2L-03

組込み向けRISC-Vのためのスケーラブルなベクトル処理に関する検討

木村 嘉毅 † 菊池 智也 †† 大津 金光 †† 大川 猛 †† 横田 隆史 ††
† 宇都宮大学工学部情報工学科 †† 宇都宮大学大学院工学研究科情報システム科学専攻

1 はじめに

FPGA (Field Programmable Gate Array) の大容量化，高性能化により組込みシステムでの利用が増加している．FPGAを用いた開発は一般に難易度が高く，開発期間が長くなることが多い．開発期間の短縮のために，処理のすべてを専用回路として開発するのではなく，ソフトコアプロセッサを併用する選択肢がある．ソフトコアプロセッサの性能が向上すれば，開発すべき専用ハードウェア回路が削減され，開発期間の短縮が可能である．

データ並列性の高い処理であれば，SIMD命令による処理により高性能化を実現できる．しかし，SIMD命令による処理は一般的に，演算能力を高めるために同時演算数を増やす場合，それに応じて機械語コードを作り直す必要がある．異なる同時演算数でも同一の機械語コードをそのまま利用可能とするためには，機械語コードが同時演算数に依存しないスケーラブルなベクトル拡張が必要である．スケーラブルなベクトル拡張により，機械語コードを変更することなく，必要に応じて容易に同時演算数を増やし高性能化することが可能となる．

一方，近年はオープンな命令セットアーキテクチャであるRISC-V[1]が注目されており，組込み分野での利用増加が期待されている．RISC-V命令セットにおいてもSIMD命令セットが定義されているが，これも前述の問題を抱えており，我々の目的には適さない．また，新しいベクトル拡張についての仕様検討も行われているが，これもスケーラブルなものではない．そこで，RISC-Vにおけるスケーラブルなベクトル拡張について検討する．

2 スケーラブルなベクトル拡張

スケーラブルなベクトル拡張の実現のために，ARM社が策定したベクトル命令拡張であるSVE(Scalable Vector Extension)[2]を参考にした．SVEはポスト京コンピュータなどのハイパフォーマンス用途を想定したベクトル拡張であるが，本研究では，そのスケーラブルなベクトル処理を実現している部分に着目し，組込み用途を想定したベクトル拡張を検討する．

SVEの特徴としては，ベクトル長を固定しないという点がある．ベクトル長は128bitから2048bitまで，128の2のべき乗倍を指定できる．また，プレディケートレジスタにより，ベクトル要素ごとのマスク制御を行うことができる．また，ギャザー，スキャターによるロード，ストア機能をもち，非連続領域へのメモリアクセスをサポートしている．

SVEでは，スケーラブルなベクトル処理を行うための特徴として，プレディケートによるループ制御がある．ループ処理を行うときに使う命令として，ループカウンタと処理する全データ数を比較し，ループカウンタがデータ数より小さい間対応するプレディケートレジスタをTrueにする命令がある．この命令により，任意のデータ数に対してベクトル処理を行うことができる．また，ベクトルの要素数をループカウンタに加えるという命令がある．この命令によりベクトル長によらない処理が可能となる．これらの命令により，データ数やベクトル長を考慮することなくベクトル化を行うことができる．

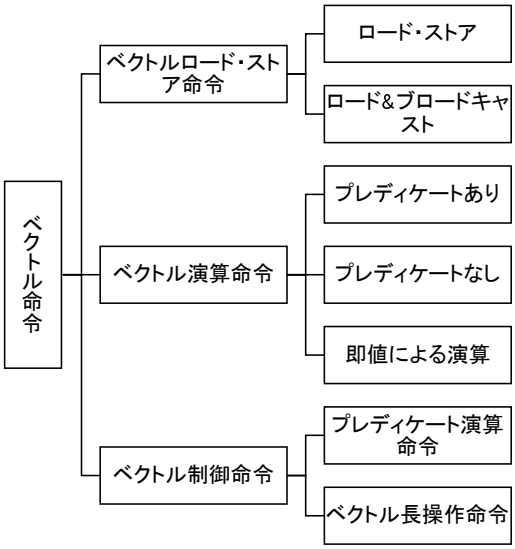

図 1: 本研究で提案するベクトル拡張の命令体系

3 ベクトル命令セット

まずレジスタ構成について検討する．既存のSVEコンパイラを活用にSVEと同様のレジスタ構成とする．本ベクトル拡張は，RISC-Vの汎用レジスタに加え，ベクトルレジスタv0-v31，ベクトルマスク制御に用いるためのプレディケートレジスタvp0-vp7，プレディケートレジスタ同士の論理演算に用いるプレディケートレジスタvp8-vp15，ベクトル長レジスタをもつ．ベクトル長は可変であり，128×2^nbitで表される．また，演算の最小単位は1byteであるので，ベク

Consideration of scalable vector processing for embedded RISC-V
†Yoshiki Kimura, ††Tomoya Kikuchi, ††Kanemitsu Ootsu, ††Takeshi Ohkawa, ††Takashi Yokota
Department of Information Science, Faculty of Engineering, Utsunomiya University (†)
Department of Information Systems Science, Graduate School of Engineering, Utsunomiya University (††)

