OVP Guide to Using Processor Models

Model Specific Information for variant renesas_v850_V850E2R

Imperas Software Limited
Imperas Buildings, North Weston
Thame, Oxfordshire, OX9 2HA, UK
docs@imperas.com

Author	Imperas Software Limited
Version	0.5
Filename	OVP_Model_Specific_Information_v850_V850E2R.pdf
Created	27 February 2018
Status	OVP Standard Release

Model Release Status

This model is released as part of OVP releases and is included in OVPworld packages. Please visit OVPworld.org.

Table of Contents

1 Overview

This document provides the details of an OVP Fast Processor Model variant.

OVP Fast Processor Models are written in C and provide a C API for use in C based platforms. The models also provide a native interface for use in SystemC TLM2 platforms.

The models are written using the OVP VMI API that provides a Virtual Machine Interface that defines the behavior of the processor. The VMI API makes a clear line between model and simulator allowing very good optimization and world class high speed performance.

Most models are provided as a binary shared object and also as source. This allows the download and use of the model binary or the use of the source to explore and modify the model.

The models are run through an extensive QA and regression testing process and most model families are validated using technology provided by the processor IP owners.

There is a companion document (OVP Guide to Using Processor Models) which explains the general concepts of OVP Fast Processor Models and their use. It is downloadable from the OVPworld website documentation pages.

1.1 Description

V850 Family Processor Model.

1.2 Licensing

Open Source Apache 2.0

1.3 Limitations

This variant is currently under development.

1.4 Verification

Models have been extensively tested by Imperas, In addition Verification suites have been supplied by Renesas for Feature Set validation

1.5 Features

2 Configuration

2.1 Location

The model source and object file is found in the VLNV tree at:
    renesas.ovpworld.org/processor/v850/1.0

2.2 GDB Path

The default GDB for this model is found at:
    $IMPERAS_HOME/lib/$IMPERAS_ARCH/gdb/v850-elf-gdb

2.3 Semi-Host Library

The default semi-host library file is found in the VLNV tree at :
    renesas.ovpworld.org/semihosting/v850Newlib/1.0

2.4 Processor Endian-ness

This is a LITTLE endian model.

2.5 QuantumLeap Support

This processor is qualified to run in a QuantumLeap enabled simulator.

2.6 Processor ELF Code

ELF codes supported by this model are: 0x57, 0x24, 0x70f1, 0x70ff and 0x747b.

3 Other Variants in this Model

Table 1. All variants in this model

Variant
V850
V850E1
V850E1F
V850ES
V850E2
V850E2M
V850E2R

4 Bus Ports

Table 2. Bus Ports

Type	Name	min	max
master (initiator)	INSTRUCTION	28	32
master (initiator)	DATA	28	32

5 Net Ports

Table 3. Net Ports

Name	Type	Description
intp	input	Interrupt Port
nmi0	input	Non-Maskable Interrupt Port
nmi1	input	Non-Maskable Interrupt Port
nmi2	input	Non-Maskable Interrupt Port
reset	input	Reset Port
mireti	output	Return from Interrupt Port
intack	output	Interrupt Acknowledge Port

6 FIFO Ports

No FIFO Ports in this model.

7 Parameters

Table 4. Parameters that can be set in the model, type:

Name	Type	Description
verbose	Boolean	Specify verbose output messages
GDBSIMMODE	Boolean	GDB Simulator Compatibility Mode
nofpu	Boolean	Disable Processor Internal FPU
RBASE	Uns32	RBASE register Reset vector Address
ucbank	Boolean	Enable the User Compatible Bank Registers (eg, VFOREST)
PEID	Uns32	Processor element number
SPID	Uns32	System Protection Number

8 Execution Modes

No execution modes.

9 Exceptions

Table 5. Exceptions handled by the model, type:

Name	Code	Description
reset	0	Reset Signal Exception
nmi0	16	Non Maskable Interrupt(0) Exception
nmi1	32	Non Maskable Interrupt(1) Exception
nmi2	48	Non Maskable Interrupt(2) Exception
intp	128	Maskable Interrupt Exception - Vector value = (0x0000ffff AND intp)
fetrap	48	FETRAP Exception
trap0	64	TRAP0 Exception
trap1	80	TRAP1 Exception
ilgop	96	Illegal OPCODE Exception
rie	48	Reserved Instruction Exception

10 Hierarchy of the model

A CPU core may allow the user to configure it to instance many processors of a Symmetrical Multi Processor (SMP). A CPU core may also have sub elements within a processor, for example hardware threading blocks.

OVP processor models can be written to include SMP blocks and to have many levels of hierarchy.

Some OVP CPU models may have a fixed hierarchy, and some may be configured by settings in a configuration register. Please see the register definitions of this model.

This model documentation shows the settings and hierarchy of the default settings for this model variant.

10.1 Level 1:

This level in the model hierarchy has 2 commands.

This level in the model hierarchy has 3 register groups:

Table 6. Register groups

Group name	**Registers**
User	32
System	65
Integration_support	8

This level in the model hierarchy has no children.

