TERMA

TEMU
Target Reference

Version 5.0, 2026-11-20

© Terma GmbH, Germany, 2015-2023. Proprietary and intellectual rights of Terma GmbH, Germany are involved in the subject-matter of this material and all manufacturing, reproduction, use, disclosure and sales rights pertaining to such subject-matter are expressly reserved. This material is submitted for a specific purpose as agreed in writing, and the recipient by accepting this material agrees that this material will not be used, copied or reproduced in whole or in part, nor its content (or any part thereof) revealed in any manner or to any third party, except own staff, to meet the purpose for which it was submitted and subject to the terms of the written agreement.

Table of Contents

1. Overview . . . 3
2. SPARC . . . 4
  2.1. Variants . . . 4
  2.2. Operating System Compatibility . . . 5
  2.3. Configuration . . . 5
  2.4. Idle Mode . . . 7
  2.5. Limitations . . . 7
  2.6. Variants . . . 8
3. PowerPC . . . 65
  3.1. Variants . . . 65
4. RISC-V . . . 86
  4.1. Variants . . . 86

The use and/or disclosure,etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.

Chapter 1. Overview

The use and/or disclosure,etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.

Chapter 2. SPARC

The SPARCv8 target comes in several variants, these include emulator cores for the ERC32 (technically a SPARCv7), LEON2, LEON3 and LEON4.

The individual targets only include the CPU core, and not any surrounding device models. The on-chip devices must be connected to the CPU core at configuration time.

2.1. Variants

These are the main variants of the SPARCv8 targets as supported by TEMU at present. Other variants can be added at request.

2.1.1. ERC32

The ERC32 core implements the SPARCv7 instruction set. It does not include the multiply and divide instructions from the SPARCv8. It also lacks the MMU.

2.1.2. LEON2

The LEON2 core implements the SPARCv8 instruction set as provided by the AT697F processor. Note that the LEON2 VHDL models also support some SPARCv8-E extensions (e.g. integer multiply accumulate instructions), but these extensions are not currently in the LEON2 core in order to be similar to the AT697F. The extensions are implemented and can be added in additional L2 models on request.

The LEON2 model supports caches. Note that it is the SoC model (not the CPU mode) that is the one implementing the cache control interfaces.

2.1.3. LEON3

The LEON3 core includes the SPARCv8 instruction set, some SPARCv8-E extensions (UMAC and SMAC instructions), the CASA instruction from the SPARCv9 ISA and the SR-MMU memory management unit.

The LEON3 model supports caches and implements the cache control interface for both instruction and data caches.

2.1.4. LEON4

The LEON3 core includes the SPARCv8 instruction set, some SPARCv8-E extensions, the CASA instruction from the SPARCv9 ISA and the SR-MMU memory management unit.

There are two differences from the LEON3:

- Instruction timing uses values from LEON4 documentation.
- Supports partial WRPSR when RD != 0. There is no real assembler syntax to express this

The use and/or disclosure,etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.

instruction (and it disassembles to the normal wrpsr format).

- Additional argument 'cputype' accepted when class is instantiated. This can be `ngmp` to ensure that `%pc` and `%npc` registers are reset with the correct values (i.e. 0c0000000 and 0xc0000004 respectively).

2.2. Operating System Compatibility

The SPARCv8 models have been executed successfully with:

- Linux
- RTEMS
- XtratuM
- XAL

2.3. Configuration

2.3.1. Arguments

When creating the processor, the `temu_create()` function accepts a number of arguments (which can be given as `args=key0:value0,key1:value1`` in the command line interface).

These arguments are:

cpuid

CPUId, this is a numeric identifier of the core in multi-core/smp systems. Defaults to 0, ignore if you want a single core machine.

freq

Clock frequency in Hz.

cputype

For the Leon4 class only. The cputype argument can be set to the string 'ngmp'. This enables NGMP reset values.

2.3.2. Properties

The following properties are important for configuration of a virtual system.

Interface References

memaccess

The interface reference reacting to the emulator core's memory accesses (whenever there is an ATC miss). This should normally refer to a memory space object or the MMU interface. Set this to `memspace:MemAccessIface` in case the CPU lacks an MMU. Set to `cpu:MmuMemAccessIface` in-case the CPU has an MMU. That is, in the case of an MMU, the iface reference refers to the object itself.

The use and/or disclosure,etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.

memAccessL2

The interface reference to an object reacting to the memory accesses. In case the system has an MMU, set this to `memspace:MemAccessIface`.

irqctrl

The interface reference to an object implementing the `IrqControl` interface. This can be used to connect external interrupt controllers which need to have interrupts acknowledged.

devices

Array of interface references to device models. The objects in this array will have a CPU reset call propagated to themselves. If your device model handles reset messages, it must be put into the devices array (in either the CPU or the machine object).

dCache

Data cache model. For high performance, omit the cache model. This property is only available in processor cores that support cache.

iCache

Data cache model. For high performance, omit the cache model. This property is only available in processor cores that support cache.

Other Properties

freq

Clock frequency in Hz. Defaults to 50000000 = 50 MHz.

cpuid

CPU id for multiprocessor configurations, defaults to 0.

2.3.3. Interfaces

The SPARCv8 emulator cores implement the following interfaces:

CpuIface

The common CPU interface. This contain functions like run and register access functions.

SparcIface

Standard SPARCv8 interface. Contains among other things functions for accessing windowed registers. One capability of the SPARC interface is the registration of ASI handlers.

IrqIface

The interrupt controller interface for raising interrupts on the processor.

InvalidMemAccessIface

Interface invoked on invalid memory accesses. This contain functions that will longjmp to the CPU trap handling logic. The interface can only be invoked from code invoked by the CPU core in one way or the other. Do not call the functions in this interface directly!

The use and/or disclosure,etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.

EventIface

Interface for posting timed events on the CPU core's event queue. Usually a reference to this event is installed in connected device models.

MemoryIface

Proxy interface which forwards to the memory space object.

MmuMemAccessIface

The memory interface provided by the CPU to do accesses through the MMU.

ICacheCtrlIface

Instruction cache control interface. Only available in LEON3 and LEON4.

DCacheCtrlIface

Data cache control interface. Only available in LEON3 and LEON4.

2.4. Idle Mode

SPARCv8 processors trigger idle mode in the following cases:

- Unconditional branch to itself
- Write to power down register

There are two unconditional branch sequences detected:

Listing 1. Annulled Unconditional Branch To Self

```
loop:   ba,a loop
```

Listing 2. Non-Annulled Unconditional Branch To Self

```
loop:   ba loop
        nop
```

In the non-annulled case, idle will only be detected if the following conditions hold:

- Both instructions are on the same page
- `%pc` points at the branch instruction
- `%npc` points at the nop-instruction

2.5. Limitations

Current limitations of the SPARCv8 target include:

- The `wrpsr` instruction is effective immediately. The up to three nops, needed in real code serves

The use and/or disclosure,etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.

no purpose in the emulator. Thus if nops are omitted you will not detect this on the emulator at present.

- Floating point traps are direct and not deferred. This is the correct behaviour for the AT697F, but may not be correct for other chips.
- Timing effects due to super-scalar execution is not simulated. Again, this is correct behaviour for the AT697F.
- Operator dependant timing effects (especially noticeable in the FPU) are not simulated. Timing for instructions is static and uses the documented typical values.
- The LEON2 model takes FPU timings from the ERC32 as no known documentation about the costs on the MEIKO FPU is available. The only known data for the MEIKO is in the same magnitude as the ERC32 FPU (which is not MEIKO), hence we assume that the ERC32 timings are roughly correct for the LEON2.
- The FPU model is based on the SPARCv8 standard, and follows the SPARCv8 recommendations for NaN-propagation. If the SPARCv8 you emulate use an FPU that is not compliant with the SPARCv8 NaN propagation recommendations, there may be slight deviation in results. If you need an FPU core that follows different rules, please contact Terma.
- The cache interface do not support line locking at present.
- The SVT trapping model is not supported at present

2.6. Variants

2.6.1. ERC32

@Erc32 Reference

Properties

Name	Type	Description
Class	*void	Class object
Component	*void	Pointer to component object if part of component
LocalName	*char	Local name (in component, if applicable)
LogMessageFlags	uint64_t	Per-object log message suppression flags
LoggingFlags	uint64_t	Flags for logging info
Name	*char	Object name
ObjectID	uint64_t	Unique ObjectID.
ReplayLoggingFlags	uint64_t	Replay-time category logging flags

The use and/or disclosure,etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.

Name	Type	Description
TimeSource	*void	Time source object

Commands

Name	Description
delete	Dispose instance of @Erc32
new	Create new instance of Erc32

Command new Arguments

Name	Type	Required	Description
freq	double	no	frequency in Hz
name	string	yes	Name of object to create

Erc32 Reference

Properties

Name	Type	Description
CPUId	uint32_t	
CPUType	int32_t	
Class	*void	Class object
Component	*void	Pointer to component object if part of component
HostFlags	uint64_t	
LocalName	*char	Local name (in component, if applicable)
LogMessageFlags	uint64_t	Per-object log message suppression flags
LoggingFlags	uint64_t	Flags for logging info
Name	*char	Object name
ObjectID	uint64_t	Unique ObjectID.
ReplayLoggingFlags	uint64_t	Replay-time category logging flags
TimeSource	*void	Time source object
asr	[uint32_t; 32]	
config.assemblerAliases	uint8_t	

The use and/or disclosure,etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.

Name	Type	Description
config.exitOnSync	uint8_t	
config.handleEventWithinBlock	uint8_t	
config.logCallAndJump	uint8_t	
config.measureExecTime	uint8_t	
cpi	double	Cycles per instruction
currentContextId	uint64_t	
currentContextValid	uint8_t	
currentThreadContextId	uint64_t	
currentThreadContextValid	uint8_t	
cwp	*void	
devices	temu_IfaceRefArray	
exitReason	int32_t	
extraRegs	[uint32_t; 32]	
fprs	[uint32_t; 32]	
freq	int64_t	Frequency in Hz
fsr	uint32_t	Floating Point State Register
fsrInit	uint32_t	
fsrSpill	uint32_t	
g	[uint32_t; 8]	
gprs	[uint32_t; 128]	
i_npc	uintptr_t	Intermediate code nPC
i_pc	uintptr_t	Intermediate code PC
idleSteps	int64_t	
ipc	double	Instructions per cycle
irqClient	temu_IfaceRef/ <unknown>	Interrupt controller (for ACKs)
logTrampolines	uint8_t	
memAccess	temu_IfaceRef/ <unknown>	Level 1 memory access interface (MMU)
memAccessL2	temu_IfaceRef/ <unknown>	Level 2 memory access interface (physical)
memReset	temu_IfaceRef/ <unknown>	Memory space reset interface
memSpace	*void	Memory space.

The use and/or disclosure,etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.

Name	Type	Description
mmuCtrl	uint32_t	
mmuCtxt	uint32_t	
mmuCtxtCount	uint32_t	
mmuCtxtPtr	uint32_t	
mmuFaultAddr	uint32_t	
mmuFaultStat	uint32_t	
nextEvent	int64_t	Next event
npc	uint32_t	Next Program Counter
parentTimeSource	*void	Parent time source
pc	uint32_t	Program Counter
pdcManager	temu_IfaceRef/ <unknown>	Pre-decode cache manager (normally memory space)
pendingInterrupts	uint16_t	
powerState	uint32_t	Power State property (deprecated)
psr	uint32_t	Processor State Register
skipIdleTags	uint64_t	
startSteps	int64_t	Start steps of the time source
state	int32_t	
stats.mmuCtrlWrites	uint64_t	
stats.mmuCtxtPtrWrites	uint64_t	
stats.mmuCtxtWrites	uint64_t	
stats.raisedTraps	uint64_t	
stats.takenInterrupts	uint64_t	
steps	int64_t	Steps of the time source
stickyFlags	uint32_t	Set bit 0 to 1 to not exit CPU on halted mode.
targetExec	temu_IfaceRef/ <unknown>	Target execution interface
targetSteps	int64_t	Target steps of the time source
tbr	uint32_t	Trap Base Register
wim	uint32_t	Window Invalidation Mask

Interfaces

The use and/or disclosure,etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.

Name	Type	Description
AssemblerIface	temu::AssemblerIface	Assembler interface
BinaryTranslationIface	temu::BinaryTranslationControlIface	
ClockIface	ClockIface	
CodePatternIface	temu::CodePatternIface	
CpuIface	temu::CpuIface	
ExecIface	temu::TargetExecutionIface	
InvalidMemAccessIface	MemAccessIface	
IrqIface	IrqCtrlIface	
ObjectIface	ObjectIface	
PhysicalMemAccessIface	MemAccessIface	
PowerIface	PowerIface	
RegisterIface	temu::RegisterIface	Auto-generated Register Interface Impl.
ResetIface	ResetIface	
SparcIface	SparcIface	
VirtualMemAccessIface	MemAccessIface	

Ports

Prop	Iface	Description
irqClient	IrqIface	interrupt controller interface

Commands

Name	Description
add-call	Adds a call at the specific address. The call operation calls a predefined method which logs that it is invoked.It is primarily useful for debugging.
add-idle	Adds an explicit idle operation at the specific address.
add-skip	Adds a skip operation at the specific address. After this, the given number of instructions will be skipped.
assemble	Assemble instruction

The use and/or disclosure,etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.

