10/4　進捗報告

佐藤孝嗣

目的:
Verilogを使いベクトル行列積を計算する回路を設計する

- “ベクトル行列積の計算の高速化”についてサーベイ
  - 演算回路の段階の高速化については見つからず
- “内積計算の高速化”についてサーベイ
  - ベクトル行列積と同様の理由で断念

加算や乗算の段階での効率化を行う
今週は加算器について

加算器

- リプルキャリーアダー(RCA)

  桁上げの計算を逐次実行するため遅延が大きい

- キャリールックアヘッドアダー(CLA)

  入力のbit数が増えると上位の桁へのファンインも増えるため、遅延が生じる

- パラレルプリフィックスアダー

  CLAのキャリー計算部分を改良し、より速い計算が可能

他にも加算器の種類があるが、

CLAより遅いというデータがあった

パラレルプリフィックスアダーを採用

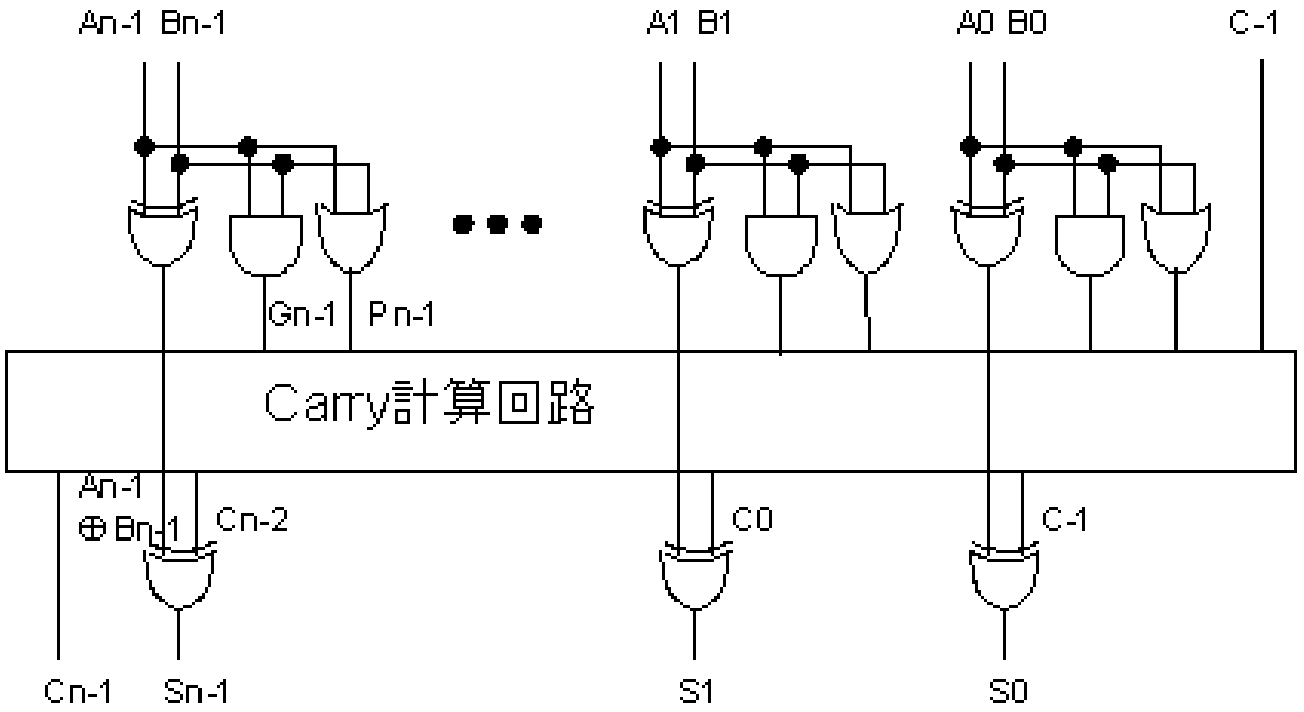
Parallel Prefix Adder

3ステップで計算

1.Pre-processing

2.Carry計算

3.Post-processing



G(generate)信号・・・この信号が1のとき桁上げが発生
P(propagate)信号・・・この信号が1のとき下位からの桁上げが上位に伝播