トルマスク制御においてベクトルレジスタ 1byte とプレディケートレジスタ 1bit が対応する．すなわち，プレディケートレジスタの長さはベクトル長の 1/8 となる．

ベクトル命令拡張の命令体系を図 1 に示す．ベクトル命令は大きく分けてベクトルロード，ストア命令，ベクトル演算命令，ベクトル制御命令の 3 つに分けられる．RISC-V にはカスタム命令のためオペコード領域が 4 つ用意されている．そのうち 1 つのみを利用する場合，オペコード領域が足りず全ての命令をエンコーディングすることができない．そこで，オペコード領域を 2 つ利用し，そのうち 1 つをベクトルロード，ストア命令，もう 1 つをベクトル演算命令とベクトル制御命令に割り当てた．

ベクトルロード，ストア命令のアドレスは，ベースとなるスカラレジスタにオフセットを加えることで計算する．オフセットはスカラレジスタ，ベクトルレジスタ，即値の 3 種類ある．オフセットにベクトルレジスタを指定することでギャザー，スキャターによる不連続なロード，ストアを実現できる．また，メモリアクセス時のデータのサイズとベクトルレジスタの要素サイズを別々に指定できる．例えば，メモリからバイトサイズのデータを読み出し，拡張してワードサイズでベクトルレジスタに格納することができる．それに加え，メモリから読み込んだデータをベクトルレジスタの各要素にブロードキャストするロード命令をサポートする．

ベクトル演算命令は，プレディケートあり演算命令，プレディケートなし演算命令，即値による演算命令に分けられる．演算命令は，基本的な算術論理演算命令，乗除算命令，ベクトルレジスタの各要素の総和を求める命令やアドレス計算のときに使うインデックスレジスタを生成する命令が含まれる．プレディケートあり演算命令は，プレディケートレジスタを指定しベクトルマスク制御を行う．

ベクトル制御命令は，プレディケート演算命令とベクトル長操作命令に分けられる．プレディケート演算命令は，プレディケートレジスタ同士の論理演算を行う．これにより，柔軟なベクトルマスク制御を行うことができる．ベクトル長操作命令は，スカラレジスタに対しベクトル長を足す等の命令がある．

次に，命令のフォーマットについて検討する．命令のフォーマットは，RISC-V の命令形式にできるだけ従った．また，デコーダの単純化のために同じフィールドにはできるだけ同じ機能をもたせた．例として，図 2 に代表的な命令のフォーマットを示す．上から，プレディケートあり VADD 命令，プレディケートなし VADD 命令，即値による VADDI 命令，ロードを行う VLW 命令である．

4　ベクトル拡張命令の使用例

図 3 に，提案するベクトル拡張によるコードの例として配列間の減算を行うプログラムのアセンブリコードを示す．ニーモニックに接頭語 v がつく命令がベクトル拡張命令である．x9 レジスタは配列の長さを格納するレジスタ，x8 レジスタはループカウンタとして使っている．3 行目ではゼロレジスタと x9 を比較し，プレディケートレジスタ vp0 を設定する．4，5 行目でそれぞれベクトルレジスタにデータをロードし，6 行目でベクトルの減算を行い，7 行目で結果を格納する．8 行目ではループカウンタをベクトル要素数でインクリメントする．9 行目でループカウンタと配列の長さを比較しプレディケートレジスタを設定し，10 行目でループの条件判定を行う．図 3 のアセンブリコードはベクトル長に依存しない．検討したベクトル拡張では，バイナリコードを変更することなくベクトル長を設定できるスケーラブルなベクトル処理が可能である．

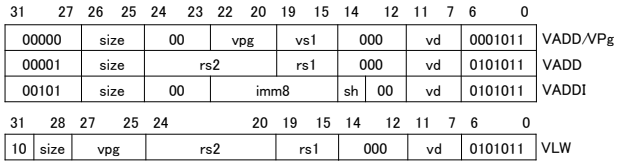

図 2: 主要なベクトル命令のフォーマット

```
1  SUB:    addi          x9,  x0, 1024
2          mv            x8,  x0
3          vwhilelo.w    vp0, x0, x9
4  LOOP:   vlw.w/vp0     v0,  x1, x8
5          vlw.w/vp0     v1,  x2, x8
6          vsub.w/vp0    v0,  v1
7          vsw.w/vp0     v0,  x3, x8
8          vincw         x8
9          vwhilelo.w    vp0, x8, x9
10         blt           x8,  x9, LOOP
```

図 3: アセンブリコード例 (配列間の減算)

5　おわりに

本稿では，既存のベクトル拡張である ARM SVE を参考にし，スケーラブルなベクトル拡張に必要な機能を検討した．さらに，それをもとに RISC-V へのベクトル拡張命令セットを検討した．今後は，検討したベクトル拡張の命令の性能評価を行い，本機能の有効性を明らかにする．

謝辞

本研究は，一部 JSPS 科研費 16K00068, 17K00072 の助成による．

参考文献

[1] Andrew Waterman, et al.: “The RISC-V Instruction Set Manual, Volume I: User-Level ISA, Document Version2.2”, RISC-V Foundation, 2017.

[2] Nigel Stephens, et al.: “The ARM scalable vector extension”, IEEE Micro, Vol.37, No.2, pp.26-39, 2017.