Name	Description
auto-log-block	Log translation of block when translating for address
auto-log-instructions	Log instructions in block with the given address
break-on-trap	Break on trap
chainBlocks	Manually chain blocks
delete	Dispose instance of Erc32
disable-block-entry-logging	Disable logging on block entries.
disable-call-target-profiling	Disable call target profiling for triggering DBT.
disable-dbt-logging	Disable logging when translating code.
disable-dbt-validation	Disable validation of emitted machine code.
disable-fallthrough-profiling	Dis profiling of fallthroughs for triggering DBT.
disable-floating-point-translation	Disable binary translation of floating point instructions.
disable-hard-code-reset	Disable hard resets of DBT emitter cache.
disable-indirect-jump-profiling	Disable profiling of indirect jump targets for triggering DBT.
disable-jit-timing	Disable jit-timing.
disable-off-page-branch-profiling	Disable profiling of off-page branch targets for triggering DBT.
disable-on-page-branch-profiling	Disable profiling of on-page branch targets for triggering DBT.
disable-translation-logging	Disable jit-logging.
disableAutoChain	Disable binary translation block chaining.
disableProfiling	Disable profiling mode
disableStats	Disable statistics
disableTranslation	Disable binary translation.
disableTraps	Disable traps.
disassemble	Disassemble code
disassembleBlock	Disassemble binary translated code block.
disassembleBlockIR	Disassemble LLVM IR of translated code block
disassembleBlockOption	Set option for disassembler. Switch between AT&T (default) and Intel (alternate) syntax. Enable / disable insruction latency in output.
enable-block-entry-logging	Enable logging on block entries.

The use and/or disclosure,etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.

Name	Description
enable-call-target-profiling	Enable call target profiling for triggering DBT.
enable-cfm-crc	Enable Code Fragment Manger CRCs
enable-dbt-logging	Enable logging when translating code.
enable-dbt-validation	Enable validation of emitted machine code.
enable-fallthrough-profiling	Enable profiling of fallthroughs for triggering DBT.
enable-floating-point-translation	Enable binary translation of floating point instructions.
enable-hard-code-reset	Enable hard resets of DBT emitter cache.
enable-indirect-jump-profiling	Enable profiling of indirect jump targets for triggering DBT.
enable-jit-timing	Enable jit-timing.
enable-off-page-branch-profiling	Enable profiling of off-page branch targets for triggering DBT.
enable-on-page-branch-profiling	Enable profiling of on-page branch targets for triggering DBT.
enable-translation-logging	Enable jit-logging.
enableAutoChain	Enable binary translation block chaining.
enableProfiling	Enable profiling mode
enableStats	Enable statistics
enableTranslation	Enable binary translation.
enableTraps	Enable traps.
find-page-table-root	Scan through the page tables starting with context zero and determine which page table roots can generate the translation.
flushProfile	Flush profile data
pfregs	Print floating point registers for CPU
pregs	Print registers for CPU
print-atc	Print ATC for CPU
print-blocks	Show block connectivity
pstat	Print CPU stats
pwin	Print register window.
resetStats	Reset statistics counters

The use and/or disclosure,etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.

Name	Description
set-binary-translation-threshold	Set number of times a call target must be executed to trigger translation.
setPC	Set PC (and nPC)
setReg	Set register
stacktrace	Print a stacktrace starting at the current PC. The trace follows firstly the register windows if %i7 points at a call or jmpl addr, %o7 instruction. After register windows have been followed, the stacktrace walks through the in memory stack. Note that leaf functions are detected if %i7 < %o7 ⇐ %pc.
translateBlock	Translate block.
translateFunc	Translate function.
validate	Validates constant data.
wake-up	Wakes up the processor.

Command add-call Arguments

Name	Type	Required	Description
context	string	no	Context
pa	int	no	Physical Address
pamask	int	no	Physical Address Mask
script	path	no	Script to run when reaching address
va	int	no	Virtual Address
vamask	int	no	Virtual Address Mask

Command add-idle Arguments

Name	Type	Required	Description
context	string	no	Context
pa	int	no	Physical Address
pamask	int	no	Physical Address Mask
tag	int	no	Tag of idle operation
va	int	no	Virtual Address
vamask	int	no	Virtual Address Mask

The use and/or disclosure,etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.

Command add-skip Arguments

Name	Type	Required	Description
context	string	no	Context
pa	int	no	Physical Address
pamask	int	no	Physical Address Mask
steps	int	yes	Steps to skip
va	int	no	Virtual Address
vamask	int	no	Virtual Address Mask

Command assemble Arguments

Name	Type	Required	Description
inst	string	yes	Instruction to assemble.
pa	int	no	Physical address
va	int	no	Virtual address

Command auto-log-block Arguments

Name	Type	Required	Description
pa	int	yes	Physical Address

Command auto-log-instructions Arguments

Name	Type	Required	Description
pa	int	yes	Physical Address

Command break-on-trap Arguments

Name	Type	Required	Description
trap	string	yes	Trap name

Command chainBlocks Arguments

Name	Type	Required	Description
arm	string	no	Name of arm in source block (nt/not-taken or t/taken). Defaults to fall through arm (not-taken)

The use and/or disclosure,etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.

Name	Type	Required	Description
source	int	yes	Physical start address of source block.
target	int	yes	Physical start address of target block.

Command disableStats Arguments

Name	Type	Required	Description
stat	string	yes	Name of statistics (executed-translated-instructions, executed-translated-blocks)

Command disassemble Arguments

Name	Type	Required	Description
count	int	no	Number of instructions
pa	int	no	Physical address
va	int	no	Virtual address

Command disassembleBlock Arguments

Name	Type	Required	Description
pa	int	no	Physical address of block
va	int	no	Virtual address of block

Command disassembleBlockIR Arguments

Name	Type	Required	Description
pa	int	no	Physical address of block
va	int	no	Virtual address of block

Command disassembleBlockOption Arguments

Name	Type	Required	Description
option	string	yes	Option to set: 'default-syntax', 'alternate-syntax', 'latency', 'no-latency'.

The use and/or disclosure,etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.

Command enable-jit-timing Arguments

Name	Type	Required	Description
print	int	no	Enable log-messages

Command enableStats Arguments

Name	Type	Required	Description
stat	string	yes	Name of statistics (executed-translated-instructions, executed-translated-blocks)

Command find-page-table-root Arguments

Name	Type	Required	Description
pa	int	no	Physical Address
va	int	yes	Virtual Address

Command pwin Arguments

Name	Type	Required	Description
win	int	no	Window ID (-1 = default == current window).

Command set-binary-translation-threshold Arguments

Name	Type	Required	Description
threshold	int	yes	Threshold

Command setPC Arguments

Name	Type	Required	Description
npc	int	no	New %npc (note if omitted %npc = %pc + 4)
pc	int	yes	New %pc

Command setReg Arguments

Name	Type	Required	Description
reg	string	yes	Register name
value	int	yes	Value

The use and/or disclosure,etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.

Command translateBlock Arguments

Name	Type	Required	Description
count	int	no	Number of instructions, omit for automatic.
pa	int	no	Physical address of block
va	int	no	Virtual address of block

Command translateFunc Arguments

Name	Type	Required	Description
pa	int	no	Physical address of block
va	int	no	Virtual address of block

2.6.2. LEON2

@Leon2 Reference

Properties

Name	Type	Description
Class	*void	Class object
Component	*void	Pointer to component object if part of component
LocalName	*char	Local name (in component, if applicable)
LogMessageFlags	uint64_t	Per-object log message suppression flags
LoggingFlags	uint64_t	Flags for logging info
Name	*char	Object name
ObjectID	uint64_t	Unique ObjectID.
ReplayLoggingFlags	uint64_t	Replay-time category logging flags
TimeSource	*void	Time source object

Commands

Name	Description
delete	Dispose instance of @Leon2

The use and/or disclosure,etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.

Name	Description
new	Create new instance of Leon2

Command new Arguments

Name	Type	Required	Description
at697f-errata	int	no	enable AT697 errata detectors (limited to IPN #384)
cpuid	int	no	cpuid in multiprocessor config
freq	double	no	frequency in Hz
name	string	yes	Name of object to create

Leon2 Reference

Properties

Name	Type	Description
CPUId	uint32_t	
CPUType	int32_t	
Class	*void	Class object
Component	*void	Pointer to component object if part of component
HostFlags	uint64_t	
LocalName	*char	Local name (in component, if applicable)
LogMessageFlags	uint64_t	Per-object log message suppression flags
LoggingFlags	uint64_t	Flags for logging info
Name	*char	Object name
ObjectID	uint64_t	Unique ObjectID.
ReplayLoggingFlags	uint64_t	Replay-time category logging flags
TimeSource	*void	Time source object
asr	[uint32_t; 32]	
config.assemblerAliases	uint8_t	

The use and/or disclosure,etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.

Name	Type	Description
config.exitOnSync	uint8_t	
config.handleEventWithinBlock	uint8_t	
config.logCallAndJump	uint8_t	
config.measureExecTime	uint8_t	
cpi	double	Cycles per instruction
currentContextId	uint64_t	
currentContextValid	uint8_t	
currentThreadContextId	uint64_t	
currentThreadContextValid	uint8_t	
cwp	*void	
dCache	temu_IfaceRef/ <unknown>	L1 data cache (optional)
devices	temu_IfaceRefArray	
exitReason	int32_t	
extraRegs	[uint32_t; 32]	
fprs	[uint32_t; 32]	
freq	int64_t	Frequency in Hz
fsr	uint32_t	Floating Point State Register
fsrInit	uint32_t	
fsrSpill	uint32_t	
g	[uint32_t; 8]	
gprs	[uint32_t; 128]	
iCache	temu_IfaceRef/ <unknown>	L1 instruction cache (optional)
i_npc	uintptr_t	Intermediate code nPC
i_pc	uintptr_t	Intermediate code PC
idleSteps	int64_t	
ipc	double	Instructions per cycle
irqClient	temu_IfaceRef/ <unknown>	Interrupt controller (for ACKs)
logTrampolines	uint8_t	
memAccess	temu_IfaceRef/ <unknown>	Level 1 memory access interface (MMU)
memAccessL2	temu_IfaceRef/ <unknown>	Level 2 memory access interface (physical)

The use and/or disclosure,etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.

Name	Type	Description
memReset	temu_IfaceRef/ <unknown>	Memory space reset interface
memSpace	*void	Memory space.
mmuCtrl	uint32_t	
mmuCtxt	uint32_t	
mmuCtxtCount	uint32_t	
mmuCtxtPtr	uint32_t	
mmuFaultAddr	uint32_t	
mmuFaultStat	uint32_t	
nextEvent	int64_t	Next event
npc	uint32_t	Next Program Counter
parentTimeSource	*void	Parent time source
pc	uint32_t	Program Counter
pdcManager	temu_IfaceRef/ <unknown>	Pre-decode cache manager (normally memory space)
pendingInterrupts	uint16_t	
powerState	uint32_t	Power State property (deprecated)
psr	uint32_t	Processor State Register
skipIdleTags	uint64_t	
startSteps	int64_t	Start steps of the time source
state	int32_t	
stats.mmuCtrlWrites	uint64_t	
stats.mmuCtxtPtrWrites	uint64_t	
stats.mmuCtxtWrites	uint64_t	
stats.raisedTraps	uint64_t	
stats.takenInterrupts	uint64_t	
steps	int64_t	Steps of the time source
stickyFlags	uint32_t	Set bit 0 to 1 to not exit CPU on halted mode.
targetExec	temu_IfaceRef/ <unknown>	Target execution interface
targetSteps	int64_t	Target steps of the time source
tbr	uint32_t	Trap Base Register
wim	uint32_t	Window Invalidation Mask

The use and/or disclosure,etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.

Interfaces

Name	Type	Description
AssemblerIface	temu::AssemblerIface	Assembler interface
BinaryTranslationIface	temu::BinaryTranslationControlIface	
ClockIface	ClockIface	
CodePatternIface	temu::CodePatternIface	
CpuIface	temu::CpuIface	
ExecIface	temu::TargetExecutionIface	
InvalidMemAccessIface	MemAccessIface	
IrqIface	IrqCtrlIface	
ObjectIface	ObjectIface	
PhysicalMemAccessIface	MemAccessIface	
PowerIface	PowerIface	
RegisterIface	temu::RegisterIface	Auto-generated Register Interface Impl.
ResetIface	ResetIface	
SparcIface	SparcIface	
VirtualMemAccessIface	MemAccessIface	

Ports

Prop	Iface	Description
irqClient	IrqIface	interrupt controller interface

Commands

Name	Description
add-call	Adds a call at the specific address. The call operation calls a predefined method which logs that it is invoked.It is primarily useful for debugging.
add-idle	Adds an explicit idle operation at the specific address.
add-skip	Adds a skip operation at the specific address. After this, the given number of instructions will be skipped.
assemble	Assemble instruction

The use and/or disclosure,etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.

