
RL78/G10

R01AN3080JJ0101

乗除算プログラム CC-RL

Rev. 1.01

2016.10.05

要旨

本アプリケーションノートでは、RL78/G10での符号なしでの乗除算の実現方法を説明します。

16bit×16bit、16bit×16bit+32bit、32bit÷32bitの3種類の演算処理についてプログラムを示します。

対象デバイス

RL78/G10

本アプリケーションノートを他のマイコンへ適用する場合、そのマイコンの仕様にあわせて変更し、十分評価してください。

目次

1. 仕様	3
2. 動作確認条件	3
3. ソフトウェア説明	4
3.1 動作概要	4
3.2 乗算の考え方	4
3.3 除算の考え方	4
3.4 符号付きのデータへの対応	6
3.5 定数一覧	6
3.6 変数一覧	6
3.7 関数（サブルーチン）一覧	7
3.8 関数（サブルーチン）仕様	7
3.9 フローチャート	9
3.9.1 main 処理	9
3.9.2 16bit データ乗算処理	10
3.9.3 積和演算処理	11
3.9.4 32bit 除算処理	12
4. サンプルコード	13
5. 参考ドキュメント	13

1. 仕様

本アプリケーションノートでは、16bit×16bit、16bit×16bit+32bit、32bit÷32bit の3種類の演算処理サブルーチンを対象として、演算の考え方から実際のプログラムとその使い方を示します。

表 1.1に本アプリケーションノートが対象とする演算処理を示します。

表 1.1 対象とする演算処理

機能	動作概要
16bit×16bit	符号なしの16bit同士の乗算処理です
16bit×16bit+32bit	符号なしの16bitの積和演算処理です 32bitからオーバフローは検出しません。
32bit÷32bit	符号なしの32bitの除算処理です

2. 動作確認条件

本アプリケーションノートのサンプルコードは、下記の条件で動作を確認しています。

表 2.1 動作確認条件

項目	内容
使用マイコン	RL78/G10 (R5F10Y16)
動作周波数	<ul style="list-style-type: none"> ● 高速オンチップオシレータ (HOCO) クロック : 20MHz ● CPU/周辺ハードウェア・クロック : 20MHz
動作電圧	5.0V (2.9V~5.5Vで動作可能) SPOR 検出電圧 : 立ち上がり 2.90V (typ.)、立ち下がり 2.84V (typ.)
統合開発環境 (CS+)	ルネサス エレクトロニクス製 CS+ for CC V3.01.00
アセンブラ (CS+)	ルネサス エレクトロニクス製 CC-RL V1.01.00
統合開発環境 (e ² studio)	ルネサス エレクトロニクス製 e ² studio V4.0.2.008
アセンブラ (e ² studio)	ルネサス エレクトロニクス製 CC-RL V1.01.00
使用環境	RL78/G10 シミュレータ

3. ソフトウェア説明

3.1 動作概要

本アプリケーションノートの演算処理では、ショートダイレクト領域に配置された変数に格納されたデータに対して、指定された演算を行い、結果をショートダイレクト領域に配置された変数に設定します。

3.2 乗算の考え方

通常の筆算のやりかたと同じく、大きいデータを小さい桁に分けて計算を行っていきます。

この場合には位取り（桁）に注意が必要です。ここでは16進数の $A_1A_2A_3A_4 \times B_1B_2B_3B_4$ の計算を考えてみます。計算方法としては8ビットの乗算命令「MULU X命令」を使うことを考えて、 $A_1A_2 \times B_1B_2$ 、 $A_1A_2 \times B_3B_4$ 、 $A_3A_4 \times B_1B_2$ 、 $A_3A_4 \times B_3B_4$ の2桁（8ビット）ずつ4回に分けて計算（乗算）し、4回の計算結果を桁に注意して加算します。

この様子を図3.1に示します。このように、4回の計算結果をその重み（桁）を考慮して加算することで答えを得ることができます。ここでは、被乗数や乗数は4桁（16ビット）ですが、それ以上の場合にも同様に、2桁ずつの計算結果の桁を考慮して加算すれば計算することができます。

