

冗長な零乗算を省略可能な非同期式直列乗算器の一構成法 An Asynchronous Serial Multiplier without Redundant Calculation for a Zero Operand

永田 将大¹⁾ 近藤 真史²⁾ 池田 大地³⁾
Masahiro NAGATA Masafumi KONDO Daichi IKEDA
茅野 功²⁾ 横川 智教¹⁾ 佐藤 洋一郎¹⁾
Isao KAYANO Tomoyuki YOKOGAWA Yoichiro SATO

1 まえがき

近年、DSP を搭載したデジタル補聴器が普及しているが、その電池寿命は数日程度に留まっている [1]。この問題に対して、単一の全加算器 (FA) で演算を行う直列乗算器にクロックを用いない非同期式回路を併用することで小面積かつ低消費電力な乗算器が提案されている [2]。しかし、非同期式回路の制御に係る休止相に起因して演算時間が増大し、これを用いたフィルタでは十分な周波数特性を得ることが困難である。そこで本研究では、さらに対の FA を並列に配置し、二対の FA を相補的に制御することにより休止相を隠蔽可能な乗算器を提案する。さらに、乗数ビットが零の場合に生じる冗長な演算を省略し、これに係る演算時間の削減を図る。

2 補聴器用 DSP を対象とした直列乗算器

乗算器は、被乗数 a_i ($0 \leq i < n$) と乗数 b_j ($0 \leq j < n$) を入力として、その積 p_k ($0 \leq k < 2n$) を求める回路であり、その動作は部分積 $a_i \cdot b_j$ (は論理積を表す) を網羅的に足し合わせるにより実現される。一般的には、FA を格子状に配置した並列乗算器 (P-MUL) として実装される。一方、直列乗算器ではクロックに同期した一对の D-FF と FA で逐次的に演算を行うため、高速なクロックに起因して消費電力が増大する。この問題に対して、クロックを用いない非同期式直列乗算器 (A-MUL) が提案されている [2]。A-MUL では、その制御に 2 線 4 相方式を採用しており、1 ビットの信号 (d) を表すために 2 ビットの信号 (d_1, d_0) を用いる。論理 '0' を表す (0, 1) と論理 '1' を表す (1, 0) を稼働相、無効な信号を表す (0, 0) を休止相といい、各相を交互に繰り返すことにより、クロックに代わる動作タイミングを生成する。

A-MUL の構成を図 1 に示す ($n=4$)。dFA は 2 線符号化された FA であり、マラーの C 素子 (入力が一一致するとその値を出力し、それ以外は直前の出力を保持する回路) を介して、動作タイミング信号 (C_e) を生成する。ASM は C_e に基づいて各レジスタ (SRa-d) へのシフト信号 (ACLK, BCLK) を生成する非同期式順序回路である。Ctrl は "01...1" を初期値とするリングカウンタとして構成され、各レジスタのシフトタイミング (SCLK) で制御信号 (CS) を生成する。CS='1' ならば $a_0 \sim a_3 \cdot b_j$ に関する部分積の足し合わせ結果が SRC へ、CS='0' ならばその時点で確定している SRC の最下位ビットの値が SRd へそれぞれ格納される。これを n 回繰り返すことにより、SRC および SRd にそれぞれ積 p の上位ビットおよび下位ビットが格納される。また、乗算の完了は、予め SRd を信号列 "10...0" で初期化し、最下位ビットに '1' が出力されたタイミングとして検出する。

- 1) 岡山県立大学, Okayama Prefectural Univ.
- 2) 川崎医療福祉大学, Kawasaki Univ. of Medical Welfare
- 3) 山陽電研株式会社, Sanyo Denken Co., Ltd.

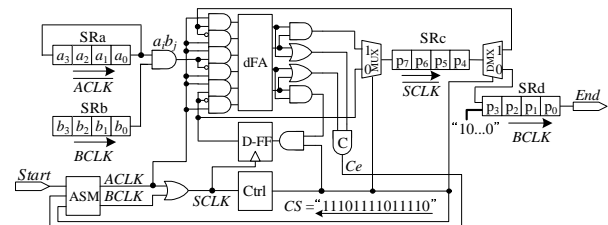


図 1: 非同期式直列乗算器 (A-MUL)

3 休止相を隠蔽可能な非同期式直列乗算器

3.1 設計方針

A-MUL の制御は、稼働相と休止相を交互に繰り返して行われるが、演算自体は稼働相のみで行われるため、既存の直列乗算器に比して演算時間が倍増することになる。この問題に対して本研究では、さらに対の dFA と D-FF を並列に配置し、これら二対の dFA を相補的に制御、すなわち稼働相と休止相を交互に処理する方針を採用。この形態によれば、実効的に稼働相が常に処理されるため、休止相に係る演算時間の劣化を隠蔽できる。

