

お客様各位

カタログ等資料中の旧社名の扱いについて

2010年4月1日を以ってNECエレクトロニクス株式会社及び株式会社ルネサステクノロジが合併し、両社の全ての事業が当社に承継されております。従いまして、本資料中には旧社名での表記が残っておりますが、当社の資料として有効ですので、ご理解の程宜しくお願ひ申し上げます。

ルネサスエレクトロニクス ホームページ (<http://www.renesas.com>)

2010年4月1日
ルネサスエレクトロニクス株式会社

【発行】ルネサスエレクトロニクス株式会社 (<http://www.renesas.com>)

【問い合わせ先】 <http://japan.renesas.com/inquiry>

ご注意書き

1. 本資料に記載されている内容は本資料発行時点のものであり、予告なく変更することがあります。当社製品のご購入およびご使用にあたりましては、事前に当社営業窓口で最新の情報をご確認いただきますとともに、当社ホームページなどを通じて公開される情報に常にご注意ください。
2. 本資料に記載された当社製品および技術情報の使用に関連し発生した第三者の特許権、著作権その他の知的財産権の侵害等に関し、当社は、一切その責任を負いません。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。
3. 当社製品を改造、改変、複製等しないでください。
4. 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。
5. 輸出に際しては、「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。本資料に記載されている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の目的で使用しないでください。また、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に使用することができません。
6. 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。
7. 当社は、当社製品の品質水準を「標準水準」、「高品質水準」および「特定水準」に分類しております。また、各品質水準は、以下に示す用途に製品が使われることを意図しておりますので、当社製品の品質水準をご確認ください。お客様は、当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途に当社製品を使用することができません。また、お客様は、当社の文書による事前の承諾を得ることなく、意図されていない用途に当社製品を使用することができません。当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途または意図されてい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生じた損害等に関し、当社は、一切その責任を負いません。なお、当社製品のデータ・シート、データ・ブック等の資料で特に品質水準の表示がない場合は、標準水準製品であることを表します。
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
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、生命維持を目的として設計されていない医療機器（厚生労働省定義の管理医療機器に相当）
特定水準： 航空機器、航空宇宙機器、海底中継機器、原子力制御システム、生命維持のための医療機器（生命維持装置、人体に埋め込み使用するもの、治療行為（患部切り出し等）を行うもの、その他直接人命に影響を与えるもの）（厚生労働省定義の高度管理医療機器に相当）またはシステム等
8. 本資料に記載された当社製品のご使用につき、特に、最大定格、動作電源電圧範囲、放熱特性、実装条件その他諸条件につきましては、当社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。
9. 当社は、当社製品の品質および信頼性の向上に努めておりますが、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、社会的損害など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、機器またはシステムとしての出荷保証をお願いいたします。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、お客様が製造された最終の機器・システムとしての安全検証をお願いいたします。
10. 当社製品の環境適合性等、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。
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固くお断りいたします。
12. 本資料に関する詳細についてのお問い合わせその他お気付きの点等がございましたら当社営業窓口までご照会ください。

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。

お客様各位

資料中の「三菱電機」、「三菱XX」等名称の株式会社ルネサス テクノロジへの変更について

2003年4月1日を以って株式会社日立製作所及び三菱電機株式会社のマイコン、ロジック、アナログ、ディスクリート半導体、及びDRAMを除くメモリ(フラッシュメモリ・SRAM等)を含む半導体事業は株式会社ルネサス テクノロジに承継されました。

従いまして、本資料中には「三菱電機」、「三菱電機株式会社」、「三菱半導体」、「三菱XX」といった表記が残っておりますが、これらの表記は全て「株式会社ルネサス テクノロジ」に変更されておりますのでご理解の程お願い致します。尚、会社商標・ロゴ・コーポレートステートメント以外の内容については一切変更しておりませんので資料としての内容更新ではありません。

注:「高周波・光素子事業、パワーデバイス事業については三菱電機にて引き続き事業運営を行います。」

2003年4月1日
株式会社ルネサス テクノロジ
カスタマサポート部

7900 シリーズ

ソフトウェアマニュアル

ルネサス16ビットシングルチップマイクロコンピュータ
7700ファミリ

本資料に記載の全ての情報は本資料発行時点のものであり、ルネサスエレクトロニクスは、予告なしに、本資料に記載した製品または仕様を変更することがあります。
ルネサスエレクトロニクスのホームページなどにより公開される最新情報をご確認ください。

安全設計に関するお願い

- ・弊社は品質、信頼性の向上に努めておりますが、半導体製品は故障が発生したり、誤動作する場合があります。弊社の半導体製品の故障又は誤動作によって結果として、人身事故、火災事故、社会的損害などを生じさせないような安全性を考慮した冗長設計、延焼対策設計、誤動作防止設計などの安全設計に十分ご注意ください。

本資料ご利用に際しての留意事項

- ・本資料は、お客様が用途に応じた適切な三菱半導体製品をご購入いただくための参考資料であり、本資料中に記載の技術情報について三菱電機が所有する知的財産権その他の権利の実施、使用を許諾するものではありません。
- ・本資料に記載の製品データ、図、表、プログラム、アルゴリズムその他応用回路例の使用に起因する損害、第三者所有の権利に対する侵害に関し、三菱電機は責任を負いません。
- ・本資料に記載の製品データ、図、表、プログラム、アルゴリズムその他全ての情報は本資料発行時点のものであり、三菱電機は、予告なしに、本資料に記載した製品または仕様を変更することがあります。三菱半導体製品のご購入に当たりますは、事前に三菱電機または特約店へ最新の情報をご確認頂きますとともに、三菱電機半導体情報ホームページ(<http://www.semicon.melco.co.jp/>) などを通じて公開される情報に常にご注意ください。
- ・本資料に記載した情報は、正確を期すため、慎重に制作したのですが万一本資料の記述誤りに起因する損害がお客様に生じた場合には、三菱電機はその責任を負いません。
- ・本資料に記載の製品データ、図、表に示す技術的な内容、プログラム及びアルゴリズムを流用する場合は、技術内容、プログラム、アルゴリズム単位で評価するだけでなく、システム全体で十分に評価し、お客様の責任において適用可否を判断してください。三菱電機は、適用可否に対する責任は負いません。
- ・本資料に記載された製品は、人命にかかわるような状況の下で使用される機器あるいはシステムに用いられることを目的として設計、製造されたものではありません。本資料に記載の製品を運輸、移動体用、医療用、航空宇宙用、原子力制御用、海底中継用機器あるいはシステムなど、特殊用途へのご利用をご検討の際には、三菱電機または特約店へご照会ください。
- ・本資料の転載、複製については、文書による三菱電機の事前の承諾が必要です。
- ・本資料に関し詳細についてのお問い合わせ、その他お気づきの点がございましたら三菱電機または特約店までご照会ください。

はじめに

このたび、CMOS16ビットマイクロコンピュータ7900シリーズのソフトウェアについて、マニュアルを作成しましたので、ご案内申し上げます。

このソフトウェアマニュアルは、ユーザの皆様に7900シリーズのソフトウェアの命令や、特長などをよく理解していただき、その機能を最大限に生かしていただくためのものです。

どうぞご利用ください。

EOL announced Product

(このページは白紙です)

EOL announced Product

ソフトウェアマニュアルご使用の前に

1. 構成

このソフトウェアマニュアルは、次の章で構成されています。

第1章 概要

7900シリーズのソフトウェアについて、簡単に紹介しています。

第2章 中央演算処理装置(CPU)

7900シリーズのCPU、アクセス空間、及びアドレッシングモードについて説明しています。

なお、アクセス空間は機種によって異なりますので、「2. 注意」を参照の上、該当資料でご確認ください。

第3章 7900命令の使用方法

メモリアクセスやレジスタなど、7900シリーズ固有の機能について説明しています。

第4章 命令

7900シリーズの各命令について説明しています。

付 録

7900シリーズの機械語命令一覧表、及び16進命令コード対応表を掲載しています。

2. 注 意

マイコンについての下記の情報は、次に示す各資料を参照ください。

これらの資料は、<http://www.infocom.maec.co.jp/>で参照いただけます。

機種展開 : 最新のデータシート又はカタログ類

電気的特性 : 最新のデータシート

ハードウェア(アクセス空間): 最新のユーザズマニュアル

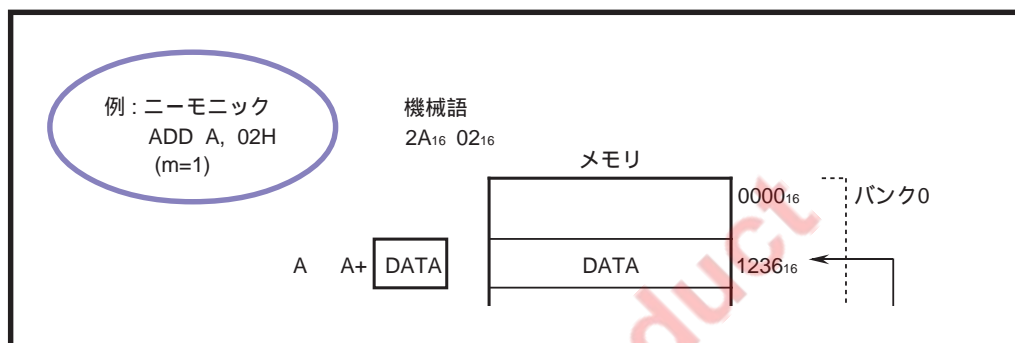
開発サポートツールについての情報は、最新のデータシート又はカタログ類を参照ください。

これらの資料は、<http://www.tool-spt.maec.co.jp/>で参照いただけます。

ソフトウェアマニュアルご使用の前に

3. 記述例について

本ソフトウェアマニュアルでは、ニーモニック例、記述例、記述方式といった形で命令の記述例を示しています。これらはアセンブリ言語記述の一例であり、使用するアセンブラによって異なります。ご使用のアセンブラのユーザーズマニュアルも併せてご参照ください。



【記述例】

CLM		
ADD.W	A,#IMM16	; A A + IMM16
ADD	B,MEM16	; B B + MEM16
SEM		
ADD.B	A,#IMM8	; AL AL + IMM8
ADD	B,MEM8	; BL BL + MEM8

アドレッシングモード	記述形式	機械語
IMM	ADD A,#imm	26 ₁₆ ,imm (81 ₁₆ ,26 ₁₆ ,imm)
DIR	ADD A,dd	2A ₁₆ ,dd (81 ₁₆ ,2A ₁₆ ,dd)
DIR,X	ADD A,dd,X	2B ₁₆ ,dd (81 ₁₆ ,2B ₁₆ ,dd)
(DIR)	ADD A,(dd)	11 ₁₆ ,20 ₁₆ ,dd (91 ₁₆ ,20 ₁₆ ,dd)
(DIR,X)	ADD A,(dd,X)	11 ₁₆ ,21 ₁₆ ,dd (91 ₁₆ ,21 ₁₆ ,dd)
(DIR),Y	ADD A,(dd),Y	11 ₁₆ ,28 ₁₆ ,dd (91 ₁₆ ,28 ₁₆ ,dd)
L(DIR)	ADD A,L(dd)	11 ₁₆ ,22 ₁₆ ,dd (91 ₁₆ ,22 ₁₆ ,dd)
⋮	⋮	⋮

目次

第1章 概要

第2章 中央演算処理装置(CPU)