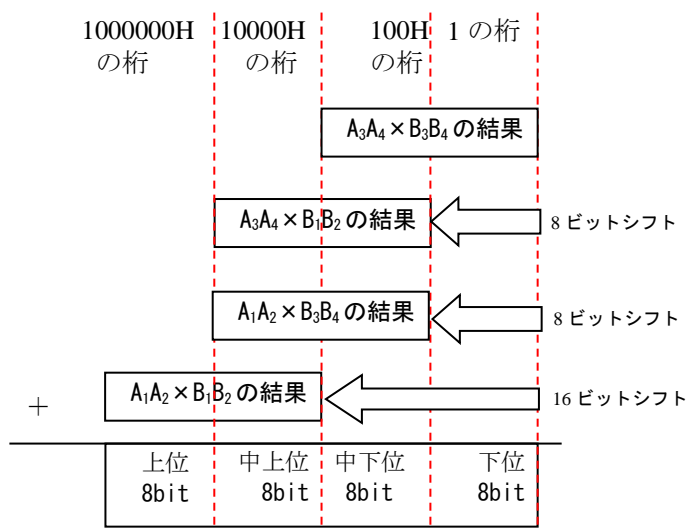


図 3.1 16bit×16bit の乗算の位取り

3.3 除算の考え方

RL78/G10には除算命令がありません。また、乗除積和演算回路も搭載されていません。従って、減算処理を繰り返すことで除算を実現する必要があります。

最も単純な実現方法は被除数から除数を何回引けたかで引けた回数を商とすることです。しかし、そのような単純な方法では処理時間が長くなってしまいます。被除数を1ビットずつシフトしながら、被除数のビット数分減算を繰り返していくことで処理時間を短縮するのが一般的です。16bit ÷ 16bit の除算を例にして、この方法を以下に説明します。

- ① 減算の繰り返しによる除算を行うためには、被除数の上位に除数と同じビット長の作業領域を準備して、0クリアしておきます（図3.2参照）。

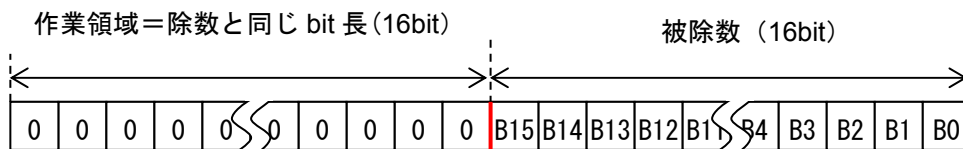


図 3.2 演算の前準備

- ② 減算処理は最初に被除数と作業領域を左に1ビットずつシフトして、減算の準備を行います。作業領域のLSBには被除数のMSBが入り、被除数のLSBは0になります(図33参照)。なお、被除数のLSBは処理と領域の効率化を図るために商領域として使用していきます。

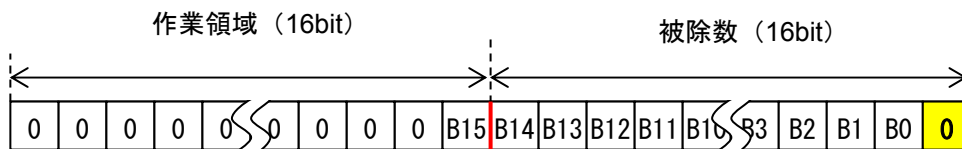


図 3.3 減算の準備

- ③ 作業領域から除数を引けるかどうかを確認し、引ければその回の商は1になり、引けなければ0になります。この商は被除数(領域)のLSBに格納します。引けた場合の余りは作業領域に格納されます(図3.4参照)。

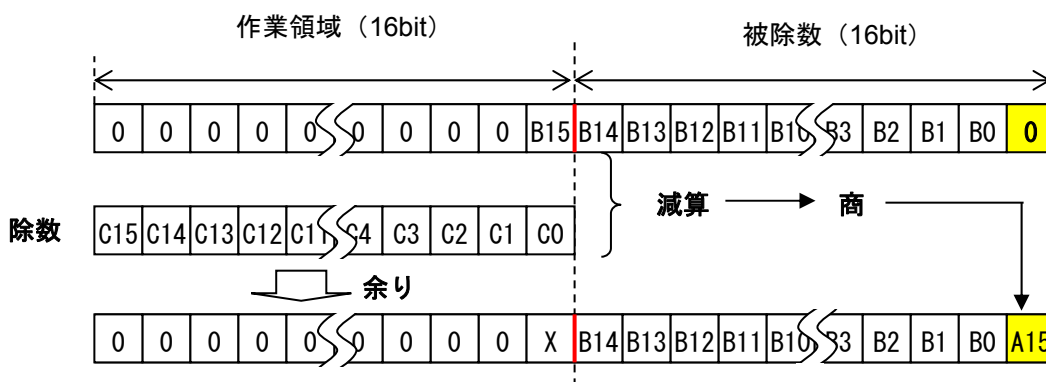


図 3.4 減算の実行

このような計算を行う場合に注意する必要があるのが、商の桁位置(ビット位置)です。減算そのものがビットをシフトしながら引いていくので、商には1回の減算ごとに1ビットシフトしながら毎回の結果を付け加えていくことになります。図3.4の場合を考えると、その結果の1ビット(商のビット)には2の15乗の重み(桁)があります。②と③がセットで1ビット分の減算が完了します。

