A reconfigurable stream compression hardware based on static symbol-lookup table

Methods

Baba Shunsuke

A. Design of LeA-SLT

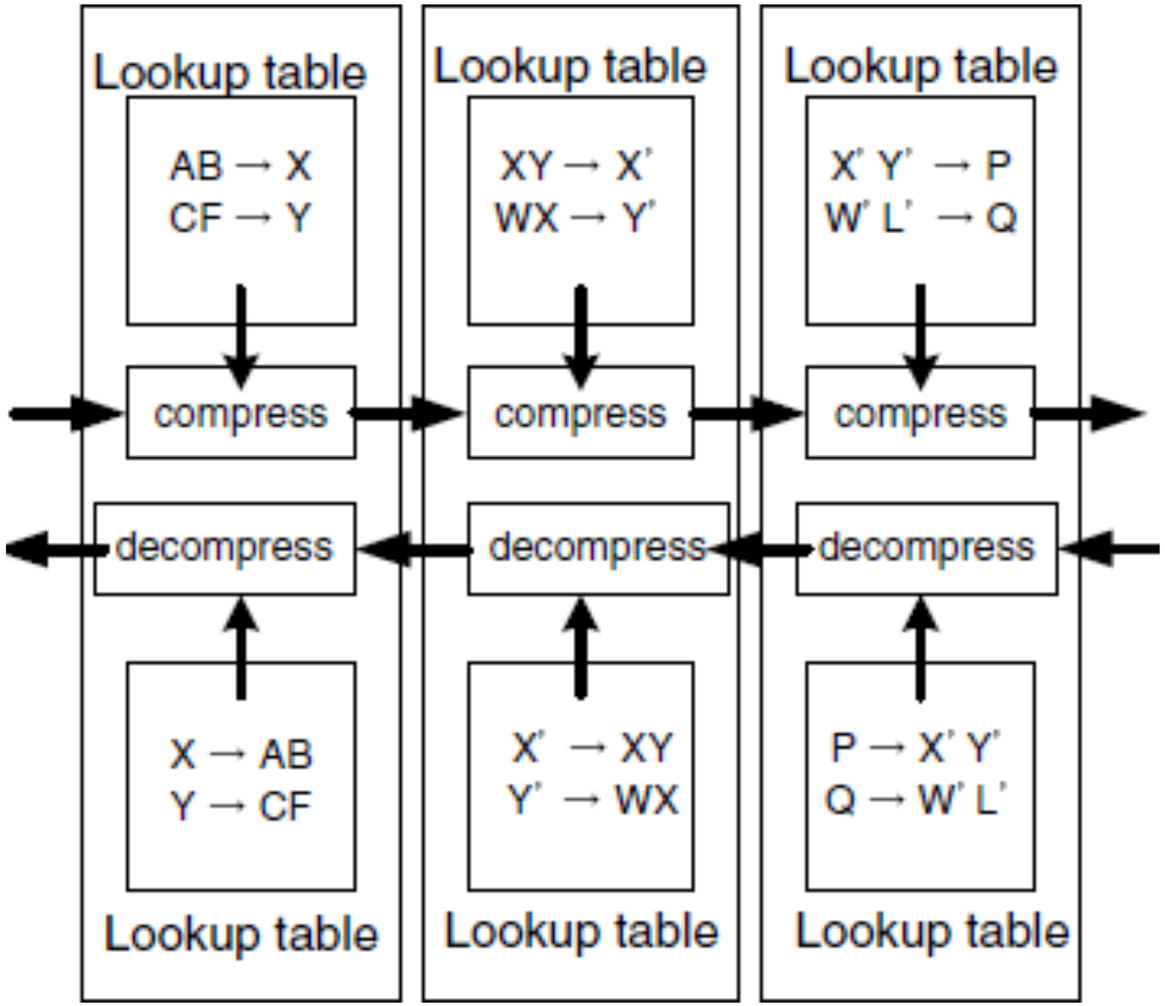
LCA-SLTの基本Module

- Compressor
- lookup table for compression
- decompressor
- lookup table for decompression

A. Design of LeA-SLT

- Compressor line
- Decompressor line

→反復回帰可能なoperationsの実装。



A. Design of LeA-SLT

・従来のオンラインLCAとの違い

→Symbolペアの変換に使用するLookup tableを静的に準備する点
モジュール自身はルックアップテーブルの内容を更新せず。テーブルの内容は、圧縮/解凍の前に静的に準備される。

A. Design of LeA-SLT

- テーブルの決め方。
  ① 対象データのテストデータセットをオンラインLCAで調べ、すべてのペアとそのマッチング記号のルックアップテーブルを生成する、
  ② ルックアップテーブルのエントリを頻度の昇順でソートする。
  ③ 使用頻度の高いエントリをテーブルの中身として使用する。
  ここで、変換後の記号(AB－＞X)が作られる。
  －＞テストデータセットによって、Look up tableがきまるので、テストデータ以外のデータでは、ミスマッチがおこってしまう。
  －＞テーブル作成にはLRUや機械学習などの統計的アプローチが必要である。