2.1 中央演算処理装置(CPU)	2-2
2.1.1 アキュムレータ (Acc)	2-3
2.1.2 インデックスレジスタX (X)	2-3
2.1.3 インデックスレジスタY (Y)	2-3
2.1.4 スタックポインタ (S)	2-4
2.1.5 プログラムカウンタ (PC)	2-5
2.1.6 プログラムバンクレジスタ (PG)	2-5
2.1.7 データバンクレジスタ (DT)	2-5
2.1.8 ダイレクトページレジスタ0~3 (DPR0~DPR3)	2-6
2.1.9 プロセッサステータスレジスタ (PS)	2-8
2.2 アクセス空間	2-10
2.3 アドレッシングモード	2-11
2.3.1 概要	2-11
2.3.2 アドレッシングモードの説明	2-11

第3章 7900命令の使用方法

3.1 メモリアクセス	3-2
3.1.1 ダイレクトアドレッシング	3-2
3.1.2 アブソリュート、アブソリュートロングアドレッシング	3-2
3.1.3 インダイレクト、インダイレクトロングアドレッシング	3-2
3.2 DPR0~DPR3の使用	3-4
3.3 8/16ビット長データ操作	3-5
3.4 インデックスレジスタX, Y	3-6
3.5 分岐命令の使用方法	3-7

第4章 命令

4.1 命令の体系	4-2
4.2 命令の説明	4-8
4.3 ソフトウェア開発上の注意	4-221
4.3.1 命令実行サイクル	4-221
4.3.2 mフラグ、xフラグの状態	4-221
4.3.3 データ領域配置の工夫	4-221
4.3.4 10進演算を行う場合	4-221
4.4 32ビット演算命令実行時の注意	4-222

付録

付録1．機械語命令一覧表	5-2
付録2．16進命令コード対応表	5-44

Memo

EOL announced Product

第 1 章
概 要

EOL announced Product

7900シリーズは7700ファミリの上位機種です。ソフトウェアは従来の7700ファミリ(7700シリーズ、7751シリーズ等)とソースレベルで互換性があります。

7900シリーズの特長は以下の通りです。

従来の7700ファミリとソースレベルで互換性があります。

7700シリーズの命令数は103、7751シリーズは109であるのに対し、7900シリーズでは203に拡張されています。追加されている命令は以下のようなものです。

- (i) 32ビット演算
- (ii) 8ビット長データ専用命令
- (iii) メモリ間データ転送命令
- (iv) レジスタ/メモリを0クリアする命令
- (v) キャリーなし加減算命令
- (vi) スタックポインタに対する加減算命令
- (vii) メモリに対する論理和・論理積・排他的論理和
- (viii) メモリに対する比較演算命令
- (ix) 符号付き条件分岐命令
- (x) 比較&条件分岐命令
- (xi) デクリメント&条件分岐命令
- (xii) PC相対によるサブルーチン呼び出し命令

命令を拡張した結果、従来の7700ファミリに比べてプログラムサイズが平均20~30%縮小できます。アクセス空間は16Mバイトです。この空間をアクセスするための豊富なアドレッシングモードを持っています。000000₁₆~00FFFF₁₆の64Kバイト空間を高速かつバイト数の小さな命令でアクセス可能です。このために使用することを目的としたダイレクトページレジスタを4本用意しています。命令実行サイクル数を従来の7700ファミリのそれよりも短縮しています。

第 2 章

中央演算処理装置 (CPU)

- 2.1 中央演算処理装置(CPU)
- 2.2 アクセス空間
- 2.3 アドレッシングモード

2.1 中央演算処理装置(CPU)

中央演算処理装置(以下CPUと称す)は図2.1.1に示す13個のレジスタで構成されています。

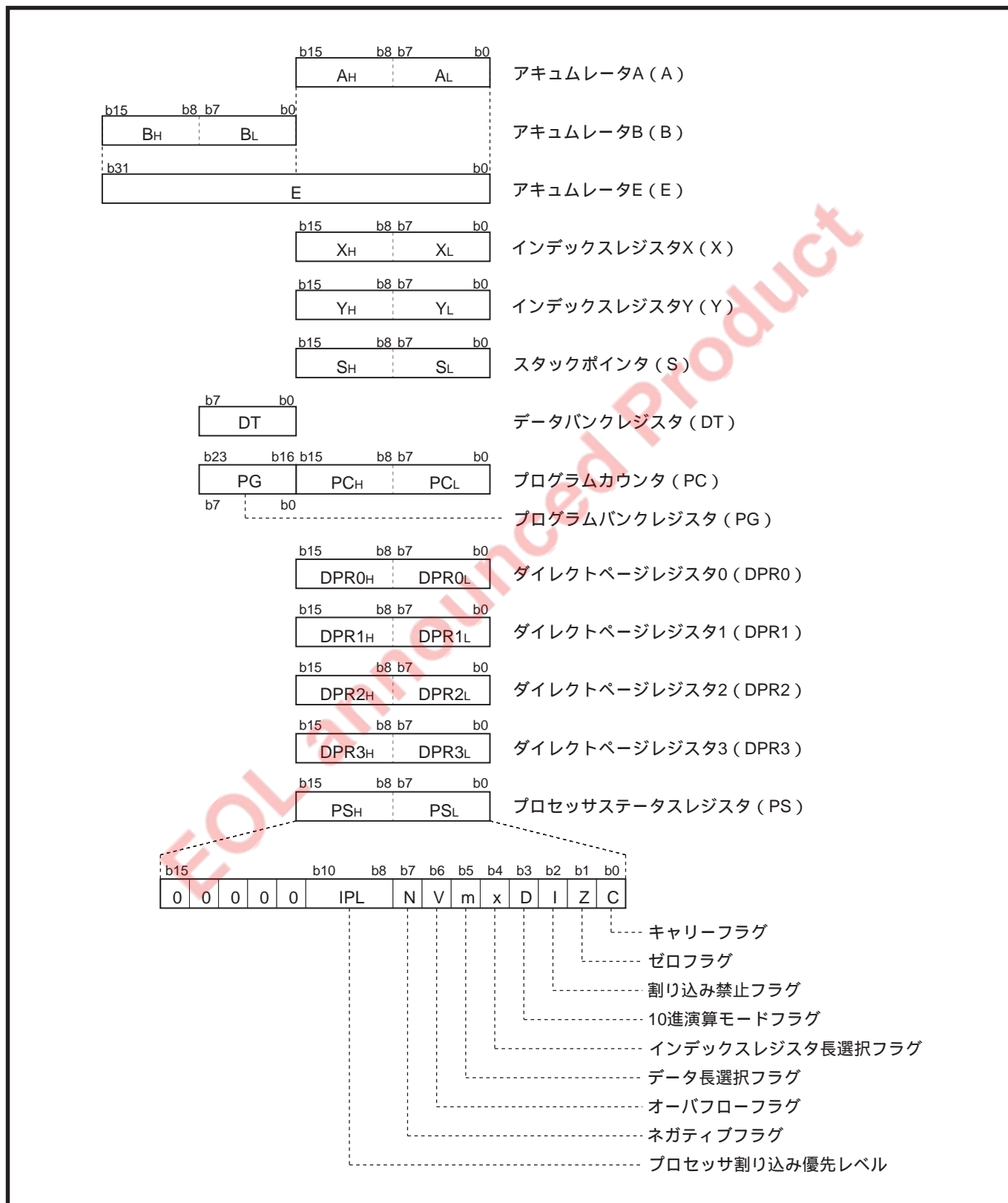


図2.1.1 CPUレジスタ

2.1.1 アキュムレータ (Acc)

アキュムレータには、アキュムレータA、及びアキュムレータBがあります。また、アキュムレータAとアキュムレータBを接続して、32ビットのアキュムレータとして使用することもできます(アキュムレータE)。

(1)アキュムレータA (A)

演算、転送、入出力などのデータ処理は、アキュムレータA(以下Aと称す)を中心に実行されます。このレジスタは16ビット構成ですが、後述のプロセッサステータスレジスタ内のデータ長選択フラグ(以下mフラグと称す)で選択することによって、下位8ビットだけを使用することもできます。

(2)アキュムレータB (B)

Aと同等の機能を持ち、Aの代わりとして使用できます。ただし、一部の命令では、Aしか使用できないものもあります。また一部の命令では、命令バイト数、及び実行サイクル数はAを使用した場合より多く必要となります。アキュムレータB(以下Bと称す)も16ビット構成ですが、Aと同様にmフラグの影響を受けます。

(3)アキュムレータE (E)

Aを下位16ビット、Bを上位16ビットとする32ビットのアキュムレータです。32ビットのデータを扱う命令で使用します。mフラグの影響は受けません。

2.1.2 インデックスレジスタX (X)

インデックスレジスタX(以下Xと称す)は16ビット構成ですが、後述のプロセッサステータスレジスタ内のインデックスレジスタ長選択フラグ(以下xフラグと称す)で選択することによって、下位8ビットだけを使用することもできます。

Xをインデックスレジスタとするアドレッシングモードでは、オペランドの内容にこのレジスタの内容を加算した番地をアクセスします。また、MVP、MVN、RMPA命令でも使用します。

2.1.3 インデックスレジスタY (Y)

Xと同等の機能を持ちます。インデックスレジスタY(以下Yと称す)も16ビット構成ですが、Xと同様にxフラグの影響を受けます。

2.1.4 スタックポインタ (S)

スタックポインタ(以下Sと称す)は16ビット構成で、割り込み及びサブルーチン呼び出し時、又はスタックを用いるアドレッシングモードを実行したときに使用されます。Sの内容は、割り込みやサブルーチン呼び出し時にレジスタ類を退避する番地(スタック領域)を示します。スタック領域はバンク0_i(「2.2 アクセス空間」参照)内に設定されます。

割り込み要求を受け付けた場合、マイクロコンピュータはSの内容が示す番地にプログラムバンクレジスタ(以下PGと称す)の内容を退避し、Sの内容を1減算します。次にプログラムカウンタ(以下PCと称す)及びプロセッサステータスレジスタ(以下PSと称す)の内容を退避します。割り込み要求受付後のSの内容は、割り込み要求受付前のSの内容から5を減算した値になります(図2.1.2参照)。

割り込みルーチンでの処理を終了し、元のルーチンに復帰するときは、RTI命令が実行されると、退避時と逆の順序(PC PG)でスタック領域内に退避していたレジスタの内容を元のレジスタに復帰します。また、Sの内容を割り込み要求受付前の状態に戻します。

サブルーチンを呼び出した場合も割り込み時と同様の動作を行います。PSの内容は自動的に退避しません(アドレッシングモードによってはPGの内容も退避しません)。

割り込み及びサブルーチン呼び出し時には上記レジスタ以外は自動的に退避しませんので、内容の保持が必要なレジスタはソフトウェアによって退避してください。

リセット時のSの内容は機種によって異なります。各機種の最新のデータシート、又はユーザーズマニュアルを参照してください。

スタック領域はサブルーチンのネスティングや多重割り込みの受付によって変化しますので、必要なデータを壊さないようにサブルーチンのネスティングの深さなどには注意してください。

