TEMU

Target Reference

Version 2.3, 2022-05-02

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 and LEON3.

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 for the NGMP based processor (the NGMP has 0c0000000 and 0xc0000004 as reset values for these 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() fun 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' in order to indicate that the LEON4 core should use the NGMP reset values.

2.3.2. Properties

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

Interface References

memaccess

The interface reference to an object 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 or 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 invoked first in the memaccess interface reference. In case the system has an MMU, set this to memspace:MemAccessIface.

memory

The interface reference to an object handling memory block read and writes, this should normally refer to a memory space object.

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 documentation about the costs on the MEIKO FPU (which is the standard FPU for the LEON2) 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

Attributes

Properties

Name	Type	Description
CPUId	uint32_t	
asr	[32 x uint32_t]	
cwp	internalptr	
cycles	int64_t	
devices	irefarray / <unknown>	
extraRegs	[32 x uint32_t]	
fprs	[32 x uint32_t]	
freq	uint64_t	
fsr	uint32_t	
g	[8 x uint32_t]	
gprs	[128 x uint32_t]	
irq	int8_t	
irqClient	iref / <unknown>	
machine	iref / <unknown>	
memAccess	iref / <unknown>	
memAccessL2	iref / <unknown>	
memory	iref / <unknown>	
mmuCtrl	uint32_t	
mmuCtxt	uint32_t	
mmuCtxtPtr	uint32_t	
mmuFaultAddr	uint32_t	
mmuFaultStat	uint32_t	
nextEvent	int64_t	

Name	Type	Description
npc	uint32_t	
object.timeSource	object	Time source object (a cpu or machine object)
pc	uint32_t	
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.
tbr	uint32_t	
wim	uint32_t	

Interfaces

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

Ports

Prop	Iface	Description
irqClient	IrqIface	interrupt controller interface

2.5.2. LEON2

Attributes

Properties

Name	Type	Description
CPUId	uint32_t	
asr	[32 x uint32_t]	
cwp	internalptr	
cycles	int64_t	
dCache	iref / <unknown>	l1 data cache model
devices	irefarray / <unknown>	
extraRegs	[32 x uint32_t]	
fprs	[32 x uint32_t]	
freq	uint64_t	
fsr	uint32_t	
g	[8 x uint32_t]	
gprs	[128 x uint32_t]	
iCache	iref / <unknown>	l1 instr cache model
irq	int8_t	
irqClient	iref / <unknown>	
machine	iref / <unknown>	
memAccess	iref / <unknown>	
memAccessL2	iref / <unknown>	
memory	iref / <unknown>	
mmuCtrl	uint32_t	
mmuCtxt	uint32_t	
mmuCtxtPtr	uint32_t	
mmuFaultAddr	uint32_t	
mmuFaultStat	uint32_t	
nextEvent	int64_t	
npc	uint32_t	
object.timeSource	object	Time source object (a cpu or machine object)
pc	uint32_t	
powerState	uint32_t	
psr	uint32_t	

Name	Type	Description
state	int32_t	
steps	int64_t	
stickyFlags	uint32_t	Set bit 0 to 1 to not exit CPU on halted mode.
tbr	uint32_t	
wim	uint32_t	

Interfaces

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

Ports

Prop	Iface	Description
irqClient	IrqIface	interrupt controller interface

2.5.3. LEON3

Attributes

Properties

Name	Type	Description
CPUId	uint32_t	
asr	[32 x uint32_t]	
cwp	internalptr	

Name	Type	Description
cycles	int64_t	
dCache	iref / <unknown>	l1 data cache model
devices	irefarray / <unknown>	
extraRegs	[32 x uint32_t]	
fprs	[32 x uint32_t]	
freq	uint64_t	
fsr	uint32_t	
g	[8 x uint32_t]	
gprs	[128 x uint32_t]	
iCache	iref / <unknown>	l1 instr cache model
irq	int8_t	
irqClient	iref / <unknown>	
machine	iref / <unknown>	
memAccess	iref / <unknown>	
memAccessL2	iref / <unknown>	
memory	iref / <unknown>	
mmuCtrl	uint32_t	
mmuCtxt	uint32_t	
mmuCtxtPtr	uint32_t	
mmuFaultAddr	uint32_t	
mmuFaultStat	uint32_t	
nextEvent	int64_t	
npc	uint32_t	
object.timeSource	object	Time source object (a cpu or machine object)
pc	uint32_t	
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.