Name	Description
auto-log-block	Log translation of block when translating for address
auto-log-instructions	Log instructions in block with the given address
break-on-trap	Break on trap
chainBlocks	Manually chain blocks
delete	Dispose instance of Leon2
disable-block-entry-logging	Disable logging on block entries.
disable-call-target-profiling	Disable call target profiling for triggering DBT.
disable-dbt-logging	Disable logging when translating code.
disable-dbt-validation	Disable validation of emitted machine code.
disable-fallthrough-profiling	Dis profiling of fallthroughs for triggering DBT.
disable-floating-point-translation	Disable binary translation of floating point instructions.
disable-hard-code-reset	Disable hard resets of DBT emitter cache.
disable-indirect-jump-profiling	Disable profiling of indirect jump targets for triggering DBT.
disable-jit-timing	Disable jit-timing.
disable-off-page-branch-profiling	Disable profiling of off-page branch targets for triggering DBT.
disable-on-page-branch-profiling	Disable profiling of on-page branch targets for triggering DBT.
disable-translation-logging	Disable jit-logging.
disableAutoChain	Disable binary translation block chaining.
disableProfiling	Disable profiling mode
disableStats	Disable statistics
disableTranslation	Disable binary translation.
disableTraps	Disable traps.
disassemble	Disassemble code
disassembleBlock	Disassemble binary translated code block.
disassembleBlockIR	Disassemble LLVM IR of translated code block
disassembleBlockOption	Set option for disassembler. Switch between AT&T (default) and Intel (alternate) syntax. Enable / disable insruction latency in output.
enable-block-entry-logging	Enable logging on block entries.

The use and/or disclosure,etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.

Name	Description
enable-call-target-profiling	Enable call target profiling for triggering DBT.
enable-cfm-crc	Enable Code Fragment Manger CRCs
enable-dbt-logging	Enable logging when translating code.
enable-dbt-validation	Enable validation of emitted machine code.
enable-fallthrough-profiling	Enable profiling of fallthroughs for triggering DBT.
enable-floating-point-translation	Enable binary translation of floating point instructions.
enable-hard-code-reset	Enable hard resets of DBT emitter cache.
enable-indirect-jump-profiling	Enable profiling of indirect jump targets for triggering DBT.
enable-jit-timing	Enable jit-timing.
enable-off-page-branch-profiling	Enable profiling of off-page branch targets for triggering DBT.
enable-on-page-branch-profiling	Enable profiling of on-page branch targets for triggering DBT.
enable-translation-logging	Enable jit-logging.
enableAutoChain	Enable binary translation block chaining.
enableProfiling	Enable profiling mode
enableStats	Enable statistics
enableTranslation	Enable binary translation.
enableTraps	Enable traps.
find-page-table-root	Scan through the page tables starting with context zero and determine which page table roots can generate the translation.
flushProfile	Flush profile data
pfregs	Print floating point registers for CPU
pregs	Print registers for CPU
print-atc	Print ATC for CPU
print-blocks	Show block connectivity
pstat	Print CPU stats
pwin	Print register window.
resetStats	Reset statistics counters

The use and/or disclosure,etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.

Name	Description
set-binary-translation-threshold	Set number of times a call target must be executed to trigger translation.
setPC	Set PC (and nPC)
setReg	Set register
stacktrace	Print a stacktrace starting at the current PC. The trace follows firstly the register windows if %i7 points at a call or jmpl addr, %o7 instruction. After register windows have been followed, the stacktrace walks through the in memory stack. Note that leaf functions are detected if %i7 < %o7 ⇐ %pc.
translateBlock	Translate block.
translateFunc	Translate function.
validate	Validates constant data.
wake-up	Wakes up the processor.

Command add-call Arguments

Name	Type	Required	Description
context	string	no	Context
pa	int	no	Physical Address
pamask	int	no	Physical Address Mask
script	path	no	Script to run when reaching address
va	int	no	Virtual Address
vamask	int	no	Virtual Address Mask

Command add-idle Arguments

Name	Type	Required	Description
context	string	no	Context
pa	int	no	Physical Address
pamask	int	no	Physical Address Mask
tag	int	no	Tag of idle operation
va	int	no	Virtual Address
vamask	int	no	Virtual Address Mask

The use and/or disclosure,etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.

Command add-skip Arguments

Name	Type	Required	Description
context	string	no	Context
pa	int	no	Physical Address
pamask	int	no	Physical Address Mask
steps	int	yes	Steps to skip
va	int	no	Virtual Address
vamask	int	no	Virtual Address Mask

Command assemble Arguments

Name	Type	Required	Description
inst	string	yes	Instruction to assemble.
pa	int	no	Physical address
va	int	no	Virtual address

Command auto-log-block Arguments

Name	Type	Required	Description
pa	int	yes	Physical Address

Command auto-log-instructions Arguments

Name	Type	Required	Description
pa	int	yes	Physical Address

Command break-on-trap Arguments

Name	Type	Required	Description
trap	string	yes	Trap name

Command chainBlocks Arguments

Name	Type	Required	Description
arm	string	no	Name of arm in source block (nt/not-taken or t/taken). Defaults to fall through arm (not-taken)

The use and/or disclosure,etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.

Name	Type	Required	Description
source	int	yes	Physical start address of source block.
target	int	yes	Physical start address of target block.

Command disableStats Arguments

Name	Type	Required	Description
stat	string	yes	Name of statistics (executed-translated-instructions, executed-translated-blocks)

Command disassemble Arguments

Name	Type	Required	Description
count	int	no	Number of instructions
pa	int	no	Physical address
va	int	no	Virtual address

Command disassembleBlock Arguments

Name	Type	Required	Description
pa	int	no	Physical address of block
va	int	no	Virtual address of block

Command disassembleBlockIR Arguments

Name	Type	Required	Description
pa	int	no	Physical address of block
va	int	no	Virtual address of block

Command disassembleBlockOption Arguments

Name	Type	Required	Description
option	string	yes	Option to set: 'default-syntax', 'alternate-syntax', 'latency', 'no-latency'.

The use and/or disclosure,etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.

Command enable-jit-timing Arguments

Name	Type	Required	Description
print	int	no	Enable log-messages

Command enableStats Arguments

Name	Type	Required	Description
stat	string	yes	Name of statistics (executed-translated-instructions, executed-translated-blocks)

Command find-page-table-root Arguments

Name	Type	Required	Description
pa	int	no	Physical Address
va	int	yes	Virtual Address

Command pwin Arguments

Name	Type	Required	Description
win	int	no	Window ID (-1 = default == current window).

Command set-binary-translation-threshold Arguments

Name	Type	Required	Description
threshold	int	yes	Threshold

Command setPC Arguments

Name	Type	Required	Description
npc	int	no	New %npc (note if omitted %npc = %pc + 4)
pc	int	yes	New %pc

Command setReg Arguments

Name	Type	Required	Description
reg	string	yes	Register name
value	int	yes	Value

The use and/or disclosure,etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.

Command translateBlock Arguments

Name	Type	Required	Description
count	int	no	Number of instructions, omit for automatic.
pa	int	no	Physical address of block
va	int	no	Virtual address of block

Command translateFunc Arguments

Name	Type	Required	Description
pa	int	no	Physical address of block
va	int	no	Virtual address of block

2.6.3. LEON3

@Leon3 Reference

Properties

Name	Type	Description
Class	*void	Class object
Component	*void	Pointer to component object if part of component
LocalName	*char	Local name (in component, if applicable)
LogMessageFlags	uint64_t	Per-object log message suppression flags
LoggingFlags	uint64_t	Flags for logging info
Name	*char	Object name
ObjectID	uint64_t	Unique ObjectID.
ReplayLoggingFlags	uint64_t	Replay-time category logging flags
TimeSource	*void	Time source object

Commands

Name	Description
delete	Dispose instance of @Leon3

The use and/or disclosure,etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.

Name	Description
new	Create new instance of Leon3

Command new Arguments

Name	Type	Required	Description
cpuid	int	no	cpuid in multiprocessor config
freq	double	no	frequency in Hz
name	string	yes	Name of object to create

Leon3 Reference

Properties

Name	Type	Description
CPUId	uint32_t	
CPUType	int32_t	
Class	*void	Class object
Component	*void	Pointer to component object if part of component
HostFlags	uint64_t	
LocalName	*char	Local name (in component, if applicable)
LogMessageFlags	uint64_t	Per-object log message suppression flags
LoggingFlags	uint64_t	Flags for logging info
Name	*char	Object name
ObjectID	uint64_t	Unique ObjectID.
ReplayLoggingFlags	uint64_t	Replay-time category logging flags
TimeSource	*void	Time source object
asr	[uint32_t; 32]	
config.assemblerAliases	uint8_t	
config.exitOnSync	uint8_t	
config.handleEventWithinBlock	uint8_t	
config.logCallAndJump	uint8_t	

The use and/or disclosure,etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.

Name	Type	Description
config.measureExecTime	uint8_t	
cpi	double	Cycles per instruction
currentContextId	uint64_t	
currentContextValid	uint8_t	
currentThreadContextId	uint64_t	
currentThreadContextValid	uint8_t	
cwp	*void	
dCache	temu_IfaceRef/ <unknown>	L1 data cache (optional)
devices	temu_IfaceRefArray	
exitReason	int32_t	
extraRegs	[uint32_t; 32]	
fprs	[uint32_t; 32]	
freq	int64_t	Frequency in Hz
fsr	uint32_t	Floating Point State Register
fsrInit	uint32_t	
fsrSpill	uint32_t	
g	[uint32_t; 8]	
gprs	[uint32_t; 128]	
iCache	temu_IfaceRef/ <unknown>	L1 instruction cache (optional)
i_npc	uintptr_t	Intermediate code nPC
i_pc	uintptr_t	Intermediate code PC
idleSteps	int64_t	
ipc	double	Instructions per cycle
irqClient	temu_IfaceRef/ <unknown>	Interrupt controller (for ACKs)
logTrampolines	uint8_t	
memAccess	temu_IfaceRef/ <unknown>	Level 1 memory access interface (MMU)
memAccessL2	temu_IfaceRef/ <unknown>	Level 2 memory access interface (physical)
memReset	temu_IfaceRef/ <unknown>	Memory space reset interface
memSpace	*void	Memory space.
mmuCtrl	uint32_t	

The use and/or disclosure,etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.

Name	Type	Description
mmuCtxt	uint32_t	
mmuCtxtCount	uint32_t	
mmuCtxtPtr	uint32_t	
mmuFaultAddr	uint32_t	
mmuFaultStat	uint32_t	
nextEvent	int64_t	Next event
npc	uint32_t	Next Program Counter
parentTimeSource	*void	Parent time source
pc	uint32_t	Program Counter
pdcManager	temu_IfaceRef/ <unknown>	Pre-decode cache manager (normally memory space)
pendingInterrupts	uint16_t	
powerState	uint32_t	Power State property (deprecated)
psr	uint32_t	Processor State Register
skipIdleTags	uint64_t	
startSteps	int64_t	Start steps of the time source
state	int32_t	
stats.mmuCtrlWrites	uint64_t	
stats.mmuCtxtPtrWrites	uint64_t	
stats.mmuCtxtWrites	uint64_t	
stats.raisedTraps	uint64_t	
stats.takenInterrupts	uint64_t	
steps	int64_t	Steps of the time source
stickyFlags	uint32_t	Set bit 0 to 1 to not exit CPU on halted mode.
targetExec	temu_IfaceRef/ <unknown>	Target execution interface
targetSteps	int64_t	Target steps of the time source
tbr	uint32_t	Trap Base Register
wim	uint32_t	Window Invalidation Mask

Interfaces

The use and/or disclosure,etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.

Name	Type	Description
AhbIface	AhbIface	
AssemblerIface	temu::AssemblerIface	Assembler interface
BinaryTranslationIface	temu::BinaryTranslationControlIface	
ClockIface	ClockIface	
CodePatternIface	temu::CodePatternIface	
CpuIface	temu::CpuIface	
DCacheCtrlIface	temu::CacheCtrlIface	
ExecIface	temu::TargetExecutionIface	
ICacheCtrlIface	temu::CacheCtrlIface	
InvalidMemAccessIface	MemAccessIface	
IrqIface	IrqCtrlIface	
MmuMemAccessIface	MemAccessIface	
ObjectIface	ObjectIface	
PhysicalMemAccessIface	MemAccessIface	
PowerIface	PowerIface	
RegisterIface	temu::RegisterIface	Auto-generated Register Interface Impl.
ResetIface	ResetIface	
SparcIface	SparcIface	
VirtualMemAccessIface	MemAccessIface	

Ports

Prop	Iface	Description
irqClient	IrqIface	interrupt controller interface

Commands

Name	Description
add-call	Adds a call at the specific address. The call operation calls a predefined method which logs that it is invoked.It is primarily useful for debugging.
add-idle	Adds an explicit idle operation at the specific address.

The use and/or disclosure,etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.

Name	Description
add-skip	Adds a skip operation at the specific address. After this, the given number of instructions will be skipped.
assemble	Assemble instruction
auto-log-block	Log translation of block when translating for address
auto-log-instructions	Log instructions in block with the given address
break-on-trap	Break on trap
chainBlocks	Manually chain blocks
delete	Dispose instance of Leon3
disable-block-entry-logging	Disable logging on block entries.
disable-call-target-profiling	Disable call target profiling for triggering DBT.
disable-dbt-logging	Disable logging when translating code.
disable-dbt-validation	Disable validation of emitted machine code.
disable-fallthrough-profiling	Dis profiling of fallthroughs for triggering DBT.
disable-floating-point-translation	Disable binary translation of floating point instructions.
disable-hard-code-reset	Disable hard resets of DBT emitter cache.
disable-indirect-jump-profiling	Disable profiling of indirect jump targets for triggering DBT.
disable-jit-timing	Disable jit-timing.
disable-off-page-branch-profiling	Disable profiling of off-page branch targets for triggering DBT.
disable-on-page-branch-profiling	Disable profiling of on-page branch targets for triggering DBT.
disable-translation-logging	Disable jit-logging.
disableAutoChain	Disable binary translation block chaining.
disableProfiling	Disable profiling mode
disableStats	Disable statistics
disableTranslation	Disable binary translation.
disableTraps	Disable traps.
disassemble	Disassemble code
disassembleBlock	Disassemble binary translated code block.
disassembleBlockIR	Disassemble LLVM IR of translated code block

The use and/or disclosure,etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.

