5E－4　高速乗算回路の構成法

佐藤証　高野光司
日本アイ・ビー・エム株式会社東京基礎研究所

1. はじめに

コンシューマープロダクトによる安全な e ビジネスの展開には，高速で小型の暗号回路技術が不可欠である．公開鍵暗号方式は RSA から楕円暗号へ，共通鍵暗号方式は DES から AES [1]へと移りつつあるが，それらの回路実装の多くにおいて乗算器の速度がボトルネックとなる．そこで筆者らは半導体デバイスのパフォーマンスを最大限に引き出すカスタムレイアウトによる暗号 LSI の開発・方式提案[1)-3)]を行ってきた．しかし様々な製品への組み込みなどアプリケーションを広げるとき，カスタムレイアウトではなく，ゲートアレイやスタンダードセルをターゲットとした IP コア化が有利である．そこで本論文では，高速乗算器をゲートアレイやスタンダードセルライブラリによって構成する手法を提案する．

2. 高速加算器

組み合わせ回路による乗算器はで，部分積加算を Wallace 木によるキャリー保存加算方式で行った後，最終段の桁上げ伝播をキャリー先見加算器で実行する方式が一般的である．本手法ではキャリー先見加算器の代わりに図 1，2 のキャリースキップ加算器[1)-3)]を用いる．図 1 の加算器の特徴は全加算器ブロック(FA)内の信号伝播遅延と，各ブロック上をスキップするキャリー *Ci* の遅延を揃えて，無駄な待ち時間が生じないようにしている点である．この加算器を階層化してキャリーを 2 重に飛ばしたのが図 2 の 2 段キャリースキップ加算器であり，こちらも加算器ブロック上を多重に飛ぶキャリーの伝播遅延がそろうように配慮している．

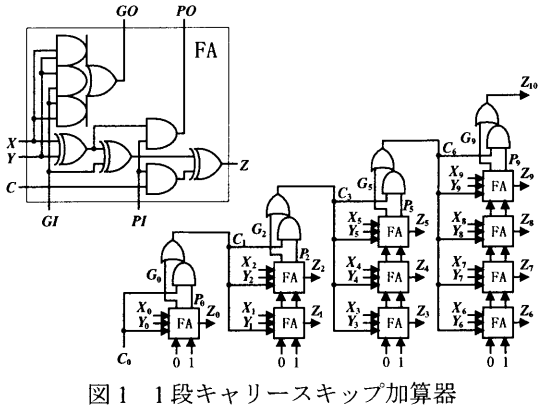

図 1　1 段キャリースキップ加算器

A Design Methodology of High-Speed Multiplier
Akashi SATOH, Kohji TAKANO
Tokyo Research Laboratory, IBM Japan Ltd.
1623-14, Shimotsuruma, Yamato-shi, Kanagawa 242-8502, Japan

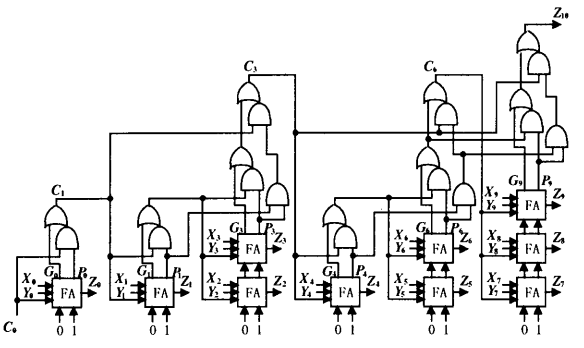

図 2　2 段キャリースキップ加算器

3. 高速乗算器

図 3 は部分積をキャリー保存方式で加えていく Wallace 木と，その出力を受ける高速加算器を示している．FA は 3 入力 2 出力の全加算器，HA は 2 入力 2 出力の半加算器で，後段に位置し遅延が大きいものほど濃いパターンで塗られている．図の左上に示したように，この木の遅延は LSB から上に向かって大きく，また中心部から MSB に向かって小さい山型となる．なお図 4 以降の遅延特性ではこれを 180° 回転させ，左から右に向かって LSB→MSB，下から上に向かって遅延が増加する形としている．本手法ではこの Wallace 木の上に遅延特性に合わせて加算器を構成する．

