ETMv4 sysfs linux driver programming reference.

Author: Mike Leach <mike.leach@linaro.org>
Date: October 11th, 2019

Supplement to existing ETMv4 driver documentation.

Sysfs files and directories

Root: `/sys/bus/coresight/devices/etm<N>`

The following paragraphs explain the association between sysfs files and the ETMv4 registers that they effect. Note the register names are given without the â€˜TRCâ€™ prefix.

File: `mode` (rw)

Trace Registers: {CONFIGR + others}

Notes: Bit select trace features. See â€˜modeâ€™ section below. Bits in this will cause equivalent programming of trace config and other registers to enable the features requested.

Syntax & eg: `echo bitfield > mode`

bitfield up to 32 bits setting trace features.

Example: `$> echo 0x012 > mode`

File: `reset` (wo)
Trace Registers: All
Notes: Reset all programming to trace nothing / no logic programmed.
Syntax: `echo 1 > reset`

File: `enable_source` (wo)
Trace Registers: PRGCTLR, All hardware regs.
Notes:
- > 0 : Programs up the hardware with the current values held in the driver and enables trace.
- = 0 : disable trace hardware.

Syntax: `echo 1 > enable_source`

File: `cpu` (ro)

Trace Registers: None.

Notes: CPU ID that this ETM is attached to.

Example: `$> cat cpu`

`$> 0`

File: `addr_idx` (rw)

Trace Registers: None.

Notes: Virtual register to index address comparator and range features. Set index for first of the pair in a range.

Syntax: `echo idx > addr_idx`

Where idx < nr_addr_cmp x 2

File: `addr_range` (rw)

Trace Registers: ACVR[idx, idx+1], VIIECTLR

Notes: Pair of addresses for a range selected by addr_idx. Include / exclude according to the optional parameter, or if omitted uses the current â€˜modeâ€™ setting. Select comparator range in control register. Error if index is odd value.

Depends: `mode, addr_idx`

Syntax: `echo addr1 addr2 [exclude] > addr_range`

Where addr1 and addr2 define the range and addr1 < addr2.

Optional exclude value:-

- 0 for include
- 1 for exclude.

Example: `$> echo 0x0000 0x2000 0 > addr_range`

File: `addr_single` (rw)
Trace Registers: ACVR[idx]
Notes: Set a single address comparator according to addr_idx. This is used if the address comparator is used as part of event generation logic etc.
Depends: `addr_idx`
Syntax: `echo addr1 > addr_single`

File: `addr_start` (rw)
Trace Registers: ACVR[idx], VISSCTLR
Notes: Set a trace start address comparator according to addr_idx. Select comparator in control register.
Depends: `addr_idx`
Syntax: `echo addr1 > addr_start`

File: `addr_stop` (rw)
Trace Registers: ACVR[idx], VISSCTLR
Notes: Set a trace stop address comparator according to addr_idx. Select comparator in control register.
Depends: `addr_idx`
Syntax: `echo addr1 > addr_stop`

File: `addr_context` (rw)

Trace Registers: ACATR[idx,{6:4}]

Notes: Link context ID comparator to address comparator addr_idx

Depends: `addr_idx`

Syntax: `echo ctxt_idx > addr_context`

Where ctxt_idx is the index of the linked context id / vmid comparator.

File: `addr_ctxtype` (rw)

Trace Registers: ACATR[idx,{3:2}]

Notes: Input value string. Set type for linked context ID comparator

Depends: `addr_idx`

Syntax: `echo type > addr_ctxtype`

Type one of {all, vmid, ctxid, none}

Example: `$> echo ctxid > addr_ctxtype`

File: `addr_exlevel_s_ns` (rw)

Trace Registers: ACATR[idx,{14:8}]

Notes: Set the ELx secure and non-secure matching bits for the selected address comparator

Depends: `addr_idx`

Syntax: `echo val > addr_exlevel_s_ns`

val is a 7 bit value for exception levels to exclude. Input value shifted to correct bits in register.

Example: `$> echo 0x4F > addr_exlevel_s_ns`

File: `addr_instdatatype` (rw)
Trace Registers: ACATR[idx,{1:0}]
Notes: Set the comparator address type for matching. Driver only supports setting instruction address type.
Depends: `addr_idx`

File: `addr_cmp_view` (ro)

Trace Registers: ACVR[idx, idx+1], ACATR[idx], VIIECTLR

Notes: Read the currently selected address comparator. If part of address range then display both addresses.