- ④ 次の桁に②と③を行った結果の作業領域と被除数は図3.5に示すようになります。

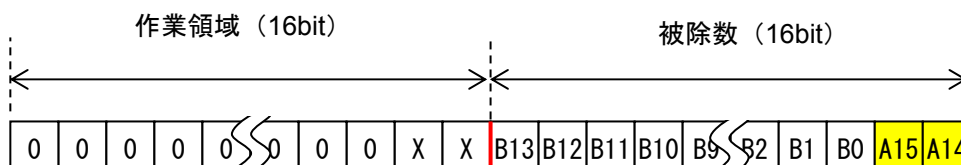


図 3.5 2回目の実行結果

- ⑤ 3回目のシフトと減算を行った結果の作業領域と被除数は図3.6に示すようになります。

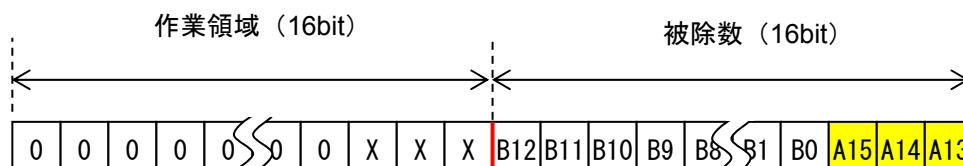


図 3.6 3 回目の実行結果

- ⑥ 同様のシフトと減算を合計で 16 回繰り返すことで、作業領域と被除数は図 3.7 に示すようになります。ここで、作業領域には剰余が格納され、被除数(領域)には商が格納されます。

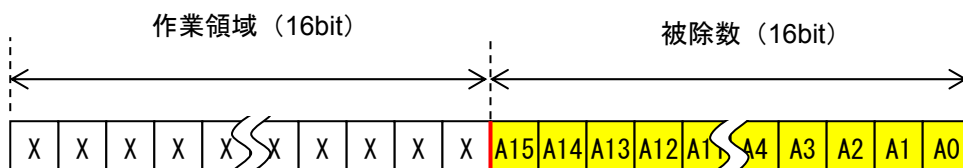


図 3.7 16 回目の実行結果

このように、被除数と商を同じ領域を共用することでシフト処理も共用でき、プログラムが短くでき、実行する命令が少なくなるので高速化も図れます。

以上の例は 16 ビットでしたが、32 ビットになっても同じ考え方ができます。

3.4 符号付きのデータへの対応

乗除算で、符号付きデータを扱うには、最初に符号を確認します。負数の場合には、その符号をフラグに記憶してから、データに対して 2 の補数をとることで、絶対値を得ます。得られた絶対値で演算処理を行い、結果に対して、2 つのデータの符号の組み合わせに対応して、必要なら結果の負数化処理を行います。

3.5 定数一覧

表 3.1 にサンプルコードで使用する定数を示します。

表 3.1 サンプルコードで使用する定数

定数名	設定値	内容
DNMULC	16/8	被乗数 (16bit) のバイト数
DNMUL	16/8	乗数 (16bit) のバイト数
DNRES	DNMUL + DNMULC	結果領域 (32bit) のバイト数
DNdivS	32/8	除数(32bit) のバイト数
DNdivD	32/8	被除数(32bit) のバイト数
DNQUO	DNdivD × 8	演算回数

3.6 変数一覧

表 3.2 にサンプルコードで使用する変数を示します。

表 3.2 サンプルコードで使用する変数

変数名	概要
RREG0	被乗数領域／除数領域下位
RREG1	乗数領域／除数領域上位
RREG2	乗算結果領域／積和結果加算領域／被除数及び商領域
RREG3	作業及び剰余領域

3.7 関数（サブルーチン）一覧

表 3.3 に関数一覧を示します。

表 3.3 関数(サブルーチン) 一覧

関数(サブルーチン) 名	概要
M16bitX16bit	16bit データの乗算（符号なし）
M16bitA32bit	16bit データの乗算を行い、その結果を 32bit データに加算（符号なし）
D32bit_32bitS	32bit データの除算（符号なし、除数チェックあり）
D32bit_32bitS2	32bit データの除算（符号なし、除数チェックなし）

3.8 関数（サブルーチン）仕様

サンプルコードの関数（サブルーチン）仕様を示します。

[関数名] M16bitX16bit

概要	16bit データの乗算（符号なし）	
説明	RREG0 に格納された被乗数に RREG1 に格納された乗数を掛けて、結果を RREG2 に格納します。	
引数	RREG0	被乗数（16bit）
	RREG1	乗数（16bit）
リターン値	RREG2	積（32bit）
備考	使用レジスタ	A,X,B,C,D,E,H,L
	使用スタック	4+4 バイト

[関数名] M16bitA32bit

概要	16bit データの乗算を行い、その結果を 32bit データに加算（符号なし）	
説明	RREG2 の値に RREG0 と RREG1 に格納されたデータの乗算結果を加算します。	
引数	RREG0	被乗数（16bit）
	RREG1	乗数（16bit）
	RREG2	被加数（32bit）
リターン値	RREG2	積和（32bit）
備考	使用レジスタ	A,X,B,C, D,E,H,L
	使用スタック	4+4 バイト

