robfinch@<remove>Finitron.ca

## Document

This document outlines the proposed vector instruction set extension included with the Thor processing core.

## Some Considerations

### Interrupt latencies.

So that vectors with a long length don’t impact interrupt latencies too adversely there are two sets of instructions. The first set specifies the vector length in a register for cases where the length is short (eg. three or four element). The second set specifies a specific element to process. When the length specification is used the vector instruction is re-queued as separate instructions for each element at the enqueue stage of the core. This allows the core to distribute the vector operations among available functional units so that processing of mulitiple elements may proceed in parallel.

When a large number of elements is present each element is processed by a separate instruction. An external loop is required to process the elements. This allows interrupts to occur during the processing of a vector operation.

### Code density.

Code density impacts cache performance. When a single instruction can be used to represent a vector operation by specifying the vector length, code density is improved.

### Load / Store

Load / store operations always specify a specific element to load or store. This is to allow potential exceptions that occur during a load or store operation to be processed in order.

### Instruction Encoding

Thor uses variable length instruction encodings in order to achieve good code density. The instruction encoding allows for up to 64 vector registers.

## Detailed Vector Instruction Set

### VADD

Synopsis

Vector register add. Vt = Va + Vb

***Description***

Two vector registers (Va and Vb) are added together and placed in the target vector register Vt. Which element of the vector to add or the vector length is specified in register Ra.

***Instruction Format (element specified)***

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 47 43 | 42 40 | 39 34 | 33 28 | 27 22 | 21 16 | 15 0 | 7 | 0 |
| 05 | T3 | Vt6 | Vb6 | Va6 | Ra6 | 57h8 | Pn4 | Pc4 |

***Instruction Format (length specified)***

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 47 43 | 42 40 | 39 34 | 33 28 | 27 22 | 21 16 | 15 0 | 7 | 0 |
| 10h5 | T3 | Vt6 | Vb6 | Va6 | Ra6 | 57h8 | Pn4 | Pc4 |

***Operation***

Vt[Ra] =Va[Ra] + Vb[Ra]

or

for x = 1 to [Ra]

Vt[x] = Va[x] + Vb[x]

Operand Type

|  |  |  |
| --- | --- | --- |
| T3 |  |  |
| 0 | Integer |  |
| 1 | Float single |  |
| 2 | float double |  |
| 4 | Float Quad |  |

### VAND

Synopsis

Vector register bitwise and. Vt = Va & Vb

***Description***

Two vector registers (Va and Vb) are anded together and placed in the target vector register Vt. Which element of the vector to and or the vector length is specified in register Ra.

***Instruction Format (element specified)***

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 47 43 | 42 40 | 39 34 | 33 28 | 27 22 | 21 16 | 15 0 | 7 | 0 |
| 85 | 03 | Vt6 | Vb6 | Va6 | Ra6 | 57h8 | Pn4 | Pc4 |

***Instruction Format (length specified)***

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 47 43 | 42 40 | 39 34 | 33 28 | 27 22 | 21 16 | 15 0 | 7 | 0 |
| 18h5 | 03 | Vt6 | Vb6 | Va6 | Ra6 | 57h8 | Pn4 | Pc4 |

***Operation***

Vt[Ra] =Va[Ra] & Vb[Ra]

or

for x = 1 to [Ra]

Vt[x] = Va[x] & Vb[x]

Operand Type

|  |  |  |
| --- | --- | --- |
| T3 |  |  |
| 0 | Integer |  |
|  |  |  |
|  |  |  |
|  |  |  |

### VBITS2V

Synopsis

Convert bits to Boolean vector.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 39 34 | 33 28 | 27 22 | 21 16 | 15 0 | 7 | 0 |
| 66 | Vt6 | Rb6 | Ra6 | 568 | Pn4 | Pc4 |

***Description***

The least significant bit of each vector element is copied to the corresponding bit in the target register. Ra specifies the length of the vector.

***Operation***

For x = 0 to [Ra]-1

Rt[x] = Va[x].LSB

***Exceptions:*** none

### VCMP

Synopsis

Vector register compare. Vt = Va ? Vb

***Description***

Two vector registers (Va and Vb) are compared together and the comparison result is placed in the target vector register Vt. Which element of the vector to compare or the vector length is specified in register Ra. A -1, 0, or +1 is placed in the target vector register element by element depending on the relationship of the elements.

