchdog counter has counted down to zero the cop_rst_o signal is asserted to reset the embedded system and the system software back to a "good" condition. The COP reset should also set any system hardware to a "safe" condition to minimize the chance of any hardware damage. To prevent the Watchdog Counter from reaching zero two specific words must be written to the Count Register, which will cause the Watchdog Counter to be reinitialized with the value in the Timeout Register.

3 Operation

The COP Module is a very simple counter that can be controlled by commands from the WISHBONE bus. The output from the module is a pulse that is one startup_osc_i period wide and should be OR'ed with the system reset signal. The COP watchdog counter is set to a value programmed into the TOUT_VAL register each time the COP is disabled with the COP_ENA control bit or when two specific words are written to the COP_CNT register. If the COP watchdog counter is allowed to decrement down to the zero value then the cop_rst_o will become active to force a system reset.

The COP module also has the capability to generate an interrupt at a programmed number of cycles before the cop_rst_o initiates a system reset. This functionality is primarily intended as a debug feature.

A very readable article about watchdog timers and their use:

http://www.ganssle.com/watchdogs.pdf

The recommended software procedure for using the COP Module:

1. Initialize COP
   a) Check COP_EVENT status bit to find out what caused the last system reset.
   b) Clear COP_ENA
   c) Set TOUT_VAL to the maximun time to run without a COP service access.
   d) Set COP_IRQ, DEBUG_ENA, STOP_ENA, WAIT_ENA as required
   e) Set COP_ENA
   f) Set CWP and CLCK as required
2. Normal Operation
   a) Service COP at regural intervals by writing the two correct service words to the COP_CNT register.

4

Registers

The COP Module can be configured through the use of the DWIDTH parameter to have a WISHBONE bus interface with either an 8-bit bus with 8-bit granularity or to use a 16-bit bus with 16-bit granularity. This document shows the resultant address and bit field for both configurations although in an actual instance of the COP Module only one of the pairs of tables will be valid. For an end user or programmer reference it may be best to simplify this document by removing the tables that reference the unused bus configuration.

List of Registers

Name	Address	Width	Access	Description
CNTRL	0x00	16	RW	COP Control Register
TOUT	0x01	16	RW	COP Timeout Register
CNT	0x02	16	R	COP Counter Value, Service Word

Table 1: List of registers, 16 bit data (default DWIDTH=16)

Name	Address	Width	Access	Description
CNTRL_0	0x00	8	RW	COP Control Register Low
CNTRL_1	0x01	8	RW	COP Control Register High
TOUT	0x02	8	RW	COP Timeout Register
Reserved	0x03	8	R	
CNT_0	0x04	8	R	COP Counter Value Low Byte
CNT_1	0x05	8	R	COP Counter Value High Byte

Table 1a: List of registers, 8 bit data (DWIDTH=8)

4.1 CNTRL Register

Bit #	Access	Description
15:9	R	Reserved, write zeros for future compatibility.
8	R/W	COP_EVENT, Status bit to record that a COP reset has happened. To clear the COP_EVENT status bit write a '1' to the COP_EVENT bit or the bit will be cleared when a correct COP service request is executed.
7:6	RW	COP_IRQ '00' No COP interrupt '01' COP interrupt 16 osc clocks before COP Reset '10' COP interrupt 32 osc clocks before COP Reset '11' COP interrupt 64 osc clocks before COP Reset
5	RW	DEBUG_ENA, Enable COP in DEBUG Mode '0' COP Counter stops in DEBUG Mode. '1' COP Counter continues to run in DEBUG Mode. DEBUG_ENA can only be changed when the CWP bit is set to zero.
4	RW	STOP_ENA, Enable COP in STOP Mode. '0' COP Counter stops in STOP Mode. '1' COP Counter continues to run in STOP Mode. STOP_ENA can only be changed when the CWP bit is set to zero.
3	RW	WAIT_ENA, Enable COP in WAIT Mode. '0' COP Counter stops in WAIT Mode. '1' COP Counter continues to run in WAIT Mode. WAIT_ENA can only be changed when the CWP bit is set to zero.
2	RW	COP_ENA, Enable COP Watchdog Counter to begin decrementing. '0' COP Watchdog Counter is disabled. The COP Watchdog Counter is set to the Timeout value. '1' COP Watchdog Counter is enabled. COP_ENA can only be changed when the CWP bit is set to zero. Other bits in this register and the Timeout value can only

Bit #	Access	Description
		be changed when the COP_ENA bit is '0'.
1	RW	CWP, COP Write Protect. When set to '1' the COP_ENA cannot be changed. When set to '0' the COP_ENA bit can be changed.
0	RW	CLCK, COP LOCK. '0' CWP and be written. '1' CWP is locked in its current state and cannot be changed except by a system reset.