11 Model Commands

11.1 Level 1:

11.1.1 isync

specify instruction address range for synchronous execution

Table 7. isync command arguments

Argument	Type	Description
-addresshi	Uns64	end address of synchronous execution range
-addresslo	Uns64	start address of synchronous execution range

11.1.2 itrace

enable or disable instruction tracing

Table 8. itrace command arguments

Argument	Type	Description
-after	Uns64	apply after this many instructions
-enable	Boolean	enable instruction tracing
-instructioncount	Boolean	include the instruction number in each trace
-off	Boolean	disable instruction tracing
-on	Boolean	enable instruction tracing
-registerchange	Boolean	show registers changed by this instruction
-registers	Boolean	show registers after each trace

12 Registers

12.1 Level 1:

12.1.1 User

Table 9. Registers at level 1, type: , register group: 'User'

Name	Bits	Initial value (Hex)		Description
R0	32	0	r-	Zero Register
R1	32	0	rw	Assembler-reserved register
R2	32	0	rw	Address/data variable register (when the real-time OS to be used is not using r2)
R3	32	0	rw	Stack pointer (SP)
R4	32	0	rw	Global pointer (GP)
R5	32	0	rw	Test pointer (TP)
R6	32	0	rw	Address/data variable registers
R7	32	0	rw	Address/data variable registers
R8	32	0	rw	Address/data variable registers
R9	32	0	rw	Address/data variable registers
R10	32	0	rw	Address/data variable registers
R11	32	0	rw	Address/data variable registers
R12	32	0	rw	Address/data variable registers
R13	32	0	rw	Address/data variable registers

R14	32	0	rw	Address/data variable registers
R15	32	0	rw	Address/data variable registers
R16	32	0	rw	Address/data variable registers
R17	32	0	rw	Address/data variable registers
R18	32	0	rw	Address/data variable registers
R19	32	0	rw	Address/data variable registers
R20	32	0	rw	Address/data variable registers
R21	32	0	rw	Address/data variable registers
R22	32	0	rw	Address/data variable registers
R23	32	0	rw	Address/data variable registers
R24	32	0	rw	Address/data variable registers
R25	32	0	rw	Address/data variable registers
R26	32	0	rw	Address/data variable registers
R27	32	0	rw	Address/data variable registers
R28	32	0	rw	Address/data variable registers
R29	32	0	rw	Address/data variable registers
R30	32	0	rw	Element pointer (EP)
R31	32	0	rw	Link pointer (LP)

12.1.2 System

Table 10. Registers at level 1, type: , register group: 'System'

Name	Bits	Initial value (Hex)		Description
EIPC	32	0	rw	Interrupt status-saving register PC
EIPSW	32	0	rw	Interrupt status-saving register PSW
FEPC	32	0	rw	NMI status-saving register PC
FEPSW	32	0	rw	NMI status-saving register PSW
ECR	32	0	rw	Exception cause register
PSW	32	20	rw	Program status word
PID	32	0	r-	_UNIMPLEMENTED_
CFG	32	0	r-	_UNIMPLEMENTED_
SCCFG	32	0	r-	_UNIMPLEMENTED_
SCBP	32	0	r-	_UNIMPLEMENTED_
EIIC	32	0	rw	EI Cause Register
FEIC	32	0	rw	FE Cause Register
DBIC	32	0	rw	DB Cause Register
CTPC	32	0	rw	CALLT status-saving register PC
CTPSW	32	0	rw	CALLT status-saving register PSW
DBPC	32	0	rw	Exception/Debug trap status-saving register PC
DBPSW	32	0	rw	Exception/Debug trap status-saving register PSW
CTBP	32	0	rw	CALLT base pointer
DIR	32	0	rw	Debug Interface register

EIWR	32	0	rw	EIWR
FEWR	32	0	rw	FEWR
DBWR	32	0	rw	DBWR
BSEL	32	0	rw	Bank Select Register
PC	32	0	rw	Program Counter
FPVIP	32	0	r-	
VMECR	32	0	r-	
VMTID	32	0	r-	
VMADR	32	0	r-	
VPECR	32	0	r-	
VPTID	32	0	r-	
VPADR	32	0	r-	
VDECR	32	0	r-	
VDTID	32	0	r-	
MPM	32	0	r-	
MPC	32	0	r-	
TID	32	0	r-	
PPA	32	0	r-	
PPM	32	0	r-	
PPC	32	0	r-	
DCC	32	0	r-	
DCV0	32	0	r-	
DCV1	32	0	r-	
SPAL	32	0	r-	
SPAU	32	0	r-	
IPA0L	32	0	r-	
IPA0U	32	0	r-	
IPA1L	32	0	r-	
IPA1U	32	0	r-	
IPA2L	32	0	r-	
IPA2U	32	0	r-	
IPA3L	32	0	r-	
IPA3U	32	0	r-	
DPA0L	32	0	r-	
DPA0U	32	0	r-	
DPA1L	32	0	r-	
DPA1U	32	0	r-	
DPA2L	32	0	r-	
DPA2U	32	0	r-	
DPA3L	32	0	r-	
DPA3U	32	0	r-	
FPSR	32	20000	rw	Floating-point configuration/status

FPEPC	32	0	rw	Floating-point exception program counter
FPST	32	0	rw	Floating-point status
FPCC	32	0	rw	Floating-point comparison result
FPCFG	32	0	rw	Floating-point configuration

12.1.3 Integration_support

Table 11. Registers at level 1, type: , register group: 'Integration_support'

Name	Bits	Initial value (Hex)		Description
stop	32	0	rw	Support Register for Stopping Processor
ResultHi	32	0	rw	External ResultHi Register
ResultLo	32	0	rw	External ResultLo Register
ResultSz	8	0	rw	External ResultSz Register
ResultId	8	0	rw	External ResultId Register
ResultFlg	32	0	rw	External ResultFlg Register
ResultCC	32	0	rw	External ResultCC Register
FLG_LL	8	0	rw	Load/Store Exclusive FLG_LL

#