Depends: `addr_idx`

Syntax: `cat addr_cmp_view`

Example:

```
$> cat addr_cmp_view
addr_cmp[0] range 0x0 0xffffffffffffffff include ctrl(0x4b00)
```

File: `nr_addr_cmp` (ro)

Trace Registers: From IDR4

Notes: Number of address comparator pairs

File: `sshot_idx` (rw)

Trace Registers: None

Notes: Select single shot register set.

File: `sshot_ctrl` (rw)

Trace Registers: SSCCR[idx]

Notes: Access a single shot comparator control register.

Depends: `sshot_idx`

Syntax: `echo val > sshot_ctrl`

Writes val into the selected control register.

File: `sshot_status` (ro)

Trace Registers: SSCSR[idx]

Notes: Read a single shot comparator status register

Depends: `sshot_idx`

Syntax: `cat sshot_status`

Read status.

Example:

```
$> cat sshot_status
0x1
```

File: `sshot_pe_ctrl` (rw)

Trace Registers: SSPCICR[idx]

Notes: Access a single shot PE comparator input control register.

Depends: `sshot_idx`

Syntax: `echo val > sshot_pe_ctrl`

Writes val into the selected control register.

File: `ns_exlevel_vinst` (rw)

Trace Registers: VICTLR{23:20}

Notes: Program non-secure exception level filters. Set / clear NS exception filter bits. Setting â€˜1â€™ excludes trace from the exception level.

Syntax: `echo bitfield > ns_exlevel_viinst`

Where bitfield contains bits to set clear for EL0 to EL2

Example:

```
%> echo 0x4 > ns_exlevel_viinst
```

Excludes EL2 NS trace.

File: `vinst_pe_cmp_start_stop` (rw)
Trace Registers: VIPCSSCTLR
Notes: Access PE start stop comparator input control registers

---

File: `bb_ctrl` (rw)
Trace Registers: BBCTLR
Notes: Define ranges that Branch Broadcast will operate in. Default (0x0) is all addresses.
Depends: BB enabled.

File: `cyc_threshold` (rw)
Trace Registers: CCCTLR
Notes: Set the threshold for which cycle counts will be emitted. Error if attempt to set below minimum defined in IDR3, masked to width of valid bits.
Depends: CC enabled.

File: `syncfreq` (rw)
Trace Registers: SYNCPR
Notes: Set trace synchronisation period. Power of 2 value, 0 (off) or 8-20. Driver defaults to 12 (every 4096 bytes).

File: `cntr_idx` (rw)

Trace Registers: none

Notes: Select the counter to access

Syntax: `echo idx > cntr_idx`

Where idx < nr_cntr

File: `cntr_ctrl` (rw)

Trace Registers: CNTCTLR[idx]

Notes: Set counter control value.

Depends: `cntr_idx`

Syntax: `echo val > cntr_ctrl`

Where val is per ETMv4 spec.

File: `cntrldvr` (rw)

Trace Registers: CNTRLDVR[idx]

Notes: Set counter reload value.

Depends: `cntr_idx`

Syntax: `echo val > cntrldvr`

Where val is per ETMv4 spec.

File: `nr_cntr` (ro)
Trace Registers: From IDR5
Notes: Number of counters implemented.

File: `ctxid_idx` (rw)

Trace Registers: None

Notes: Select the context ID comparator to access

Syntax: `echo idx > ctxid_idx`

Where idx < numcidc

File: `ctxid_pid` (rw)

Trace Registers: CIDCVR[idx]
Notes: Set the context ID comparator value
Depends: `ctxid_idx`

File: `ctxid_masks` (rw)

Trace Registers: CIDCCTLR0, CIDCCTLR1, CIDCVR<0-7>

Notes: Pair of values to set the byte masks for 1-8 context ID comparators. Automatically clears masked bytes to 0 in CID value registers.

Syntax: `echo m3m2m1m0 [m7m6m5m4] > ctxid_masks`

32 bit values made up of mask bytes, where mN represents a byte mask value for Context ID comparator N.

