TEMU
Target Reference

Version 3.0-pre, 2022-04-12

Table of Contents

Chapter 1. Overview

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

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.

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!

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. Limitations

Current limitations of the SPARCv8 target include:

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

2.5. Variants

2.5.1. ERC32

@Erc32 Reference

Properties

Name	Type	Description
Class	*void	Class object
Component	*void	Pointer to component object if part of component
LoggingFlags	uint64_t	Flags for logging info
Name	*char	Object name
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
LoggingFlags	uint64_t	Flags for logging info
Name	*char	Object name
TimeSource	*void	Time source object
asr	[uint32_t; 32]	
cwp	*void	

Name	Type	Description
cycles	int64_t	
devices	temu_IfaceRefArray	
extraRegs	[uint32_t; 32]	
fprs	[uint32_t; 32]	
freq	uint64_t	
fsr	uint32_t	
g	[uint32_t; 8]	
gprs	[uint32_t; 128]	
i_npc	uintptr_t	Intermediate code nPC
i_pc	uintptr_t	Intermediate code PC
irq	int8_t	
irqClient	temu_IfaceRef/ <unknown>	Interrupt controller (for ACKs)
machine	temu_IfaceRef/ <unknown>	Machine interface
memAccess	temu_IfaceRef/ <unknown>	Level 1 memory access interface (MMU)
memAccessL2	temu_IfaceRef/ <unknown>	Level 2 memory access interface (physical)
memSpace	*void	.
memory	temu_IfaceRef/ <unknown>	Memory interface
mmuCtrl	uint32_t	
mmuCtxt	uint32_t	
mmuCtxtPtr	uint32_t	
mmuFaultAddr	uint32_t	
mmuFaultStat	uint32_t	
nextEvent	int64_t	
npc	uint32_t	Next program counter register (%npc)
pc	uint32_t	Program counter register (%pc)
pdcManager	temu_IfaceRef/ <unknown>	Pre-decode cache manager (normally memory space)
powerState	uint32_t	
psr	uint32_t	
state	int32_t	

Name	Type	Description
steps	int64_t	
stickyFlags	uint32_t	Set bit 0 to 1 to not exit CPU on halted mode.
targetExec	temu_IfaceRef/ <unknown>	Target execution interface
tbr	uint32_t	
wim	uint32_t	

Interfaces

Name	Type	Description
ClockIface	ClockIface	
CpuIface	CpuIface	
EventIface	EventIface	
InvalidMemAccessIface	MemAccessIface	
IrqIface	IrqCtrlIface	
MemoryIface	MemoryIface	
ObjectIface	ObjectIface	
PowerIface	PowerIface	
ResetIface	ResetIface	
SparcIface	SparcIface	

Ports

Prop	Iface	Description
irqClient	IrqIface	interrupt controller interface

Commands

Name	Description
assemble	Assemble instruction
delete	Dispose instance of Erc32
disableProfiling	Disable profiling mode
disableTraps	Disable traps.
disassemble	Disassemble code
enableProfiling	Enable profiling mode
enableTraps	Enable traps.

Name	Description
flushProfile	Flush profile data
pregs	Print registers for CPU
pstat	Print CPU stats
pwin	Print register window.
resetStats	Reset statistics counters
setPC	Set PC (and nPC)
setReg	Set register

Command assemble Arguments

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

Command disassemble Arguments

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

Command pwin Arguments

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

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

2.5.2. LEON2

@Leon2 Reference

Properties

Name	Type	Description
Class	*void	Class object
Component	*void	Pointer to component object if part of component
LoggingFlags	uint64_t	Flags for logging info
Name	*char	Object name
TimeSource	*void	Time source object