A. Design of LeA-SLT

静的LUTのメリット

静的LUTを用いたアプローチでは

①圧縮データにテーブルを必要としない

②圧縮率に応じて、エントリ数を決めることができる

エントリ数多くすれば、圧縮率があがり、

エントリ数をさげれば、圧縮率下がる。

A. Design of LeA-SLT

応用

ネットワークへの応用

・NICれべるでの圧縮率の観察により、圧縮率の異常があった際は全ノードのテーブルを最新のデータセットを使って生成し直すことによって、改善することができる。

B.Validity of compression mechanism of static lookup tables

データセット

エントリ数を32、64、128、256と変え
エントリ数の制限による
圧縮効果の影響を調べた

	Size	Describe
Sources.50MB	50MB	ソースコード linux-2.6.11.6
Sources.50MB	200MB	ソースコード gcc-4.0.0
dna	50MB	遺伝子のDNA配列を何個かならべたもの

B.Validity of compression mechanism of static lookup tables

圧縮ビット

バイト単位で圧縮をお行なう場合
圧縮された記号を未使用の記号に割り当てる
必要がある

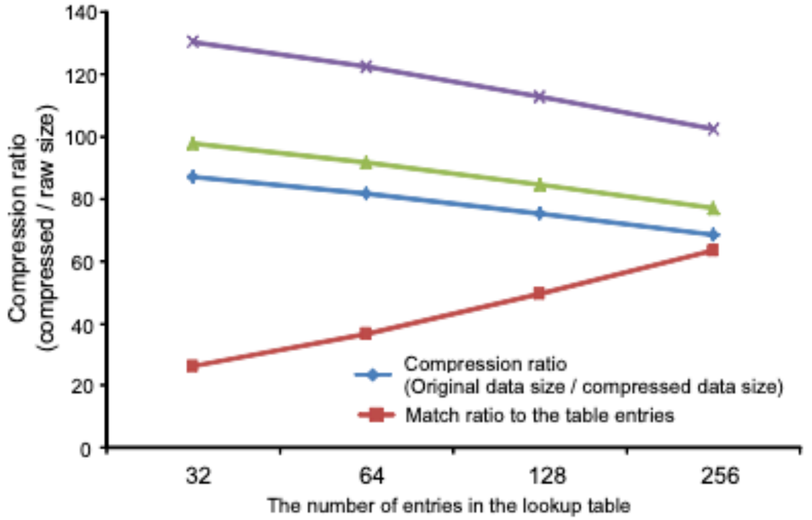
0000000~11111111(2)
は使われている

ー＞ビット拡張を行なって、
未使用の記号を表現

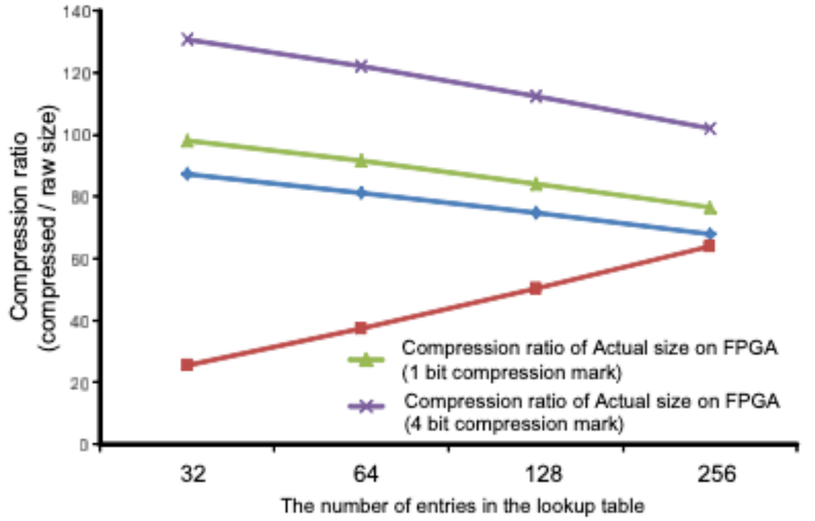
	Size	Describe
Sources.50MB	50MB	ソースコード linux-2.6.11.6
Sources.50MB	200MB	ソースコード gcc-4.0.0
dna	50MB	遺伝子のDNA配列を何個かならべたもの

B.Validity of compression mechanism of static lookup tables
結果

- (a)データセット Source.50MB 圧縮対象 Source.50MBで実験
- (b)データセット Source.50MB 圧縮対象 Source.200MBで実験



(a) The compression ratio and the matching ratio of the entries in the table, applying the static table of source.50MB to source.50MB.