Reset Value:

CNTRL: 0004h

Table 2: CNTRL Register Bits

The COP CNTRL register has three bits that can limit the ability to change other bits in the CNTRL register when they are set. These bits are COP_ENA, CWP, and CLCK. There is a hierarchy of protection, the COP_ENA inhibits changes to the WAIT_ENA, STOP_ENA, DEBUG_ENA and TOUT_VAL registers, the CWP inhibits changes to the COP_ENA bit, and the CLCK inhibits changes to the CWP bit. By using the different levels of protection different levels of system security can be obtained while still allowing some flexibility in controlling the COP functionality. The final step of initializing the COP should be to set the CLCK bit.

The protection mechanism is arranged such that a protection bit and the bits that it protects can be set in a single write command. To change a protected bit after its associated protection control bit is set requires two write commands, the first to clear the protection control bit and the second to operate on the protected bit targeted for change.

CLCK	CWP	COP_ENA	Status
1	1	1	Most protected, least control: The COP is always enabled and no changes can be made to the control bits.
0	1	1	Some protection, some control: The COP is enabled and cannot be disabled without first clearing the CWP bit which takes two writes to the CNTRL register. The first write is to clear the CWP bit and the second write is to clear the COP_ENA bit. Because CLCK is not set, if CLCK does get unintentionally set then control becomes a function of whatever the settings of the CWP bit was at the time that CLCK was set.
1	0	X	Least protected, most control: Software always has control over the COP_ENA bit and can enable and disable COP functionality on demand.
1	1	0	COP disabled till next system reset.

Bit #	Access	Description
7:1	R	Reserved, write zeros for future compatibility.
0	R/W	COP_EVENT, Status bit to record that a COP reset has happened. To clear the COP_EVENT status bit write a '1' to the COP_EVENT bit or the bit will be cleared when a correct COP service request is executed.

Reset Value:

CNTRL_1: 00h

Table 2a: CNTRL_1 Register Bits

Bit #	Access	Description
7:6	RW	COP_IRQ '00' No COP interrupt '01' COP interrupt 16 osc clocks before COP Reset '10' COP interrupt 32 osc clocks before COP Reset '11' COP interrupt 64 osc clocks before COP Reset

Bit #	Access	Description
5	RW	DEBUG_ENA, Enable COP in DEBUG Mode '0' COP Counter stops in DEBUG Mode. '1' COP Counter continues to run in DEBUG Mode. DEBUG_ENA can only be changed when the CWP bit is set to zero.
4	RW	STOP_ENA, Enable COP in STOP Mode. '0' COP Counter stops in STOP Mode. '1' COP Counter continues to run in STOP Mode. STOP_ENA can only be changed when the CWP bit is set to zero.
3	RW	WAIT_ENA, Enable COP in WAIT Mode. '0' COP Counter stops in WAIT Mode. '1' COP Counter continues to run in WAIT Mode. WAIT_ENA can only be changed when the CWP bit is set to zero.
2	RW	COP_ENA, Enable COP Watchdog Counter to begin decrementing. '0' COP Watchdog Counter is disabled. The COP Watchdog Counter is set to the Timeout value. '1' COP Watchdog Counter is enabled. COP_ENA can only be changed when the CWP bit is set to zero. Other bits in this register and the Timeout value can only be changed when the COP_ENA bit is '0'.
1	RW	CWP, COP Write Protect. When set to '1' the COP_ENA cannot be changed. When set to '0' the COP_ENA bit can be changed.
0	RW	CLCK, COP LOCK. '0' CWP and be written. '1' CWP is locked in its current state and cannot be changed except by a system reset.

