TERMA

TEMU
MEC Device Model Manual

Mattias Holm

Version 1.1, 2016-05-12

Table of Contents

Table 1. Record of Changes

Rev	Date	Author	Note
1.2	2018-08-29	MH	Add note about IRQ behaviour.
1.1	2016-05-12	MH	Auto gen tables.
1.0	2015-03-01	MH	Initial version.

1. Introduction

The MEC (Memory Controller) device is used with the ERC32 processor. The device provides two UARTs, two timers and an interrupt interface. The interrupt interface allows for the raising and lowering of the 5 external interrupts provided by the ERC32 (IRQ 0 through (including) 4). The device model takes care of converting these to the relevant internal interrupts (i.e. SPARC IRQs 2,3,10,11 and 14). When raising (or lowering) a MEC interrupt you need to use numbers 0-4.

2. Configuration

2.1. Interrupt Delivery

The property irqControl should be connected to the device which the MEC raises interrupts on, this is normally a CPU object. The connection should be made to the CPU-object's interface of type IrqIface. Note that the CPU must support interrupts 1 through 15, this is in general case correct for SPARC based processors, but other CPUs may not be compatible.

2.2. UART Connections

Two serial interfaces exist, the UartAIface and the UartBIface, these can be connected to in order to receive data from remote serial port terminals (i.e. this is the RX direction). The uarta and uartb properties can be used to connect the TX direction of the UARTs.

2.3. Infinite UART Speed

Set config.infiniteUartSpeed to nonzero to enable infinite speed on the Tx channels. With infinite speed, a written byte is immediately forwarded to the destination device, with limited UART speed (the variable being zero) the timing due to UART scaler bits (upper 8 bits of the MecCtrlReg) will be simulated, leading to realistic byte rates over the serial port device. Note that individual bits are not transmitted only the bytes.

3. Attributes

3.1. Properties

Name	Type	Description
accessProtSegment1Base	uint32_t	
accessProtSegment1End	uint32_t	
accessProtSegment2Base	uint32_t	
accessProtSegment2End	uint32_t	
config.infiniteUartSpeed	uint32_t	
cpu	iref / <unknown>	
errorAndResetStatus	uint32_t	
failingAddr	uint32_t	
gpiConfig	uint32_t	
gpiData	uint32_t	
gptCounter	uint32_t	
gptCounterProgramReg	uint32_t	
gptScaler	uint32_t	
gptScalerProgramReg	uint32_t	
ioConfig	uint32_t	
irqClear	uint32_t	
irqControl	iref / <unknown>	
irqForce	uint32_t	
irqMask	uint32_t	
irqPending	uint32_t	
irqShape	uint32_t	
mecCtrl	uint32_t	
memoryConfig	uint32_t	
object.timeSource	object	Time source object (a cpu or machine object)
outSignals	[8 x iref / SignalIface]	
powerDown	uint32_t	
rtcCounter	uint32_t	
rtcCounterProgramReg	uint32_t	
rtcScaler	uint32_t	
rtcScalerProgramReg	uint32_t	
softwareReset	uint32_t	

Name	Type	Description
systemFaultStatus	uint32_t	
testControl	uint32_t	
timerControl	uint32_t	
uartChanARxTx	uint32_t	
uartChanBRxTx	uint32_t	
uartStatus	uint32_t	
uarta	iref / <unknown>	
uartb	iref / <unknown>	
waitStateConfig	uint32_t	
wdogProgAndTimeoutAck	uint32_t	
wdogTrapDoorSet	uint32_t	

3.2. Interfaces

Name	Type	Description
DeviceIface	DeviceIface	
IrqClientIface	IrqClientIface	
IrqIface	IrqIface	
MemAccessIface	MemAccessIface	
ResetIface	ResetIface	
SignalIface	SignalIface	Incomming signals
UartAIface	SerialIface	
UartBIface	SerialIface	

3.3. Ports

Prop	Iface	Description
irqControl	IrqClientIface	uart a
uarta	UartAIface	uart a
uartb	UartBIface	uart b

4. Notes

The MEC sets the interrupt pending register bit when an interrupt is raised even when the interrupt is masked. The mask is only applied when evaluating whether to raise an IRQ with the

CPU.

5. Limitations

The following deviations from real hardware are known to exist, if you need the correct behaviour (or simulation of it, contact us for more info):

- The UARTs do not support external (watchdog) clocks.
- The UARTs do not support parity, framing errors, break signals or stop bit configuration (although the transmission times are computed based on stop bit count and parity bit embedding).
- Write protection registers have no effect
- Timer values are lazily computed on reads, the content in the case a timer is disabled is estimated on disabling time. This is in principle correct. However, the prescaler counter write has no effect, only the reload value has an effect when written. This may cause an offset of 1024 cycles when re-enabling a timer.