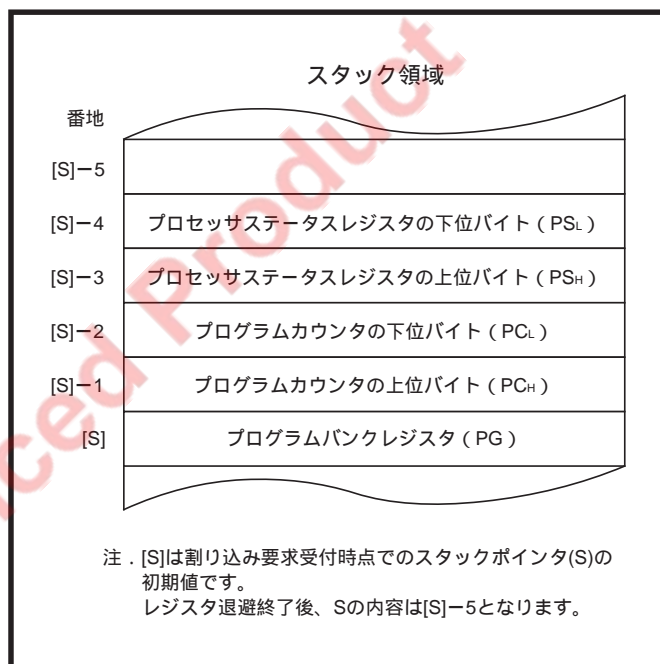


図2.1.2 スタック領域へのレジスタ退避

2.1.5 プログラムカウンタ (PC)

PCは16ビット構成で、次に実行する命令の格納番地(24ビット)の下位16ビットを示します。

リセット時、PCの上位バイト(PC_H)には“FF₁₆”、下位バイト(PC_L)には“FE₁₆”が設定されます。そして、リセット直後はリセットのベクトル番地の内容が設定されます(リセットのベクトル番地は機種によって異なります。各機種の最新のデータシート、又はユーザーズマニュアルを参照してください)。

図2.1.3にPG及びPCを示します。

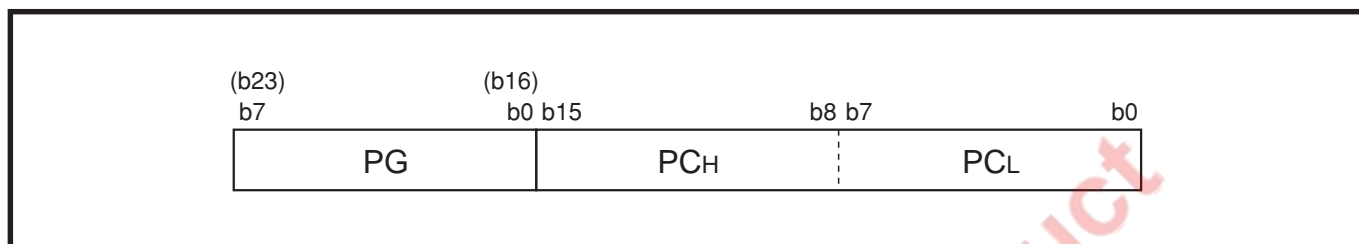


図2.1.3 PG及びPC

2.1.6 プログラムバンクレジスタ (PG)

アクセス空間は64Kバイトごとに分けられており、その単位をバンクと呼びます(「2.2 アクセス空間」参照)。

PGは8ビット構成で、次に実行する命令(次に命令キューバッファから読み出す命令)の格納番地(24ビット)の上位8ビット(バンク)を示します。PGの内容は、PCの内容が加算されキャリーが生じたり、分岐命令などで変位をPCの内容に加算してキャリーが生じた場合には、自動的に1加算されます。また、PCの内容から減算してポローが生じた場合には、自動的に1減算されます。したがって、通常はプログラミングの際にバンクの境界を意識する必要はありません。

リセット時のPGの値は機種によって異なります。各機種の最新のデータシート、又はユーザーズマニュアルを参照してください。

2.1.7 データバンクレジスタ (DT)

データバンクレジスタ(以下DTと称す)は8ビット構成です。DTを使用する下記のアドレッシングモードにおいて、このレジスタの内容はアクセスする番地(24ビット)の上位8ビット(バンク)として扱われます。このレジスタへの値の設定にはLDT命令を使用してください。

リセット時、このレジスタは“00₁₆”になります。

DTを使用するアドレッシングモード

- ダイレクト・インダイレクト
- ダイレクト・インデクストX・インダイレクト
- ダイレクト・インダイレクト・インデクストY
- アブソリュート
- アブソリュート・インデクストX
- アブソリュート・インデクストY
- アブソリュート・ビット・レラティブ
- スタックポインタ・レラティブ・インダイレクト・インデクストY
- 積和

2.1.8 ダイレクトページレジスタ0~3 (DPR0~DPR3)

ダイレクトページレジスタ(以下DPRnと称す)は、従来の7700ファミリと比べて強化されています。

ダイレクトページレジスタは16ビット構成です。バンク0内の64Kバイト空間を効率良くアクセスするために用います。

プロセッサモードレジスタ1のダイレクトページレジスタ切り替えビットによって、DPR0だけを使用するか、DPR0~DPR3を使用するかを選択できます。このビットの機能は以下の通りです。

表2.1.1 ダイレクトページレジスタの切り替え

	ダイレクトページレジスタ切り替えビット	
	0	1
使用可能なDPRn	DPR0	DPR0~DPR3
DPRnを基底としてアクセス可能なブロックサイズ	256バイト	64バイト
備考	従来の7700との互換モード	-

ダイレクトページレジスタ切り替えビットの値は、一旦設定した後は変更しないでください。

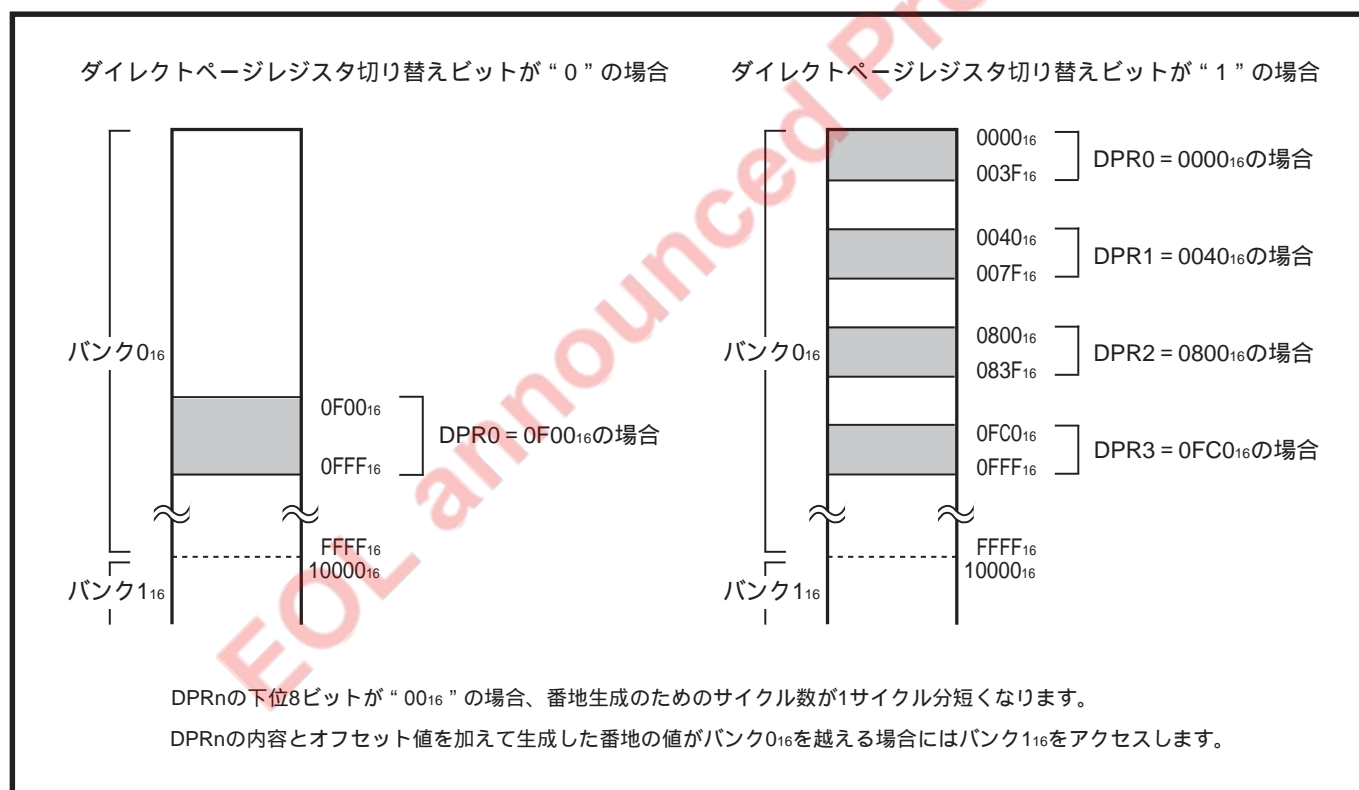


図2.1.4 ダイレクトページ領域の指定例

DPRnの下位8ビットの内容を“ 00₁₆ ”にすると、番地生成のためのサイクル数が“ 00₁₆ ”以外の場合よりも1サイクル分短くなりますので、効率良くアクセスできます。

リセット時、DPR0は“ 0000₁₆ ”、DPR1 ~ DPR3は不定になります。

DPRnを使用するアドレッシングモード

ダイレクト

ダイレクト・インデクストX

ダイレクト・インデクストY

ダイレクト・インダイレクト

ダイレクト・インデクストX・インダイレクト

ダイレクト・インダイレクト・インデクストY

ダイレクト・インダイレクトロング

ダイレクト・インダイレクトロング・インデクストY

ダイレクト・ビット・レラティブ

EOL announced Product

2.1.9 プロセッサステータスレジスタ (PS)

プロセッサステータスレジスタ(以下PSと称す)は11ビット構成です。図2.1.5にPSのレジスタ構成を示します。

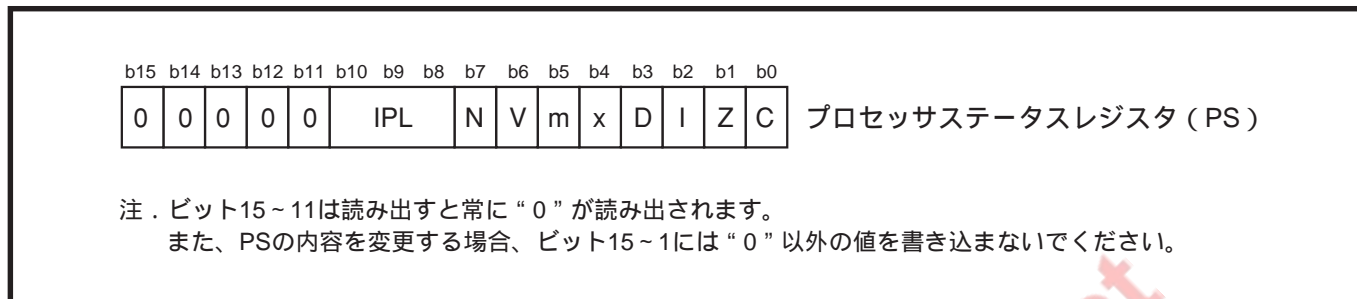


図2.1.5 PSのレジスタ構成

(1)ビット0：キャリーフラグ (C)

演算処理中、算術論理ユニットで発生したキャリー、又はボローを保持します。シフト命令、ローテート命令によっても変化します。キャリーフラグ(以下Cフラグと称す)を“1”にするときはSEC、SEP命令、“0”にするときはCLC、CLP命令を使用してください。

リセット時、Cフラグの内容は不定です。

(2)ビット1：ゼロフラグ (Z)