Commands

Name	Description
delete	Dispose instance of @Leon2
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
LoggingFlags	uint64_t	Flags for logging info
Name	*char	Object name
TimeSource	*void	Time source object
asr	[uint32_t; 32]	
cwp	*void	
cycles	int64_t	
dCache	temu_IfaceRef/ <unknown>	L1 data cache (optional)
devices	temu_IfaceRefArray	
extraRegs	[uint32_t; 32]	
fprs	[uint32_t; 32]	
freq	uint64_t	
fsr	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
irq	int8_t	
irqClient	temu_IfaceRef/ <unknown>	Interrupt controller (for ACKs)
machine	temu_IfaceRef/ <unknown>	Machine interface
memAccess	temu_IfaceRef/ <unknown>	Level 1 memory access interface (MMU)
memAccessL2	temu_IfaceRef/ <unknown>	Level 2 memory access interface (physical)
memSpace	*void	.
memory	temu_IfaceRef/ <unknown>	Memory interface
mmuCtrl	uint32_t	
mmuCtxt	uint32_t	

Name	Type	Description
mmuCtxtPtr	uint32_t	
mmuFaultAddr	uint32_t	
mmuFaultStat	uint32_t	
nextEvent	int64_t	
npc	uint32_t	Next program counter register (%npc)
pc	uint32_t	Program counter register (%pc)
pdcManager	temu_IfaceRef/ <unknown>	Pre-decode cache manager (normally memory space)
powerState	uint32_t	
psr	uint32_t	
state	int32_t	
steps	int64_t	
stickyFlags	uint32_t	Set bit 0 to 1 to not exit CPU on halted mode.
targetExec	temu_IfaceRef/ <unknown>	Target execution interface
tbr	uint32_t	
wim	uint32_t	

Interfaces

Name	Type	Description
ClockIface	ClockIface	
CpuIface	CpuIface	
EventIface	EventIface	
InvalidMemAccessIface	MemAccessIface	
IrqIface	IrqCtrlIface	
MemoryIface	MemoryIface	
ObjectIface	ObjectIface	
PowerIface	PowerIface	
ResetIface	ResetIface	
SparcIface	SparcIface	

Ports

Prop	Iface	Description
irqClient	IrqIface	interrupt controller interface

Commands

Name	Description
assemble	Assemble instruction
delete	Dispose instance of Leon2
disableProfiling	Disable profiling mode
disableTraps	Disable traps.
disassemble	Disassemble code
enableProfiling	Enable profiling mode
enableTraps	Enable traps.
flushProfile	Flush profile data
pregs	Print registers for CPU
pstat	Print CPU stats
pwin	Print register window.
resetStats	Reset statistics counters
setPC	Set PC (and nPC)
setReg	Set register

Command assemble Arguments

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

Command disassemble Arguments

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

Command pwin Arguments

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

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

2.5.3. LEON3

@Leon3 Reference

Properties

Name	Type	Description
Class	*void	Class object
Component	*void	Pointer to component object if part of component
LoggingFlags	uint64_t	Flags for logging info
Name	*char	Object name
TimeSource	*void	Time source object

Commands

Name	Description
delete	Dispose instance of @Leon3
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
LoggingFlags	uint64_t	Flags for logging info
Name	*char	Object name
TimeSource	*void	Time source object
asr	[uint32_t; 32]	
cwp	*void	
cycles	int64_t	
dCache	temu_IfaceRef/ <unknown>	L1 data cache (optional)
devices	temu_IfaceRefArray	
extraRegs	[uint32_t; 32]	
fprs	[uint32_t; 32]	
freq	uint64_t	
fsr	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
irq	int8_t	
irqClient	temu_IfaceRef/ <unknown>	Interrupt controller (for ACKs)

Name	Type	Description
machine	temu_IfaceRef/ <unknown>	Machine interface
memAccess	temu_IfaceRef/ <unknown>	Level 1 memory access interface (MMU)
memAccessL2	temu_IfaceRef/ <unknown>	Level 2 memory access interface (physical)
memSpace	*void	.
memory	temu_IfaceRef/ <unknown>	Memory interface
mmuCtrl	uint32_t	
mmuCtxt	uint32_t	
mmuCtxtPtr	uint32_t	
mmuFaultAddr	uint32_t	
mmuFaultStat	uint32_t	
nextEvent	int64_t	
npc	uint32_t	Next program counter register (%npc)
pc	uint32_t	Program counter register (%pc)
pdcManager	temu_IfaceRef/ <unknown>	Pre-decode cache manager (normally memory space)
powerState	uint32_t	
psr	uint32_t	
state	int32_t	
steps	int64_t	
stickyFlags	uint32_t	Set bit 0 to 1 to not exit CPU on halted mode.
targetExec	temu_IfaceRef/ <unknown>	Target execution interface
tbr	uint32_t	
wim	uint32_t	