(b) The compression ratio and the matching ratio of the entries in the table, applying the static table of source.50MB to source.200MB.

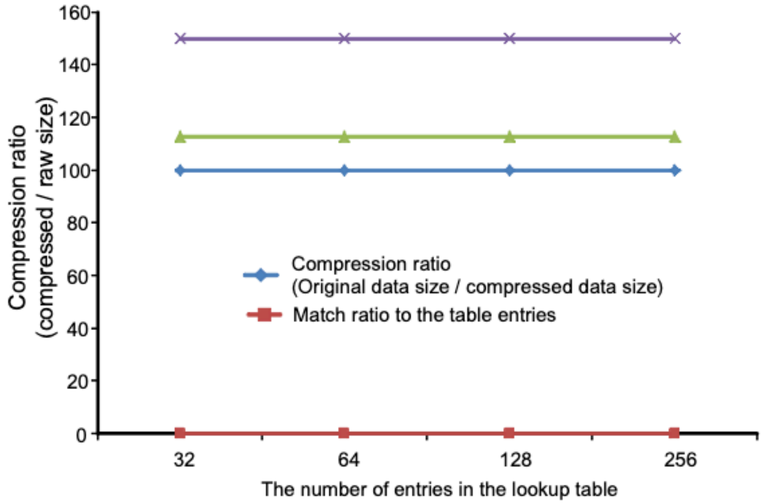
Figure 4. Comparing the compression ratio and the matching ratio of table entries applying *source.50MB* benchmark.

B.Validity of compression mechanism of static lookup tables
結果

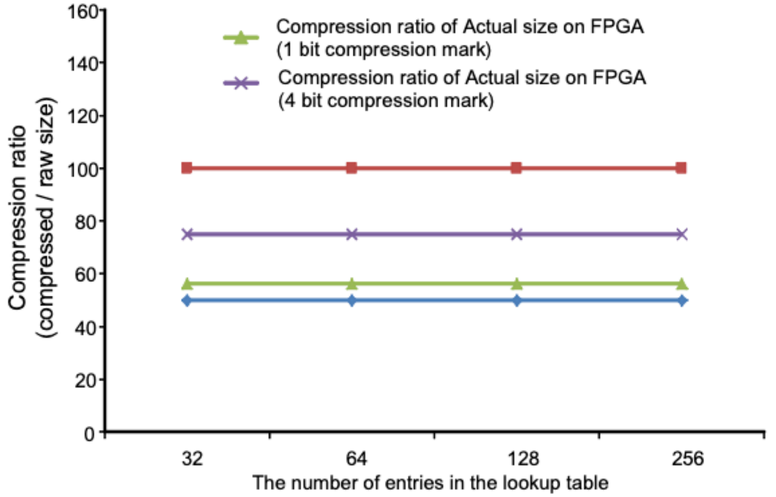
- (a)データセット Source.50MB
  圧縮対象 dna.50MBで実験
- (b)データセット dna.50MB
  圧縮対象 dna.200MBで実験

  上位32エントリには、すでに全シンボルペアの99%が

  含まれてる



(a) The compression ratio and the matching ratio of the entries in the table, applying the static table of source.50MB to dna.50MB



(b) The compression ratio and the matching ratio of the entries in the table, applying the static table of dna.50MB to dna.50MB

Figure 5. Comparing the compression ratio and the matching ratio of table entries applying *dna.50MB* benchmark.

FPGAでの実装

実装の概要

- 圧縮先の記号を圧縮してない記号と区別するためにビットを一個追加する実装。
  CAM（Content Addressable Memory）[16]
  MEM（Normal Data Memory）