[関数名] D32bit_32bitS

概要	32bit データの除算（符号なし、除数チェックあり）	
説明	除数が0でないかチェックし、0ならエラーで戻ります。0でないなら、RREG2のデータをRREG0、RREG0のデータで除算を行い、商をRREG2に剰余をRREG3に格納して終了します。	
引数	RREG0	除数（下位 16bit）
	RREG1	除数（上位 16bit）
	RREG2	被除数（32bit）
リターン値	RREG2	商（32bit）
	RREG3	剰余（32bit）
	キャリーフラグ	0 : 正常終了 1 : エラー（除数が0）
備考	使用レジスタ	A,X,B,C,L
	使用スタック	4 バイト（CALL 命令分のみ）

[関数名] D32bit_32bitS2

概要	32bit データの除算（符号なし、除数チェックなし）	
説明	RREG2のデータをRREG0、RREG1のデータで除算を行い、商をRREG1に剰余をRREG2に格納して終了します。（D32bit_32bitSの除数チェックを省略）	
引数	RREG0	除数（下位 16bit）
	RREG1	除数（上位 16bit）
	RREG2	被除数（32bit）
リターン値	RREG2	商（32bit）
	RREG3	剰余（32bit）
	キャリーフラグ	0 : 正常終了
備考	使用レジスタ	A,X,B,C,L
	使用スタック	4 バイト（CALL 命令分のみ）

