OVP Guide to Using Processor Models

Model specific information for Renesas_RL78-S3

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

Author	Imperas Software Limited
Version	20211118.0
Filename	OVP_Model_Specific_Information_rl78_RL78-S3.pdf
Created	18 November 2021
Status	OVP Standard Release

Copyright Notice

Right to Copy Documentation

Destination Control Statement

Disclaimer

Model Release Status

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

Contents

1 Overview **1**
- 1.1 Description . . . 1
- 1.2 Licensing . . . 1
- 1.3 Reference . . . 1
- 1.4 Limitations . . . 2
- 1.5 Verification . . . 2
- 1.6 Features . . . 2

2 Configuration **3**
- 2.1 Location . . . 3
- 2.2 GDB Path . . . 3
- 2.3 Semi-Host Library . . . 3
- 2.4 Processor Endian-ness . . . 3
- 2.5 QuantumLeap Support . . . 3
- 2.6 Processor ELF code . . . 3

3 All Variants in this model **4**

4 Bus Master Ports **5**

5 Bus Slave Ports **6**

6 Net Ports **7**

7 FIFO Ports **8**

8 Formal Parameters **9**
- 8.1 Parameter values . . . 9

9 Execution Modes **10**

10 Exceptions **11**

11 Hierarchy of the model **12**
- 11.1 Level 1 . . . 12

12 Model Commands **13**
- 12.1 Level 1 . . . 13
  - 12.1.1 isync . . . 13

12.1.2 itrace 13

13 Registers **14**
13.1 Level 1 14
13.1.1 GPR 14
13.1.2 Bank1 14
13.1.3 Bank2 14
13.1.4 Bank3 15
13.1.5 Bank4 15
13.1.6 System 15

Chapter 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

RL78 Family Processor Model.

1.2 Licensing

Open Source Apache 2.0

1.3 Reference

RL78 User Manual: Software, Single-Chip microcontrollers, http://documentation.renesas.com/doc/products/mpumcu/doc/rl78/r01us0015ej0220_rl78.pdf

1.4 Limitations

The PMC (Processor Model Control) register behavior is not modeled.

This processor model requires that RAM is available at the address range of the memory mapped registers

Address ranges 0xFFEE0 to 0xFFEFF for General purpose registers (e.g. X, A)

Address ranges 0xFFFF0 to 0xFFFFF for special function registers (e.g. SP)

This processor model should be started with a reset signal. The processor reads from the reset vector 0x0000 on reset and uses this value for the initial PC

1.5 Verification

Models have been tested by eSOL TRINITY and Imperas

1.6 Features

All instructions are supported

Banked registers are supported

External exceptions are supported

The BRK instruction (internal trap) is supported

Memory mirroring is supported

Memory mapped registers is supported

Chapter 2

Configuration

2.1 Location

This model's VLNV is renesas.ovpworld.org/processor/rl78/1.0.
The model source is usually at:
$IMPERAS_HOME/ImperasLib/source/renesas.ovpworld.org/processor/rl78/1.0
The model binary is usually at:
$IMPERAS_HOME/lib/$IMPERAS_ARCH/ImperasLib/renesas.ovpworld.org/processor/rl78/1.0

2.2 GDB Path

The default GDB for this model is: $IMPERAS_HOME/lib/$IMPERAS_ARCH/gdb/rl78-elf-gdb.

2.3 Semi-Host Library

The default semi-host library file is renesas.ovpworld.org/semihosting/rl78Newlib/1.0

2.4 Processor Endian-ness

This is a LITTLE endian model.

2.5 QuantumLeap Support

A simulator using this processor will not be able to use QuantumLeap.

2.6 Processor ELF code

The ELF code supported by this model is: 0xc5.

Chapter 3

All Variants in this model

This model has these variants

Variant	Description
RL78-S1	RL78-S1
RL78-S2	RL78-S2
RL78-S3	RL78-S3

Table 3.1: All Variants in this model

Chapter 4

Bus Master Ports

This model has these bus master ports.

Name	min	max	Connect?	Description
INSTRUCTION	20	32	mandatory	
DATA	20	32	mandatory	

Table 4.1: Bus Master Ports

Chapter 5

Bus Slave Ports

This model has these bus slave ports.

Name	Size(bytes)	Connect?	Description
GPRSP	0x100000	optional	
SFRSP	0x100000	optional	

Table 5.1: Bus Slave Ports

Chapter 6

Net Ports

This model has these net ports.

Name	Type	Connect?	Description
reset	input	optional	
extint	input	optional	
intAck	output	optional	

Table 6.1: Net Ports

Chapter 7

FIFO Ports

This model has no FIFO ports.

Chapter 8

Formal Parameters

Name	Type	Description
verbose	Boolean	Verbose mode
variant	Enumeration	processor variant
sim_ac_flag	Boolean	simulate PSW.AC flag
exit_on_halt	Boolean	simulation will exit on HALT instruction
mirror_rom_addr	Uns32	mirror rom addr
mirror_start_addr	Uns32	mirror start addr
mirror_end_addr	Uns32	mirror end addr

Table 8.1: Parameters

8.1 Parameter values

These are the current parameter values.