演算処理、データ転送の結果が“0”のとき“1”になり、“0”以外のとき“0”になります。10進演算実行時、ゼロフラグ(以下Zフラグと称す)の内容は無効です。Zフラグを“1”にするときはSEP命令、“0”にするときはCLP命令を使用してください。

リセット時、Zフラグの内容は不定です。

(3)ビット2：割り込み禁止フラグ (I)

すべてのマスクブル割り込みを禁止するフラグです。割り込み禁止フラグ(以下Iフラグと称す)が“1”のときは割り込み禁止状態です。また、割り込み要求を受け付けると自動的に“1”になり、多重割り込みを禁止します。Iフラグを“1”にするときはSEI、SEP命令、“0”にするときはCLI、CLP命令を使用してください。

リセット時、Iフラグは“1”になります。

(4)ビット3：10進演算モードフラグ (D)

加減算を2進で行うか、10進で行うかを選択します。10進演算モードフラグ(以下Dフラグと称す)が“0”のときは通常の2進演算を行います。“1”のときは8ビットを2桁(m=1のとき)又は16ビットを4桁(m=0のとき)の10進数として演算します。10進補正は自動的に行われます(ただし、10進演算が行えるのはADC、ADCB、SBC、SBCB命令です)。Dフラグを“1”にするときはSEP命令、“0”にするときはCLP命令を使用してください。

リセット時、Dフラグは“0”になります。

(5)ビット4：インデックスレジスタ長選択フラグ (x)

X、又はYを16ビット長で使用するか、8ビット長で使用するかを選択します。xフラグが“0”のときは16ビット長、“1”のときは8ビット長になります(注)。xフラグを“1”にするときはSEP命令、“0”にするときはCLP命令を使用してください。

リセット時、xフラグは“0”になります。

(6)ビット5：データ長選択フラグ (m)

データ長を16ビット長で使用するか、8ビット長で使用するかを選択します。mフラグが“0”のときは16ビット長、“1”のときは8ビット長になります(注)。mフラグを“1”にするときはSEM、SEP命令、“0”にするときはCLM、CLP命令を使用してください。

リセット時、mフラグは“0”になります。

注．ビット長の異なるレジスタ間で転送を行った場合は、転送先のレジスタのデータ長で転送されます。ただしTXA、TYA、TXB、TYB、及びTXS命令を使用した場合は除きます。

(7)ビット6：オーバフローフラグ (V)

1語を符号付きの2進数として加減算するときに意味を持ちます。

加減算の結果が-2147483648～+2147483647(32ビット長演算時)、-32768～+32767(16ビット長演算時)、-128～+127(8ビット長演算時)の範囲を越えると、オーバフローフラグ(以下Vフラグと称す)は“1”になります。また、DIV、DIVS命令実行結果が、それを格納するレジスタ長を越えた場合にも、Vフラグは“1”になります。

10進演算実行時、Vフラグの内容は無効です。Vフラグを“1”にするときはSEP命令、“0”にするときはCLV、CLP命令を使用してください。

リセット時、Vフラグの内容は不定です。

(8)ビット7：ネガティブフラグ (N)

演算処理、データ転送の結果が負(演算結果の最上位ビットが“1”)のとき“1”になり、それ以外では“0”になります。10進演算実行時、ネガティブフラグ(以下Nフラグと称す)の内容は無効です。Nフラグを“1”にするときはSEP命令、“0”にするときはCLP命令を使用してください。

リセット時、Nフラグの内容は不定です。

(9)ビット10～8：プロセッサ割り込み優先レベル (IPL)

プロセッサ割り込み優先レベル(以下IPLと称す)によって、レベル0からレベル7までの8段階のIPLを決めることができます。各割り込みの割り込み優先レベル(割り込み制御レジスタ内に設定)がIPLより高い場合に、その割り込みは許可されます。割り込み要求が受け付けられると、IPLの内容はスタック領域に退避され、受け付けられた割り込みの優先レベルがIPLに入ります。

IPLを直接“1”、又は“0”にする命令はありません。IPLを変更する場合は、一度スタック領域に設定したいIPLの値を格納し、PUL、又はPLP命令でPSの内容を変更してください。

リセット時、IPLは“000₂”になります。

2.2 アクセス空間

7900シリーズのアクセス空間は、 0_{16} ~ $FFFFFF_{16}$ 番地の16Mバイトです(図2.2.1参照)。ただし、 $FF0000_{16}$ ~ $FFFFFF_{16}$ 番地は予約領域のため、使用しないでください。

PCは16ビット構成ですが、8ビットのPGと組み合わせて、24ビットのアドレスを生成します。

7900シリーズのアクセス空間を64Kバイトずつ区切ったものをバンクと呼びます。PGがバンク番号を示します。

メモリとI/Oは同一のアクセス空間内に配置されています。したがって、メモリとI/Oを区別することなく、同じ命令を使用して転送、演算などが行えます。

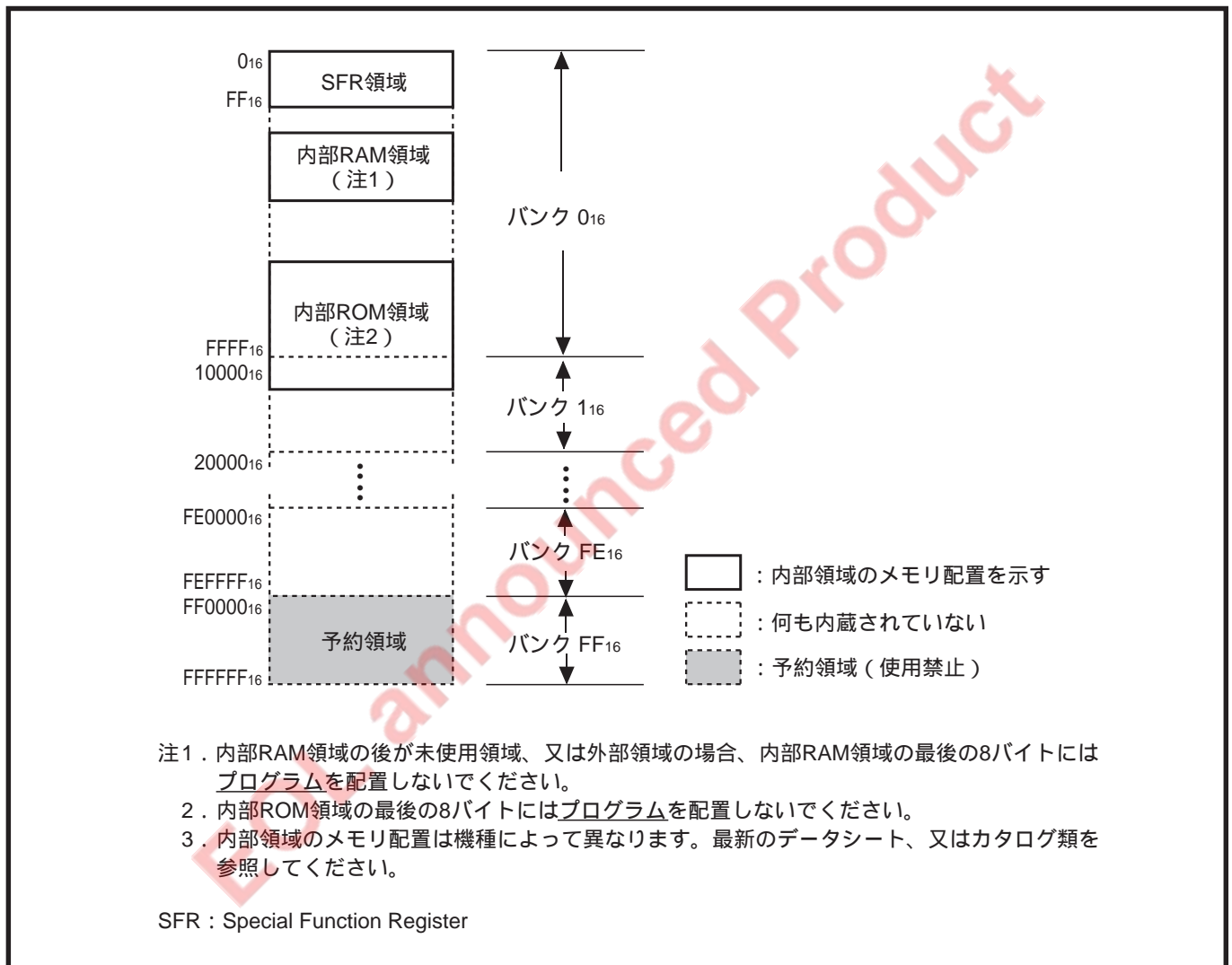


図2.2.1 7900シリーズのアクセス空間

2.3 アドレッシングモード

2.3.1 概要

命令を実行する上で、演算に必要なデータをメモリから取り出したり、逆に演算結果をあるメモリに格納する場合には、必ずそのメモリの番地を指定しておかなければなりません。また、プログラムの実行中にある番地にジャンプする場合にも、飛び先番地を指定する必要があります。このメモリ番地の指定を行う方法のことをアドレッシングと呼びます。

7900シリーズは27種類のアドレッシングモードを持っています。

2.3.2 アドレッシングモードの説明

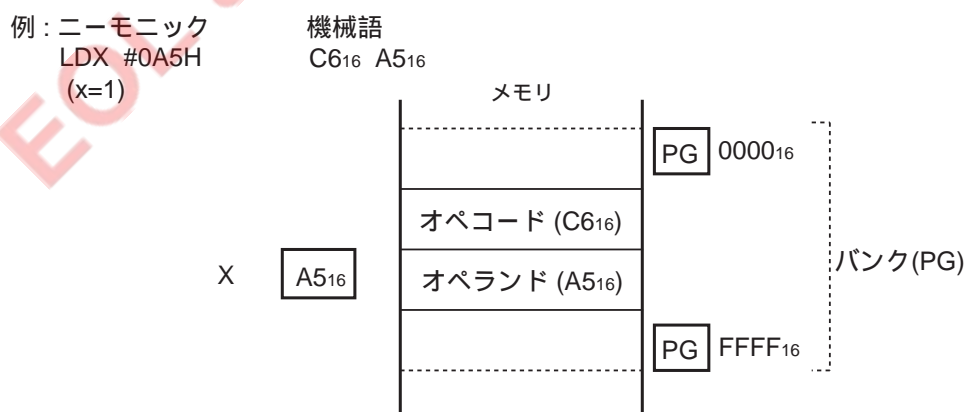
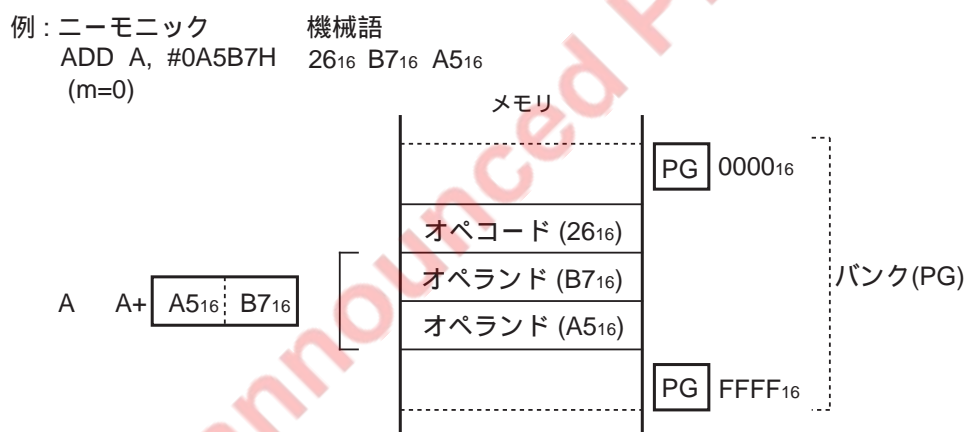
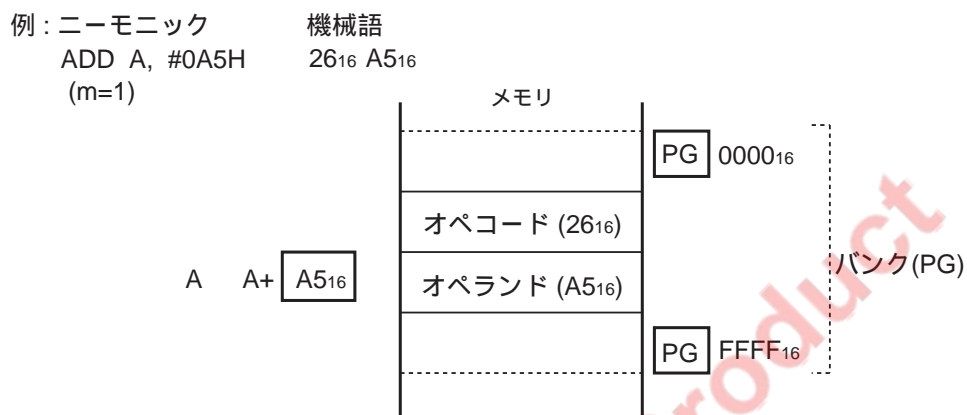
各アドレッシングモードの説明を以下のページに示します。