Name	Type	Description
tbr	uint32_t	
wim	uint32_t	

Interfaces

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

Ports

Prop	Iface	Description
irqClient	IrqIface	interrupt controller interface

2.5.4. LEON4

Attributes

Properties

Name	Type	Description
CPUId	uint32_t	
asr	[32 x uint32_t]	
cwp	internalptr	

Name	Type	Description
cycles	int64_t	
dCache	iref / <unknown>	l1 data cache model
devices	irefarray / <unknown>	
extraRegs	[32 x uint32_t]	
fprs	[32 x uint32_t]	
freq	uint64_t	
fsr	uint32_t	
g	[8 x uint32_t]	
gprs	[128 x uint32_t]	
iCache	iref / <unknown>	l1 instr cache model
irq	int8_t	
irqClient	iref / <unknown>	
machine	iref / <unknown>	
memAccess	iref / <unknown>	
memAccessL2	iref / <unknown>	
memory	iref / <unknown>	
mmuCtrl	uint32_t	
mmuCtxt	uint32_t	
mmuCtxtPtr	uint32_t	
mmuFaultAddr	uint32_t	
mmuFaultStat	uint32_t	
nextEvent	int64_t	
npc	uint32_t	
object.timeSource	object	Time source object (a cpu or machine object)
pc	uint32_t	
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.

Name	Type	Description
tbr	uint32_t	
wim	uint32_t	

Interfaces

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

Ports

Prop	Iface	Description
irqClient	IrqIface	interrupt controller interface

Chapter 3. ARM

The ARMv7 target in TEMU comes with support for the ARMv7-R sub target at the moment. A number of on-chip devices based on existing ARM based CPUs are available. Currently this includes a subset of the TMS570 devices.

3.1. Variants

3.1.1. ARMv7-R

The ARMv7-R core is available, it comes with an PMSA compliant MPU modelled on the Cortex-R5 MPU. Note that this differs slightly from the ARMv7-R architecture description. Especially, it lacks dedicated executable protection and uses read protection to handle fetch permission.

While the model is still in beta state, the model has been successfully used to run actual time and space partitioned hypervisors with partitions using different kernels (e.g. XtratuM / RTEMS / XAL is known to run).

3.2. Operating System Compatibility

The ARMv7 models have been executed successfully with:

- RTEMS
- XtratuM
- XAL

3.3. Configuration

3.3.1. Interface References

memAccess

The interface reference to an object 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 or 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.

memory

The interface reference to an object handling memory block read and writes, this should normally refer to a memory space object.

irqClient

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.

coproc

The interface reference array contains references to coprocessor interfaces. The coprocessor interface is currently unstable and it is adviced that it at this moment not implemented by third parties.

3.3.2. Other Properties

freq

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

cpuid

CPU id for multiprocessor configurations, defaults to 0.

3.3.3. Interfaces

The SPARCv8 emulator cores implement the following interfaces:

CpuIface

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

ArmIface

Standard ARMv7 interface. Contains among other things functions for accessing banked registers.

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.

3.4. Limitations

Current known limitations of the ARMv7 target include:

- Performance is limited due to decoding logic needed for ARMv7 and Thumb2 ISAs. Current performance is around 30 % of the SPARCv8 model. This will be addressed in the future.
- No static timing model is defined at this moment. That means that one instruction take one

cycle to finish.

- The built-in assemblers and disassemblers are not working at this moment.
- Cache control interfaces are not implemented or supported, this can be addressed if needed.
- NEON (vector) instructions are not implemented at this moment.
- A co-processor interface exists, but it is currently not stable.