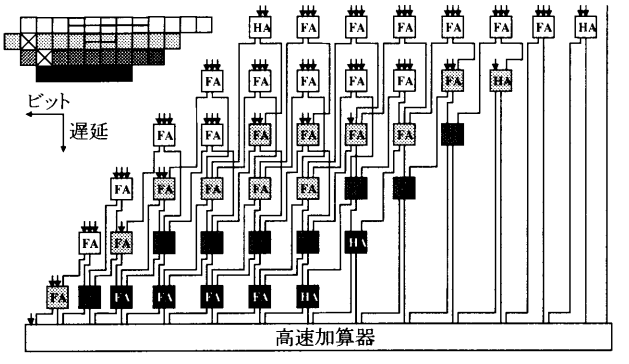

図 3　部分積加算における Wallace 木の遅延

Wallace 木において LSB からすぐの部分（図 4）はスロープが急，つまり木の遅延が大きく高速加算器の使用が無意味なのでリップルキャリー加算器を用いる．そしてスロープが緩やかになった bit 3 からキャリースキップ加算器を用いる．なお図で CG はキャリージェネレータで，図 1 の AND-OR ゲートのことであり，HA と CG は FA 一個と等価なのでこれで置き換えると，図 4(b)のように bit 0-4 の 5 ビットがリップルキャリー加算器となる．なお図 4 はクリティカルパスであるキャリ

一伝播のイメージを示したものなので，いくつかの信号が省略されている．

このアプローチでは，もしWallace木のスロープに急な凸凹があると，その上にきれいに加算器をのせることができず，無駄な信号待ちが生じてしまう．そこで，最大遅延である山の頂上を低くすることはもちろん，上り坂ができるだけなだらかになるようにWallace木を組むことがポイントとなる．しかしWallace木の遅延の値はFAやHAセルの遅延の整数倍とはならないので，どのように加算器を構成しても図5(a)(b)のようにキャリーまたはFA出力PのいずれかがセルCGで待たされることになる．そこでブロック上をスキップするキャリーの“遅延時間／ビット数”が小さくなるように，つまりキャリーができるだけ低く飛ぶように加算器ブロックの構成を調節する．