***Instruction Format (element specified)***

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 47 43 | 42 40 | 39 34 | 33 28 | 27 22 | 21 16 | 15 0 | 7 | 0 |
| 65 | T3 | Vt6 | Vb6 | Va6 | Ra6 | 57h8 | Pn4 | Pc4 |

***Instruction Format (length specified)***

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 47 43 | 42 40 | 39 34 | 33 28 | 27 22 | 21 16 | 15 0 | 7 | 0 |
| 16h5 | T3 | Vt6 | Vb6 | Va6 | Ra6 | 57h8 | Pn4 | Pc4 |

***Operation***

Vt[Ra] =Va[Ra] + Vb[Ra]

or

for x = 1 to [Ra]

Vt[x] = Va[x] + Vb[x]

Operand Type

|  |  |  |
| --- | --- | --- |
| T3 |  |  |
| 0 | Integer |  |
| 1 | Float single |  |
| 2 | float double |  |
| 4 | Float Quad |  |

### VEINS

Synopsis

Vector element insert.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 39 34 | 33 28 | 27 22 | 21 16 | 15 0 | 7 | 0 |
| 16 | Vt6 | Rb6 | Ra6 | 568 | Pn4 | Pc4 |

***Description***

A general purpose register Rb is transferred into one element of a vector register Vt. The element to insert is identified by Ra.

***Operation***

Rt = Va[Ra]

***Exceptions:*** none

### VEOR

Synopsis

Vector register bitwise exclusive or. Vt = Va ^ Vb

***Description***

Two vector registers (Va and Vb) are exclusively or’ed together and placed in the target vector register Vt. Which element of the vector to exclusive or, or the vector length is specified in register Ra.

***Instruction Format (element specified)***

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 47 43 | 42 40 | 39 34 | 33 28 | 27 22 | 21 16 | 15 0 | 7 | 0 |
| A5 | 03 | Vt6 | Vb6 | Va6 | Ra6 | 57h8 | Pn4 | Pc4 |

***Instruction Format (length specified)***

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 47 43 | 42 40 | 39 34 | 33 28 | 27 22 | 21 16 | 15 0 | 7 | 0 |
| 1Ah5 | 03 | Vt6 | Vb6 | Va6 | Ra6 | 57h8 | Pn4 | Pc4 |

***Operation***

Vt[Ra] =Va[Ra] ^ Vb[Ra]

or

for x = 1 to [Ra]

Vt[x] = Va[x] & Vb[x]

Operand Type

|  |  |  |
| --- | --- | --- |
| T3 |  |  |
| 0 | Integer |  |
|  |  |  |
|  |  |  |
|  |  |  |

### VDIV

Synopsis

Vector register add. Vt = Va / Vb

***Description***

Two vector registers (Va and Vb) are divided and placed in the target vector register Vt. Which element of the vector to divide or the vector length is specified in register Ra.

***Instruction Format (element specified)***

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 47 43 | 42 40 | 39 34 | 33 28 | 27 22 | 21 16 | 15 0 | 7 | 0 |
| 35 | T3 | Vt6 | Vb6 | Va6 | Ra6 | 578 | Pn4 | Pc4 |

***Instruction Format (length specified)***

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 47 43 | 42 40 | 39 34 | 33 28 | 27 22 | 21 16 | 15 0 | 7 | 0 |
| 13h5 | T3 | Vt6 | Vb6 | Va6 | Ra6 | 578 | Pn4 | Pc4 |

***Operation***

Vt[Ra] =Va[Ra] + Vb[Ra]

or

for x = 1 to [Ra]

Vt[x] = Va[x] + Vb[x]

Operand Type

|  |  |  |
| --- | --- | --- |
| T3 |  |  |
| 0 | Integer |  |
| 1 | Float single |  |
| 2 | float double |  |
| 4 | Float Quad |  |

### VEX

Synopsis

Vector element extract.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 39 34 | 33 28 | 27 22 | 21 16 | 15 0 | 7 | 0 |
| 06 | Rt6 | Va6 | Ra6 | 568 | Pn4 | Pc4 |

***Description***

One element of a vector register Va is transferred to a general purpose register Rt. The element to extract is identified by Ra.

***Operation***

Rt = Va[Ra]

***Exceptions:*** none

### VFLT2INT

Synopsis

Vector float to integer.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 39 34 | 33 28 | 27 22 | 21 16 | 15 0 | 7 | 0 |
| 46 | Vt6 | Va6 | Ra6 | 568 | Pn4 | Pc4 |

***Description***

Elements of the vector are converted from floating point to integer.

***Operation***

For x = 0 to [Ra]-1

Vt[x] = (int)Va[x]

***Exceptions:*** none

### VINT2FLT

Synopsis

Vector float to integer.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 39 34 | 33 28 | 27 22 | 21 16 | 15 0 | 7 | 0 |
| 56 | Vt6 | Va6 | Ra6 | 568 | Pn4 | Pc4 |

***Description***

Elements of the vector are converted from integer to floating point.

***Operation***

For x = 0 to [Ra]-1

Vt[x] =(float) Va[x]

***Exceptions:*** none

### VLD

Synopsis

Load vector element

**Description:**

Load one element of a vector register from memory using register indirect with displacement addressing. The element to load is specified by register Rb.