Name	Description
disassembleBlockOption	Set option for disassembler. Switch between AT&T (default) and Intel (alternate) syntax. Enable / disable insruction latency in output.
enable-block-entry-logging	Enable logging on block entries.
enable-call-target-profiling	Enable call target profiling for triggering DBT.
enable-cfm-crc	Enable Code Fragment Manger CRCs
enable-dbt-logging	Enable logging when translating code.
enable-dbt-validation	Enable validation of emitted machine code.
enable-fallthrough-profiling	Enable profiling of fallthroughs for triggering DBT.
enable-floating-point-translation	Enable binary translation of floating point instructions.
enable-hard-code-reset	Enable hard resets of DBT emitter cache.
enable-indirect-jump-profiling	Enable profiling of indirect jump targets for triggering DBT.
enable-jit-timing	Enable jit-timing.
enable-off-page-branch-profiling	Enable profiling of off-page branch targets for triggering DBT.
enable-on-page-branch-profiling	Enable profiling of on-page branch targets for triggering DBT.
enable-translation-logging	Enable jit-logging.
enableAutoChain	Enable binary translation block chaining.
enableProfiling	Enable profiling mode
enableStats	Enable statistics
enableTranslation	Enable binary translation.
enableTraps	Enable traps.
find-page-table-root	Scan through the page tables starting with context zero and determine which page table roots can generate the translation.
flushProfile	Flush profile data
pfregs	Print floating point registers for CPU
pregs	Print registers for CPU
print-atc	Print ATC for CPU
print-blocks	Show block connectivity

The use and/or disclosure,etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.

Name	Description
pstat	Print CPU stats
pwin	Print register window.
resetStats	Reset statistics counters
set-binary-translation-threshold	Set number of times a call target must be executed to trigger translation.
setPC	Set PC (and nPC)
setReg	Set register
stacktrace	Print a stacktrace starting at the current PC. The trace follows firstly the register windows if %i7 points at a call or jmpl addr, %o7 instruction. After register windows have been followed, the stacktrace walks through the in memory stack. Note that leaf functions are detected if %i7 < %o7 ⇐ %pc.
translateBlock	Translate block.
translateFunc	Translate function.
validate	Validates constant data.
wake-up	Wakes up the processor.

Command add-call Arguments

Name	Type	Required	Description
context	string	no	Context
pa	int	no	Physical Address
pamask	int	no	Physical Address Mask
script	path	no	Script to run when reaching address
va	int	no	Virtual Address
vamask	int	no	Virtual Address Mask

Command add-idle Arguments

Name	Type	Required	Description
context	string	no	Context
pa	int	no	Physical Address
pamask	int	no	Physical Address Mask
tag	int	no	Tag of idle operation

The use and/or disclosure,etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.

Name	Type	Required	Description
va	int	no	Virtual Address
vamask	int	no	Virtual Address Mask

Command add-skip Arguments

Name	Type	Required	Description
context	string	no	Context
pa	int	no	Physical Address
pamask	int	no	Physical Address Mask
steps	int	yes	Steps to skip
va	int	no	Virtual Address
vamask	int	no	Virtual Address Mask

Command assemble Arguments

Name	Type	Required	Description
inst	string	yes	Instruction to assemble.
pa	int	no	Physical address
va	int	no	Virtual address

Command auto-log-block Arguments

Name	Type	Required	Description
pa	int	yes	Physical Address

Command auto-log-instructions Arguments

Name	Type	Required	Description
pa	int	yes	Physical Address

Command break-on-trap Arguments

Name	Type	Required	Description
trap	string	yes	Trap name

Command chainBlocks Arguments

The use and/or disclosure,etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.

Name	Type	Required	Description
arm	string	no	Name of arm in source block (nt/not-taken or t/taken). Defaults to fall through arm (not-taken)
source	int	yes	Physical start address of source block.
target	int	yes	Physical start address of target block.

Command disableStats Arguments

Name	Type	Required	Description
stat	string	yes	Name of statistics (executed-translated-instructions, executed-translated-blocks)

Command disassemble Arguments

Name	Type	Required	Description
count	int	no	Number of instructions
pa	int	no	Physical address
va	int	no	Virtual address

Command disassembleBlock Arguments

Name	Type	Required	Description
pa	int	no	Physical address of block
va	int	no	Virtual address of block

Command disassembleBlockIR Arguments

Name	Type	Required	Description
pa	int	no	Physical address of block
va	int	no	Virtual address of block

Command disassembleBlockOption Arguments

The use and/or disclosure,etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.

Name	Type	Required	Description
option	string	yes	Option to set: 'default-syntax', 'alternate-syntax', 'latency', 'no-latency'.

Command enable-jit-timing Arguments

Name	Type	Required	Description
print	int	no	Enable log-messages

Command enableStats Arguments

Name	Type	Required	Description
stat	string	yes	Name of statistics (executed-translated-instructions, executed-translated-blocks)

Command find-page-table-root Arguments

Name	Type	Required	Description
pa	int	no	Physical Address
va	int	yes	Virtual Address

Command pwin Arguments

Name	Type	Required	Description
win	int	no	Window ID (-1 = default == current window).

Command set-binary-translation-threshold Arguments

Name	Type	Required	Description
threshold	int	yes	Threshold

Command setPC Arguments

Name	Type	Required	Description
npc	int	no	New %npc (note if omitted %npc = %pc + 4)
pc	int	yes	New %pc

The use and/or disclosure,etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.

Command setReg Arguments

Name	Type	Required	Description
reg	string	yes	Register name
value	int	yes	Value

Command translateBlock Arguments

Name	Type	Required	Description
count	int	no	Number of instructions, omit for automatic.
pa	int	no	Physical address of block
va	int	no	Virtual address of block

Command translateFunc Arguments

Name	Type	Required	Description
pa	int	no	Physical address of block
va	int	no	Virtual address of block

2.6.4. LEON4

@Leon4 Reference

Properties

Name	Type	Description
Class	*void	Class object
Component	*void	Pointer to component object if part of component
LocalName	*char	Local name (in component, if applicable)
LogMessageFlags	uint64_t	Per-object log message suppression flags
LoggingFlags	uint64_t	Flags for logging info
Name	*char	Object name
ObjectID	uint64_t	Unique ObjectID.
ReplayLoggingFlags	uint64_t	Replay-time category logging flags

The use and/or disclosure,etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.

Name	Type	Description
TimeSource	*void	Time source object

Commands

Name	Description
delete	Dispose instance of @Leon4
new	Create new instance of Leon4

Command new Arguments

Name	Type	Required	Description
cpuid	int	no	cpuid in multiprocessor config
cputype	string	no	CPU type (none, ngmp, gr740)
freq	double	no	frequency in Hz
name	string	yes	Name of object to create

Leon4 Reference

Properties

Name	Type	Description
CPUId	uint32_t	
CPUType	int32_t	
Class	*void	Class object
Component	*void	Pointer to component object if part of component
HostFlags	uint64_t	
LocalName	*char	Local name (in component, if applicable)
LogMessageFlags	uint64_t	Per-object log message suppression flags
LoggingFlags	uint64_t	Flags for logging info
Name	*char	Object name
ObjectID	uint64_t	Unique ObjectID.

The use and/or disclosure,etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.

Name	Type	Description
ReplayLoggingFlags	uint64_t	Replay-time category logging flags
TimeSource	*void	Time source object
asr	[uint32_t; 32]	
config.assemblerAliases	uint8_t	
config.exitOnSync	uint8_t	
config.handleEventWithinBlock	uint8_t	
config.logCallAndJump	uint8_t	
config.measureExecTime	uint8_t	
cpi	double	Cycles per instruction
currentContextId	uint64_t	
currentContextValid	uint8_t	
currentThreadContextId	uint64_t	
currentThreadContextValid	uint8_t	
cwp	*void	
dCache	temu_IfaceRef/ <unknown>	L1 data cache (optional)
devices	temu_IfaceRefArray	
exitReason	int32_t	
extraRegs	[uint32_t; 32]	
fprs	[uint32_t; 32]	
freq	int64_t	Frequency in Hz
fsr	uint32_t	Floating Point State Register
fsrInit	uint32_t	
fsrSpill	uint32_t	
g	[uint32_t; 8]	
gprs	[uint32_t; 128]	
iCache	temu_IfaceRef/ <unknown>	L1 instruction cache (optional)
i_npc	uintptr_t	Intermediate code nPC
i_pc	uintptr_t	Intermediate code PC
idleSteps	int64_t	
ipc	double	Instructions per cycle
irqClient	temu_IfaceRef/ <unknown>	Interrupt controller (for ACKs)

The use and/or disclosure,etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.

Name	Type	Description
logTrampolines	uint8_t	
memAccess	temu_IfaceRef/ <unknown>	Level 1 memory access interface (MMU)
memAccessL2	temu_IfaceRef/ <unknown>	Level 2 memory access interface (physical)
memReset	temu_IfaceRef/ <unknown>	Memory space reset interface
memSpace	*void	Memory space.
mmuCtrl	uint32_t	
mmuCtxt	uint32_t	
mmuCtxtCount	uint32_t	
mmuCtxtPtr	uint32_t	
mmuFaultAddr	uint32_t	
mmuFaultStat	uint32_t	
nextEvent	int64_t	Next event
npc	uint32_t	Next Program Counter
parentTimeSource	*void	Parent time source
pc	uint32_t	Program Counter
pdcManager	temu_IfaceRef/ <unknown>	Pre-decode cache manager (normally memory space)
pendingInterrupts	uint16_t	
powerState	uint32_t	Power State property (deprecated)
psr	uint32_t	Processor State Register
resetNpc	uint32_t	Reset nPC (for LEON4)
resetPc	uint32_t	Reset PC (for LEON4)
skipIdleTags	uint64_t	
startSteps	int64_t	Start steps of the time source
state	int32_t	
stats.mmuCtrlWrites	uint64_t	
stats.mmuCtxtPtrWrites	uint64_t	
stats.mmuCtxtWrites	uint64_t	
stats.raisedTraps	uint64_t	
stats.takenInterrupts	uint64_t	

The use and/or disclosure,etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.

Name	Type	Description
steps	int64_t	Steps of the time source
stickyFlags	uint32_t	Set bit 0 to 1 to not exit CPU on halted mode.
targetExec	temu_IfaceRef/ <unknown>	Target execution interface
targetSteps	int64_t	Target steps of the time source
tbr	uint32_t	Trap Base Register
wim	uint32_t	Window Invalidation Mask

Interfaces

Name	Type	Description
AhbIface	AhbIface	
AssemblerIface	temu::AssemblerIface	Assembler interface
BinaryTranslationIface	temu::BinaryTranslationControlIface	
ClockIface	ClockIface	
CodePatternIface	temu::CodePatternIface	
CpuIface	temu::CpuIface	
DCacheCtrlIface	temu::CacheCtrlIface	
DynamicResetAddressIface	temu::DynamicResetAddressIface	
ExecIface	temu::TargetExecutionIface	
ICacheCtrlIface	temu::CacheCtrlIface	
InvalidMemAccessIface	MemAccessIface	
IrqIface	IrqCtrlIface	
MmuMemAccessIface	MemAccessIface	
ObjectIface	ObjectIface	
PhysicalMemAccessIface	MemAccessIface	
PowerIface	PowerIface	
RegisterIface	temu::RegisterIface	Auto-generated Register Interface Impl.
ResetIface	ResetIface	
SparcIface	SparcIface	
VirtualMemAccessIface	MemAccessIface	

The use and/or disclosure,etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.

Ports

Prop	Iface	Description
irqClient	IrqIface	interrupt controller interface

Commands

Name	Description
add-call	Adds a call at the specific address. The call operation calls a predefined method which logs that it is invoked.It is primarily useful for debugging.
add-idle	Adds an explicit idle operation at the specific address.
add-skip	Adds a skip operation at the specific address. After this, the given number of instructions will be skipped.
assemble	Assemble instruction
auto-log-block	Log translation of block when translating for address
auto-log-instructions	Log instructions in block with the given address
break-on-trap	Break on trap
chainBlocks	Manually chain blocks
delete	Dispose instance of Leon4
disable-block-entry-logging	Disable logging on block entries.
disable-call-target-profiling	Disable call target profiling for triggering DBT.
disable-dbt-logging	Disable logging when translating code.
disable-dbt-validation	Disable validation of emitted machine code.
disable-fallthrough-profiling	Dis profiling of fallthroughs for triggering DBT.
disable-floating-point-translation	Disable binary translation of floating point instructions.
disable-hard-code-reset	Disable hard resets of DBT emitter cache.
disable-indirect-jump-profiling	Disable profiling of indirect jump targets for triggering DBT.
disable-jit-timing	Disable jit-timing.
disable-off-page-branch-profiling	Disable profiling of off-page branch targets for triggering DBT.

The use and/or disclosure,etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.