キャリー計算回路を変えることで性能が変化

Parallel Prefix Adder

- Pre-processingでの計算

  入力Ai，Biに対し

  S'i＝Ai XOR Bi

  Gi＝Ai×Bi　　Pi＝Ai + Bi

- Carry計算回路での計算

  i番目からk番目のbitのG信号をG[i:k]、P信号をP[i:k]とすると

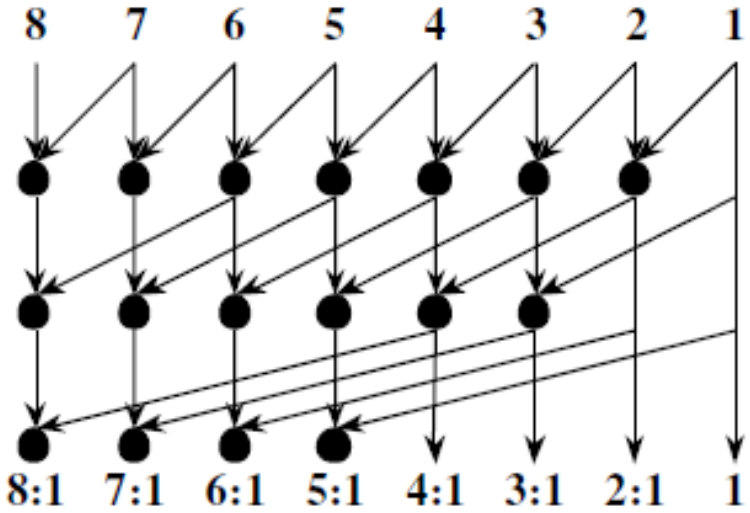
$$G[i:k] = \begin{cases} G[i:j] + P[i:j]G[j-1:k] & (i \neq k) \\ Gi & (i=k) \end{cases}$$

$$P[i:k] = \begin{cases} P[i:j]P[j-1:k] & (i \neq k) \\ Pi & (i=k) \end{cases}$$

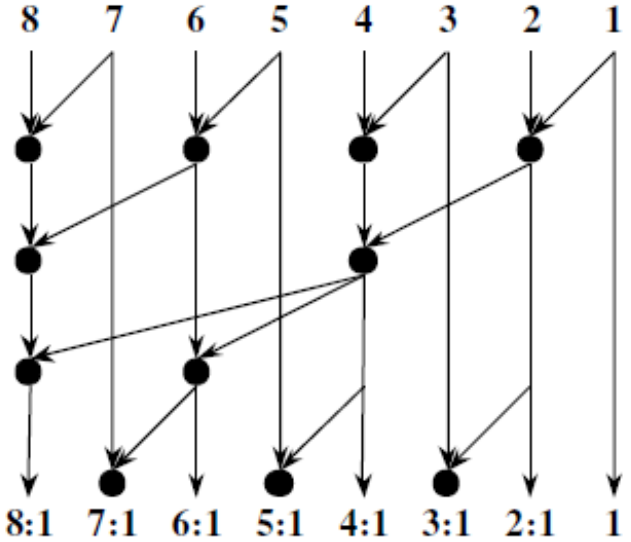
この計算を各ノードで行う

Parallel Prefix Adder

Carry計算回路の例



(a) Kogge-Stone prefix adder



(b) Brent-Kung prefix adder

- Post-processingでの計算

  Ci＝G[i:0]