3.5. Variants

3.5.1. ARMv7-R

Attributes

Properties

Name	Type	Description
CPUId	uint32_t	
abtregs	[2 x uint32_t]	
coproc	[16 x iref / ARMCoProcessor]	
cpsr	uint32_t	CPSR register
cycles	int64_t	
fiqregs	[7 x uint32_t]	
freq	uint64_t	
gprs	[16 x uint32_t]	
irqClient	iref / <unknown>	
irqregs	[2 x uint32_t]	
machine	iref / <unknown>	
memAccess	iref / MemAccessIface	
memory	iref / MemoryIface	
monregs	[2 x uint32_t]	
nextEvent	int64_t	
object.timeSource	object	Time source object (a cpu or machine object)
powerState	uint32_t	
sctlr	uint32_t	
sctlr_rst	uint32_t	
spsr	[7 x uint32_t]	

Name	Type	Description
state	int32_t	
steps	int64_t	
svcregs	[2 x uint32_t]	
undregs	[2 x uint32_t]	

Interfaces

Name	Type	Description
ArmIface	ARMCpu	
ClockIface	ClockIface	
CpuIface	CpuIface	
EventIface	EventIface	
InvalidMemAccessIface	MemAccessIface	
IrqIface	IrqCtrlIface	
MemoryIface	MemoryIface	
PowerIface	PowerIface	
ResetIface	ResetIface	

Ports

Prop	Iface	Description
irqClient	IrqIface	interrupt controller interface

Registers

Register support is currently experimental!

Register Bank registers

Register cpsr

CPSR register

Cold reset value: 0x13

Warm reset value: 0x13

Field	Mask	Cold	Warm	Description
	0x80000000	0x0	0x0	Negative condition flag

Field	Mask	Cold	Warm	Description
	0x40000000	0x0	0x0	Zero condition flag
	0x20000000	0x0	0x0	Carry condition flag
	0x10000000	0x0	0x0	Overflow condition flag
	0x8000000	0x0	0x0	Cumulative saturation bit
it_1_0	0x6000000	0x0	0x0	If-Then execution state
	0x3000000	0x0	0x0	Jazelle bit
ge	0xf0000	0x0	0x0	Greater than or equal flags
it_7_2	0xfc00	0x0	0x0	If-Then execution state
	0x200	0x1	0x1	Endianess
	0x100	0x1	0x1	Asynchronous abort mask
	0x80	0x1	0x1	IRQ mask
	0x40	0x1	0x1	FIQ mask
	0x20	0x0	0x0	Thumb bit
	`0x1f`	0x13	0x13	Mode

Chapter 4. PowerPC

The PowerPC target in TEMU comes with support for the 32 bit PowerPC architecture. It currently implements the PPC750 CPU model. The target is in beta-state at time of writing.

4.1. Variants

4.1.1. PowerPC 750

The PPC 750 is a 32 bit PowerPC without AltiVec support.

4.2. Configuration

4.2.1. Properties

Interface References

memaccess

The interface reference to an object 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 or 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.

memory

The interface reference to an object handling memory block read and writes, this should normally refer to a memory space object.

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.

Other Properties

freq

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

cpuid

CPU id for multiprocessor configurations, defaults to 0.

4.2.2. Interfaces

The SPARCv8 emulator cores implement the following interfaces:

CpuIface

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

PowerPCIface

Standard PowerPC interface. Exposes convenient functions for accessing SPRs.

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.

4.3. Limitations

Current known limitations of the PowerPC target include:

- No static timing model is defined at this moment. That means that one instruction take one cycle to finish.
- The built-in assemblers and disassemblers are not working at this have problems with split field instructions (e.g. SPR IDs).
- Cache control interfaces are not implemented or supported, this can be addressed if needed.
- 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.

4.4. Variants

4.4.1. PPC750

Attributes

Properties

Name	Type	Description
CPUId	uint32_t	
cycles	int64_t	
devices	irefarray / <unknown>	

Name	Type	Description
fprs	[32 x uint64_t]	
freq	uint64_t	
gprs	[32 x uint32_t]	
irqClient	iref / IrqClientIface	
machine	iref / MachineIface	
memAccess	iref / MemAccessIface	
memAccessL2	iref / MemAccessIface	
memory	iref / MemoryIface	
nextEvent	int64_t	
object.timeSource	object	Time source object (a cpu or machine object)
pc	uint32_t	
powerState	uint32_t	
state	int32_t	
steps	int64_t	

Interfaces

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

Ports

Prop	Iface	Description
irqClient	IrqIface	interrupt controller interface