Interfaces

Name	Type	Description
AhbIface	AhbIface	
ClockIface	ClockIface	
CpuIface	CpuIface	
DCacheCtrlIface	CacheCtrlIface	

Name	Type	Description
EventIface	EventIface	
ICacheCtrlIface	CacheCtrlIface	
InvalidMemAccessIface	MemAccessIface	
IrqIface	IrqCtrlIface	
MemoryIface	MemoryIface	
MmuMemAccessIface	MemAccessIface	
ObjectIface	ObjectIface	
PowerIface	PowerIface	
ResetIface	ResetIface	
SparcIface	SparcIface	

Ports

Prop	Iface	Description
irqClient	IrqIface	interrupt controller interface

Commands

Name	Description
assemble	Assemble instruction
delete	Dispose instance of Leon3
disableProfiling	Disable profiling mode
disableTraps	Disable traps.
disassemble	Disassemble code
enableProfiling	Enable profiling mode
enableTraps	Enable traps.
flushProfile	Flush profile data
pregs	Print registers for CPU
pstat	Print CPU stats
pwin	Print register window.
resetStats	Reset statistics counters
setPC	Set PC (and nPC)
setReg	Set register

Command assemble Arguments

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

Command disassemble Arguments

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

Command pwin Arguments

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

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

2.5.4. LEON4

@Leon4 Reference

Properties

Name	Type	Description
Class	*void	Class object

Name	Type	Description
Component	*void	Pointer to component object if part of component
LoggingFlags	uint64_t	Flags for logging info
Name	*char	Object name
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
LoggingFlags	uint64_t	Flags for logging info
Name	*char	Object name
TimeSource	*void	Time source object
asr	[uint32_t; 32]	
cwp	*void	

Name	Type	Description
cycles	int64_t	
dCache	temu_IfaceRef/ <unknown>	L1 data cache (optional)
devices	temu_IfaceRefArray	
extraRegs	[uint32_t; 32]	
fprs	[uint32_t; 32]	
freq	uint64_t	
fsr	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
irq	int8_t	
irqClient	temu_IfaceRef/ <unknown>	Interrupt controller (for ACKs)
machine	temu_IfaceRef/ <unknown>	Machine interface
memAccess	temu_IfaceRef/ <unknown>	Level 1 memory access interface (MMU)
memAccessL2	temu_IfaceRef/ <unknown>	Level 2 memory access interface (physical)
memSpace	*void	.
memory	temu_IfaceRef/ <unknown>	Memory interface
mmuCtrl	uint32_t	
mmuCtxt	uint32_t	
mmuCtxtPtr	uint32_t	
mmuFaultAddr	uint32_t	
mmuFaultStat	uint32_t	
nextEvent	int64_t	
npc	uint32_t	Next program counter register (%npc)
pc	uint32_t	Program counter register (%pc)
pdcManager	temu_IfaceRef/ <unknown>	Pre-decode cache manager (normally memory space)
powerState	uint32_t	

Name	Type	Description
psr	uint32_t	
resetNpc	uint32_t	Reset nPC (for LEON4)
resetPc	uint32_t	Reset PC (for LEON4)
state	int32_t	
steps	int64_t	
stickyFlags	uint32_t	Set bit 0 to 1 to not exit CPU on halted mode.
targetExec	temu_IfaceRef/ <unknown>	Target execution interface
tbr	uint32_t	
wim	uint32_t	