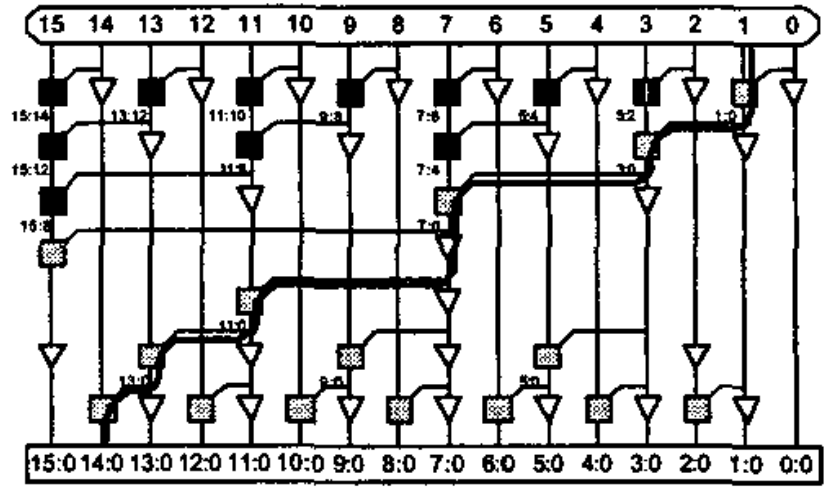
  Si＝S'i XOR Ci-1

主なParallel Prefix加算器

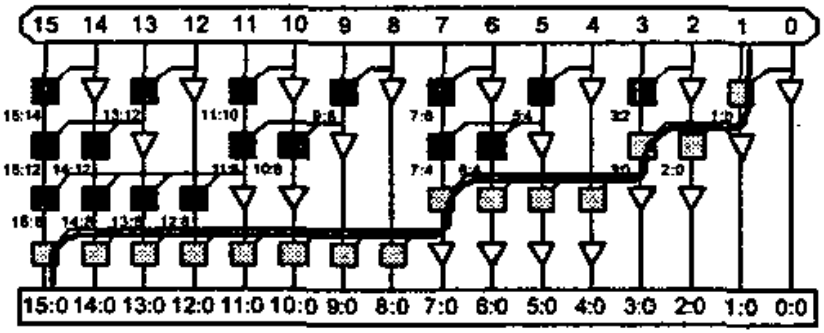
- Sklansky Adder
- Ladner-Fischer Adder
- Kogge-Stone Adder
- Han-Carlson Adder
- Brent-Kung Adder
- Knowles Adder

Carry計算回路の回路図(1)