**Instruction Format:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 47 45 | 44 34 | 33 28 | 27 22 | 21 16 | 15 0 | 7 | 0 |
| Seg3 | Displacement11 | Vt6 | Rb6 | Ra6 | 89h8 | Pn4 | Pc4 |

**Operation**

Vt[Rc] = memory64[displacement+Ra]

**Exceptions:** DBE, DBG, LMT, TLB

### VLDX

Synopsis

Load vector element

**Description:**

Load one element of a vector register from memory using scaled indexed addressing. The element to load is specified by register Rc.

**Instruction Format:**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 47 45 | 44 42 | 41 40 | 39 34 | 33 28 | 27 22 | 21 16 | 15 0 | 7 | 0 |
| Seg3 | ~3 | Sc2 | Vt6 | Rc6 | Rb6 | Ra6 | B9h8 | Pn4 | Pc4 |

**Operation**

Vt[Rc] = memory64[Ra+Rb \* Sc]

**Exceptions:** DBE, DBG, LMT, TLB

### VMAC

Synopsis

Vector register multiply accumulate. Vt = +-( +- Va \*+- Vb +- Vc)

**Description**

Vector registers Va and Vb are multiplied together then accumulated with vector register Vc. The sign of Va, Vb, and Vc may be inverted. The sign of the entire result may be inverted.

Which element or the length of the vectors to MAC is specified in register Ra.

**Instruction Format**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 55 | 52 49 | 48 46 | 45 40 | 39 34 | 33 28 | 27 22 | 21 16 | 15 0 | 7 | 0 |
| O3 | Sn4 | T3 | Vt6 | Vc6 | Vb6 | Va6 | Ra6 | 5A8 | Pn4 | Pc4 |

**Operation**

Vt[Ra] =Va[Ra] \* Vb[Ra] + Vc

**Operand Type**

|  |  |  |
| --- | --- | --- |
| T3 |  |  |
| 0 | Integer |  |
| 1 | Float single |  |
| 2 | float double |  |
| 4 | Float Quad |  |

### VMUL

Synopsis

Vector register add. Vt = Va \* Vb

***Description***

Two vector registers (Va and Vb) are multiplied together and placed in the target vector register Vt. Which element of the vector to multiply or the vector length is specified in register Ra.

***Instruction Format (element specified)***

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 47 43 | 42 40 | 39 34 | 33 28 | 27 22 | 21 16 | 15 0 | 7 | 0 |
| 25 | T3 | Vt6 | Vb6 | Va6 | Ra6 | 57h8 | Pn4 | Pc4 |

***Instruction Format (length specified)***

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 47 43 | 42 40 | 39 34 | 33 28 | 27 22 | 21 16 | 15 0 | 7 | 0 |
| 12h5 | T3 | Vt6 | Vb6 | Va6 | Ra6 | 57h8 | Pn4 | Pc4 |

***Operation***

Vt[Ra] =Va[Ra] + Vb[Ra]

or

for x = 1 to [Ra]

Vt[x] = Va[x] + Vb[x]

Operand Type

|  |  |  |
| --- | --- | --- |
| T3 |  |  |
| 0 | Integer |  |
| 1 | Float single |  |
| 2 | float double |  |
| 4 | Float Quad |  |

### VOR

Synopsis

Vector register bitwise or. Vt = Va | Vb

***Description***

Two vector registers (Va and Vb) are or’ed together and placed in the target vector register Vt. Which element of the vector to or, or the vector length is specified in register Ra.

***Instruction Format (element specified)***

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 47 43 | 42 40 | 39 34 | 33 28 | 27 22 | 21 16 | 15 0 | 7 | 0 |
| 95 | 03 | Vt6 | Vb6 | Va6 | Ra6 | 57h8 | Pn4 | Pc4 |

***Instruction Format (length specified)***

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 47 43 | 42 40 | 39 34 | 33 28 | 27 22 | 21 16 | 15 0 | 7 | 0 |
| 19h5 | 03 | Vt6 | Vb6 | Va6 | Ra6 | 57h8 | Pn4 | Pc4 |

***Operation***

Vt[Ra] =Va[Ra] | Vb[Ra]

or

for x = 1 to [Ra]

Vt[x] = Va[x] | Vb[x]

Operand Type

|  |  |  |
| --- | --- | --- |
| T3 |  |  |
| 0 | Integer |  |
|  |  |  |
|  |  |  |
|  |  |  |

### VSCALE

Synopsis

Scale Vector. Vt = Va \* Ra

***Description***

A vector register Va is scaled by a general purpose register Rb and placed in the target vector register Vt. Which element of the vector to add or the vector length is specified in register Ra.