Interfaces

Name	Type	Description
AhbIface	AhbIface	
ClockIface	ClockIface	
CpuIface	CpuIface	
DCacheCtrlIface	CacheCtrlIface	
DynamicResetAddressIface	temu::DynamicResetAddressIface	
EventIface	EventIface	
ICacheCtrlIface	CacheCtrlIface	
InvalidMemAccessIface	MemAccessIface	
IrqIface	IrqCtrlIface	
MemoryIface	MemoryIface	
MmuMemAccessIface	MemAccessIface	
ObjectIface	ObjectIface	
PowerIface	PowerIface	
ResetIface	ResetIface	
SparcIface	SparcIface	

Ports

Prop	Iface	Description
irqClient	IrqIface	interrupt controller interface

Commands

Name	Description
assemble	Assemble instruction
delete	Dispose instance of Leon4
disableProfiling	Disable profiling mode
disableTraps	Disable traps.
disassemble	Disassemble code
enableProfiling	Enable profiling mode
enableTraps	Enable traps.
flushProfile	Flush profile data
pregs	Print registers for CPU
pstat	Print CPU stats
pwin	Print register window.
resetStats	Reset statistics counters
setPC	Set PC (and nPC)
setReg	Set register

Command assemble Arguments

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

Command disassemble Arguments

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

Command pwin Arguments

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

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

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
LoggingFlags	uint64_t	Flags for logging info
Name	*char	Object name
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

e500v2 Reference

Properties

Name	Type	Description
CPUId	uint32_t	
Class	*void	Class object

Name	Type	Description
Component	*void	Pointer to component object if part of component
LoggingFlags	uint64_t	Flags for logging info
Name	*char	Object name
TimeSource	*void	Time source object
acc	uint64_t	Accumulator register
atbl	uint32_t	
atbu	uint32_t	
bbear	uint32_t	
bbtar	uint32_t	
bucsr	uint32_t	
cr	uint32_t	
csrrs	[uint32_t; 2]	
ctr	uint32_t	
cycles	int64_t	
dac1	uint32_t	
dac2	uint32_t	
dbcrs	[uint32_t; 3]	
dbsr	uint32_t	
dear	uint32_t	
dec	uint32_t	
decar	uint32_t	
devices	temu_IfaceRefArray	
dvc1	uint32_t	
dvc2	uint32_t	
esr	uint32_t	
freq	uint64_t	
gprs	[uint64_t; 32]	
hids	[uint32_t; 2]	
iac1	uint32_t	
iac2	uint32_t	
irq	int8_t	

Name	Type	Description
irqClient	temu_IfaceRef/ <unknown>	
ivors	[uint32_t; 64]	
ivpr	uint32_t	
l1cfg0	uint32_t	
l1cfg1	uint32_t	
lr	uint32_t	
machine	temu_IfaceRef/ <unknown>	
mass	[uint32_t; 7]	
mcar	uint32_t	
mcsr	uint32_t	
mcsrrs	[uint32_t; 2]	
memAccess	temu_IfaceRef/ <unknown>	
memAccessL2	temu_IfaceRef/ <unknown>	
memory	temu_IfaceRef/ <unknown>	
mmucfg	uint32_t	
mmucsr0	uint32_t	
msr	uint32_t	
nextEvent	int64_t	
pc	uint32_t	
pid0	uint32_t	
pid1	uint32_t	
pid2	uint32_t	
pir	uint32_t	
pmc0	uint32_t	
pmc1	uint32_t	
pmc2	uint32_t	
pmc3	uint32_t	
pmgc0	uint32_t	
pmlca0	uint32_t	
pmlca1	uint32_t	
pmlca2	uint32_t	
pmlca3	uint32_t	