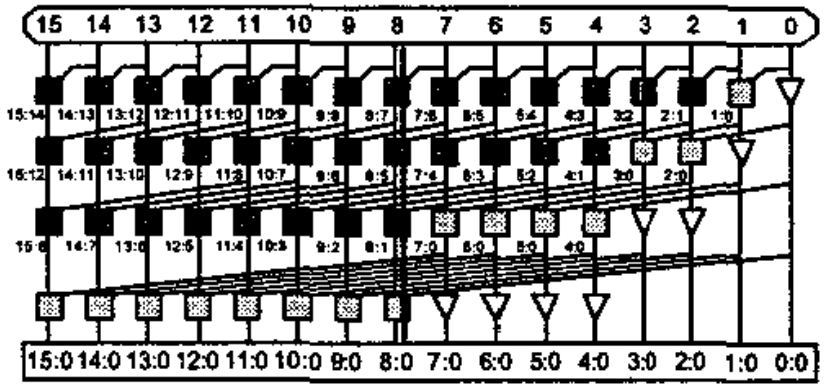
(a) Brent-Kung



(b) Sklansky

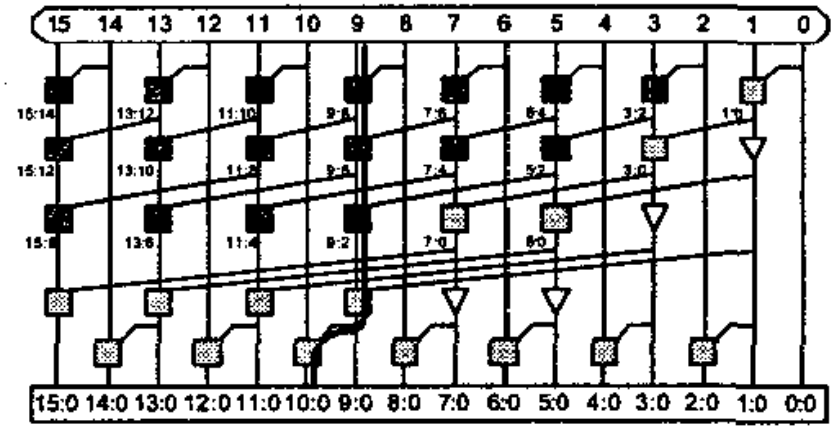


(c) Kogge-Stone

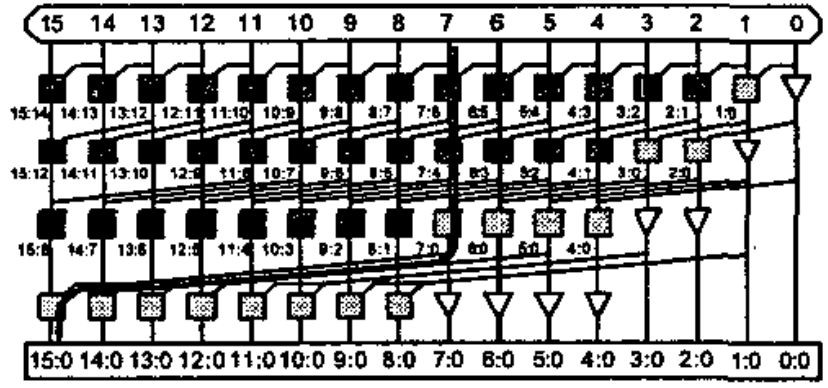


Carry計算回路の回路図(2)

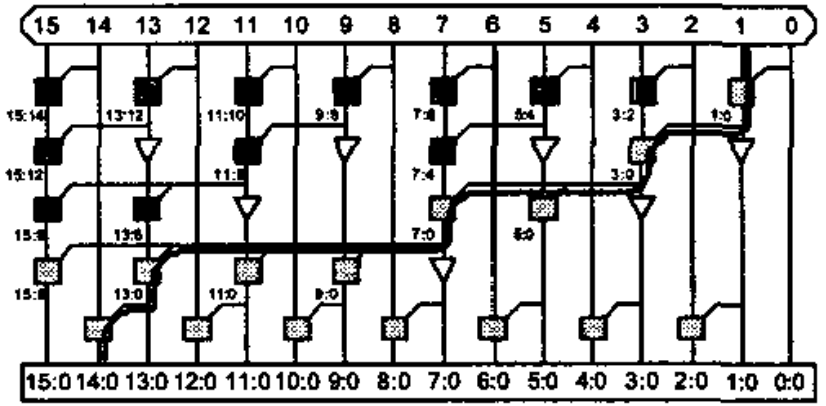
(d) Han-Carlson



(e) Knowles [2,1,1,1]



(f) Ladner-Fischer



各加算器のCarry計算回路の比較

Nを入力のbit数とする

- LogicLevel・・・通過するノードの最大数　(L=$\log_2 N$)
- MaxFanout・・・ひとつのノードの出力から枝分かれする最大の数
- Tracks・・・ノード間を通る導線の最大数

Architecture	Classification	Logic Levels	Max Fanout	Tracks
Brent-Kung	(L-1, 0, 0)	$L + (L - 1)$	2	1
Sklansky	(0, L-1, 0)	L	$N/2 + 1$	1
Kogge-Stone	(0, 0, L-1)	L	2	$N/2$
Han-Carlson	(1, 0, L-2)	$L + 1$	2	$N/4$
Knowles [2,1,...,1]	(0, 1, L-2)	L	3	$N/4$
Ladner-Fischer	(1, L-2, 0)	$L + 1$	$N/4 + 1$	1
(1, 1, 1)	(1, 1, L-3)	$L + 1$	3	$N/8$

Table 1. Comparison of parallel prefix network architectures

加算器の比較

	$N = 16$	$N = 32$	$N = 64$	$N = 128$
Brent-Kung	10.4 / 9.9	13.7 / 13.0	18.1 / 17.4	24.9 / 24.2
Sklansky	13.0 / 8.8	21.6 / 12.4	38.2 / 18.3	70.8 / 28.2
Kogge-Stone	9.4 / 7.4	12.4 / 10.0	17.0 / 14.1	24.8 / 21.5
Han-Carlson	9.9 / 7.7	12.1 / 9.4	15.1 / 12.0	19.7 / 16.1
Knowles [2,1,...,1]	9.7 / 7.9	12.7 / 10.3	17.3 / 14.5	25.1 / 21.8
Ladner-Fischer	10.6 / 8.4	15.2 / 10.8	23.8 / 14.5	40.4 / 20.3
(1, 1, 1)	10.7 / 8.1	12.9 / 9.8	15.9 / 12.4	20.5 / 16.5

Table 2. Adder delays: w=0.5; inverting static CMOS / footed domino

入力のbit数が
16bitではKogge-Stoneが最も速く
32bit以上ではHan-Carlsonが最も速い

今週の予定

- Optisystemのチュートリアルを実施
  - まず、用意されているチュートリアルを実施
- 並行してVerilogで加算器の設計
- 乗算器についてサーベイ