Name	Description
disable-on-page-branch-profiling	Disable profiling of on-page branch targets for triggering DBT.
disable-translation-logging	Disable jit-logging.
disableAutoChain	Disable binary translation block chaining.
disableProfiling	Disable profiling mode
disableStats	Disable statistics
disableTranslation	Disable binary translation.
disableTraps	Disable traps.
disassemble	Disassemble code
disassembleBlock	Disassemble binary translated code block.
disassembleBlockIR	Disassemble LLVM IR of translated code block
disassembleBlockOption	Set option for disassembler. Switch between AT&T (default) and Intel (alternate) syntax. Enable / disable insruction latency in output.
enable-block-entry-logging	Enable logging on block entries.
enable-call-target-profiling	Enable call target profiling for triggering DBT.
enable-cfm-crc	Enable Code Fragment Manger CRCs
enable-dbt-logging	Enable logging when translating code.
enable-dbt-validation	Enable validation of emitted machine code.
enable-fallthrough-profiling	Enable profiling of fallthroughs for triggering DBT.
enable-floating-point-translation	Enable binary translation of floating point instructions.
enable-hard-code-reset	Enable hard resets of DBT emitter cache.
enable-indirect-jump-profiling	Enable profiling of indirect jump targets for triggering DBT.
enable-jit-timing	Enable jit-timing.
enable-off-page-branch-profiling	Enable profiling of off-page branch targets for triggering DBT.
enable-on-page-branch-profiling	Enable profiling of on-page branch targets for triggering DBT.
enable-translation-logging	Enable jit-logging.
enableAutoChain	Enable binary translation block chaining.
enableProfiling	Enable profiling mode

The use and/or disclosure,etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.

Name	Description
enableStats	Enable statistics
enableTranslation	Enable binary translation.
enableTraps	Enable traps.
find-page-table-root	Scan through the page tables starting with context zero and determine which page table roots can generate the translation.
flushProfile	Flush profile data
pfregs	Print floating point registers for CPU
pregs	Print registers for CPU
print-atc	Print ATC for CPU
print-blocks	Show block connectivity
pstat	Print CPU stats
pwin	Print register window.
resetStats	Reset statistics counters
set-binary-translation-threshold	Set number of times a call target must be executed to trigger translation.
setPC	Set PC (and nPC)
setReg	Set register
stacktrace	Print a stacktrace starting at the current PC. The trace follows firstly the register windows if %i7 points at a call or jmpl addr, %o7 instruction. After register windows have been followed, the stacktrace walks through the in memory stack. Note that leaf functions are detected if %i7 < %o7 ⇐ %pc.
translateBlock	Translate block.
translateFunc	Translate function.
validate	Validates constant data.
wake-up	Wakes up the processor.

Command add-call Arguments

Name	Type	Required	Description
context	string	no	Context
pa	int	no	Physical Address
pamask	int	no	Physical Address Mask

The use and/or disclosure,etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.

Name	Type	Required	Description
script	path	no	Script to run when reaching address
va	int	no	Virtual Address
vamask	int	no	Virtual Address Mask

Command add-idle Arguments

Name	Type	Required	Description
context	string	no	Context
pa	int	no	Physical Address
pamask	int	no	Physical Address Mask
tag	int	no	Tag of idle operation
va	int	no	Virtual Address
vamask	int	no	Virtual Address Mask

Command add-skip Arguments

Name	Type	Required	Description
context	string	no	Context
pa	int	no	Physical Address
pamask	int	no	Physical Address Mask
steps	int	yes	Steps to skip
va	int	no	Virtual Address
vamask	int	no	Virtual Address Mask

Command assemble Arguments

Name	Type	Required	Description
inst	string	yes	Instruction to assemble.
pa	int	no	Physical address
va	int	no	Virtual address

Command auto-log-block Arguments

Name	Type	Required	Description
pa	int	yes	Physical Address

The use and/or disclosure,etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.

Command auto-log-instructions Arguments

Name	Type	Required	Description
pa	int	yes	Physical Address

Command break-on-trap Arguments

Name	Type	Required	Description
trap	string	yes	Trap name

Command chainBlocks Arguments

Name	Type	Required	Description
arm	string	no	Name of arm in source block (nt/not-taken or t/taken). Defaults to fall through arm (not-taken)
source	int	yes	Physical start address of source block.
target	int	yes	Physical start address of target block.

Command disableStats Arguments

Name	Type	Required	Description
stat	string	yes	Name of statistics (executed-translated-instructions, executed-translated-blocks)

Command disassemble Arguments

Name	Type	Required	Description
count	int	no	Number of instructions
pa	int	no	Physical address
va	int	no	Virtual address

Command disassembleBlock Arguments

Name	Type	Required	Description
pa	int	no	Physical address of block

The use and/or disclosure,etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.

Name	Type	Required	Description
va	int	no	Virtual address of block

Command disassembleBlockIR Arguments

Name	Type	Required	Description
pa	int	no	Physical address of block
va	int	no	Virtual address of block

Command disassembleBlockOption Arguments

Name	Type	Required	Description
option	string	yes	Option to set: 'default-syntax', 'alternate-syntax', 'latency', 'no-latency'.

Command enable-jit-timing Arguments

Name	Type	Required	Description
print	int	no	Enable log-messages

Command enableStats Arguments

Name	Type	Required	Description
stat	string	yes	Name of statistics (executed-translated-instructions, executed-translated-blocks)

Command find-page-table-root Arguments

Name	Type	Required	Description
pa	int	no	Physical Address
va	int	yes	Virtual Address

Command pwin Arguments

Name	Type	Required	Description
win	int	no	Window ID (-1 = default == current window).

The use and/or disclosure,etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.

Command set-binary-translation-threshold Arguments

Name	Type	Required	Description
threshold	int	yes	Threshold

Command setPC Arguments

Name	Type	Required	Description
npc	int	no	New %npc (note if omitted %npc = %pc + 4)
pc	int	yes	New %pc

Command setReg Arguments

Name	Type	Required	Description
reg	string	yes	Register name
value	int	yes	Value

Command translateBlock Arguments

Name	Type	Required	Description
count	int	no	Number of instructions, omit for automatic.
pa	int	no	Physical address of block
va	int	no	Virtual address of block

Command translateFunc Arguments

Name	Type	Required	Description
pa	int	no	Physical address of block
va	int	no	Virtual address of block

2.6.5. LEON5

@Leon5 Reference

Properties

Name	Type	Description
Class	*void	Class object

The use and/or disclosure,etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.

Name	Type	Description
Component	*void	Pointer to component object if part of component
LocalName	*char	Local name (in component, if applicable)
LogMessageFlags	uint64_t	Per-object log message suppression flags
LoggingFlags	uint64_t	Flags for logging info
Name	*char	Object name
ObjectID	uint64_t	Unique ObjectID.
ReplayLoggingFlags	uint64_t	Replay-time category logging flags
TimeSource	*void	Time source object

Commands

Name	Description
delete	Dispose instance of @Leon5
new	Create new instance of Leon5

Command new Arguments

Name	Type	Required	Description
cpuid	int	no	cpuid in multiprocessor config
freq	double	no	frequency in Hz
name	string	yes	Name of object to create

Leon5 Reference

Properties

Name	Type	Description
CPUId	uint32_t	
CPUType	int32_t	
Class	*void	Class object
Component	*void	Pointer to component object if part of component

The use and/or disclosure,etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.

Name	Type	Description
HostFlags	uint64_t	
LocalName	*char	Local name (in component, if applicable)
LogMessageFlags	uint64_t	Per-object log message suppression flags
LoggingFlags	uint64_t	Flags for logging info
Name	*char	Object name
ObjectID	uint64_t	Unique ObjectID.
ReplayLoggingFlags	uint64_t	Replay-time category logging flags
TimeSource	*void	Time source object
asr	[uint32_t; 32]	
config.assemblerAliases	uint8_t	
config.exitOnSync	uint8_t	
config.handleEventWithinBlock	uint8_t	
config.logCallAndJump	uint8_t	
config.measureExecTime	uint8_t	
cpi	double	Cycles per instruction
cpuToCpu	temu_IfaceRef/ <unknown>	CPU to CPU Interface
currentContextId	uint64_t	
currentContextValid	uint8_t	
currentThreadContextId	uint64_t	
currentThreadContextValid	uint8_t	
cwp	*void	
dCache	temu_IfaceRef/ <unknown>	L1 data cache (optional)
devices	temu_IfaceRefArray	
exitReason	int32_t	
extraRegs	[uint32_t; 32]	
fprs	[uint32_t; 32]	
freq	int64_t	Frequency in Hz
fsr	uint32_t	Floating Point State Register
fsrInit	uint32_t	
fsrSpill	uint32_t	

The use and/or disclosure,etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.

Name	Type	Description
g	[uint32_t; 8]	
gprs	[uint32_t; 128]	
iCache	temu_IfaceRef/ <unknown>	L1 instruction cache (optional)
i_npc	uintptr_t	Intermediate code nPC
i_pc	uintptr_t	Intermediate code PC
idleSteps	int64_t	
ipc	double	Instructions per cycle
irqClient	temu_IfaceRef/ <unknown>	Interrupt controller (for ACKs)
logTrampolines	uint8_t	
memAccess	temu_IfaceRef/ <unknown>	Level 1 memory access interface (MMU)
memAccessL2	temu_IfaceRef/ <unknown>	Level 2 memory access interface (physical)
memReset	temu_IfaceRef/ <unknown>	Memory space reset interface
memSpace	*void	Memory space.
mmuCtrl	uint32_t	
mmuCtxt	uint32_t	
mmuCtxtCount	uint32_t	
mmuCtxtPtr	uint32_t	
mmuFaultAddr	uint32_t	
mmuFaultStat	uint32_t	
nextEvent	int64_t	Next event
npc	uint32_t	Next Program Counter
parentTimeSource	*void	Parent time source
pc	uint32_t	Program Counter
pdcManager	temu_IfaceRef/ <unknown>	Pre-decode cache manager (normally memory space)
pendingInterrupts	uint16_t	
powerState	uint32_t	Power State property (deprecated)
psr	uint32_t	Processor State Register
resetNpc	uint32_t	Reset nPC (for LEON5)
resetPc	uint32_t	Reset PC (for LEON5)

The use and/or disclosure,etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.

Name	Type	Description
skipIdleTags	uint64_t	
startSteps	int64_t	Start steps of the time source
state	int32_t	
stats.mmuCtrlWrites	uint64_t	
stats.mmuCtxtPtrWrites	uint64_t	
stats.mmuCtxtWrites	uint64_t	
stats.raisedTraps	uint64_t	
stats.takenInterrupts	uint64_t	
steps	int64_t	Steps of the time source
stickyFlags	uint32_t	Set bit 0 to 1 to not exit CPU on halted mode.
targetExec	temu_IfaceRef/ <unknown>	Target execution interface
targetSteps	int64_t	Target steps of the time source
tbr	uint32_t	Trap Base Register
wim	uint32_t	Window Invalidation Mask

Interfaces

Name	Type	Description
AhbIface	AhbIface	
AssemblerIface	temu::AssemblerIface	Assembler interface
BinaryTranslationIface	temu::BinaryTranslationControlIface	
ClockIface	ClockIface	
CodePatternIface	temu::CodePatternIface	
CpuIface	temu::CpuIface	
DCacheCtrlIface	temu::CacheCtrlIface	
DynamicResetAddressIface	temu::DynamicResetAddressIface	
ExecIface	temu::TargetExecutionIface	
ICacheCtrlIface	temu::CacheCtrlIface	
InvalidMemAccessIface	MemAccessIface	
IrqIface	IrqCtrlIface	
Leon5Iface	temu::Leon5Iface	

The use and/or disclosure,etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.

Name	Type	Description
MmuMemAccessIface	MemAccessIface	
ObjectIface	ObjectIface	
PhysicalMemAccessIface	MemAccessIface	
PowerIface	PowerIface	
RegisterIface	temu::RegisterIface	Auto-generated Register Interface Impl.
ResetIface	ResetIface	
SparcIface	SparcIface	
VirtualMemAccessIface	MemAccessIface	

Ports

Prop	Iface	Description
irqClient	IrqIface	interrupt controller interface

Commands

Name	Description
add-call	Adds a call at the specific address. The call operation calls a predefined method which logs that it is invoked.It is primarily useful for debugging.
add-idle	Adds an explicit idle operation at the specific address.
add-skip	Adds a skip operation at the specific address. After this, the given number of instructions will be skipped.
assemble	Assemble instruction
auto-log-block	Log translation of block when translating for address
auto-log-instructions	Log instructions in block with the given address
break-on-trap	Break on trap
chainBlocks	Manually chain blocks
delete	Dispose instance of Leon5
disable-block-entry-logging	Disable logging on block entries.
disable-call-target-profiling	Disable call target profiling for triggering DBT.
disable-dbt-logging	Disable logging when translating code.

The use and/or disclosure,etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.