Name	Value
(Others)	
verbose	F
variant	RL78-S3
sim_ac_flag	T
exit_on_halt	F
mirror_rom_addr	0x2000
mirror_start_addr	0xf2000
mirror_end_addr	0xfbfff

Table 8.2: Parameter values

Chapter 9

Execution Modes

Mode	Code
RB0	0
RB1	1
RB2	2
RB3	3

Table 9.1: Modes implemented in this processor

Chapter 10

Exceptions

Exception	Code
RST	0
TRP	0
IAW	0
BRK	126
IRQ	65535

Table 10.1: Exceptions implemented by this processor

Chapter 11

Hierarchy of the model

A CPU core may be configured 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.

11.1 Level 1

This level in the model hierarchy has 2 commands.
This level in the model hierarchy has 6 register groups:

Group name	Registers
GPR	8
Bank1	8
Bank2	8
Bank3	8
Bank4	8
System	7

Table 11.1: Register groups

This level in the model hierarchy has no children.

Chapter 12

Model Commands

A Processor model can implement one or more **Model Commands** available to be invoked from the simulator command line, from the OP API or from the Imperas Multiprocessor Debugger.

12.1 Level 1

12.1.1 isync

specify instruction address range for synchronous execution

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

Table 12.1: isync command arguments

12.1.2 itrace

enable or disable instruction tracing

Argument	Type	Description
-after	Uns64	apply after this many instructions
-enable	Boolean	enable instruction tracing
-instructioncount	Boolean	include the instruction number in each trace
-memory	String	show memory accesses by this instruction. Argument can be any combination of X (execute), L (load or store access) and S (system)
-off	Boolean	disable instruction tracing
-on	Boolean	enable instruction tracing
-processorname	Boolean	Include processor name in all trace lines
-registerchange	Boolean	show registers changed by this instruction
-registers	Boolean	show registers after each trace

Table 12.2: itrace command arguments

Chapter 13

Registers

13.1 Level 1

13.1.1 GPR

Registers at level:1, group:GPR

Name	Bits	Initial-Hex	RW	Description
X	8	0	rw	X
A	8	0	rw	A
C	8	0	rw	C
B	8	0	rw	B
E	8	0	rw	E
D	8	0	rw	D
L	8	0	rw	L
H	8	0	rw	H

Table 13.1: Registers at level 1, group:GPR

13.1.2 Bank1

Registers at level:1, group:Bank1

Name	Bits	Initial-Hex	RW	Description
X[RB0]	8	0	rw	R00
A[RB0]	8	0	rw	R01
C[RB0]	8	0	rw	R02
B[RB0]	8	0	rw	R03
E[RB0]	8	0	rw	R04
D[RB0]	8	0	rw	R05
L[RB0]	8	0	rw	R06
H[RB0]	8	0	rw	R07

Table 13.2: Registers at level 1, group:Bank1

13.1.3 Bank2

Registers at level:1, group:Bank2

Name	Bits	Initial-Hex	RW	Description
X[RB1]	8	0	rw	R08

A[RB1]	8	0	rw	R09
C[RB1]	8	0	rw	R10
B[RB1]	8	0	rw	R11
E[RB1]	8	0	rw	R12
D[RB1]	8	0	rw	R13
L[RB1]	8	0	rw	R14
H[RB1]	8	0	rw	R15

Table 13.3: Registers at level 1, group:Bank2

13.1.4 Bank3

Registers at level:1, group:Bank3

Name	Bits	Initial-Hex	RW	Description
X[RB2]	8	0	rw	R16
A[RB2]	8	0	rw	R17
C[RB2]	8	0	rw	R18
B[RB2]	8	0	rw	R19
E[RB2]	8	0	rw	R20
D[RB2]	8	0	rw	R21
L[RB2]	8	0	rw	R22
H[RB2]	8	0	rw	R23

Table 13.4: Registers at level 1, group:Bank3

13.1.5 Bank4

Registers at level:1, group:Bank4

Name	Bits	Initial-Hex	RW	Description
X[RB3]	8	0	rw	R24
A[RB3]	8	0	rw	R25
C[RB3]	8	0	rw	R26
B[RB3]	8	0	rw	R27
E[RB3]	8	0	rw	R28
D[RB3]	8	0	rw	R29
L[RB3]	8	0	rw	R30
H[RB3]	8	0	rw	R31

Table 13.5: Registers at level 1, group:Bank4

13.1.6 System

Registers at level:1, group:System

Name	Bits	Initial-Hex	RW	Description
PSW	8	6	r-	status register
ES	8	f	rw	ES
CS	8	0	rw	CS
PC	20	0	rw	program counter
PMC	8	0	r-	processor mode control
MEM	8	0	r-	MEM
SP	16	0	rw	stack pointer

Table 13.6: Registers at level 1, group:System