Second value not required on systems that have fewer than 4 context ID comparators

---

File: `numcidc` (ro)
Trace Registers: From IDR4
Notes: Number of Context ID comparators

File: `vmid_idx` (rw)

Trace Registers: None

Notes: Select the VM ID comparator to access.

Syntax: `echo idx > vmid_idx`

Where idx < numvmidc

File: `vmid_val` (rw)
Trace Registers: VMIDCVR[idx]
Notes: Set the VM ID comparator value
Depends: `vmid_idx`

File: `vmid_masks` (rw)

Trace Registers: VMIDCCTLR0, VMIDCCTLR1, VMIDCVR<0-7>

Notes: Pair of values to set the byte masks for 1-8 VM ID comparators. Automatically clears masked bytes to 0 in VMID value registers.

Syntax: `echo m3m2m1m0 [m7m6m5m4] > vmid_masks`

Where mN represents a byte mask value for VMID comparator N. Second value not required on systems that have fewer than 4 VMID comparators.

File: `numvmidc` (ro)
Trace Registers: From IDR4
Notes: Number of VMID comparators

File: `res_idx` (rw)

Trace Registers: None.

Notes: Select the resource selector control to access. Must be 2 or higher as selectors 0 and 1 are hardwired.

Syntax: `echo idx > res_idx`

Where 2 <= idx < nr_resource x 2

File: `res_ctrl` (rw)

Trace Registers: RSCTLR[idx]

Notes: Set resource selector control value. Value per ETMv4 spec.

Depends: `res_idx`

Syntax: `echo val > res_cntr`

Where val is per ETMv4 spec.

File: `nr_resource` (ro)
Trace Registers: From IDR4
Notes: Number of resource selector pairs

File: `event` (rw)

Trace Registers: EVENTCTRL0R

Notes: Set up to 4 implemented event fields.

Syntax: `echo ev3ev2ev1ev0 > event`

Where evN is an 8 bit event field. Up to 4 event fields make up the 32-bit input value. Number of valid fields is implementation dependent, defined in IDR0.

File: `event_instren` (rw)

Trace Registers: EVENTCTRL1R

Notes: Choose events which insert event packets into trace stream.

Depends: EVENTCTRL0R

Syntax: `echo bitfield > event_instren`

Where bitfield is up to 4 bits according to number of event fields.

File: `event_ts` (rw)

Trace Registers: TSCTLR

Notes: Set the event that will generate timestamp requests.

Depends: `TS activated`

Syntax: `echo evfield > event_ts`

Where evfield is an 8 bit event selector.

File: `seq_idx` (rw)
Trace Registers: None
Notes: Sequencer event register select - 0 to 2

File: `seq_state` (rw)
Trace Registers: SEQSTR
Notes: Sequencer current state - 0 to 3.

File: `seq_event` (rw)

Trace Registers: SEQEVR[idx]

Notes: State transition event registers

Depends: `seq_idx`

Syntax: `echo evBevF > seq_event`

Where evBevF is a 16 bit value made up of two event selectors,

- evB : back
- evF : forwards.

File: `seq_reset_event` (rw)

Trace Registers: SEQRSTEVR

Notes: Sequencer reset event

Syntax: `echo evfield > seq_reset_event`

Where evfield is an 8 bit event selector.

File: `nrseqstate` (ro)
Trace Registers: From IDR5
Notes: Number of sequencer states (0 or 4)

File: `nr_pe_cmp` (ro)
Trace Registers: From IDR4
Notes: Number of PE comparator inputs

File: `nr_ext_inp` (ro)
Trace Registers: From IDR5
Notes: Number of external inputs

File: `nr_ss_cmp` (ro)
Trace Registers: From IDR4
Notes: Number of Single Shot control registers

Note: When programming any address comparator the driver will tag the comparator with a type used - i.e. RANGE, SINGLE, START, STOP. Once this tag is set, then only the values can be changed using the same sysfs file / type used to program it.

Thus:

```
% echo 0 > addr_idx           ; select address comparator 0
% echo 0x1000 0x5000 0 > addr_range ; set address range on comparators 0, 1.
% echo 0x2000 > addr_start    ; error as comparator 0 is a range comparator
% echo 2 > addr_idx           ; select address comparator 2
% echo 0x2000 > addr_start    ; this is OK as comparator 2 is unused.
% echo 0x3000 > addr_stop     ; error as comparator 2 set as start address.
% echo 2 > addr_idx           ; select address comparator 3
% echo 0x3000 > addr_stop     ; this is OK
```

To remove programming on all the comparators (and all the other hardware) use the reset parameter:

```
% echo 1 > reset
```

The â€˜modeâ€™ sysfs parameter.