Name	Description
disable-dbt-validation	Disable validation of emitted machine code.
disable-fallthrough-profiling	Dis profiling of fallthroughs for triggering DBT.
disable-floating-point-translation	Disable binary translation of floating point instructions.
disable-hard-code-reset	Disable hard resets of DBT emitter cache.
disable-indirect-jump-profiling	Disable profiling of indirect jump targets for triggering DBT.
disable-jit-timing	Disable jit-timing.
disable-off-page-branch-profiling	Disable profiling of off-page branch targets for triggering DBT.
disable-on-page-branch-profiling	Disable profiling of on-page branch targets for triggering DBT.
disable-translation-logging	Disable jit-logging.
disableAutoChain	Disable binary translation block chaining.
disableProfiling	Disable profiling mode
disableStats	Disable statistics
disableTranslation	Disable binary translation.
disableTraps	Disable traps.
disassemble	Disassemble code
disassembleBlock	Disassemble binary translated code block.
disassembleBlockIR	Disassemble LLVM IR of translated code block
disassembleBlockOption	Set option for disassembler. Switch between AT&T (default) and Intel (alternate) syntax. Enable / disable insruction latency in output.
enable-block-entry-logging	Enable logging on block entries.
enable-call-target-profiling	Enable call target profiling for triggering DBT.
enable-cfm-crc	Enable Code Fragment Manger CRCs
enable-dbt-logging	Enable logging when translating code.
enable-dbt-validation	Enable validation of emitted machine code.
enable-fallthrough-profiling	Enable profiling of fallthroughs for triggering DBT.
enable-floating-point-translation	Enable binary translation of floating point instructions.
enable-hard-code-reset	Enable hard resets of DBT emitter cache.

The use and/or disclosure,etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.

Name	Description
enable-indirect-jump-profiling	Enable profiling of indirect jump targets for triggering DBT.
enable-jit-timing	Enable jit-timing.
enable-off-page-branch-profiling	Enable profiling of off-page branch targets for triggering DBT.
enable-on-page-branch-profiling	Enable profiling of on-page branch targets for triggering DBT.
enable-translation-logging	Enable jit-logging.
enableAutoChain	Enable binary translation block chaining.
enableProfiling	Enable profiling mode
enableStats	Enable statistics
enableTranslation	Enable binary translation.
enableTraps	Enable traps.
find-page-table-root	Scan through the page tables starting with context zero and determine which page table roots can generate the translation.
flushProfile	Flush profile data
pfregs	Print floating point registers for CPU
pregs	Print registers for CPU
print-atc	Print ATC for CPU
print-blocks	Show block connectivity
pstat	Print CPU stats
pwin	Print register window.
resetStats	Reset statistics counters
set-binary-translation-threshold	Set number of times a call target must be executed to trigger translation.
setPC	Set PC (and nPC)
setReg	Set register
stacktrace	Print a stacktrace starting at the current PC. The trace follows firstly the register windows if %i7 points at a call or jmpl addr, %o7 instruction. After register windows have been followed, the stacktrace walks through the in memory stack. Note that leaf functions are detected if %i7 < %o7 ⇐ %pc.

The use and/or disclosure,etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.

Name	Description
translateBlock	Translate block.
translateFunc	Translate function.
validate	Validates constant data.
wake-up	Wakes up the processor.

Command add-call Arguments

Name	Type	Required	Description
context	string	no	Context
pa	int	no	Physical Address
pamask	int	no	Physical Address Mask
script	path	no	Script to run when reaching address
va	int	no	Virtual Address
vamask	int	no	Virtual Address Mask

Command add-idle Arguments

Name	Type	Required	Description
context	string	no	Context
pa	int	no	Physical Address
pamask	int	no	Physical Address Mask
tag	int	no	Tag of idle operation
va	int	no	Virtual Address
vamask	int	no	Virtual Address Mask

Command add-skip Arguments

Name	Type	Required	Description
context	string	no	Context
pa	int	no	Physical Address
pamask	int	no	Physical Address Mask
steps	int	yes	Steps to skip
va	int	no	Virtual Address
vamask	int	no	Virtual Address Mask

Command assemble Arguments

The use and/or disclosure,etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.

Name	Type	Required	Description
inst	string	yes	Instruction to assemble.
pa	int	no	Physical address
va	int	no	Virtual address

Command auto-log-block Arguments

Name	Type	Required	Description
pa	int	yes	Physical Address

Command auto-log-instructions Arguments

Name	Type	Required	Description
pa	int	yes	Physical Address

Command break-on-trap Arguments

Name	Type	Required	Description
trap	string	yes	Trap name

Command chainBlocks Arguments

Name	Type	Required	Description
arm	string	no	Name of arm in source block (nt/not-taken or t/taken). Defaults to fall through arm (not-taken)
source	int	yes	Physical start address of source block.
target	int	yes	Physical start address of target block.

Command disableStats Arguments

Name	Type	Required	Description
stat	string	yes	Name of statistics (executed-translated-instructions, executed-translated-blocks)

Command disassemble Arguments

The use and/or disclosure,etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.

Name	Type	Required	Description
count	int	no	Number of instructions
pa	int	no	Physical address
va	int	no	Virtual address

Command disassembleBlock Arguments

Name	Type	Required	Description
pa	int	no	Physical address of block
va	int	no	Virtual address of block

Command disassembleBlockIR Arguments

Name	Type	Required	Description
pa	int	no	Physical address of block
va	int	no	Virtual address of block

Command disassembleBlockOption Arguments

Name	Type	Required	Description
option	string	yes	Option to set: 'default-syntax', 'alternate-syntax', 'latency', 'no-latency'.

Command enable-jit-timing Arguments

Name	Type	Required	Description
print	int	no	Enable log-messages

Command enableStats Arguments

Name	Type	Required	Description
stat	string	yes	Name of statistics (executed-translated-instructions, executed-translated-blocks)

Command find-page-table-root Arguments

The use and/or disclosure,etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.

Name	Type	Required	Description
pa	int	no	Physical Address
va	int	yes	Virtual Address

Command pwin Arguments

Name	Type	Required	Description
win	int	no	Window ID (-1 = default == current window).

Command set-binary-translation-threshold Arguments

Name	Type	Required	Description
threshold	int	yes	Threshold

Command setPC Arguments

Name	Type	Required	Description
npc	int	no	New %npc (note if omitted %npc = %pc + 4)
pc	int	yes	New %pc

Command setReg Arguments

Name	Type	Required	Description
reg	string	yes	Register name
value	int	yes	Value

Command translateBlock Arguments

Name	Type	Required	Description
count	int	no	Number of instructions, omit for automatic.
pa	int	no	Physical address of block
va	int	no	Virtual address of block

Command translateFunc Arguments

The use and/or disclosure,etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.

Name	Type	Required	Description
pa	int	no	Physical address of block
va	int	no	Virtual address of block

The use and/or disclosure,etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.

Chapter 3. PowerPC

The PowerPC target comes with support for the 32 bit PowerPC architecture. It currently implements the PPC750 and E500v2 CPU models.

3.1. Variants

3.1.1. E500v2

The E500v2 models the NXP processor core of the same name. This includes the SPE instruction set.

@e500v2 Reference

Properties

Name	Type	Description
Class	*void	Class object
Component	*void	Pointer to component object if part of component
LocalName	*char	Local name (in component, if applicable)
LogMessageFlags	uint64_t	Per-object log message suppression flags
LoggingFlags	uint64_t	Flags for logging info
Name	*char	Object name
ObjectID	uint64_t	Unique ObjectID.
ReplayLoggingFlags	uint64_t	Replay-time category logging flags
TimeSource	*void	Time source object

Commands

Name	Description
delete	Dispose instance of @e500v2
new	Create new instance of e500v2

Command new Arguments

Name	Type	Required	Description
name	string	yes	Name of object to create

The use and/or disclosure,etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.

e500v2 Reference

Properties

Name	Type	Description
CPUId	uint32_t	
CPUType	int32_t	
Class	*void	Class object
ClockDivider	uint64_t	Current clock divider (relation CPU freq and CCB or TB freq).
Component	*void	Pointer to component object if part of component
HostFlags	uint64_t	
LocalName	*char	Local name (in component, if applicable)
LogMessageFlags	uint64_t	Per-object log message suppression flags
LoggingFlags	uint64_t	Flags for logging info
Name	*char	Object name
ObjectID	uint64_t	Unique ObjectID.
ReplayLoggingFlags	uint64_t	Replay-time category logging flags
TimeSource	*void	Time source object
acc	uint64_t	Accumulator register
atbl	uint32_t	Alternate Time Base Lower
atbu	uint32_t	Alternate Time Base Upper
bbear	uint32_t	Branch Buffer Entry Address Register
bbtar	uint32_t	Branch Buffer Target Address Register
bucsr	uint32_t	Branch Unit Control and Status Register
config.CCBFrequency	uint64_t	CCB Clock Frequency
config.TBFrequency	uint64_t	Time Base Clock Frequency (same as RTC on e500)
config.assemblerAliases	uint8_t	
config.exitOnSync	uint8_t	

The use and/or disclosure,etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.

Name	Type	Description
config.handleEventWithinBlock	uint8_t	
config.measureExecTime	uint8_t	
cpi	double	Cycles per instruction
cr	uint32_t	
csrrs	[uint32_t; 2]	
ctr	uint32_t	Count Register
currentContextId	uint64_t	
currentContextValid	uint8_t	
currentThreadContextId	uint64_t	
currentThreadContextValid	uint8_t	
dac1	uint32_t	Data Address Compare Register 1
dac2	uint32_t	Data Address Compare Register 2
dbcrs	[uint32_t; 3]	
dbsr	uint32_t	Debug Status Register
dear	uint32_t	Data Exception Address Register
dec	uint32_t	Decrementer
decar	uint32_t	Decrementer Auto-Reload Register
devices	temu_IfaceRefArray	
dvc1	uint32_t	Data Value Compare Register 1
dvc2	uint32_t	Data Value Compare Register 2
esr	uint32_t	Exception Syndrome Register
exitReason	int32_t	
freq	int64_t	Frequency in Hz
gprs	[uint64_t; 32]	
hid0	uint32_t	Hardware Implementation-Dependent Register 0
hid1	uint32_t	Hardware Implementation-Dependent Register 1
iac1	uint32_t	Instruction Address Compare 1 Register

The use and/or disclosure,etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.

Name	Type	Description
iac2	uint32_t	Instruction Address Compare 2 Register
idleSteps	int64_t	
ipc	double	Instructions per cycle
irq	int8_t	
irqClient	temu_IfaceRef/ <unknown>	
ivors	[uint32_t; 64]	
ivpr	uint32_t	Interrupt Vector Prefix Register
l1cfg0	uint32_t	L1 Cache Configuration Register 0
l1cfg1	uint32_t	L1 Cache Configuration Register 1
l1csr0	uint32_t	L1 Cache Control and Status Register 0
l1csr1	uint32_t	L1 Cache Control and Status Register 1
logTrampolines	uint8_t	
lr	uint32_t	Link register
mass	[uint32_t; 7]	
mcar	uint32_t	Machine Check Address Register
mcsr	uint32_t	Machine Check Syndrome Register
mcsrrs	[uint32_t; 2]	
memAccess	temu_IfaceRef/ <unknown>	Level 1 memory access interface (MMU)
memAccessL2	temu_IfaceRef/ <unknown>	Level 2 memory access interface (physical)
memReset	temu_IfaceRef/ <unknown>	Memory space reset interface
memSpace	*void	Memory space.
mmucfg	uint32_t	
mmucsr0	uint32_t	
msr	uint32_t	
nextEvent	int64_t	Next event

The use and/or disclosure,etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.

Name	Type	Description
parentTimeSource	*void	Parent time source
pc	uint32_t	Program counter
pdcManager	temu_IfaceRef/ <unknown>	Pre-decode cache manager (normally memory space)
pid0	uint32_t	Process ID Register 0
pid1	uint32_t	Process ID Register 1
pid2	uint32_t	Process ID Register 2
pir	uint32_t	Processor ID Register
pmc0	uint32_t	Performance Monitor Counter 0
pmc1	uint32_t	Performance Monitor Counter 1
pmc2	uint32_t	Performance Monitor Counter 2
pmc3	uint32_t	Performance Monitor Counter 3
pmgc0	uint32_t	Performance Monitor Global Control Register 0
pmlca0	uint32_t	Performance Monitor Local Control a0
pmlca1	uint32_t	Performance Monitor Local Control a1
pmlca2	uint32_t	Performance Monitor Local Control a2
pmlca3	uint32_t	Performance Monitor Local Control a3
pmlcb0	uint32_t	Performance Monitor Local Control b0
pmlcb1	uint32_t	Performance Monitor Local Control b1
pmlcb2	uint32_t	Performance Monitor Local Control b2
pmlcb3	uint32_t	Performance Monitor Local Control b3
powerState	uint32_t	Power State property (deprecated)
pvr	uint32_t	Processor Version Register
skipIdleTags	uint64_t	
spefscr	uint32_t	

The use and/or disclosure,etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.

Name	Type	Description
spr976	uint32_t	
sprgs	[uint32_t; 8]	
srrs	[uint32_t; 2]	
startSteps	int64_t	Start steps of the time source
state	int32_t	
steps	int64_t	Steps of the time source
stickyFlags	uint32_t	Set bit 0 to 1 to not exit CPU on halted mode.
svr	uint32_t	System Version Register
targetExec	temu_IfaceRef/ <unknown>	Target execution interface
targetSteps	int64_t	Target steps of the time source
tcr	uint32_t	Timer Control Register
tlb0Entries	[uint32_t; 8192]	TLB0 Entries
tlb0Nv	uint32_t	TLB0 NV
tlb0cfg	uint32_t	TLB0 Configuration Register
tlb1Entries	[uint32_t; 64]	TLB1 Entries
tlb1cfg	uint32_t	TLB1 Configuration Register
tsr	uint32_t	Timer Status Register
upmc0	uint32_t	User Performance Monitor Counter 0
upmc1	uint32_t	User Performance Monitor Counter 1
upmc2	uint32_t	User Performance Monitor Counter 2
upmc3	uint32_t	User Performance Monitor Counter 3
upmgc0	uint32_t	User Performance Monitor Global Control Register 0
upmlca0	uint32_t	User Performance Monitor Local Control a0
upmlca1	uint32_t	User Performance Monitor Local Control a1
upmlca2	uint32_t	User Performance Monitor Local Control a2

The use and/or disclosure,etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.