インプライド(IMP)	2-12
イミディエイト(IMM)	2-13
アキュムレータ(A)	2-15
ダイレクト(DIR)	2-16
ダイレクト・インデクストX(DIR, X)	2-18
ダイレクト・インデクストY(DIR, Y)	2-21
ダイレクト・インダイレクト(DIR)	2-22
ダイレクト・インデクストX・インダイレクト(DIR, X)	2-24
ダイレクト・インダイレクト・インデクストY(DIR, Y)	2-27
ダイレクト・インダイレクトロング(L(DIR))	2-30
ダイレクト・インダイレクトロング・インデクストY(L(DIR), Y)	2-32
アブソリュート(ABS)	2-35
アブソリュート・インデクストX(ABS, X)	2-38
アブソリュート・インデクストY(ABS, Y)	2-41
アブソリュートロング(ABL)	2-44
アブソリュートロング・インデクストX(ABL, X)	2-46
アブソリュート・インダイレクト(ABS)	2-48
アブソリュート・インダイレクトロング(L(ABS))	2-49
アブソリュート・インデクストX・インダイレクト(ABS, X)	2-50
スタック(STK)	2-51
レラティブ(REL)	2-54
ダイレクト・ビット・レラティブ(DIR, b, R)	2-55
アブソリュート・ビット・レラティブ(ABS, b, R)	2-57
スタックポインタ・レラティブ(SR)	2-59
スタックポインタ・レラティブ・インダイレクト・インデクストY(SR, Y)	2-60
ブロック転送(BLK)	2-63
積和(積和)	2-65

注．特に断らない限り、「ダイレクト」の説明図にはDPR0のみを使用する場合を示します。

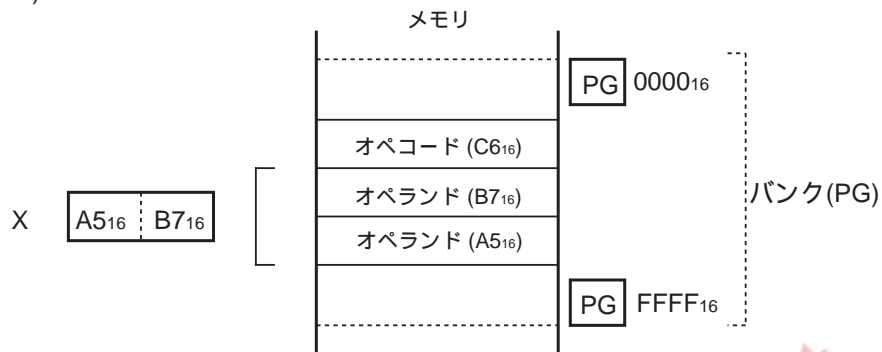
【名称】 イミディエイト アドレッシングモード

【機能】 レジスタと即値との演算を行います。



例: ニーモニック
LDX #0A5B7H
(x=0)

機械語
C6₁₆ B7₁₆ A5₁₆



EOL announced Product

【名称】 アキュムレータ アドレッシングモード

【機能】 アキュムレータの内容を操作します。

例：ニーモニック

ROL A
(m=1)

機械語

13₁₆

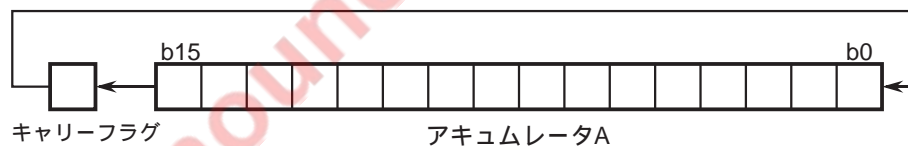


例：ニーモニック

ROL A
(m=0)

機械語

13₁₆



【名称】 ダイレクト アドレッシングモード

【機能】 命令のオペランドとダイレクトページレジスタの内容を加算した値で指定されるバンク0内のメモリの内容が演算データとなります。ただし、命令のオペランドとダイレクトページレジスタの内容を加算した値がバンク0を越える場合は、バンク1内のメモリが指定されます。プロセッサモードレジスタ1の「ダイレクトページレジスタ切り替えビット」により、以下のいずれかを選択できます。

ダイレクトページレジスタ切り替えビットが“0”の場合

ダイレクトページレジスタ(DPR0)のみを使用します。

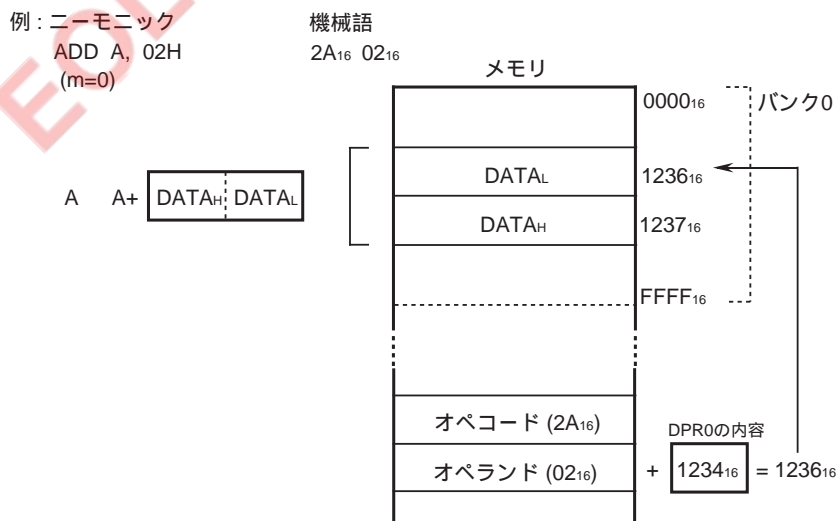
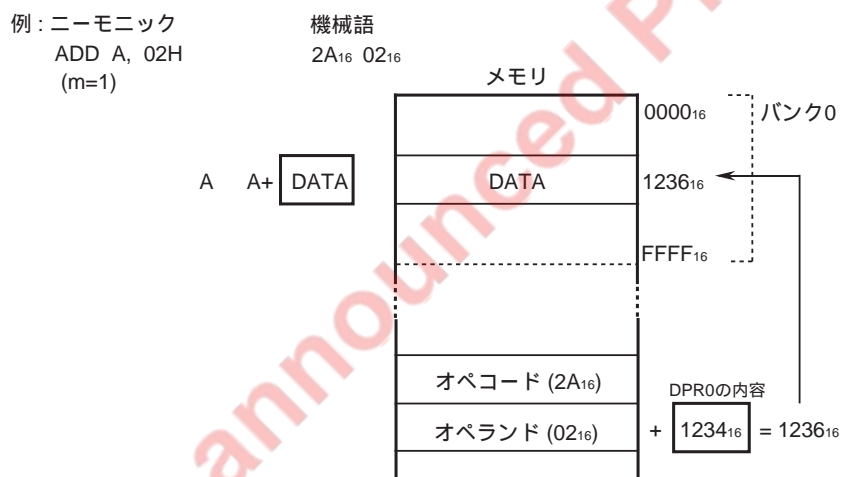
この時、DPR0からのオフセットは8ビット長で指定します。

ダイレクトページレジスタ切り替えビットが“1”の場合

ダイレクトページレジスタ0~3(DPR0~3)を使用します。

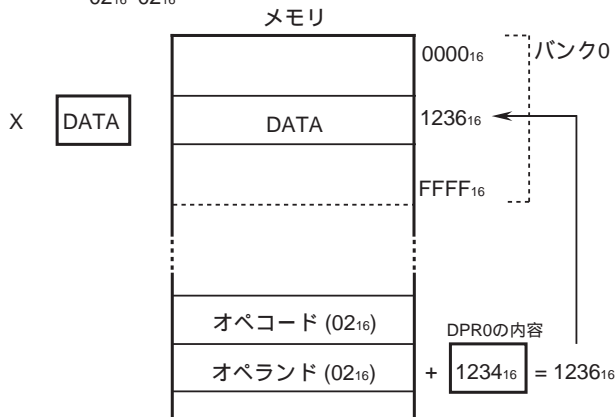
この時、オペランド(8ビット)の上位2ビットでダイレクトページレジスタを指定し、下位6ビットでオフセットを指定します。

<ダイレクトページレジスタ切り替えビットが“0”の場合>



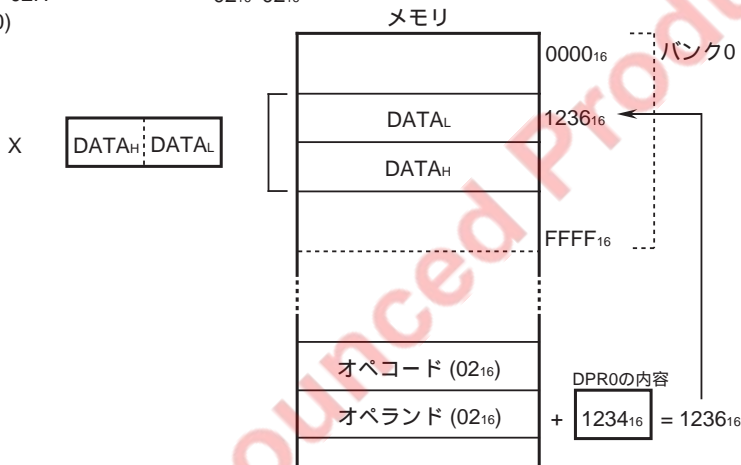
例：ニーモニック
LDX 02H
(x=1)

機械語
02₁₆ 02₁₆



例：ニーモニック
LDX 02H
(x=0)