上述の相補制御を実現するためには、演算データを奇数ビット目と偶数ビット目に分割した上で、各 dFA へ交互に入力する必要がある。データの分割については、単に被乗数を格納する SRa の奇数番目と偶数番目の D-FF をカスケード接続することで実現できる。一方、 b_j に係る部分積の足し合わせは、桁上げを含めると $n+1$ の奇数回行う必要があり、単に分割されたレジスタのデータを dFA へ交互に入力するだけではデータの整合性を保つことができない。したがって、 b_j に係る部分積の足し合わせが完了する度に、入力先となる dFA を動的に切り換える必要がある。この制御について本研究では、dFA への入力をマルチプレクサ (MUX) により選択する形態を前提として、その選択信号を T-FF で生成する方針を採用。特に、T-FF のトリガ信号に b_j のシフト信号 (BCLK) を用いることで、 b_j がシフトするタイミングで入力先となる dFA の切り換えを実現する。

3.2 回路構成

提案する非同期式直列乗算器 (D-MUL) の構成を図 2 に示し、各構成要素の機能と意味を以下に記す。各構成要素の添え字 e および o はそれぞれ偶数および奇数ビット目のデータに対応することを表す。

SRa および SRb: 被乗数 (a) および乗数 (b) を格納するシフトレジスタ。

SRc および SRd: 積 p の上位ビット (c) および下位ビット (d) を格納するシフトレジスタ。なお、SRc は演算途中の部分積の保持にも利用される。

SI/SO は直列の入力/出力、**PI/PO** は並列の入力/出力を表し、 WEn をトリガとして PI の値をロードする。

dFA: 2 線符号化を施した全加算器。

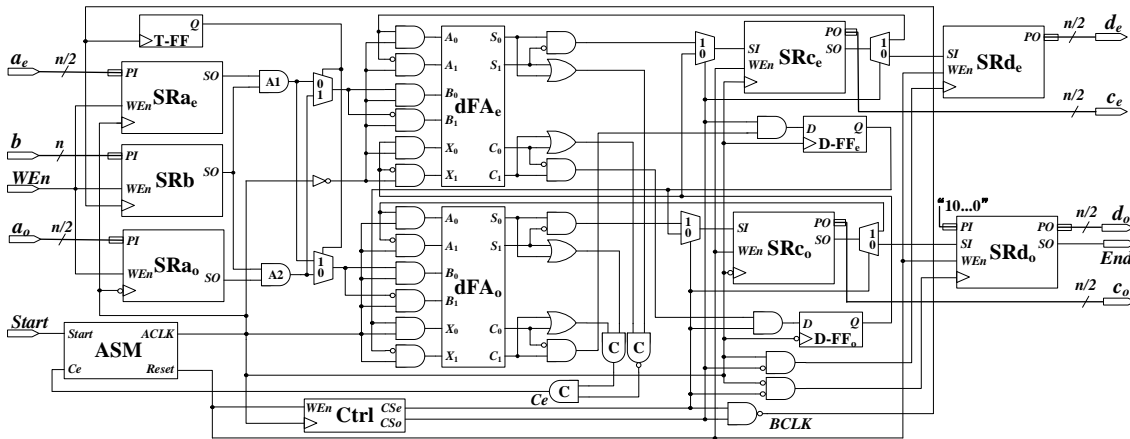


図2: 休止相を隠蔽可能な非同期式直列乗算器 (D-MUL)

D-FF: 桁上げを保持するための記憶回路。

Ctrl: 制御信号 CS_e および CS_o を生成する制御回路。 $CS_e = "1...101...1"$, $CS_o = "01...1"$ からなる $n+1$ ビットの信号列を繰り返し生成するリングカウンタで構成される。

ASM: $ACLK$ の生成を担う非同期式順序回路。

3.3 動作手順

手順1: データの格納: WE_n のアサートにより、被乗数 a が SRa_e と SRa_o , 乗数 b が SRb にそれぞれ格納される。また, SRc_e , SRc_o および SRd_e には "00...00", SRd_o には "10...00" がそれぞれ初期値として格納される。

手順2: 制御信号の生成: WE_n のネゲート後, 演算の開始を指示する信号 $Start$ がアサートされると, ASM により $ACLK$ がアサートされる。それに伴い $Ctrl$ から CS_e および CS_o がアサートされる。