3.9 フローチャート

3.9.1 main 処理

図 3.8 に main 処理フローを示します。

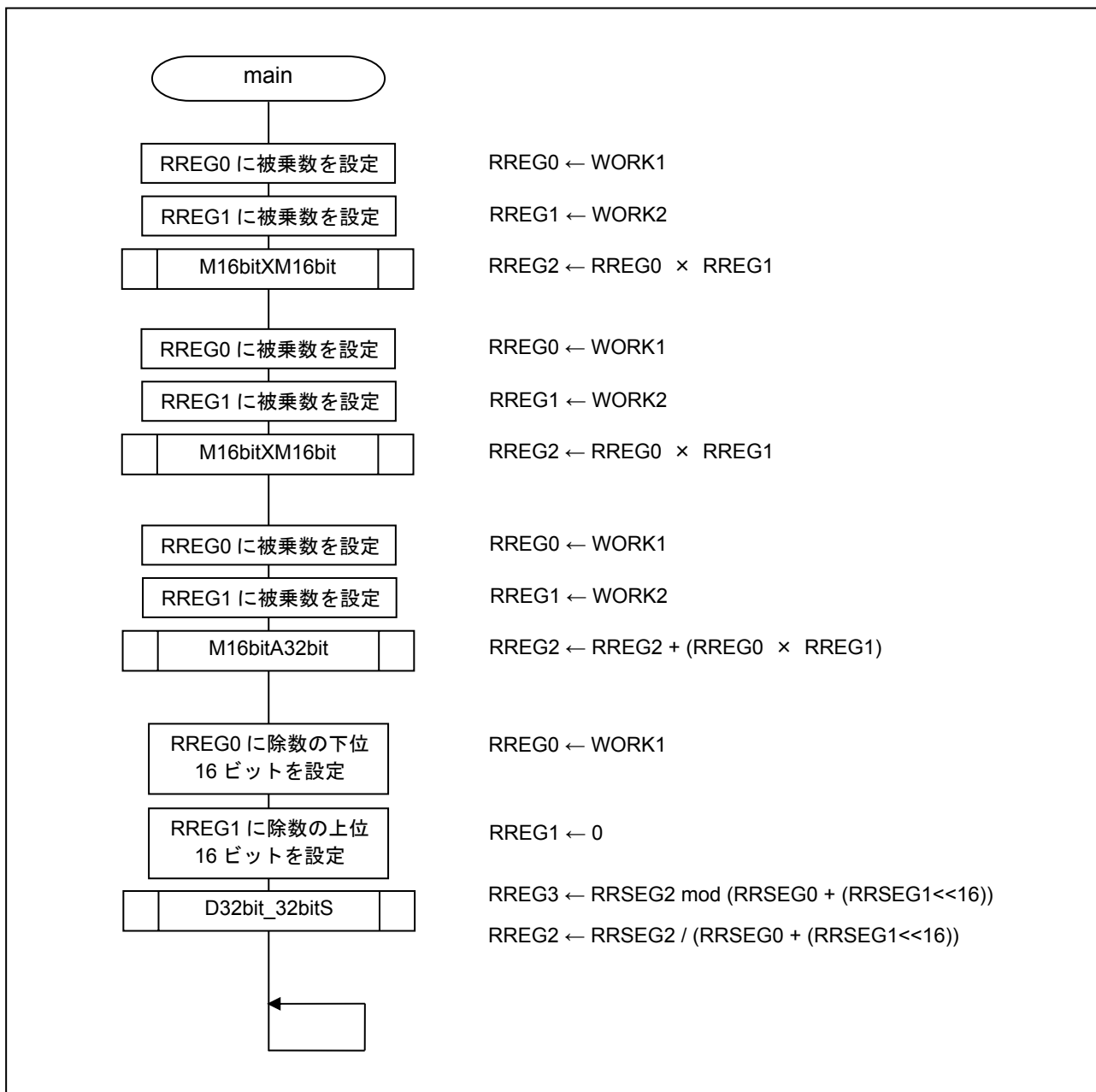


図 3.8 main 処理フロー

3.9.2 16bit データ乗算処理

図 3.9 に 16bit データ乗算処理フローを示します。

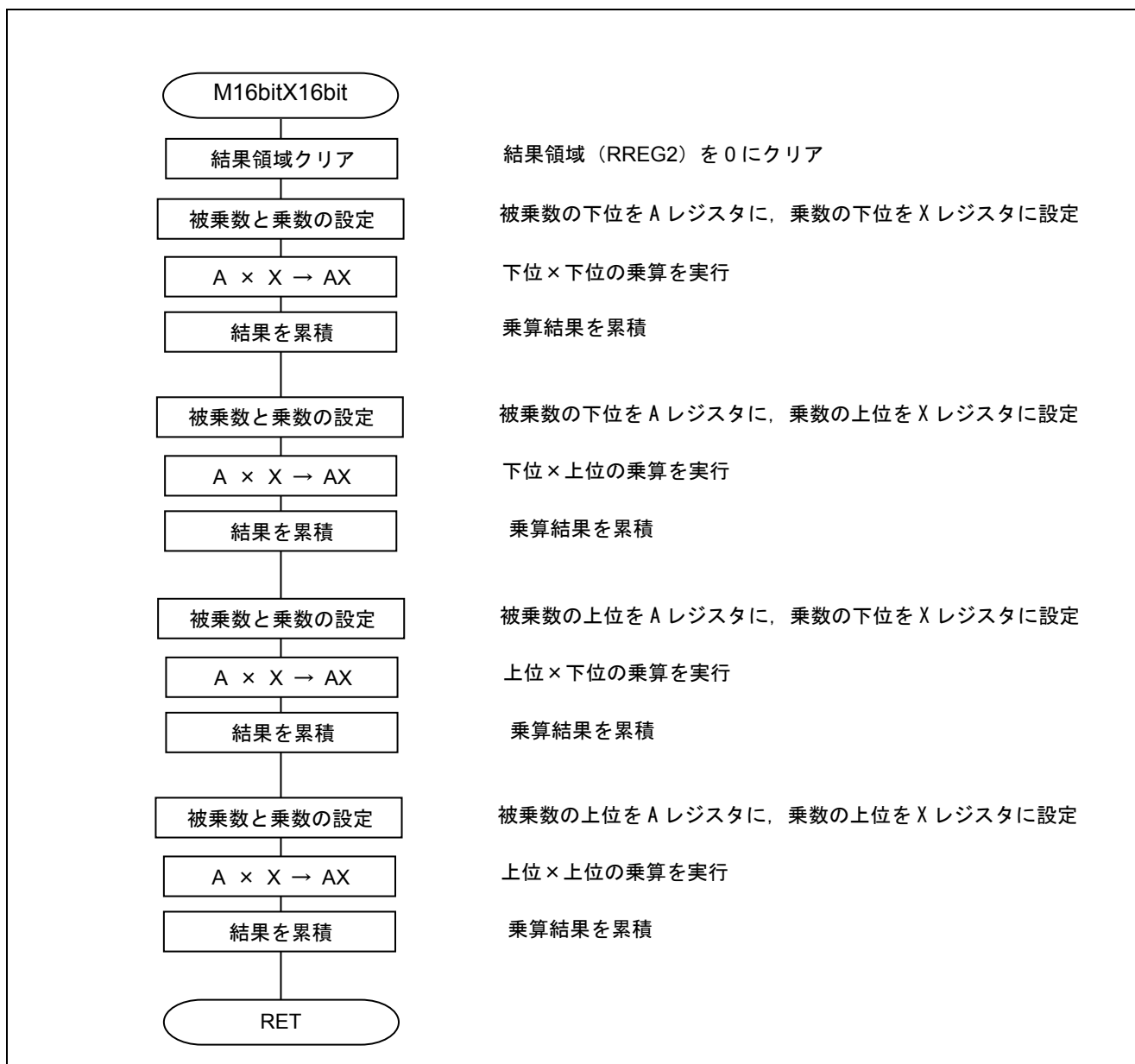


図 3.9 16bit 乗算処理フロー

3.9.3 積和演算処理

図 3.10 に積和演算処理フローを示します。

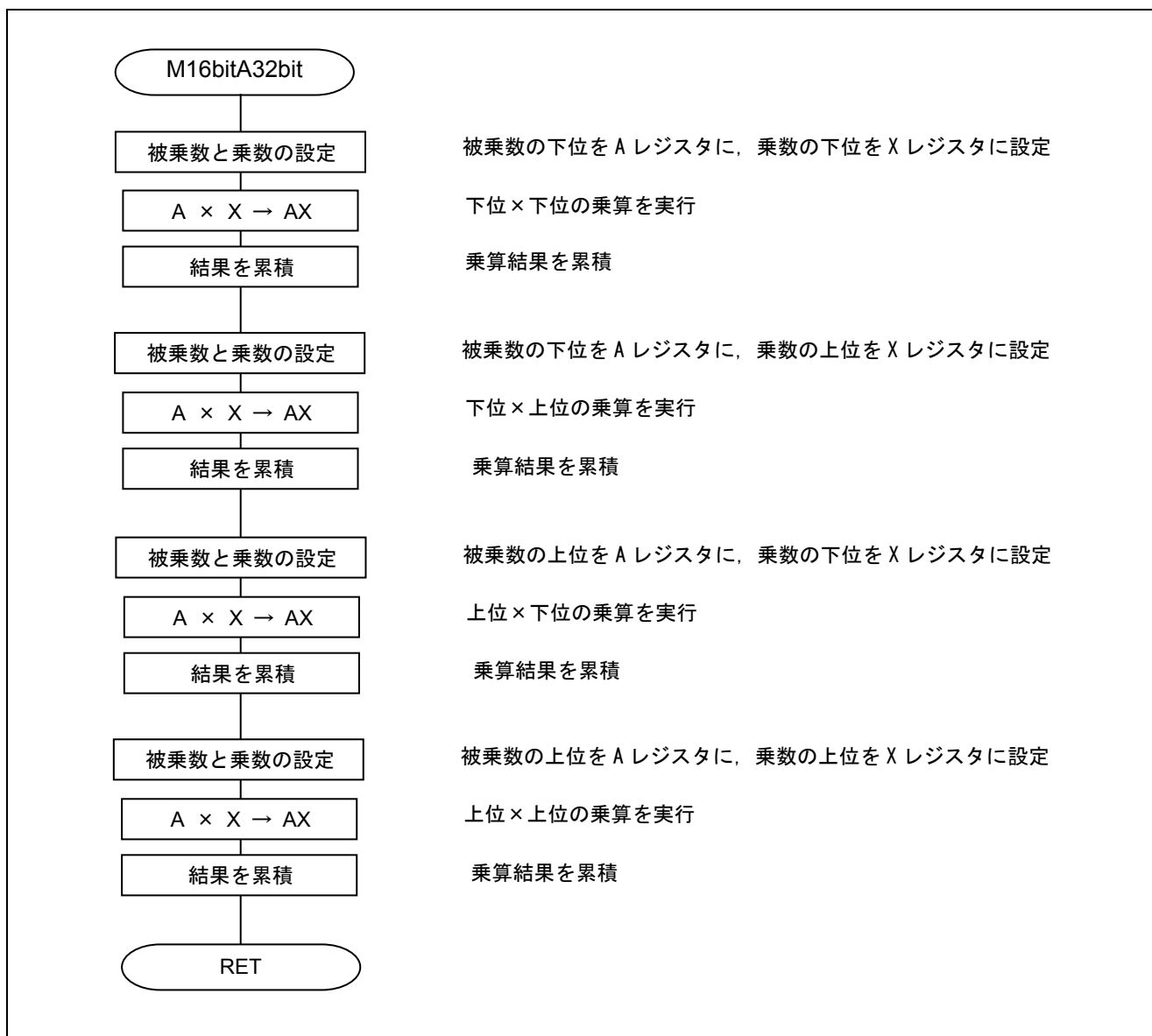


図 3.10 積和演算処理フロー

3.9.4 32bit 除算処理

図 3.11 に 32bit 除算処理フローを示します。

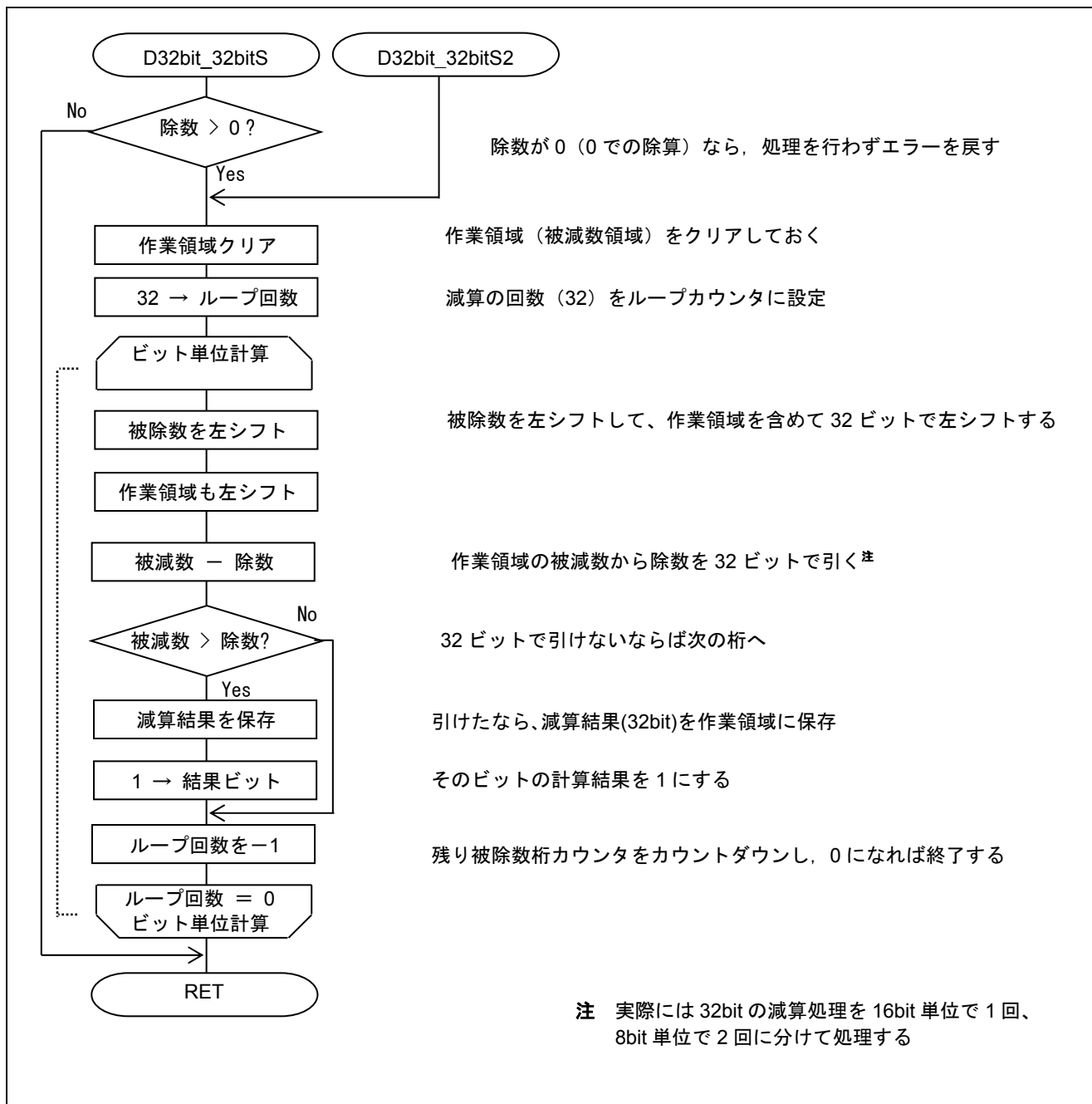


図 3.11 32bit 除算処理フロー

4. サンプルコード

サンプルコードは、ルネサス エレクトロニクスホームページから入手してください。