Name	Type	Description
upmlca3	uint32_t	User Performance Monitor Local Control a3
upmlcb0	uint32_t	User Performance Monitor Local Control b0
upmlcb1	uint32_t	User Performance Monitor Local Control b1
upmlcb2	uint32_t	User Performance Monitor Local Control b2
upmlcb3	uint32_t	User Performance Monitor Local Control b3
usrpg0	uint32_t	User SPR General Register 0
xer	uint32_t	Integer exception register

Interfaces

Name	Type	Description
AssemblerIface	temu::AssemblerIface	Assembler interface
BinaryTranslationIface	temu::BinaryTranslationControlIface	
ClockIface	ClockIface	
CodePatternIface	temu::CodePatternIface	
CpuIface	temu::CpuIface	
E500MMUIface	temu::E500MMUIface	MMU interface for the E500
ExecIface	temu::TargetExecutionIface	
InvalidMemAccessIface	MemAccessIface	
IrqIface	IrqCtrlIface	
MmuMemAccessIface	MemAccessIface	MMU access interface
ObjectIface	ObjectIface	
PhysicalMemAccessIface	MemAccessIface	
PowerIface	PowerIface	
PowerPCIface	temu::PowerPCIface	
RegisterIface	temu::RegisterIface	Auto-generated Register Interface Impl.
ResetIface	ResetIface	
VirtualMemAccessIface	MemAccessIface	

The use and/or disclosure,etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.

Ports

Prop	Iface	Description
irqClient	IrqIface	interrupt controller interface

Commands

Name	Description
add-call	Adds a call at the specific address. The call operation calls a predefined method which logs that it is invoked.It is primarily useful for debugging.
add-idle	Adds an explicit idle operation at the specific address.
add-skip	Adds a skip operation at the specific address. After this, the given number of instructions will be skipped.
assemble	Assemble instruction
delete	Dispose instance of e500v2
disable-dbt-logging	Disable logging when translating code.
disable-dbt-validation	Disable validation of emitted machine code.
disable-hard-code-reset	Disable hard resets of DBT emitter cache.
disable-stats	Disable statistics
disable-translation-logging	Disable jit-logging.
disassemble-block	Disassemble binary translated code block.
disassemble-block-option	Set option for disassembler. Switch between AT&T (default) and Intel (alternate) syntax. Enable / disable insruction latency in output.
enable-dbt-logging	Enable logging when translating code.
enable-dbt-validation	Enable validation of emitted machine code.
enable-hard-code-reset	Enable hard resets of DBT emitter cache.
enable-stats	Enable statistics
enable-translation-logging	Enable jit-logging.
pfregs	Print floating point registers for CPU
pregs	Print registers for CPU
printTLB0	Print TLB0
printTLB1	Print TLB1

The use and/or disclosure,etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.

Name	Description
pstat	Print CPU stats
raiseCritical	Raise critical interrupt
raiseExternal	Raise external interrupt
set-reg	Set register
setPC	Set PC
setTLB0Entry	Add entry to TLB0
setTLB1Entry	Add entry to TLB1
stacktrace	ABI stack trace
translate-block	Translate block.
translate-func	Translate function.
wake-up	Wakes up the processor.

Command add-call Arguments

Name	Type	Required	Description
context	string	no	Context
pa	int	no	Physical Address
pamask	int	no	Physical Address Mask
script	path	no	Script to run when reaching address
va	int	no	Virtual Address
vamask	int	no	Virtual Address Mask

Command add-idle Arguments

Name	Type	Required	Description
context	string	no	Context
pa	int	no	Physical Address
pamask	int	no	Physical Address Mask
tag	int	no	Tag of idle operation
va	int	no	Virtual Address
vamask	int	no	Virtual Address Mask

Command add-skip Arguments

The use and/or disclosure,etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.

Name	Type	Required	Description
context	string	no	Context
pa	int	no	Physical Address
pamask	int	no	Physical Address Mask
steps	int	yes	Steps to skip
va	int	no	Virtual Address
vamask	int	no	Virtual Address Mask

Command assemble Arguments

Name	Type	Required	Description
inst	string	yes	Instruction to assemble.
pa	int	no	Physical address
va	int	no	Virtual address

Command disable-stats Arguments

Name	Type	Required	Description
stat	string	yes	Name of statistics (executed-translated-instructions, executed-translated-blocks)

Command disassemble-block Arguments

Name	Type	Required	Description
pa	int	no	Physical address of block
va	int	no	Virtual address of block

Command disassemble-block-option Arguments

Name	Type	Required	Description
option	string	yes	Option to set: 'default-syntax', 'alternate-syntax', 'latency', 'no-latency'.

Command enable-stats Arguments

The use and/or disclosure,etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.

Name	Type	Required	Description
stat	string	yes	Name of statistics (executed-translated-instructions, executed-translated-blocks)

Command set-reg Arguments

Name	Type	Required	Description
reg	string	yes	Register name
value	int	yes	Value

Command setPC Arguments

Name	Type	Required	Description
pc	int	yes	New pc

Command setTLB0Entry Arguments

Name	Type	Required	Description
mas1	int	yes	MAS1
mas2	int	yes	MAS2
mas3	int	yes	MAS3
mas7	int	yes	MAS7
set	int	yes	Set [0-127]
way	int	yes	Way [0-3]

Command setTLB1Entry Arguments

Name	Type	Required	Description
entry	int	yes	Entry [0-15]
mas1	int	yes	MAS1
mas2	int	yes	MAS2
mas3	int	yes	MAS3
mas7	int	yes	MAS7

Command translate-block Arguments

The use and/or disclosure,etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.

Name	Type	Required	Description
count	int	no	Number of instructions, omit for automatic.
pa	int	no	Physical address of block
va	int	no	Virtual address of block

Command translate-func Arguments

Name	Type	Required	Description
pa	int	no	Physical address of block
va	int	no	Virtual address of block

Note on Trap Support

TEMU supports the raising of traps, via either the API or the command line. Some Book-E PowerPC traps are parametarized, and the standard interfaces sets the parameters to zero.

The parameters need to be applied manually after using the interfaces or command line.

Table 1. Traps Requiring Manual Setting of Registers Afterwards

Trap Name	Note
`data_storage`	`dear` and `esr` registers need to be set.
`instruction_storage`	`esr` register needs to be set.
`alignment_interrupt`	`dear` and `esr` registers need to be set.
`program_interrupt`	`esr` needs to be set (upper 8 bits).
`data_TLB_error`	`dear` and `esr` registers need to be set.

In addition, the e500v2 core does not support all Book-E traps, please consult the hardware documentation for further information.

Limitations

- No static timing model is defined at this moment. That means that one instruction take one cycle to finish.
- MMU model is not yet validated against hardware.
- Cache control interfaces are not implemented or supported, this can be addressed if needed.
- The errata described in the e500CORERMAD is not correctly simulated. the following instructions deviates from hardware errata and are implemented as documented instead:
  - `evmwlssiaaw`

The use and/or disclosure,etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.

- evmwlssianw
- evmwlusiaaw
- evmwlusianw

3.1.2. PPC750

The TEMU PowerPC 750 model, models a PPC750CX processor core.

@ppc750 Reference

Properties

Name	Type	Description
Class	*void	Class object
Component	*void	Pointer to component object if part of component
LocalName	*char	Local name (in component, if applicable)
LogMessageFlags	uint64_t	Per-object log message suppression flags
LoggingFlags	uint64_t	Flags for logging info
Name	*char	Object name
ObjectID	uint64_t	Unique ObjectID.
ReplayLoggingFlags	uint64_t	Replay-time category logging flags
TimeSource	*void	Time source object

Commands

Name	Description
delete	Dispose instance of @ppc750
new	Create new instance of ppc750

Command new Arguments

Name	Type	Required	Description
name	string	yes	Name of object to create

ppc750 Reference

Properties

The use and/or disclosure,etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.

Name	Type	Description
CPUId	uint32_t	
CPUType	int32_t	
Class	*void	Class object
Component	*void	Pointer to component object if part of component
HostFlags	uint64_t	
LocalName	*char	Local name (in component, if applicable)
LogMessageFlags	uint64_t	Per-object log message suppression flags
LoggingFlags	uint64_t	Flags for logging info
Name	*char	Object name
ObjectID	uint64_t	Unique ObjectID.
ReplayLoggingFlags	uint64_t	Replay-time category logging flags
TimeSource	*void	Time source object
config.assemblerAliases	uint8_t	
config.exitOnSync	uint8_t	
config.handleEventWithinBlock	uint8_t	
config.measureExecTime	uint8_t	
cpi	double	Cycles per instruction
cr	uint32_t	
ctr	uint32_t	
currentContextId	uint64_t	
currentContextValid	uint8_t	
currentThreadContextId	uint64_t	
currentThreadContextValid	uint8_t	
dabr	uint32_t	
dar	uint32_t	
dbats	[uint32_t; 16]	
dec	uint32_t	
devices	temu_IfaceRefArray	
dmal	uint32_t	

The use and/or disclosure,etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.

Name	Type	Description
dmau	uint32_t	
dsisr	uint32_t	
ear	uint32_t	
exitReason	int32_t	
fprs	[uint64_t; 32]	
freq	int64_t	Frequency in Hz
gprs	[uint32_t; 32]	
gqrs	[uint32_t; 8]	
hids	[uint32_t; 4]	
iabr	uint32_t	
ibats	[uint32_t; 16]	
ictc	uint32_t	
idleSteps	int64_t	
ipc	double	Instructions per cycle
irq	int8_t	
irqClient	temu_IfaceRef/ <unknown>	
l2cr	uint32_t	
logTrampolines	uint8_t	
lr	uint32_t	
memAccess	temu_IfaceRef/ <unknown>	Level 1 memory access interface (MMU)
memAccessL2	temu_IfaceRef/ <unknown>	Level 2 memory access interface (physical)
memReset	temu_IfaceRef/ <unknown>	Memory space reset interface
memSpace	*void	Memory space.
mmcrs	[uint32_t; 2]	
msr	uint32_t	
nextEvent	int64_t	Next event
parentTimeSource	*void	Parent time source
pc	uint32_t	Program counter
pdcManager	temu_IfaceRef/ <unknown>	Pre-decode cache manager (normally memory space)
pmcs	[uint32_t; 4]	

The use and/or disclosure,etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.

Name	Type	Description
powerState	uint32_t	Power State property (deprecated)
pvr	uint32_t	
sdr1	uint32_t	
sia	uint32_t	
skipIdleTags	uint64_t	
sprgs	[uint32_t; 4]	
srrs	[uint32_t; 2]	
startSteps	int64_t	Start steps of the time source
state	int32_t	
steps	int64_t	Steps of the time source
stickyFlags	uint32_t	Set bit 0 to 1 to not exit CPU on halted mode.
targetExec	temu_IfaceRef/ <unknown>	Target execution interface
targetSteps	int64_t	Target steps of the time source
tdch	uint32_t	
tdcl	uint32_t	
thrms	[uint32_t; 3]	
uisa	uint32_t	
ummcrs	[uint32_t; 2]	
upmcs	[uint32_t; 4]	
wpar	uint32_t	
xer	uint32_t	

Interfaces

Name	Type	Description
AssemblerIface	temu::AssemblerIface	Assembler interface
BinaryTranslationIface	temu::BinaryTranslationControlIface	
ClockIface	ClockIface	
CodePatternIface	temu::CodePatternIface	
CpuIface	temu::CpuIface	
ExecIface	temu::TargetExecutionIface	

The use and/or disclosure,etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.

Name	Type	Description
InvalidMemAccessIface	MemAccessIface	
IrqIface	IrqCtrlIface	
MmuMemAccessIface	MemAccessIface	
ObjectIface	ObjectIface	
PhysicalMemAccessIface	MemAccessIface	
PowerIface	PowerIface	
PowerPCIface	temu::PowerPCIface	
RegisterIface	temu::RegisterIface	Auto-generated Register Interface Impl.
ResetIface	ResetIface	
VirtualMemAccessIface	MemAccessIface	

Ports

Prop	Iface	Description
irqClient	IrqIface	interrupt controller interface

Commands

Name	Description
add-call	Adds a call at the specific address. The call operation calls a predefined method which logs that it is invoked.It is primarily useful for debugging.
add-idle	Adds an explicit idle operation at the specific address.
add-skip	Adds a skip operation at the specific address. After this, the given number of instructions will be skipped.
assemble	Assemble instruction
delete	Dispose instance of ppc750
disable-dbt-logging	Disable logging when translating code.
disable-dbt-validation	Disable validation of emitted machine code.
disable-hard-code-reset	Disable hard resets of DBT emitter cache.
disable-stats	Disable statistics
disable-translation-logging	Disable jit-logging.

The use and/or disclosure,etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.