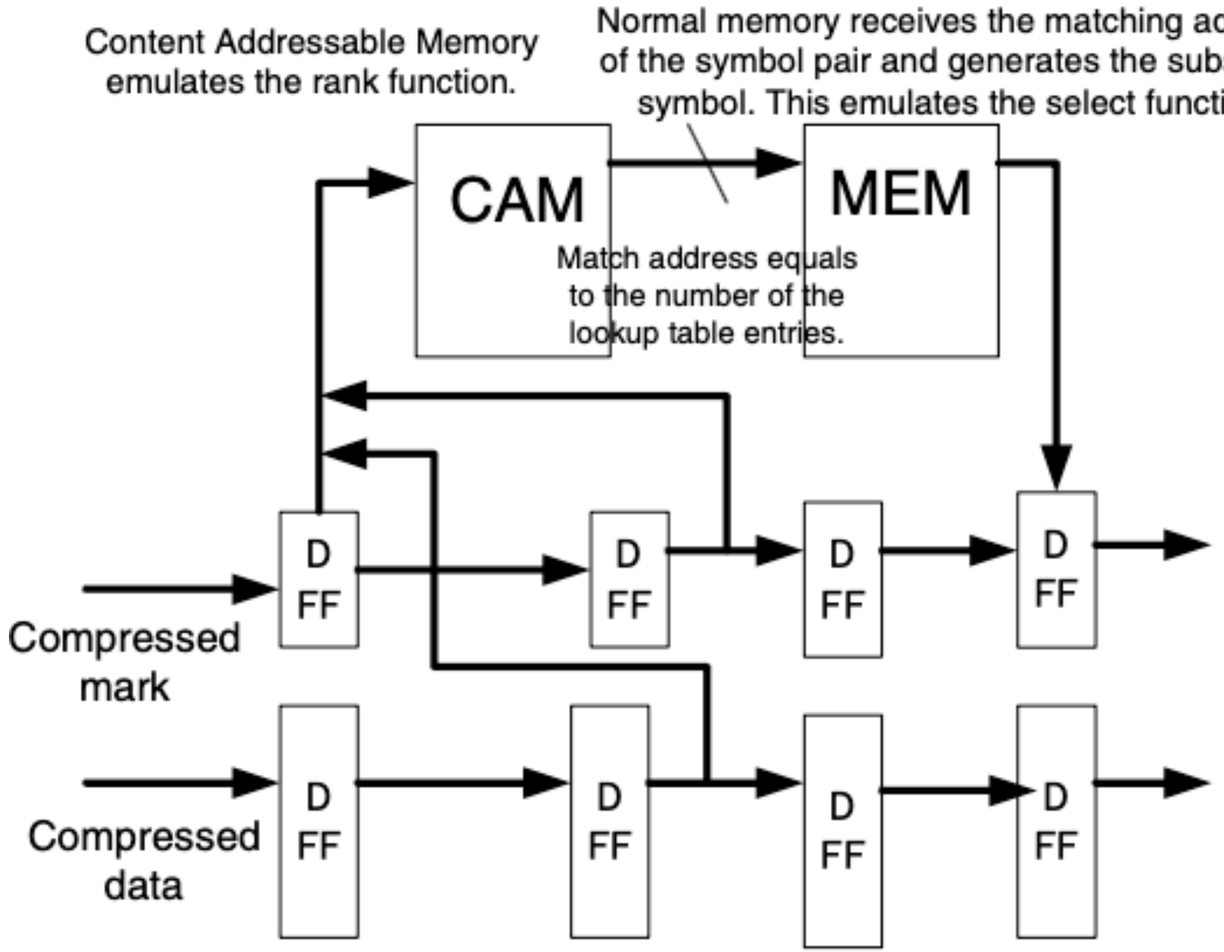


Figure 6. Organization of compresssion part of LCA-SLT.

FPGAでの実装
ソースサイズ

・CAMの必要なビット数

Compression

（L_{symbol} ×2＋L_{mark}×2）×D＋ $log2D$×D bitsが必要

Decompression

（ L_{symbol}＋L_{mark} ）×D＋$log2D$ ×D bitsが必要

L_{symbol}記号の長さ L_{mark} 圧縮ビット D エントリの数

FPGAでの実装
ソースサイズ

・MEMの必要なビット数

Compression

L_{symbol} × D

Decompression

（L_{symbol}×2 L_{mark}×2）×Dビット

L_{symbol}記号の長さ L_{mark} 圧縮ビット D エントリの数

FPGAでの実装
ソースサイズ

・CAM MEMの実装方法

BRAM(ブロックメモリ)

SRL(リセットラッチ)

の二つの実装方法がある。どちらとも実験してみた。

FPGAでの実装
ソースサイズ

Table I
RESOURCE USAGE AND MAXIMUM FREQUENCY OF AN LCA-SLT MODULE IN THE CASE OF 8 BIT SYMBOL FOR FOUR STEPS.

128 entries	# of slice registers	# of slice LUTs	# of BRAMs	Max Freq.
Using SRL-based CAM	141	2224	1	93 MHz
Using BRAM-based CAM	380	738	21	93 MHz
256 entries	# of slice registers	# of slice LUTs	# of BRAMs	Max Freq.
Using SRL-based CAM	144	4135	1	75 MHz
Using BRAM based CAM	638	1546	41	101 MHz
512 entries	# of slice registers	# of slice LUTs	# of BRAMs	Max Freq.
Using SRL-based CAM	152	8128	1	51 MHz
Using BRAM-based CAM	1152	2567	81	81 MHz

FPGAでの実装

実装の結果

・LUTのエントリを制限すると圧縮するとリソースサイズも小さくなる。

・データごとに非決定的な処理を行わなくても済む

・LUTをデータに含めずに圧縮Moduleに含めることができている。