

ECC回路内蔵16ビットマイコンの設計

Design of 16 bit Microcomputer with ECC Circuit

姜珂 野平江梨 鈴木五郎
Ke Jiang Eri Nohira Goro Suzuki

北九州市立大学情報メディア工学科
Department of Information Science, University of Kitakyushu

1. はじめに

現在、携帯電話、デジタルカメラをはじめ、様々なデジタル家電製品の普及が著しくなっている。100nmを切るdeep sub-μを使った信号処理用LSIでは、(1)α線によるソフトエラーに対する高信頼性化[1]、(2)量込み演算の高速化、(3)低消費電力化、が重大な課題となっている。これらの課題を解決する16ビットマイコンの設計を行った[2]ので報告する。

2. アーキテクチャ

16ビットマイコンの基本的なアーキテクチャは次の通りである。

- register方式：汎用的な8種類のregister (R0～R7)
- 命令長：16bit 固定命令長（図1）

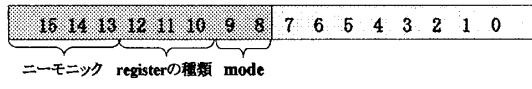


図1 1命令の構成

- アドレッシングモード：direct/immediate/indirect (register)

- CPI (cycle per instruction) : 8

1-4サイクル：命令フェッチ/命令デコード
5-8サイクル：データフェッチ/命令実行

- Machine cycle (clock cycle) : 5 nsec (200MHz)

- 命令セット：148種類

- ブロックダイアグラム(図2)

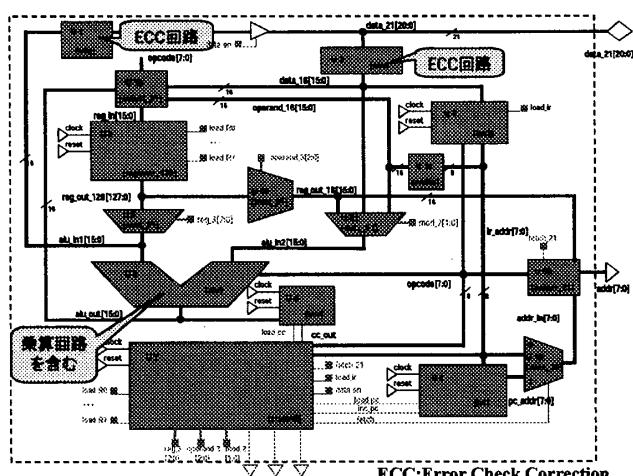


図2 ブロックダイアグラム

3. 高信頼性化

ソフトエラーの対策にはECC(Error Correction Code)回路を具備することとした。バースト誤りの訂正が可能な巡回符号方式[3]とし、割り算回路にはサイクル数の増加が避けられる組み合わせ回路方式を採用した。

ECC回路[4]は、誤り箇所の特定を行うために必要な5bitのcheck codeを生成するECG(Error Check-code Generator)回路(図2のモジュールU1)と、メモリから読み込んだデータに誤りがないかを計算し、誤りがあった場合にはその検出と訂正を行うCAC(Check And Correction)回路(図2のモジュールU2)から構成することにした。

巡回符号を扱うには、1,0を多項式で表現し、dataを $W_d(x)$ 、check dataを $W_c(x)$ 、メモリに書き込むデータを $W(x)$ とし、generator polynomialを $G(x)$ 、quotient polynomialを $Q(x)$ とする。ECGでは、

$$\overbrace{W_d(x) \quad W_c(x)}^{n \text{ bit}} \quad m \text{ bit} \quad n = 21, \quad m = 16$$

$$x^5 \cdot W_d(x)/G(x) = Q(x) + W_c(x)$$

$$(W(x) = x^5 \cdot W_d(x) + W_c(x) = G(x) \cdot Q(x))$$

$$G(x) = x^5 + x^2 + 1$$

のように、check dataを決定した。

メモリから読み込んだデータ $R(x)$ にエラー $e(x)$ が加わったと仮定する。CACでは、

$$\begin{aligned} R(x)/G(x) &= \{W(x) + e(x)\}/G(x) \\ &= W(x)/G(x) + e(x)/G(x) \\ &= e(x)/G(x) \end{aligned}$$

の式より余りが0なら誤りなしと判定し、0でない場合は誤りがあると判定できる。余りと誤り箇所とは一対一と対応することから、エラー箇所が同定でき、該当ビット(バーストエラーを含む)の修正を行う。

cyclic codeで必要となる割り算回路には組み合わせ論理を用いて、検討した。組み合わせ論理を用いた回路構造を図3に示す。

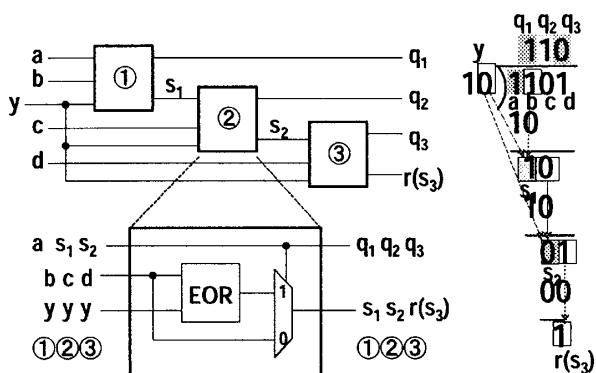


図3 組み合わせ論理を用いた割り算回路

$D(x)/Y(x)=Q(x) \cdot Y(x)+R(x)$ の割り算において

$$D(x) = x^4 + x^3 + 1, Y(x) = x + 1$$

のとき上の式のような計算が行われた。

図3からも分かるように、まずデータの高次ビットを取り出して商として立て、商が1のときはyと2番目の1ビットとexclusiveORをとる。また、商が0のときには演算を行わず通過させる。この処理はセレクタとexclusiveORを用いて、図3の回路のように実現した。

