XR/17032 Processor Specification

Draft 8

April 1, 2023

SPECIFICATION COPYRIGHT

REGISTERS

The XR/17032 processor architecture contains 32 GPRs (general-purpose registers), and 16 CRs (control registers).

Control Registers

For control registers, there are 2 instructions, `MTCR` and `MFCR`, that alter the control registers, and for these 2 instructions the control registers are encoded in 4 bits of the opcode. The `TBLO` and `TBINDEX` control registers specifically are modified by the `TBRD` and `TBFN` instructions.

Name	Encoding	Description
RS	0000	Processor Status
TBLO	0010	TLB Entry
EPC	0011	Exception Program Counter
EVEC	0100	Exception Vector
PGTB	0101	Page Table Base
TBINDEX	0110	TLB Index
EBADADDR	0111	Bad Address
TBVEC	1000	TLB Miss Handler
FWVEC	1001	Firmware Call Vector
TBSCRATCH	1010	TLB Miss Scratch
TBHI	1011	ASID & TLB Miss Address
K0	1100	
K1	1101	
K2	1110	
K3	1111	

General Purpose Registers

General purpose registers are encoded with 5 bits in instruction opcodes using registers.

Name	Encoding	Description
ZERO	00000 Zero Register: always reads zero, ignores writes.	
T0	00001	
T1	00010	
T2	00011	
T3	00100	
T4	00101	
T5	00110	
A0	00111	
A1	01000	
A2	01001	
A3	01010	
S0	01011	
S1	01100	
S2	01101	
S3	01110	
S4	01111	
S5	10000	

Name	Encoding	Description
S6	10001	
S7	10010	
S8	10011	
S9	10100	
S10	10101	
S11	10110	
S12	10111	
S13	11000	
S14	11001	
S15	11010	
S16	11011	
S17	11100	
S18 / TP	11101	**AISIX:** Scratch register 18. **MINTIA:** Thread Local Storage Pointer
SP	11110	Stack Pointer
LR	11111	Link Register

MEMORY

The memory layout of the address space of the XR/17032 architecture are discussed here.

TODO:

- finish writing this section

OPCODES

Example - EX imm16

```
nnnn nnnn nnnn nnnn ssss sddd dd11 1100
```

This is where the full English explanation of an opcode goes.

Implementation

```
// put the exact code of how its implemented in xremu here. md.py adds backticks.
```

Notes

Any auxiliary notes (such as referring to other opcodes or document sections) goes here.

This is the only optional opcode toml key.

Example 2 - EX2 imm16

```
nnnn nnnn nnnn nnnn ssss sddd dd11 1111
```

This is where the full English explanation of an opcode goes.

Implementation

```
// put the exact code of how its implemented in xremu here. md.py adds backticks.
```

Notes

Any auxiliary notes (such as referring to other opcodes or document sections) goes here.

This is the only optional opcode toml key.