手順3: 部分積の加算処理: 上段の dFA_e には $ACLK$ が入力されているため, $ACLK$ がネゲートされている期間 dFA_e は稼働相となる。 SRa_e および SRb から出力された a_i と b_j はゲート A1 を経て部分積 $a_i \cdot b_j$ となり, MUX を介して dFA_e にアサートされて加算が行われる。そして, その演算結果は $ACLK$ のアサートにより SRc_e にシフトされる。また, この時 dFA_e は休止相に移り, dFA_o にて稼働相が開始される。この手順を $n/2$ 回繰り返すことで b_j に関する部分積が全て SRc に格納される。

手順4: 演算結果の格納: $CS_e = '0'$ により, 演算結果として確定している $SRc_e(0)$ は DMX を介して, $SRd_e(n/2-1)$ にシフトされる。また同時に, D-FF_e に格納されていた桁上げが $SRc_e(n/2-1)$ に格納される。同様に $CS_o = '0'$ で $SRc_o(0)$ が $SRd_o(n/2-1)$ にシフトされるが, $SRc_o(n/2-1)$ には D-FF_e に保持されている桁上げが格納される点に注意する。

手順5: 演算処理の終了: 手順3, 4 を n 回繰り返すと, SRc と SRd に演算結果が格納される。それに伴い, SRd_o の初期値として最上位ビットに格納していた信号 '1' が演算の終了を表す信号 End としてアサートされる。

4 冗長な零乗算の省略による高性能化

逐次的な演算形態である直列乗算器においては, FA による加算回数の削減は, 単に演算時間を短縮するだけでなく低消費電力化の効果も期待できる。ここで, 乗数ビットが '0' であればその部分積は全て '0' であるため, これに係る冗長な演算は省略可能であり, D-MUL にその制御を導入することを考える。

冗長演算を省略するのは $b_j = '0'$, すなわち $SRb(0) = '0'$

が出力されたタイミングで, その部分積に係る演算を実行することなく SRb を連続シフトし, $SRc(0)$ を確定データとして直ちに $SRd(n/2-1)$ にシフトすればよい。具体的には, 各レジスタのシフトの起点となる信号 CS_o および CS_e は, 互いに $n/2$ ビット分だけ位相がずれたビット列であるため, $Ctrl$ のリングカウンタの値を $n/2$ ビット分シフトする必要がある。これはついで本研究では, カウンタ内の D-FF のプリセット・クリア信号を利用して, シフトを伴うことなくより簡便にカウンタの更新を行う。以上の制御により, $SRb(0) = '0'$ を条件として $b_j = '0'$ に係る CS_o および CS_e のビット列が省略され, b_{j+1} に係る部分積の演算を開始することができる。特に, CS_o および CS_e の省略に伴って $BCLK$ が連続生成されるため, 結局 dFA への入力を制御する T-FF は連続で反転してその出力は変化せず, データの整合性を保ったまま b_{j+1} に係る部分積を適切に dFA へ入力できる。

5 設計と評価

提案する D-MUL を Xilinx ISE14.7 により設計し, 同社製 FPGA Zynq XC7Z020 を対象としたタイミングシミュレーションを通じて所望の動作を確認した。続いて, 各種 16bit 乗算器を設計し, Xilinx XPA を用いて消費電力の解析を行った。ランダムなビット列 5000 パターンについて評価を行った結果, D-MUL の消費電力は, P-MUL に比して約 48%, A-MUL に比して約 55% 削減できることを確認した。さらに, A-MUL に対しては平均演算時間を約 32% 削減可能であり, 休止相の隠蔽と零乗算を省略することの有効性も確認できる。

6 あとがき

本研究では, デジタル補聴器用 DSP への応用を前提として, 休止相に係る無効な演算を隠蔽し, かつ冗長な零乗算を省略可能な非同期式直列乗算器を提案した。設計と評価の結果, 既存の乗算器に比して消費電力および演算時間の面での優位性を確認した。今後はこの制御方式を積和演算器などへ拡張する予定である。

謝辞 本研究の一部は, JSPS 科研費 19K20238 の支援を受けて実施されたものである。

参考文献

- [1] J. DiCristina, "Introduction to Hearing Aids and Important Design Considerations," Maxim integrated (2010)
- [2] 小林タ莉, 近藤真史, 他, "デジタル補聴器用 DSP を対象とした非同期式直列乗算器の一構成法," 第 68 回電気・情報関連学会中国支部連合大会, R17-19-12 (2017)