***Instruction Format (element specified)***

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 47 43 | 42 40 | 39 34 | 33 28 | 27 22 | 21 16 | 15 0 | 7 | 0 |
| 55 | T3 | Vt6 | Va6 | Rb6 | Ra6 | 57h8 | Pn4 | Pc4 |

***Instruction Format (length specified)***

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 47 43 | 42 40 | 39 34 | 33 28 | 27 22 | 21 16 | 15 0 | 7 | 0 |
| 15h5 | T3 | Vt6 | Va6 | Rb6 | Ra6 | 57h8 | Pn4 | Pc4 |

***Operation***

Vt[Ra] =Va[Ra] + Vb[Ra]

or

for x = 1 to [Ra]

Vt[x] = Va[x] + Vb[x]

Operand Type

|  |  |  |
| --- | --- | --- |
| T3 |  |  |
| 0 | Integer |  |
| 1 | Float single |  |
| 2 | float double |  |
| 4 | Float Quad |  |

### VSHL

Synopsis

Vector shift left.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 39 34 | 33 28 | 27 22 | 21 16 | 15 0 | 7 | 0 |
| 26 | Vt6 | Va6 | Ra6 | 568 | Pn4 | Pc4 |

***Description***

Elements of the vector are transferred upwards to the next element position. Element #0 is loaded with the value zero.

***Operation***

For x = 0 to [Ra]-1

Vt[x+1] = Va[x]

Vt[0] = 0

***Exceptions:*** none

### VSHR

Synopsis

Vector shift right.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 39 34 | 33 28 | 27 22 | 21 16 | 15 0 | 7 | 0 |
| 36 | Vt6 | Va6 | Ra6 | 568 | Pn4 | Pc4 |

***Description***

Elements of the vector are transferred downwards to the next element position. The last is loaded with the value zero.

***Operation***

For x = 0 to [Ra]-1

Vt[x] = Va[x+1]

Vt[Ra-1] = 0

***Exceptions:*** none

### VST

Synopsis

Store vector element

**Description:**

Store one element of a vector register to memory using register indirect with displacement addressing. The element to store is specified by register Rb.

**Instruction Format:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 47 45 | 44 34 | 33 28 | 27 22 | 21 16 | 15 0 | 7 | 0 |
| Seg3 | Displacement11 | Vs6 | Rb6 | Ra6 | 89h8 | Pn4 | Pc4 |

**Operation**

memory64[displacement+Ra] = Vs[Rb]

**Exceptions:** DBE, DBG, LMT, TLB

### VSTX

Synopsis

Store vector element

**Description:**

Store one element of a vector register to memory using scaled indexed addressing. The element to store is specified by register Rc.

**Instruction Format:**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 47 45 | 44 42 | 41 40 | 39 34 | 33 28 | 27 22 | 21 16 | 15 0 | 7 | 0 |
| Seg3 | ~3 | Sc2 | Vs6 | Rc6 | Rb6 | Ra6 | 89h8 | Pn4 | Pc4 |

**Operation**

memory64[Ra+Rb \* Sc] = Vs[Rc]

**Exceptions:** DBE, DBG, LMT, TLB

### VSUB

Synopsis

Vector register add. Vt = Va - Vb

**Description**

Two vector registers (Va and Vb) are subtracted and placed in the target vector register Vt. Which element of the vector to subtract or the vector length is specified in register Ra.

**Instruction Format (element specified)**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 47 43 | 42 40 | 39 34 | 33 28 | 27 22 | 21 16 | 15 0 | 7 | 0 |
| 15 | T3 | Vt6 | Vb6 | Va6 | Ra6 | 57h8 | Pn4 | Pc4 |

**Instruction Format (length specified)**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 47 43 | 42 40 | 39 34 | 33 28 | 27 22 | 21 16 | 15 0 | 7 | 0 |
| 11h5 | T3 | Vt6 | Vb6 | Va6 | Ra6 | 57h8 | Pn4 | Pc4 |

**Operation**

Vt[Ra] =Va[Ra] + Vb[Ra]

or

for x = 1 to [Ra]

Vt[x] = Va[x] + Vb[x]

**Operand Type**

|  |  |  |
| --- | --- | --- |
| T3 |  |  |
| 0 | Integer |  |
| 1 | Float single |  |
| 2 | float double |  |
| 4 | Float Quad |  |

### V2BITS

Synopsis

Convert Boolean vector to bits.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 39 34 | 33 28 | 27 22 | 21 16 | 15 0 | 7 | 0 |
| 66 | Rt6 | Va6 | Ra6 | 568 | Pn4 | Pc4 |

***Description***

The least significant bit of each vector element is copied to the corresponding bit in the target register. Ra specifies the length of the vector.

***Operation***

For x = 0 to [Ra]-1

Rt[x] = Va[x].LSB

***Exceptions:*** none