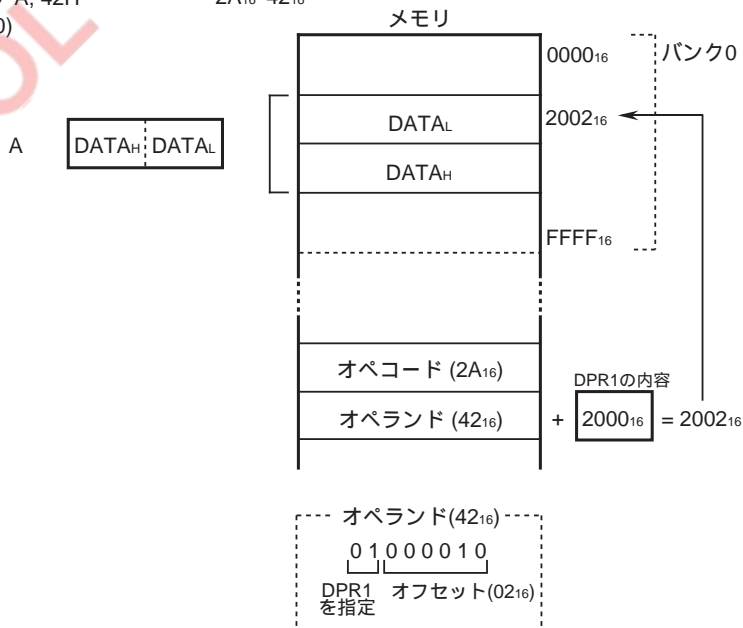
機械語
02₁₆ 02₁₆



<ダイレクトページレジスタ切り替えビットが“1”の場合>

例：ニーモニック
ADD A, 42H
(x=0)

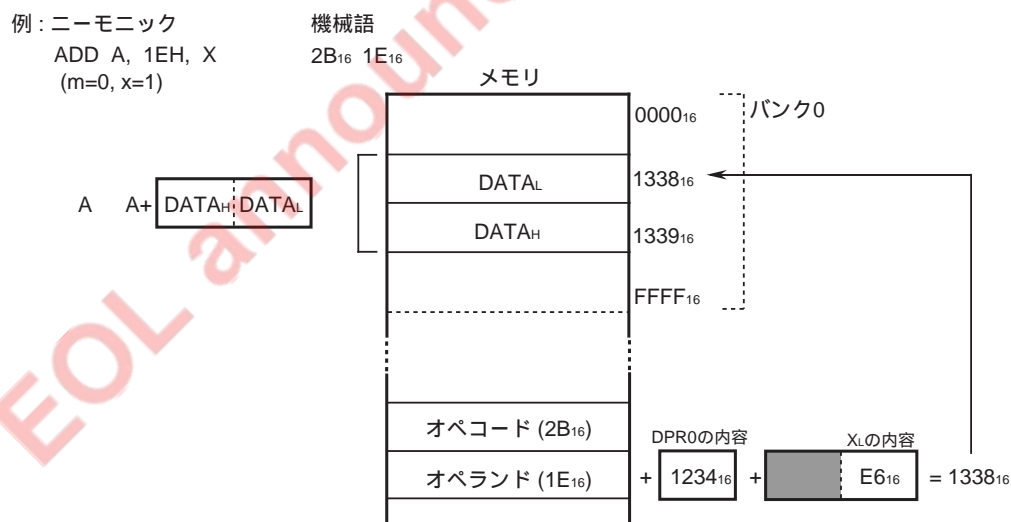
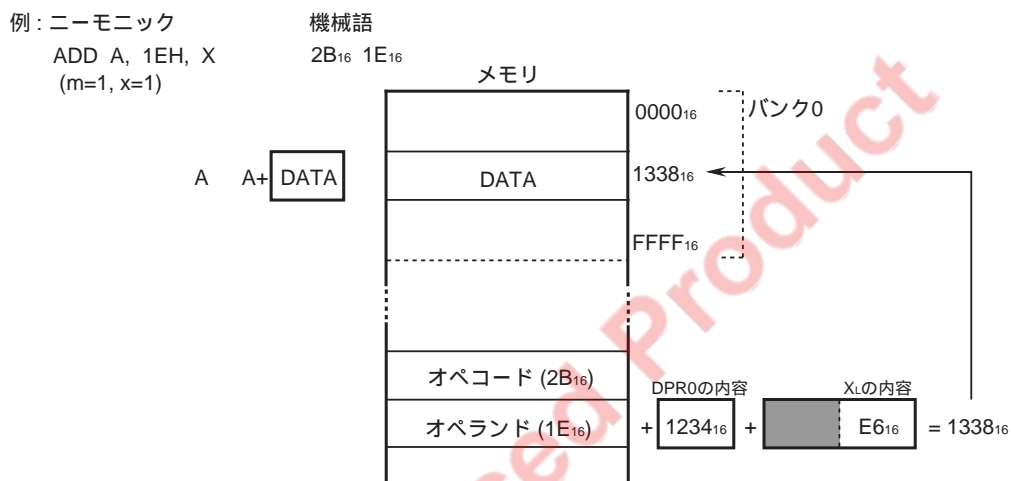
機械語
2A₁₆ 42₁₆



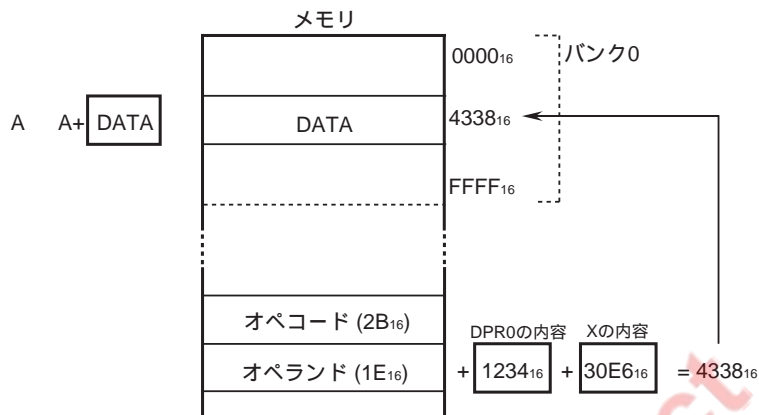
【名称】 ダイレクト・インデクストX アドレッシングモード

【機能】 命令のオペランドとダイレクトページレジスタの内容とインデックスレジスタXの内容を加算した値で指定されるバンク0内のメモリの内容が演算データとなります。ただし、命令のオペランドとダイレクトページレジスタの内容とインデックスレジスタXの内容を加算した値がバンク0あるいはバンク1を越える場合は、バンク1あるいはバンク2内のメモリが指定されます。

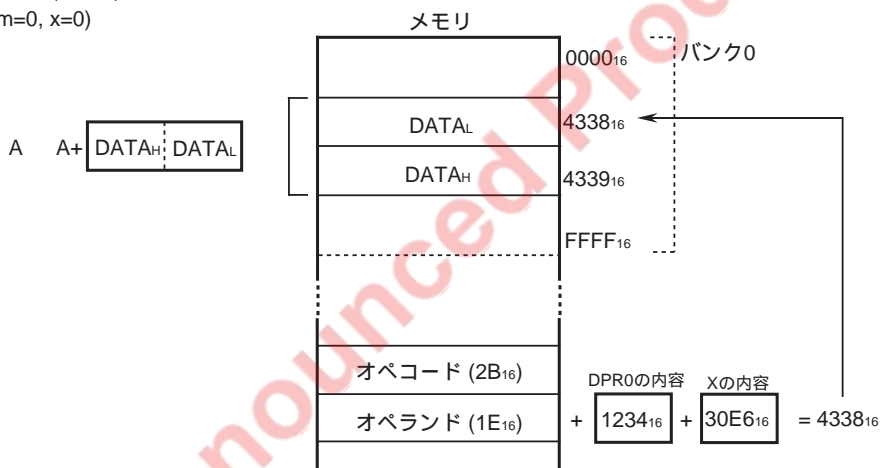
<ダイレクトページレジスタ切り替えビットが“0”の場合>



例：ニーモニック 機械語
 ADD A, 1EH, X 2B₁₆ 1E₁₆
 (m=1, x=0)



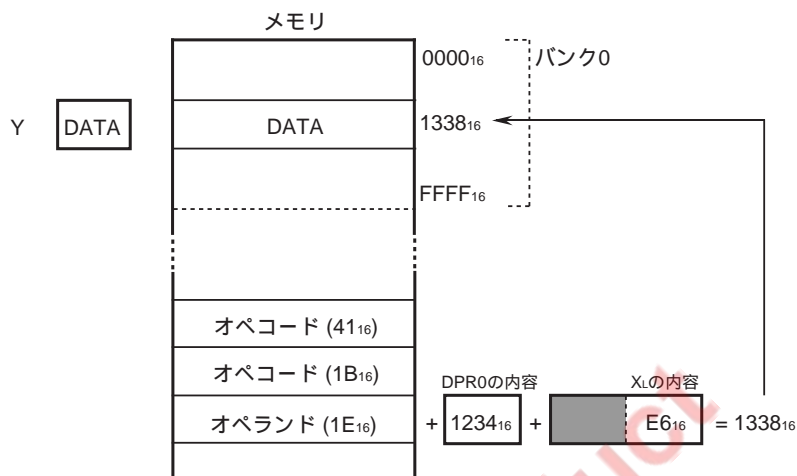
例：ニーモニック 機械語
 ADD A, 1EH, X 2B₁₆ 1E₁₆
 (m=0, x=0)



EOL announced Product

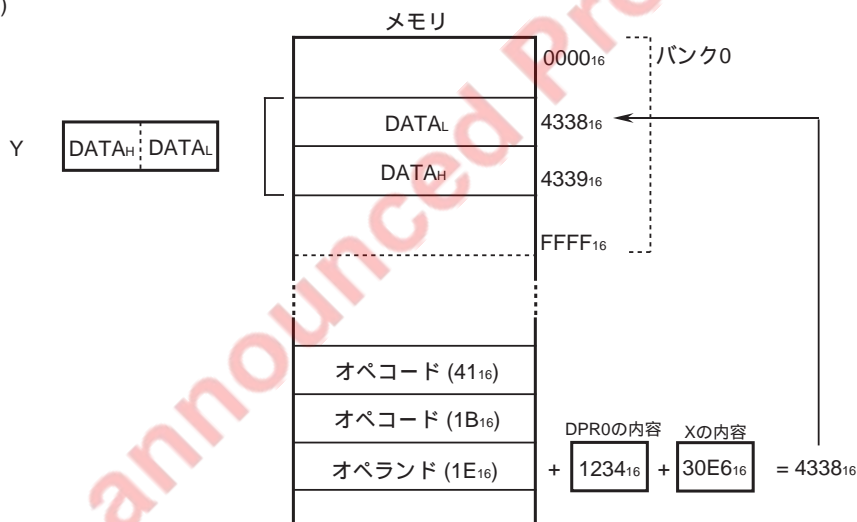
例：ニーモニック
LDY 1EH, X
(x=1)

機械語
41₁₆ 1B₁₆ 1E₁₆



例：ニーモニック
LDY 1EH, X
(x=0)

