Thor2023 - Coherence Point Processor

© 2023 Robert Finch

# Instruction Set

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | -0 | -1 | -2 | -3 | -4 | -5 | -6 | -7 |
| 0- | END | NOP | MIN | MAX | ADD | CMP | SHL | SHR |
| 1- | ANDC | OR | XOR |  | LOAD | STORE |  | DBR |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| Instructions128 | OperandB96 | OperandA96 | OperandT96 | Memory address96 |

Operands / Registers

|  |  |
| --- | --- |
| Opx3 |  |
| 0 | Memory address |
| 1 | OperandT |
| 2 | OperandA |
| 3 | OperandB |
| 4 | OperandC |
| 5 | Register 5 |
| 6 | Register 6 |
| 7 | Register 7 |
|  |  |
|  | Ccr – two bits |

OperandT is returned to the memory system.

## ADD

**Description:**

Add two source operands and place result in target.

**Instruction Format:**

|  |  |  |  |
| --- | --- | --- | --- |
| 12 10 | 9 7 | 6 4 | 3 0 |
| Rb3 | Ra3 | Rt3 | 44 |

Load ‘add’ and store:

|  |
| --- |
| LOAD Rt,[ma]  ADD r5,Rt,Ra  STORE r5,[ma]  END |

## CMP

**Description:**

Compare two source operands and place result in ccr. If Opa and Opb are equal then the ccr is set to zero. If Opa is less than Opb then the ccr is set to 1 otherwise the ccr is set to 3.

**Instruction Format:**

|  |  |  |  |
| --- | --- | --- | --- |
| 12 10 | 9 7 | 6 4 | 3 0 |
| Opb3 | Opa3 | ~3 | 54 |

## DBR

**Description:**

Decrement and branch. Register #7 is decremented and the program loops backwards by ‘D’ instructions unless r7 is zero.

**Instruction Format:**

|  |  |  |
| --- | --- | --- |
| 12 8 | 7 4 | 3 0 |
| 05 | D4 | 154 |

## END

**Description:**

End of atomic sequence.

**Instruction Format:**

|  |  |  |  |
| --- | --- | --- | --- |
| 12 10 | 9 7 | 6 4 | 3 0 |
| 03 | 03 | 03 | 04 |

## NOP

**Description:**

Do not perform any operation

**Instruction Format:**

|  |  |  |  |
| --- | --- | --- | --- |
| 12 10 | 9 7 | 6 4 | 3 0 |
| 03 | 03 | 03 | 14 |

## LOAD

**Description:**

Load operand from memory address. The memory address is specified by operand A.

**Instruction Format:**

|  |  |  |  |
| --- | --- | --- | --- |
| 12 10 | 9 7 | 6 4 | 3 0 |
| ~3 | Ra3 | Rt3 | 124 |

**Operation:**

Rt = memory[Ra]

## STORE

**Description:**

Conditionally store operand to memory address.

|  |  |
| --- | --- |
| Cond3 | Store Condition |
| 0 | Store if eq |
| 1 | Store if lt |
| 2 | Store of le |
| 3 | Always store |
| 4 | Store if !eq (ne) |
| 5 | Store if !lt (ge) |
| 6 | Store if !le (gt) |
| 7 | Reserved |

**Instruction Format:**

|  |  |  |  |
| --- | --- | --- | --- |
| 12 10 | 9 7 | 6 4 | 3 0 |
| Opb3 | 03 | Cond3 | 134 |

Compare and Exchange

|  |
| --- |
| LOAD OpT,[mar]  CMP OpT,OpA  STORE.EQ OpB,[mar]  MOV OpT,ccr  END |

Memcpy

|  |
| --- |
| LOAD r6,[Opa]  STORE r6,[Opb]  ADD Opa,Opa,Opc  ADD Opb,Opb,Opc  DBR Opt,-4  END |

CMP

OR

ANDC

Load

|  |
| --- |
| LOAD OpT,[ma]  END |

Store

|  |
| --- |
| STORE OpA,[ma]  END |

Load ‘add’ and store:

|  |
| --- |
| LOAD OpT,[ma]  ADD r4,OpT,OpA  STORE r4,[ma]  END |

Load ‘or’ and store:

|  |
| --- |
| LOAD OpT,[ma]  OR r4,OpT,OpA  STORE r4,[ma]  END |

Load ‘andc’ and store:

|  |
| --- |
| LOAD OpT,[ma]  ANDC r4,OpT,OpA  STORE r4,[ma]  END |

AMO MIN:

|  |
| --- |
| LOAD OpT,[ma]  MIN OpT,OpT,OpA  STORE OpT,[ma]  END |