Name	Type	Description
pmlcb0	uint32_t	
pmlcb1	uint32_t	
pmlcb2	uint32_t	
pmlcb3	uint32_t	
powerState	uint32_t	
pvr	uint32_t	
spefscr	uint32_t	
sprgs	[uint32_t; 8]	
srrs	[uint32_t; 2]	
state	int32_t	
steps	int64_t	
svr	uint32_t	
tcr	uint32_t	
tlb0cfg	uint32_t	
tlb1cfg	uint32_t	
tsr	uint32_t	
upmc0	uint32_t	
upmc1	uint32_t	
upmc2	uint32_t	
upmc3	uint32_t	
upmgc0	uint32_t	
upmlca0	uint32_t	
upmlca1	uint32_t	
upmlca2	uint32_t	
upmlca3	uint32_t	
upmlcb0	uint32_t	
upmlcb1	uint32_t	
upmlcb2	uint32_t	
upmlcb3	uint32_t	
usrpg0	uint32_t	
xer	uint32_t	

Interfaces

Name	Type	Description
ClockIface	ClockIface	
CpuIface	CpuIface	
EventIface	EventIface	
InvalidMemAccessIface	MemAccessIface	
IrqIface	IrqCtrlIface	
MemoryIface	MemoryIface	
MmuMemAccessIface	MemAccessIface	MMU access interface
ObjectIface	ObjectIface	
PowerIface	PowerIface	
PowerPCIface	PowerPCIface	
ResetIface	ResetIface	

Ports

Prop	Iface	Description
irqClient	IrqIface	interrupt controller interface

Commands

Name	Description
delete	Dispose instance of e500v2
printTLB0	Print TLB0
printTLB1	Print TLB1
raiseCritical	Raise critical interrupt
raiseExternal	Raise external interrupt
setPC	Set PC
setTLB0Entry	Add entry to TLB0
setTLB1Entry	Add entry to TLB1

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

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.
- 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`
  - `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
LoggingFlags	uint64_t	Flags for logging info
Name	*char	Object name
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

Name	Type	Description
CPUId	uint32_t	
Class	*void	Class object
Component	*void	Pointer to component object if part of component
LoggingFlags	uint64_t	Flags for logging info
Name	*char	Object name
TimeSource	*void	Time source object
cr	uint32_t	
ctr	uint32_t	
cycles	int64_t	
dabr	uint32_t	
dar	uint32_t	
dbats	[uint32_t; 16]	

Name	Type	Description
dec	uint32_t	
devices	temu_IfaceRefArray	
dmal	uint32_t	
dmau	uint32_t	
dsisr	uint32_t	
ear	uint32_t	
fprs	[uint64_t; 32]	
freq	uint64_t	
gprs	[uint32_t; 32]	
gqrs	[uint32_t; 8]	
hids	[uint32_t; 4]	
iabr	uint32_t	
ibats	[uint32_t; 16]	
ictc	uint32_t	
irq	int8_t	
irqClient	temu_IfaceRef/ <unknown>	
l2cr	uint32_t	
lr	uint32_t	
machine	temu_IfaceRef/ <unknown>	
memAccess	temu_IfaceRef/ <unknown>	
memAccessL2	temu_IfaceRef/ <unknown>	
memory	temu_IfaceRef/ <unknown>	
mmcrs	[uint32_t; 2]	
msr	uint32_t	
nextEvent	int64_t	
pc	uint32_t	
pmcs	[uint32_t; 4]	
powerState	uint32_t	
pvr	uint32_t	
sdr1	uint32_t	
sia	uint32_t	
sprgs	[uint32_t; 4]	

Name	Type	Description
srrs	[uint32_t; 2]	
state	int32_t	
steps	int64_t	
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
ClockIface	ClockIface	
CpuIface	CpuIface	
EventIface	EventIface	
InvalidMemAccessIface	MemAccessIface	
IrqIface	IrqCtrlIface	
MemoryIface	MemoryIface	
MmuMemAccessIface	MemAccessIface	
ObjectIface	ObjectIface	
PowerIface	PowerIface	
PowerPCIface	PowerPCIface	
ResetIface	ResetIface	

Ports

Prop	Iface	Description
irqClient	IrqIface	interrupt controller interface

Commands

Name	Description
delete	Dispose instance of ppc750

Name	Description
setPC	Set PC

Command setPC Arguments

Name	Type	Required	Description
pc	int	yes	New pc

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.