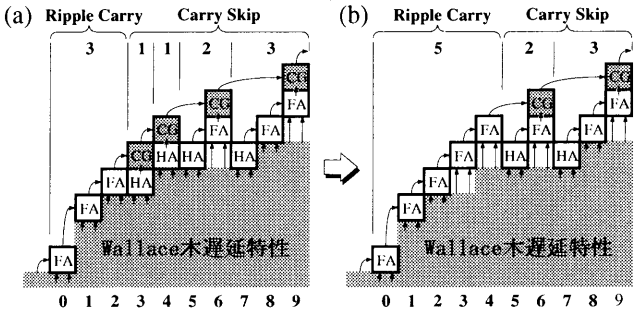

図4　遅延特性に応じた加算器の選択（上り坂）

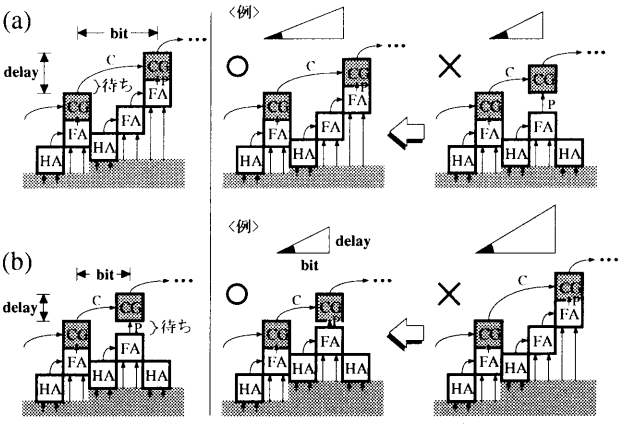

図5　加算器ブロック構成の最適化

また，Wallace木の下り坂上（MSB側）に1段キャリースキップ加算器を組むと，図5(a)のようにほとんどのFAセルが下からのキャリーを待つことになる．そこで本手法では(b)のように1段キャリースキップ加算器を階層化し，その上にさらにキャリーを飛ばす2段スキップ方式をとることで無駄な待ちを低減する．図6(b)は分かりやすいようにやや冗長な回路構成になっているが，MSB側では信号の伝播に十分な余裕があるので，Wallace木の組み方や加算器ブロックの分割方法にはそれほど注意を払う必要はない．

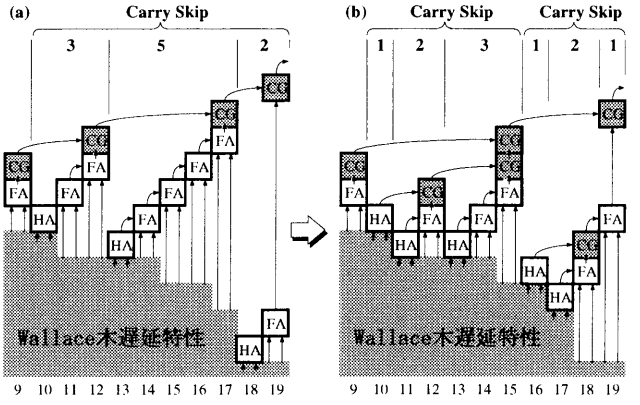

図6　遅延特性に応じた加算器の選択（下り坂）

図7に0.18 μm CMOSスタンダードセルライブラリを用いた本方式の32ビット乗算器の遅延特性を示す．加算器はbit 1-5をリップルキャリー，bit 6-30を1段スキップ，bit 31-63を2段スキップ方式で構成している．なだらかな山（Wallace木）の上に見える鋸歯状の部分は，各加算器ブロックの内部遅延を表し，その上をキャリーが飛ぶ様子がわかる．

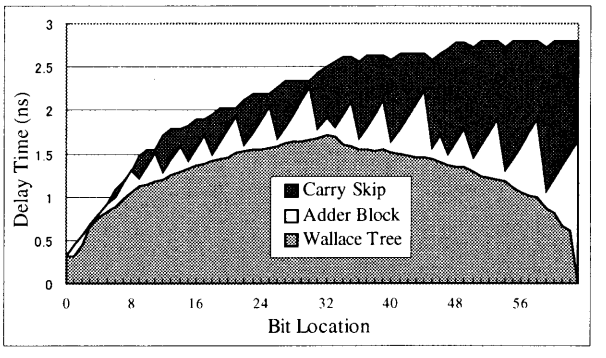

図7　実際の32ビット乗算器の遅延特性

4. むすび

乗算器の最終桁上げ加算において，Wallace木の遅延のスロープを急な上り坂，ゆるやかな上り坂，下り坂の3つに分け，それぞれに適した加算器の構成法を提案し，各ブロック間の信号遅延を合わせることの重要性について論じた．また本方式により処理時間2.8nsと高速な32ビット乗算器が構成できることを示した．

文献

1) Satoh et al.: “High-Speed MARS Hardware,” Third AES Candidate Conference, http://csrc.nist.gov/encryption/aes (2000).
2) 佐藤: “高速乗算器の一構成法”, 第61回情処全大, 5H-10 (1999).
3) Satoh et al.: “A High-Speed Small RSA Encryption LSI with Low Power Dissipation,” LCNS-1369, pp.174-187 (1998).