機械語
41₁₆ 1B₁₆ 1E₁₆



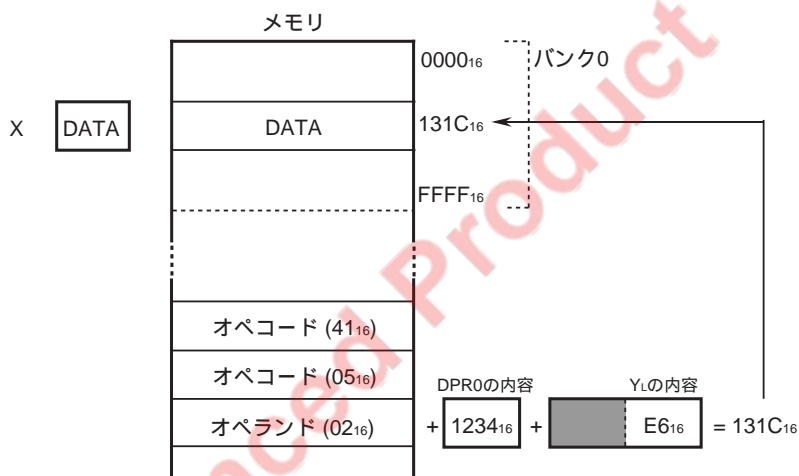
【名称】 ダイレクト・インデクストY アドレッシングモード

【機能】 命令のオペランドとダイレクトページレジスタの内容とインデックスレジスタYの内容を加算した値で指定されるバンク0内のメモリの内容が演算データとなります。ただし、命令のオペランドとダイレクトページレジスタの内容とインデックスレジスタYの内容を加算した値がバンク0あるいはバンク1を越える場合は、バンク1あるいはバンク2内のメモリが指定されます。

<ダイレクトページレジスタ切り替えビットが“0”の場合>

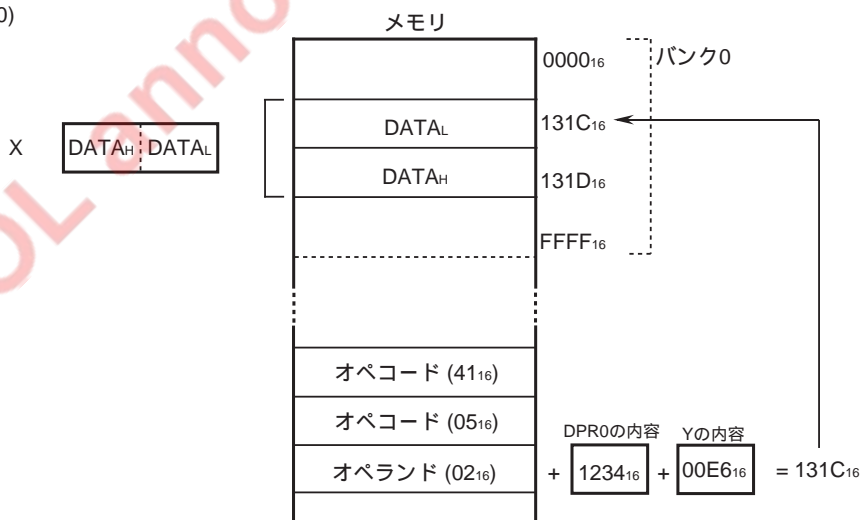
例：ニーモニック
LDX 02H, Y
(x=1)

機械語
41₁₆ 05₁₆ 02₁₆



例：ニーモニック
LDX 02H, Y
(x=0)

機械語
41₁₆ 05₁₆ 02₁₆

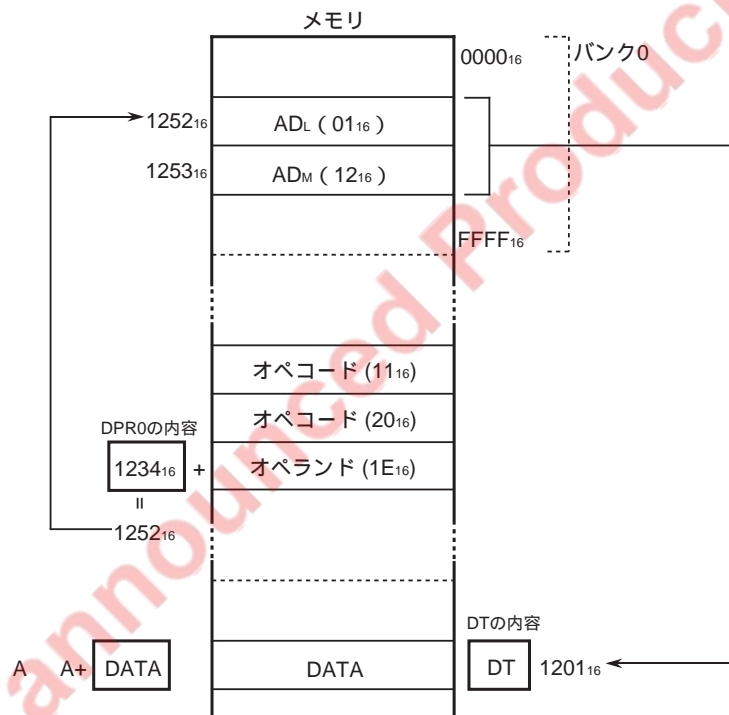


【名称】 ダイレクト・インダイレクト アドレッシングモード

【機能】 命令のオペランドとダイレクトページレジスタの内容を加算した値でバンク0内の連続する2バイトのメモリを指定します。指定されたメモリの内容とデータバンクレジスタの内容で示される番地のメモリの内容が演算データとなります。ただし、命令のオペランドとダイレクトページレジスタの内容を加算した値がバンク0を越える場合は、バンク1内のメモリが指定されます。

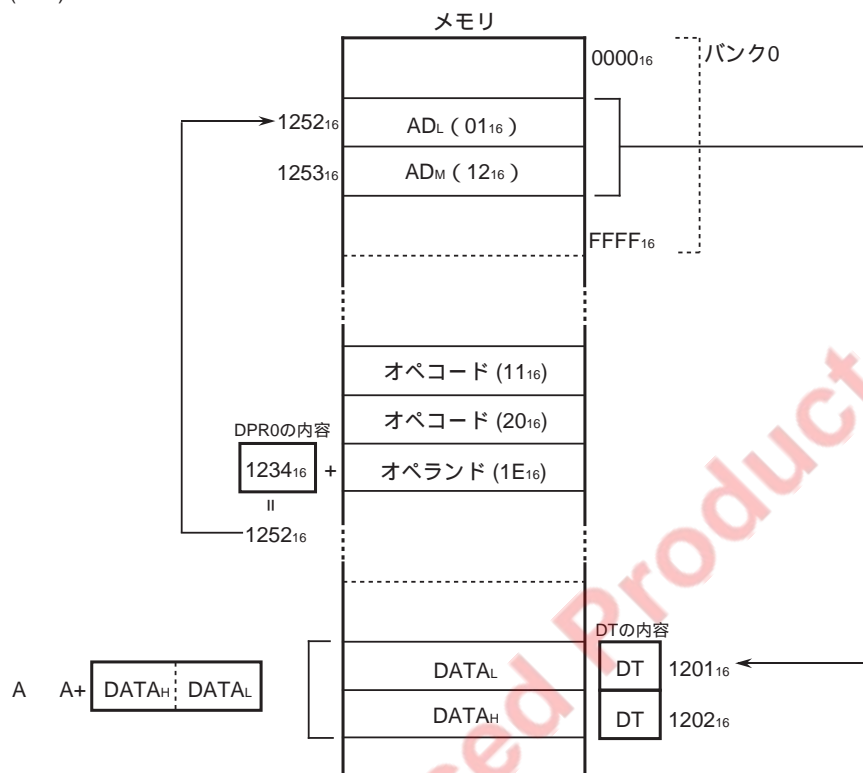
<ダイレクトページレジスタ切り替えビットが“0”の場合>

例：ニーモニック 機械語
 ADD A, (1EH) 11₁₆ 20₁₆ 1E₁₆
 (m=1)



例：ニーモニック
ADD A, (1EH)
(m=0)