Reset Value:

CNTRL_0: 04h

Table 2b: CNTRL_0 Register Bits

4.1 Timeout Register

16 Bit Data Bus

Bit #	Access	Description
15:0	RW	TOUT_VAL, COP Counter Watchdog value Sets the initial value of the COP Watchdog Counter after a proper COP service request or the changing of the COP_ENA bit to zero. The Watchdog Counter counts down from this value to zero when the COP reset is activated. These bits can only be changed when COP_ENA is zero.

Reset Value:
TOUT_VAL: FFFFh

Table 4: TOUT Register Bits

8 Bit Data Bus

Bit #	Access	Description
7:0	RW	TOUT_VAL[15:8], COP Counter Watchdog value Sets the initial value of the COP Watchdog Counter after a proper COP service request or the changing of the COP_ENA bit to zero. The Watchdog Counter counts down from this value to zero when the COP reset is activated. These bits can only be changed when COP_ENA is zero.

Reset Value:
TOUT_VAL_1: FFh

Table 4a: TOUT_VAL_1 Register Bits

Bit #	Access	Description
7:0	RW	TOUT_VAL[7:0], COP Counter Watchdog value Sets the initial value of the COP Watchdog Counter after a proper COP service request or the changing of the COP_ENA bit to zero. The Watchdog Counter counts down from this value to zero when the COP reset is activated. These bits can only be changed when COP_ENA is zero.

Reset Value:
TOUT_VAL_0: FFh

Table 4a: TOUT_VAL_0 Register Bits

4.2 CNT Register

16 Bit Data Bus

Bit #	Access	Description
15:0	RW	COUNT_VAL, Read returns the current state of the COP Watchdog Counter. Writes to the CNT register access the COP service words. To reset the COP Watchdog Counter to the TOUT_VAL state two specific service words must be written in the correct order to prevent a COP reset.

Reset Value:
CNT: FFFFh

Table 5: CNT Register Bits

8 Bit Data Bus

Bit #	Access	Description
7:0	RW	COUNT_VAL, Read returns the current state of the COP Watchdog Counter. Note: To minimize the gates there is no register to capture the full value of COUNT_VAL when only a byte is read. This means that the value that the processor sees is only the approximate value of COUNT_VAL because one of the bytes will have changed between the read of the first byte of COUNT_VAL and the second byte of COUNT_VAL. Writes to the CNT register access the COP service words. To reset the COP Watchdog Counter to the TOUT_VAL state two specific service words must be written in the correct order to prevent a COP reset.

Reset Value:
CNT_0: FFh

Table 5a: CNT_0 Register Bits

Bit #	Access	Description
7:0	RW	COUNT_VAL, Read returns the current state of the COP Watchdog Counter.

Reset Value:

CNT_1: FFh

Table 5b: CNT_1 Register Bits

5

Clocks

Name	Source	Rates (MHz)			Remarks	Description
		Max	Min	Resolution		
wb_clk_i	System	200	-	-	Master clock for all COP bus registers. Positive edge active.	System clock.
startup_osc_i	System					COP counter clock
run_osc_i	System					COP counter cock

Table 3: List of clocks

The wb_clk_i has no timing constraints based on the RTL implementation although there may be constrains applied for synthesis results to be compatible with the target physical implementation. If the COP is targeted for an ASIC implementation then [wb_clk_i] should be used as the scan clock, any clock multiplexing required to make [wb_clk_i] the scan clock should be done at the system level external to the COP Module.

The startup_osc_i is the clock used to decrement the Watchdog Counter. The frequency of this clock is assumed to at least half the speed of wb_clk_i. The phase of startup_osc_i relative to wb_clk_i is assumed to be unknown. Resynchronization of data and control signals crossing clock domains will be required to reliably transfer signals between clock wb_clk_i and startup_osc_i clock domains.