Name	Description
disassemble-block	Disassemble binary translated code block.
disassemble-block-option	Set option for disassembler. Switch between AT&T (default) and Intel (alternate) syntax. Enable / disable insruction latency in output.
enable-dbt-logging	Enable logging when translating code.
enable-dbt-validation	Enable validation of emitted machine code.
enable-hard-code-reset	Enable hard resets of DBT emitter cache.
enable-stats	Enable statistics
enable-translation-logging	Enable jit-logging.
pfregs	Print floating point registers for CPU
pregs	Print registers for CPU
pstat	Print CPU stats
set-reg	Set register
setPC	Set PC
stacktrace	ABI stack trace
translate-block	Translate block.
translate-func	Translate function.
wake-up	Wakes up the processor.

Command add-call Arguments

Name	Type	Required	Description
context	string	no	Context
pa	int	no	Physical Address
pamask	int	no	Physical Address Mask
script	path	no	Script to run when reaching address
va	int	no	Virtual Address
vamask	int	no	Virtual Address Mask

Command add-idle Arguments

Name	Type	Required	Description
context	string	no	Context
pa	int	no	Physical Address

The use and/or disclosure,etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.

Name	Type	Required	Description
pamask	int	no	Physical Address Mask
tag	int	no	Tag of idle operation
va	int	no	Virtual Address
vamask	int	no	Virtual Address Mask

Command add-skip Arguments

Name	Type	Required	Description
context	string	no	Context
pa	int	no	Physical Address
pamask	int	no	Physical Address Mask
steps	int	yes	Steps to skip
va	int	no	Virtual Address
vamask	int	no	Virtual Address Mask

Command assemble Arguments

Name	Type	Required	Description
inst	string	yes	Instruction to assemble.
pa	int	no	Physical address
va	int	no	Virtual address

Command disable-stats Arguments

Name	Type	Required	Description
stat	string	yes	Name of statistics (executed-translated-instructions, executed-translated-blocks)

Command disassemble-block Arguments

Name	Type	Required	Description
pa	int	no	Physical address of block
va	int	no	Virtual address of block

Command disassemble-block-option Arguments

The use and/or disclosure,etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.

Name	Type	Required	Description
option	string	yes	Option to set: 'default-syntax', 'alternate-syntax', 'latency', 'no-latency'.

Command enable-stats Arguments

Name	Type	Required	Description
stat	string	yes	Name of statistics (executed-translated-instructions, executed-translated-blocks)

Command set-reg Arguments

Name	Type	Required	Description
reg	string	yes	Register name
value	int	yes	Value

Command setPC Arguments

Name	Type	Required	Description
pc	int	yes	New pc

Command translate-block Arguments

Name	Type	Required	Description
count	int	no	Number of instructions, omit for automatic.
pa	int	no	Physical address of block
va	int	no	Virtual address of block

Command translate-func Arguments

Name	Type	Required	Description
pa	int	no	Physical address of block
va	int	no	Virtual address of block

Note on Trap Support

The use and/or disclosure,etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.

TEMU supports the raising of traps, via either the API or the command line. Some Power traps are parametarized, and the standard interfaces sets the parameters to zero.

The parameters need to be applied manually after using the interfaces or command line.

Table 2. Traps Requiring Manual Setting of Registers Afterwards

Trap Name	Note
`data_storage`	`dear` and `esr` registers need to be set.
`data_segment`	`dear` and `esr` registers need to be set.
`instruction_storage`	`esr` register needs to be set.
`alignment`	`dear` and `esr` registers need to be set.
`program`	`esr` needs to be set (upper 8 bits).
`data_TLB_error`	`dear` and `esr` registers need to be set.

Limitations

- No static timing model is defined at this moment. That means that one instruction take one cycle to finish.
- AltiVec instructions are not implemented at this moment. These can be added if such a PowerPC model is requested.
- MMU model is not yet validated against hardware.

The use and/or disclosure,etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.

Chapter 4. RISC-V

The RISC-V target comes with support for the 32- and 64-bit RISC-V architecture. It currently implements the NOEL-V CPU model.

4.1. Variants

4.1.1. NOEL-V

The NOEL-V models the NOEL-V RISC-V processor core of the same name. This includes the following instruction set extensions:

- RV64I: 64-bit base instruction set
- RV32I: 32-bit base instruction set
- M: Multiply and Divide
- A: Atomics
- FD: Single/Double precision floating point
- B: Bit-manipulation
- C: Compressed Instructions
- H: Hypervisor
- Zicntr: Counters
- Zicsr: Control and status register
- Zifencei: Instruction-fetch fence

The model is validated against the RISC-V conformance test suite.

@NOELV Reference

Properties

Name	Type	Description
Class	*void	Class object
Component	*void	Pointer to component object if part of component
LocalName	*char	Local name (in component, if applicable)
LogMessageFlags	uint64_t	Per-object log message suppression flags
LoggingFlags	uint64_t	Flags for logging info
Name	*char	Object name

The use and/or disclosure,etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.

Name	Type	Description
ObjectID	uint64_t	Unique ObjectID.
ReplayLoggingFlags	uint64_t	Replay-time category logging flags
TimeSource	*void	Time source object

Commands

Name	Description
delete	Dispose instance of @NOELV
new	Create new instance of NOELV

Command new Arguments

Name	Type	Required	Description
hartid	int	no	Hart ID (mhartid) in multiprocessor config
name	string	yes	Name of object to create

NOELV Reference

Properties

Name	Type	Description
CPUId	uint32_t	
CPUType	int32_t	
Class	*void	Class object
Component	*void	Pointer to component object if part of component
HostFlags	uint64_t	
LocalName	*char	Local name (in component, if applicable)
LogMessageFlags	uint64_t	Per-object log message suppression flags
LoggingFlags	uint64_t	Flags for logging info
Name	*char	Object name
ObjectID	uint64_t	Unique ObjectID.

The use and/or disclosure,etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.

Name	Type	Description
ReplayLoggingFlags	uint64_t	Replay-time category logging flags
TimeSource	*void	Time source object
aclintCsrClient	temu_IfaceRef/ <unknown>	Aclint Csr interface
aiaCsrClient	temu_IfaceRef/ <unknown>	Interrupt controller Csr interface
config.MMULogging	uint8_t	Enable MMU-related logging
config.PMPLogging	uint8_t	Enable PMP-related logging
config.assemblerAliases	uint8_t	
config.exitOnSync	uint8_t	
config.handleEventWithinBlock	uint8_t	
config.logTrampolines	uint8_t	
config.measureExecTime	uint8_t	
config.numPMPEntries	uint8_t	Number of PMP entries (0, 16 or 64).
cpi	double	Cycles per instruction
currentContextId	uint64_t	
currentContextValid	uint8_t	
currentThreadContextId	uint64_t	
currentThreadContextValid	uint8_t	
exitReason	int32_t	
fcsr	uint32_t	
freq	int64_t	Frequency in Hz
gprs	[uint64_t; 32]	
hcounteren	uint32_t	
hedeleg	uint64_t	
henvcfg	uint64_t	
hgatp	uint64_t	
hgeie	uint64_t	
hgeip	uint64_t	
hideleg	uint64_t	
hie	uint64_t	
hip	uint64_t	

The use and/or disclosure,etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.

Name	Type	Description
hstatus	uint64_t	
htinst	uint64_t	
htval	uint64_t	
hvip	uint64_t	
idleSteps	int64_t	
ipc	double	Instructions per cycle
irqClient	temu_IfaceRef/ <unknown>	Interrupt controller (for ACKs)
marchid	uint64_t	
mcause	uint64_t	
mconfigptr	uint64_t	
mcounteren	uint32_t	
mcountinhibit	uint32_t	
mcycle	uint64_t	
medeleg	uint64_t	
memAccess	temu_IfaceRef/ <unknown>	Level 1 memory access interface (MMU)
memAccessL2	temu_IfaceRef/ <unknown>	Level 2 memory access interface (physical)
memReset	temu_IfaceRef/ <unknown>	Memory space reset interface
memSpace	*void	Memory space.
menvcfg	uint64_t	
mepc	uint64_t	
mhartid	uint64_t	
mhpmcounter	[uint64_t; 29]	
mhpmevent	[uint64_t; 29]	
mideleg	uint64_t	
mie	uint64_t	
mimpid	uint64_t	
minstret	uint64_t	
mip	uint64_t	
misa	uint64_t	
miselect	uint64_t	

The use and/or disclosure,etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.

Name	Type	Description
mncause	uint64_t	
mnepc	uint64_t	
mnscratch	uint64_t	
mnstatus	uint64_t	
mscratch	uint64_t	
mstatus	uint64_t	
mtinst	uint64_t	
mtval	uint64_t	
mtval2	uint64_t	
mtvec	uint64_t	
mvendorid	uint32_t	
nextEvent	int64_t	Next event
parentTimeSource	*void	Parent time source
pc	uint64_t	Program counter register (pc)
pdcManager	temu_IfaceRef/ <unknown>	Pre-decode cache manager (normally memory space)
pmpaddr	[uint64_t; 64]	PMP Address CSRs
pmpcfg	[uint64_t; 16]	PMP Config Entry CSRs
powerState	uint32_t	Power State property (deprecated)
privilege	uint8_t	RISC-V Mode: 0=U, 1=S, 3=M
reservationSet	[uint64_t; 3]	
satp	uint64_t	
scause	uint64_t	
scounteren	uint32_t	
senvcfg	uint64_t	
sepc	uint64_t	
siselect	uint64_t	
skipIdleTags	uint64_t	
sscratch	uint64_t	
startSteps	int64_t	Start steps of the time source
state	int32_t	

The use and/or disclosure,etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.

Name	Type	Description
steps	int64_t	Steps of the time source
stickyFlags	uint32_t	Set bit 0 to 1 to not exit CPU on halted mode.
stval	uint64_t	
stvec	uint64_t	
targetExec	temu_IfaceRef/ <unknown>	Target execution interface
targetSteps	int64_t	Target steps of the time source
virtualization	uint8_t	RISC-V Mode: 0=U, 1=S, 3=M
vsatp	uint64_t	
vscause	uint64_t	
vsepc	uint64_t	
vsie	uint64_t	
vsip	uint64_t	
vsscratch	uint64_t	
vsstatus	uint64_t	
vstval	uint64_t	
vstvec	uint64_t	

Interfaces

Name	Type	Description
AhbIface	AhbIface	
AssemblerIface	temu::AssemblerIface	Assembler interface
BinaryTranslationIface	temu::BinaryTranslationControlIface	
ClockIface	ClockIface	
CodePatternIface	temu::CodePatternIface	
CpuIface	temu::CpuIface	
ExecIface	temu::TargetExecutionIface	
InvalidMemAccessIface	MemAccessIface	
IrqIface	IrqCtrlIface	
PhysicalMemAccessIface	MemAccessIface	
PowerIface	PowerIface	

The use and/or disclosure,etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.

Name	Type	Description
RegisterIface	temu::RegisterIface	Auto-generated Register Interface Impl.
ResetIface	ResetIface	
VirtualMemAccessIface	MemAccessIface	

Ports

Prop	Iface	Description
irqClient	IrqIface	interrupt controller interface

Commands

Name	Description
assemble	Assemble instruction
delete	Dispose instance of NOELV
disable-dbt	Disable binary translation.
disable-dbt-logging	Disable logging when translating code.
disable-dbt-validation	Disable validation of emitted machine code.
disable-hard-code-reset	Disable hard resets of DBT emitter cache.
disable-stats	Disable statistics
disable-translation-logging	Disable jit-logging.
disassemble	Disassemble code
disassemble-block	Disassemble binary translated code block.
enable-dbt	Enable binary translation.
enable-dbt-logging	Enable logging when translating code.
enable-dbt-validation	Enable validation of emitted machine code.
enable-hard-code-reset	Enable hard resets of DBT emitter cache.
enable-stats	Enable statistics
enable-translation-logging	Enable jit-logging.
enableHpm	Enable performance monitor
enableTraps	Enable traps.
pregs	Print registers for CPU
pstat	Print CPU stats
setPC	Set PC
setReg	Set register

The use and/or disclosure,etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.

Name	Description
translate-block	Translate block.
translate-func	Translate function.

Command assemble Arguments

Name	Type	Required	Description
inst	string	yes	Instruction to assemble.
pa	int	no	Physical address
va	int	no	Virtual address

Command disable-stats Arguments

Name	Type	Required	Description
stat	string	yes	Name of statistics (executed-translated-instructions, executed-translated-blocks)

Command disassemble Arguments

Name	Type	Required	Description
count	int	no	Number of instructions
pa	int	no	Physical address
va	int	no	Virtual address

Command disassemble-block Arguments

Name	Type	Required	Description
pa	int	no	Physical address of block
va	int	no	Virtual address of block

Command enable-stats Arguments

Name	Type	Required	Description
stat	string	yes	Name of statistics (executed-translated-instructions, executed-translated-blocks)

The use and/or disclosure,etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.

Command enableHpm Arguments

Name	Type	Required	Description
hpm-id	int	no	HPM index

Command setPC Arguments

Name	Type	Required	Description
pc	int	yes	New %pc

Command setReg Arguments

Name	Type	Required	Description
reg	string	yes	Register name
value	int	yes	Value

Command translate-block Arguments

Name	Type	Required	Description
count	int	no	Number of instructions, omit for automatic.
pa	int	no	Physical address of block
va	int	no	Virtual address of block

Command translate-func Arguments

Name	Type	Required	Description
pa	int	no	Physical address of block
va	int	no	Virtual address of block

Limitations

- The NOEL-V model is not yet validated against hardware.

The use and/or disclosure,etc. of the contents of this document (or any part thereof) is subject to the restrictions referenced on the front page.