System Message: WARNING/2 (`D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\trace\coresight\[linux-master][Documentation][trace][coresight]coresight-etm4x-reference.rst`, line 632)

Title underline too short.

```
The â€˜modeâ€™ sysfs parameter.
---------------------------
```

This is a bitfield selection parameter that sets the overall trace mode for the ETM. The table below describes the bits, using the defines from the driver source file, along with a description of the feature these represent. Many features are optional and therefore dependent on implementation in the hardware.

Bit assignments shown below:-

---

bit (0):
ETM_MODE_EXCLUDE
description:
This is the default value for the include / exclude function when setting address ranges. Set 1 for exclude range. When the mode parameter is set this value is applied to the currently indexed address range.
bit (4):
ETM_MODE_BB
description:
Set to enable branch broadcast if supported in hardware [IDR0].
bit (5):
ETMv4_MODE_CYCACC
description:
Set to enable cycle accurate trace if supported [IDR0].
bit (6):

ETMv4_MODE_CTXID

description:

Set to enable context ID tracing if supported in hardware [IDR2].

bit (7):

ETM_MODE_VMID

description:

Set to enable virtual machine ID tracing if supported [IDR2].

bit (11):

ETMv4_MODE_TIMESTAMP

description:

Set to enable timestamp generation if supported [IDR0].

bit (12):

ETM_MODE_RETURNSTACK

description:

Set to enable trace return stack use if supported [IDR0].

bit (13-14):

ETM_MODE_QELEM(val)

description:

â€˜valâ€™ determines level of Q element support enabled if implemented by the ETM [IDR0]

bit (19):

ETM_MODE_ATB_TRIGGER

description:

Set to enable the ATBTRIGGER bit in the event control register [EVENTCTLR1] if supported [IDR5].

bit (20):

ETM_MODE_LPOVERRIDE

description:

Set to enable the LPOVERRIDE bit in the event control register [EVENTCTLR1], if supported [IDR5].

bit (21):

ETM_MODE_ISTALL_EN

description:

Set to enable the ISTALL bit in the stall control register [STALLCTLR]

bit (23):

ETM_MODE_INSTPRIO

description:

Set to enable the INSTPRIORITY bit in the stall control register [STALLCTLR] , if supported [IDR0].

bit (24):

ETM_MODE_NOOVERFLOW

description:

Set to enable the NOOVERFLOW bit in the stall control register [STALLCTLR], if supported [IDR3].

bit (25):

ETM_MODE_TRACE_RESET

description:

Set to enable the TRCRESET bit in the viewinst control register [VICTLR] , if supported [IDR3].

bit (26):

ETM_MODE_TRACE_ERR

description:

Set to enable the TRCCTRL bit in the viewinst control register [VICTLR].

bit (27):

ETM_MODE_VIEWINST_STARTSTOP

description:

Set the initial state value of the ViewInst start / stop logic in the viewinst control register [VICTLR]

bit (30):

ETM_MODE_EXCL_KERN

description:

Set default trace setup to exclude kernel mode trace (see note a)

bit (31):

ETM_MODE_EXCL_USER

description:

Set default trace setup to exclude user space trace (see note a)

Note a) On startup the ETM is programmed to trace the complete address space using address range comparator 0. â€˜modeâ€™ bits 30 / 31 modify this setting to set EL exclude bits for NS state in either user space (EL0) or kernel space (EL1) in the address range comparator. (the default setting excludes all secure EL, and NS EL2)

Once the reset parameter has been used, and/or custom programming has been implemented - using these bits will result in the EL

bits for address comparator 0 being set in the same way.

Note b) Bits 2-3, 8-10, 15-16, 18, 22, control features that only work with data trace. As A-profile data trace is architecturally prohibited in ETMv4, these have been omitted here. Possible uses could be where a kernel has support for control of R or M profile infrastructure as part of a heterogeneous system.

Bits 17, 28-29 are unused.