6

IO Ports

Port	Width	Direction	Description
wb_clk_i	1	Input	WISHBONE Bus Clock Input, Master Clock
wb_rst_i	1	Input	WISHBONE Synchronous Reset
wb_adr_i	3	Input	WISHBONE Lower address bits
wb_dat_i	8/16	Input	WISHBONE Bus Data
wb_dat_o	8/16	Output	WISHBONE Bus Data
wb_we_i	1	Input	WISHBONE Write enable
wb_stb_i	1	Input	WISHBONE Strobe signal/Core select
wb_cyc_i	1	Input	WISHBONE Valid bus cycle
wb_sel_i	2	Input	WISHBONE Data Bus Byte Select
wb_ack_o	1	Output	WISHBONE Bus cycle acknowledge
cop_irq_o	1	Output	COP Interrupt signal
cop_rst_o	1	Output	COP output signal
startup_osc_i	1	Input	Watchdog Counter Clock
stop_mode_i	1	Input	System is in STOP Mode
wait_mode_i	1	Input	System is in WAIT Mode
debug_mode_i	1	Input	System is in DEBUG Mode
por_reset_i	1	Input	Power on Reset
arst_i	1	Input	Asynchronous Reset

Port	Width	Direction	Description
scantestmode_i	1	Input	Scan Test Mode Enable

Table 4: List of IO ports

6.1 WISHBONE Interface

The core features a WISHBONE RevB.3 compliant WISHBONE Classic interface that operates in SLAVE mode. All output signals are registered. Each access takes 2 clock cycles. To limit a WISHBONE access to just two clock cycles the following synthesis rules should be used:

- Single cycle timing for wb_cyc_i and wb_stb_i
- Two cycle timing for wb_adr_i, wb_data_i, and wb_data_o. (Single cycle timing could be used but it would be a waste of resources to meet an over constrained timing path.)

Note: Use the "SINGLE_CYCLE" parameter to do a WISHBONE bus access in one clock cycle.

WISHBONE DATASHEET		
Description	Specification	
General description:	8-bit SLAVE	
Supported Cycles:	SLAVE, READ/WRITE	
Data port, size: Data port, granularity: Data port, maximum operand size: Data transfer ordering: Data transfer sequencing:	Default: 16, option 8 bit Default: 16, option to match 8 bit port size	
Supported signal list and cross reference to equivalent WISHBONE signals:	Signal Name	WISHBONE Equiv.
	wb_clk_i	CLK_I
	wb_rst_i	RST_I
	wb_adr_i	ADR_I()

	wb_dat_i	DAT_I()
	wb_dat_o	DAT_O()
	wb_we_i	WE_I
	wb_stb_i	STB_I
	wb_cyc_i	CYC_I
	wb_sel_i	SEL_I
	wb_ack_o	ACK_O

6.1.1 wb_rst_i

The synchronous reset signal has a minimum pulse width requirement of one [wb_clk_i] clock period. It will take two [wb_clk_i] clock cycles for all registers in the COP Module to initialize. Also see information on pin [arst_i]. It is assumed that elsewhere in the system cop_rst_o will be OR'ed with this signal and it will be active for some time during and after a COP watchdog timeout.

6.1.2 wb_adr_i

Connections to the WISHBONE address pin will depend on the size of the WISHBONE data bus that is set by the DWIDTH parameter. If DWIDTH=8 the all address pins should be connected, if DWIDTH=16 then [wb_adr_i(2)] should be tied low.

6.1.3 wb_sel_i

The [wb_sel_i] is the WISHBONE byte lane select signal. It is currently unimplemented in the PIT module and should be tied hi.

6.2 COP signals

6.2.1 cop_rst_o

The COP output signal. This is a one oscillator clock, [startup_osc_i], wide pulse that is output when the Watchdog Counter reaches zero.

6.2.2 cop_irq_o

This signal is an optional output that is activated at some number of [startup_osc_i] clocks before the Watchdog Counter reaches zero and resets the system. The [cop_irq_o] signal is primarily intended as a debug aid and its use is discouraged in final production code. Except as part of a temporary software debugging patch it is recommended not to put code to service the COP into the interrupt service routine. This is because the state of the system is unknown and the state of the stack and stack pointer may be corrupted. Code to put the system into a "safe" state and store debug information should be included in the interrupt service routine. This signal may be occasionally useful when an output signal with a longer pulse width is needed to be resynchronized to a slower clock domain than the COP master clock.

6.2.3 startup_osc_i

This signal is the input clock for the COP Watchdog Counter. This clock is assumed to be running at all times.