5. 参考ドキュメント

RL78/G10 ユーザーズマニュアル ハードウェア編 (R01UH0384J)

RL78 ファミリ ユーザーズマニュアル ソフトウェア編 (R01US0015J)

(最新版をルネサス エレクトロニクスホームページから入手してください。)

テクニカルアップデート/テクニカルニュース

(最新の情報をルネサス エレクトロニクスホームページから入手してください。)

ホームページとサポート窓口

ルネサス エレクトロニクスホームページ

<http://japan.renesas.com/>

お問合せ先

<http://japan.renesas.com/inquiry>

改訂記録	RL78/G10 乗除算プログラム CC-RL
------	----------------------------

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2015.11.27	—	初版発行
1.01	2016.10.05	-	誤記改訂

すべての商標および登録商標は、それぞれの所有者に帰属します。

製品ご使用上の注意事項

ここでは、マイコン製品全体に適用する「使用上の注意事項」について説明します。個別の使用上の注意事項については、本文を参照してください。なお、本マニュアルの本文と異なる記載がある場合は、本文の記載が優先するものとします。

1. 未使用端子の処理

【注意】未使用端子は、本文の「未使用端子の処理」に従って処理してください。

CMOS製品の入力端子のインピーダンスは、一般に、ハイインピーダンスとなっています。未使用端子を開放状態で動作させると、誘導現象により、LSI周辺のノイズが印加され、LSI内部で貫通電流が流れたり、入力信号と認識されて誤動作を起こす恐れがあります。未使用端子は、本文「未使用端子の処理」で説明する指示に従い処理してください。

2. 電源投入時の処置

【注意】電源投入時は、製品の状態は不定です。

電源投入時には、LSIの内部回路の状態は不確定であり、レジスタの設定や各端子の状態は不定です。外部リセット端子でリセットする製品の場合、電源投入からリセットが有効になるまでの期間、端子の状態は保証できません。