4. 乗算回路の高速化

高速化を実現するために、部分積生成に Booth-2 のアルゴリズム、部分積加算に Wallace tree を採用した[5]。8bit×8bit の乗算回路（図2のモジュール U5 alu の一部）の構造を図4に示す。

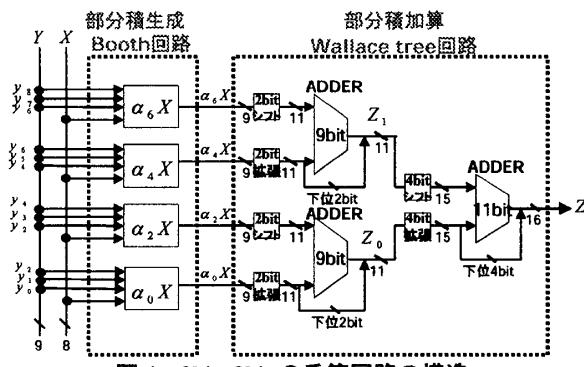


図4 8bit×8bit の乗算回路の構造

Wallace tree は加算器をツリー状に配置して加算の段数を減らす方法を採用した。

$$\begin{aligned} Z &= 2^6 \alpha_6 X + 2^4 \alpha_4 X + 2^2 \alpha_2 X + 2^0 \alpha_0 X \\ &= 2^4(2^2 \alpha_6 X + 2^0 \alpha_4 X) + (2^2 \alpha_2 X + 2^0 \alpha_0 X) \\ &= 2^4 Z_1 + 2^0 Z_0 \end{aligned}$$

ここで、Zは16ビットの積である。図2のように $2^2 \alpha_6 X + 2^0 \alpha_4 X$ と $2^2 \alpha_2 X + 2^0 \alpha_0 X$ の回路構造が等しくなる。そこで、加算の1段目で、 $2^2 \alpha_6 X + 2^0 \alpha_4 X$ と $2^2 \alpha_2 X + 2^0 \alpha_0 X$ の演算を行った。

$Z_1 = 2^2 \alpha_6 X + 2^0 \alpha_4 X$ と $Z_0 = 2^2 \alpha_2 X + 2^0 \alpha_0 X$ として、2段目で $2^4 Z_1 + 2^0 Z_0$ 行った。

今回設計した Wallace tree の構造は、加算すべき2個1組のデータのうちで乗数Yの下位ビットに対応する方のデータを符号拡張する構造を採用了。

5. 低消費電力化

F/Fの出力信号のトグルを抑制するだけでなく、クロック信号のトグルをも抑制し、大幅に消費電力を削減するゲートドックロック（gated clock）方式（図5に示す）[6]を採用了。

チップ外から供給されるクロック信号（clock）と制御論理で作り出されるロード信号（load_ir）とで論理積をとり、その結果（ld_clk）をF/Fに入力する直接のクロックとする。一つの命令を実行する8サイクルの中で1サイクルだけactive（信号値“1”）になるように、制御論理でロード信号（load_ir）を作り出すことにより、F/Fに直接印加されるクロックは1サイクル分だけ有効となる。そこで、F/F出力信号とクロック信号のトグルを最大1/8に低減することになった。F/Fを内蔵する汎用registerファイル（図2のモジュールU3）、プログラムカウンタ（図2のモジュールU8）、及びコンディションコードregister（図2のモジュールU6）に、本方式を採用し、低消費電力化を図った。

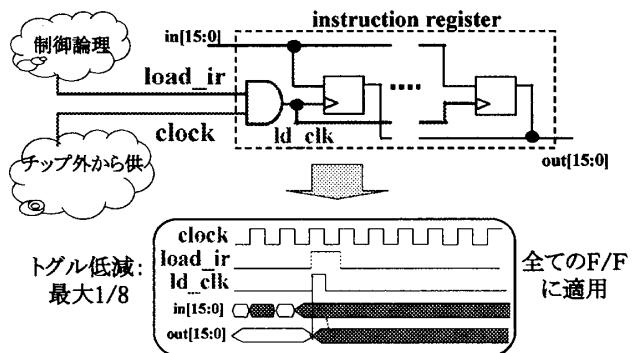


図5 消費電力の制御回路

6. 評価とまとめ

QUARTUS-II™を用いて、ALTERA-FPGA-cyclone™上へのインプリメンテーションを行った。本16bitマイコンに関して、Verilog-HDLを用いた記述を行ったが、その総ライン数約1963行であり、Model Sim™を用いた動作確認を行った。また論理合成後の総LE数は約953であった。

高信頼性化、量込み演算の高速化、及び低消費電力を特徴とする16bitマイコンの設計を行った。

参考文献

- [1] EDN "宇宙線によるシングルイベントエラーがLSIを襲う", EDN Japan, No.56, Oct.2005
- [2] 姫珂野平江梨 鈴木五郎, "ECC回路内蔵16ビットマイコンの設計", 通信学会誌大, C-12-8, 2006.
- [3] 笠原正雄, "誤り訂正符号と暗号の基礎数理", コロナ社, 2004
- [4] 今井秀樹, "情報理論", 昭晃堂, 1984.
- [5] 高木直史, "乗算回路のアルゴリズム—高速計算と回路の規則性の両立—", 情報処理学会論文誌 Vol.37, No.2, Feb.1996
- [6] 鈴木五郎, "システムLSI設計入門", コロナ社, 2003