6.2.4 por_reset_i

The [por_reset_i] signal is active low. This signal is used to initialize a limited number flip-flops that need to maintain state when [arst_i] or [wb_rst_i] are active. A separate reset signal is required for these flip_flops because it is assumed that [cop_rst_o] will effect [arst_i] and [wb_rst_i] and may cause a shortening of the [cop_rst_o] pulse width because of a combinational feedback path.

6.2.5 arst_i

The signal [arst_i] is an asynchronous reset signal that goes to all flops in the COP. It is provided for FPGA implementations and test methodologies that require this function for initialization. Using [arst_i] instead of [wb_rst_i] can result in lower cell-usage and higher performance for a FPGAs implementation because the standard FPGA cell already provides a dedicated asynchronous reset path. Using [wb_rst_i] for an ASIC implementation might synthesize to a smaller module because smaller non_reset flops can be used. Use either [arst_i] or [wb_rst_i], tie the other to a negated state. The active level of [arst_i] is determined by the parameter ARST_LVL that defaults to active low.

6.2.6 stop_mode_i

The Watchdog Counter can be frozen in its current state if this signal is active and the STOP_ENA bit is set to zero. If there is no stop mode signal available from the system then this input signal should be tied low.

6.2.7 wait_mode_i

The Watchdog Counter can be frozen in its current state if this signal is active and the WAIT_ENA bit is set to zero. If there is no stop mode signal available from the system then this input signal should be tied low.

6.2.8 debug_mode_i

The Watchdog Counter can be frozen in its current state if this signal is active and the DEBUG_ENA bit is set to zero. If there is no stop mode signal available from the system then this input signal should be tied low.

6.2.9 scantestmode_i

The [scantestmode_i] input is an optional signal used to put the module into scan test mode. When [scantestmode_i] is active the [startup_osc_clk_i] is replaced by the [wb_clk_i] clock so all register are clocked by a common clock source.

6.3 COP Core Parameters

Parameter	Type	Default	Description
ARST_LVL	Bit	1’b0	Asynchronous reset level
INIT_ENA	Bit	1’b1	COP on/off at power up
SERV_WD_0	Word	16’h5555	Compare value for first service word
SERV_WD_1	Word	16’hAAAA	Compare value for second service word
SINGLE_CYCLE	Bit	1’b0	WISHBONE wait state
DWIDTH	Int	16	Data Bus size

6.3.1 ARST_LVL

The asynchronous reset level can be set to either active high (1’b1) or active low (1’b0).

Allowed values: 1’b0, 1’b1

6.3.2 INIT_ENA

The initial value of the COP_ENA control bit can be set with this parameter. Some applications may prefer that the COP be initialized to the OFF state in which case this bit can be set to 1’b0. If the COP is not initialized to the ON state then there is no protection for a runaway code situation during the system initialization period.

Allowed values: 1’b0, 1’b1

6.3.3 SERV_WD_0

The value of the first service word to be written to the COP to reset the Watchdog timer. The default value of 16’h5555 may be used by some memory test routines so there might be better security if another value is chosen. If DWIDTH=8 then only the least significant byte is used.

Allowed values: any 16-bit value (16’hxxxx)

6.3.4 SERV_WD_1

The value of the second service word to be written to the COP to reset the Watchdog timer. The default value of 16’hAAAA may be used by some memory test routines so there might be better security if another value is chosen. If DWIDTH=8 then only the least significant byte is used.

Allowed values: any 16-bit value (16’hxxxx)

6.3.5 SINGLE_CYCLE

The default operation of the COP WISHBONE bus interface is to insert one wait state by delaying the assertion of the wb_ack_o by one wb_clk_i period. Setting the SINGLE_CYCLE parameter generates the wb_ack_o combinationaly so a WISHBONE bus cycle can be completed in one wb_clk_i period.

Allowed values: 1’b0, 1’b1

6.3.6 DWIDTH

The width of the microcontroller data buses connected to COP Module. The COP Module can support either an 8-bit data bus with 8-bit resolution or a 16-bit data bus with 16-bit resolution.

Allowed values: 8, 16

Appendix A

Name

[This section may be added to outline different specifications.]

Index

[This section contains an alphabetical list of helpful document entries with their corresponding page numbers.]