同様に、内蔵パワーオンリセット機能を使用してリセットする製品の場合、電源投入からリセットのかかる一定電圧に達するまでの期間、端子の状態は保証できません。

3. リザーブアドレスのアクセス禁止

【注意】リザーブアドレスのアクセスを禁止します。

アドレス領域には、将来の機能拡張用に割り付けられているリザーブアドレスがあります。これらのアドレスをアクセスしたときの動作については、保証できませんので、アクセスしないようにしてください。

4. クロックについて

【注意】リセット時は、クロックが安定した後、リセットを解除してください。

プログラム実行中のクロック切り替え時は、切り替え先クロックが安定した後に切り替えてください。リセット時、外部発振子（または外部発振回路）を用いたクロックで動作を開始するシステムでは、クロックが十分安定した後、リセットを解除してください。また、プログラムの途中で外部発振子（または外部発振回路）を用いたクロックに切り替える場合は、切り替え先のクロックが十分安定してから切り替えてください。

5. 製品間の相違について

【注意】型名の異なる製品に変更する場合は、事前に問題ないことをご確認下さい。

同じグループのマイコンでも型名が違っていると、内部メモリ、レイアウトパターンの相違などにより、特性が異なる場合があります。型名の異なる製品に変更する場合は、製品型名ごとにシステム評価試験を実施してください。

ご注意書き

1. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器・システムの設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因して、お客様または第三者に生じた損害に関し、当社は、一切その責任を負いません。
2. 本資料に記載されている情報は、正確を期すため慎重に作成したのですが、誤りがないことを保証するものではありません。万一、本資料に記載されている情報の誤りに起因する損害がお客様に生じた場合においても、当社は、一切その責任を負いません。
3. 本資料に記載された製品データ、図、表、プログラム、アルゴリズム、応用回路例等の情報の使用に起因して発生した第三者の特許権、著作権その他の知的財産権に対する侵害に関し、当社は、何らの責任を負うものではありません。当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
4. 当社製品を改造、改変、複製等しないでください。かかる改造、改変、複製等により生じた損害に関し、当社は、一切その責任を負いません。
5. 当社は、当社製品の品質水準を「標準水準」および「高品質水準」に分類しており、各品質水準は、以下に示す用途に製品が使用されることを意図しております。
標準水準： コンピュータ、OA機器、通信機器、計測機器、AV機器、
家電、工作機械、パーソナル機器、産業用ロボット等
高品質水準： 輸送機器（自動車、電車、船舶等）、交通用信号機器、
防災・防犯装置、各種安全装置等
当社製品は、直接生命・身体に危害を及ぼす可能性のある機器・システム（生命維持装置、人体に埋め込み使用するもの等）、もしくは多大な物的損害を発生させるおそれのある機器・システム（原子力制御システム、軍事機器等）に使用されることを意図しておらず、使用することはできません。たとえ、意図しない用途に当社製品を使用したことによりお客様または第三者に損害が生じても、当社は一切その責任を負いません。なお、ご不明点がある場合は、当社営業にお問い合わせください。
6. 当社製品をご使用の際は、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他の保証範囲内でご使用ください。当社保証範囲を超えて当社製品をご使用された場合の故障および事故につきましては、当社は、一切その責任を負いません。
7. 当社は、当社製品の品質および信頼性の向上に努めていますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は耐放射線設計については行っておりません。当社製品の故障または誤動作が生じた場合も、人身事故、火災事故、社会的損害等を生じさせないよう、お客様の責任において、冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、お客様の機器・システムとしての出荷保証を行ってください。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様の機器・システムとしての安全検証をお客様の責任で行ってください。
8. 当社製品の環境適合性等の詳細につきましては、製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制するRoHS指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。お客様がかかる法令を遵守しないことにより生じた損害に関し、当社は、一切その責任を負いません。
9. 本資料に記載されている当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器・システムに使用することはできません。また、当社製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的その他軍事用途に使用しないでください。当社製品または技術を輸出する場合は、「外国為替及び外国貿易法」その他輸出関連法令を遵守し、かかる法令の定めるところにより必要な手続を行ってください。
10. お客様の転売等により、本ご注意書き記載の諸条件に抵触して当社製品が使用され、その使用から損害が生じた場合、当社は何らの責任も負わず、お客様にてご負担して頂きますのでご了承ください。
11. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを禁じます。

注1. 本資料において使用されている「当社」とは、ルネサス エレクトロニクス株式会社およびルネサス エレクトロニクス株式会社とその総株主の議決権の過半数を直接または間接に保有する会社をいいます。

注2. 本資料において使用されている「当社製品」とは、注1において定義された当社の開発、製造製品をいいます。



ルネサス エレクトロニクス株式会社

■営業お問合せ窓口

<http://www.renesas.com>

※営業お問合せ窓口の住所は変更になることがあります。最新情報につきましては、弊社ホームページをご覧ください。

ルネサス エレクトロニクス株式会社 〒135-0061 東京都江東区豊洲3-2-24（豊洲フォレシア）

■技術的なお問合せおよび資料のご請求は下記へどうぞ。
総合お問合せ窓口：<http://japan.renesas.com/contact/>