機械語
11₁₆ 20₁₆ 1E₁₆



EOL announced Product

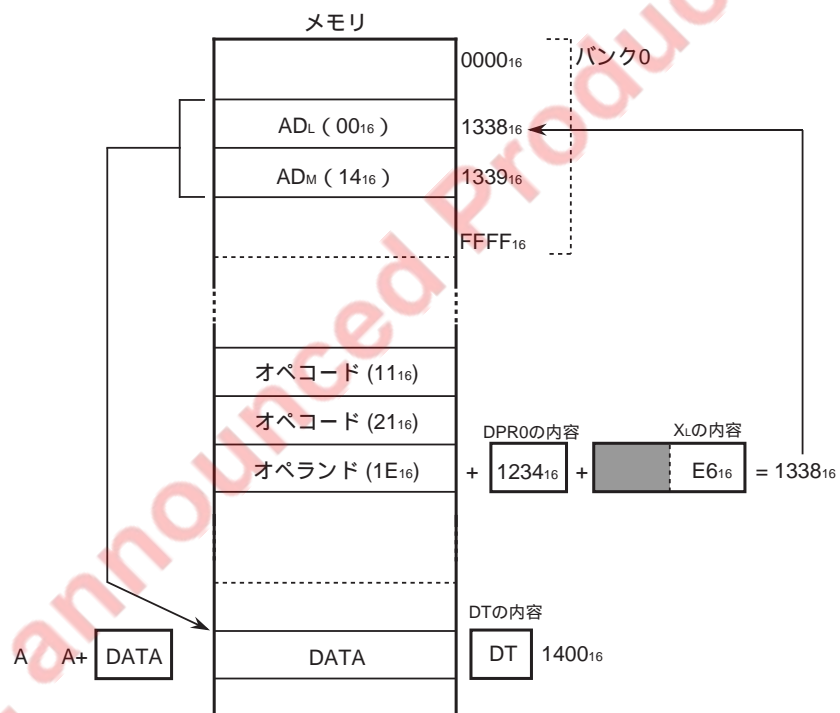
【名称】 ダイレクト・インデクストX・インダイレクト アドレッシングモード

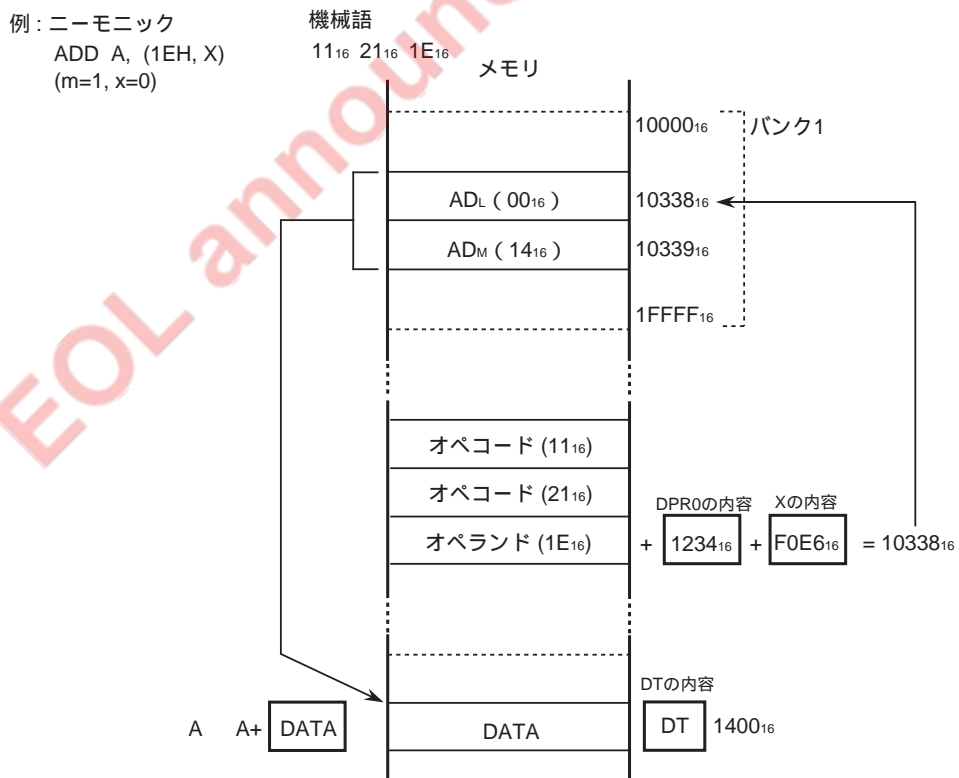
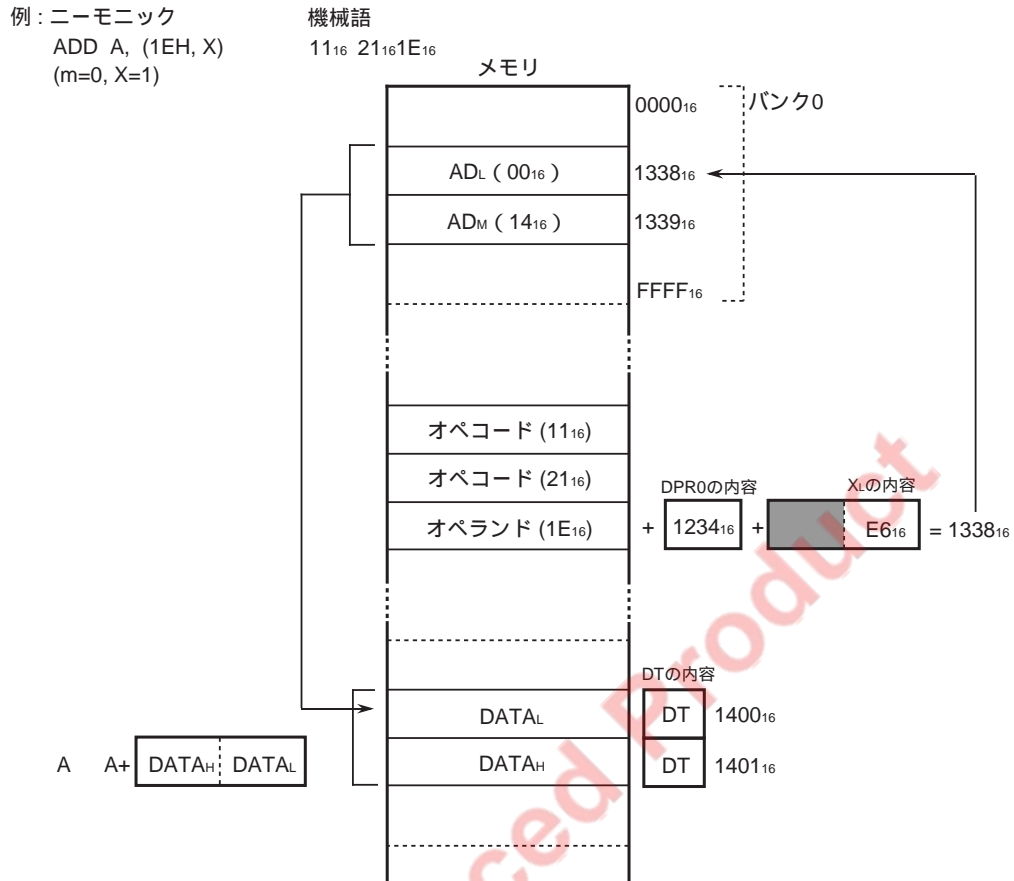
【機能】 命令のオペランドとダイレクトページレジスタの内容とインデックスレジスタXの内容を加算した値で指定されるバンク0内の連続する2バイトのメモリを指定します。指定されたメモリの内容とデータバンクレジスタの内容で示される番地のメモリの内容が演算データとなります。ただし、命令のオペランドとダイレクトページレジスタの内容とインデックスレジスタXの内容を加算した値がバンク0あるいはバンク1を越える場合は、バンク1あるいはバンク2内のメモリが指定されます。

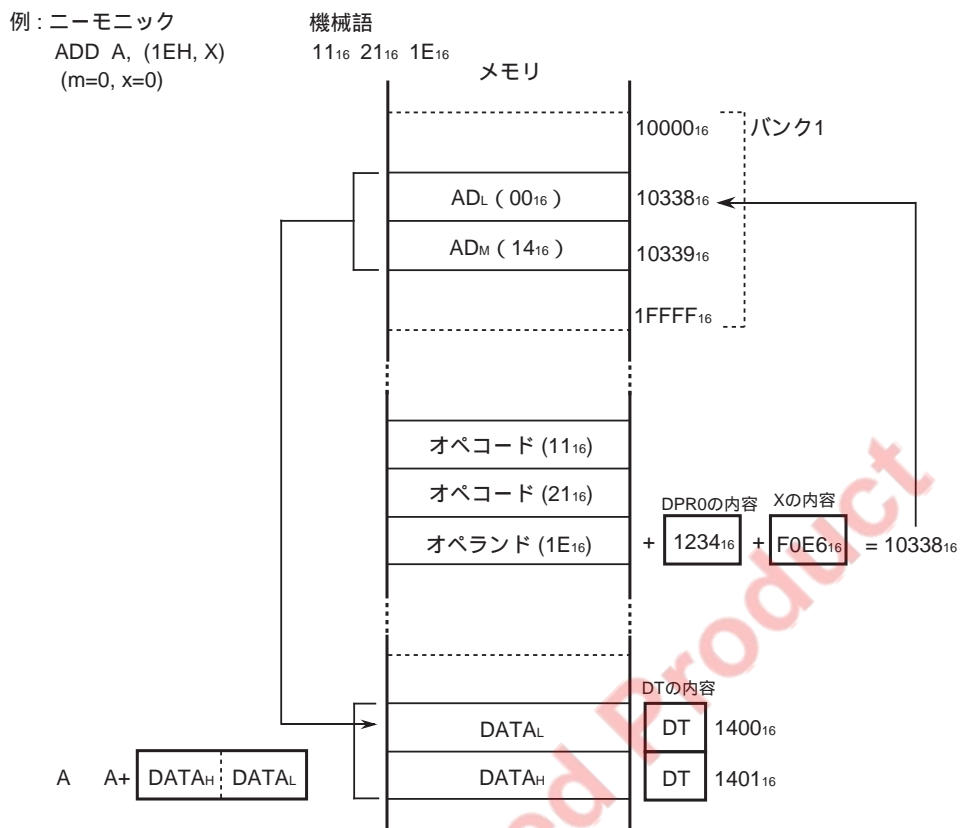
<ダイレクトページレジスタ切り替えビットが“0”の場合>

例：ニーモニック
ADD A, (1EH, X)
(m=1, x=1)

機械語
11₁₆ 21₁₆ 1E₁₆







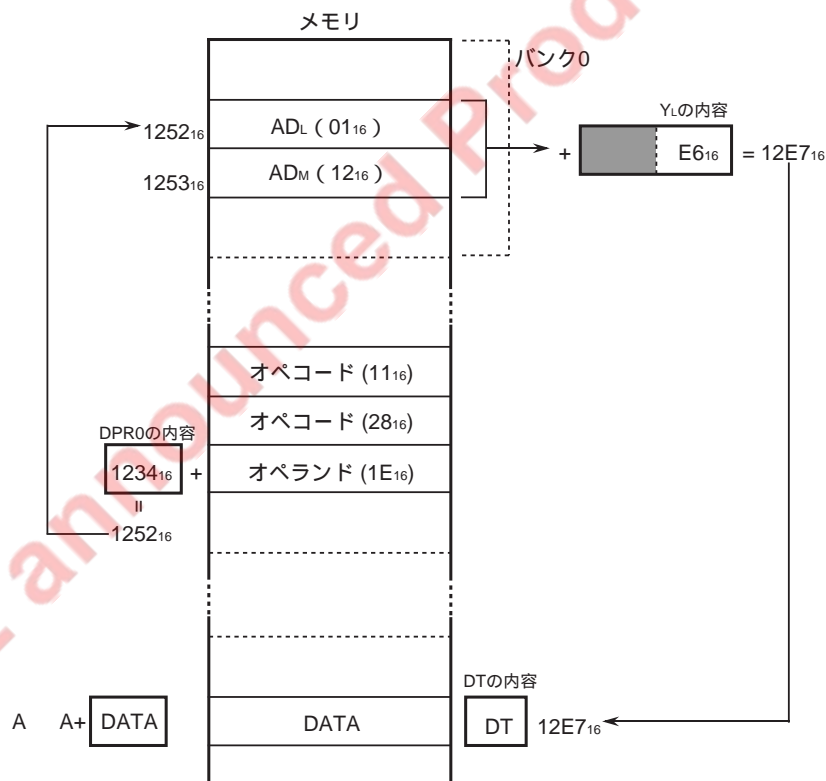
EOL announced product

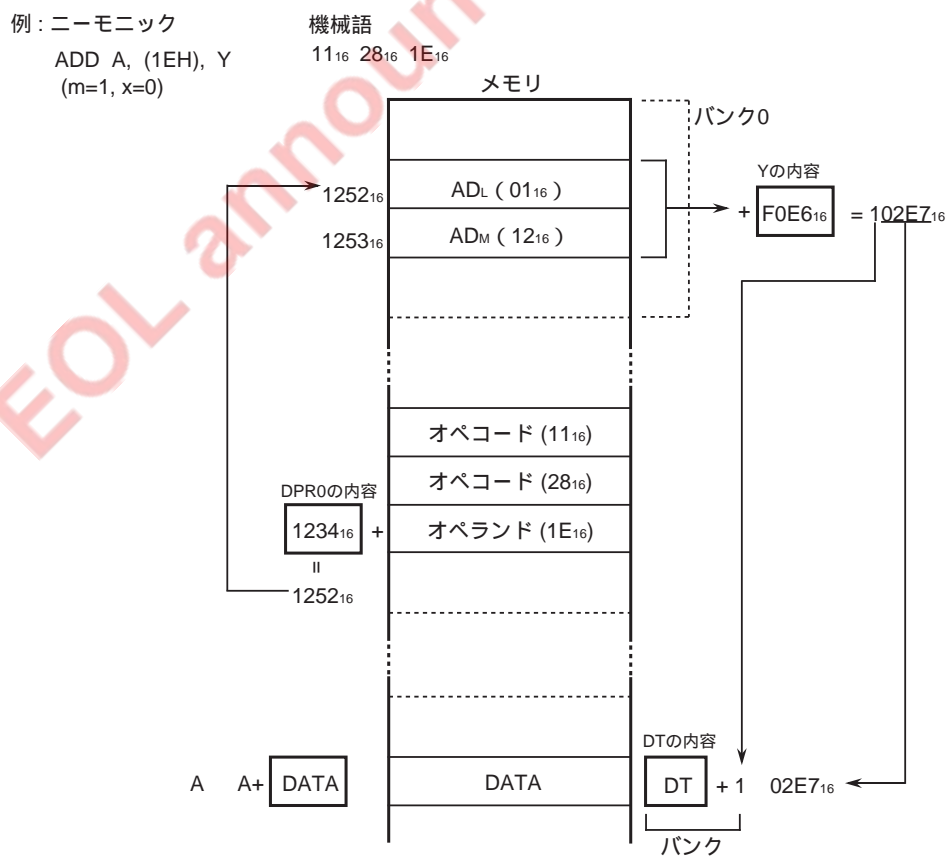
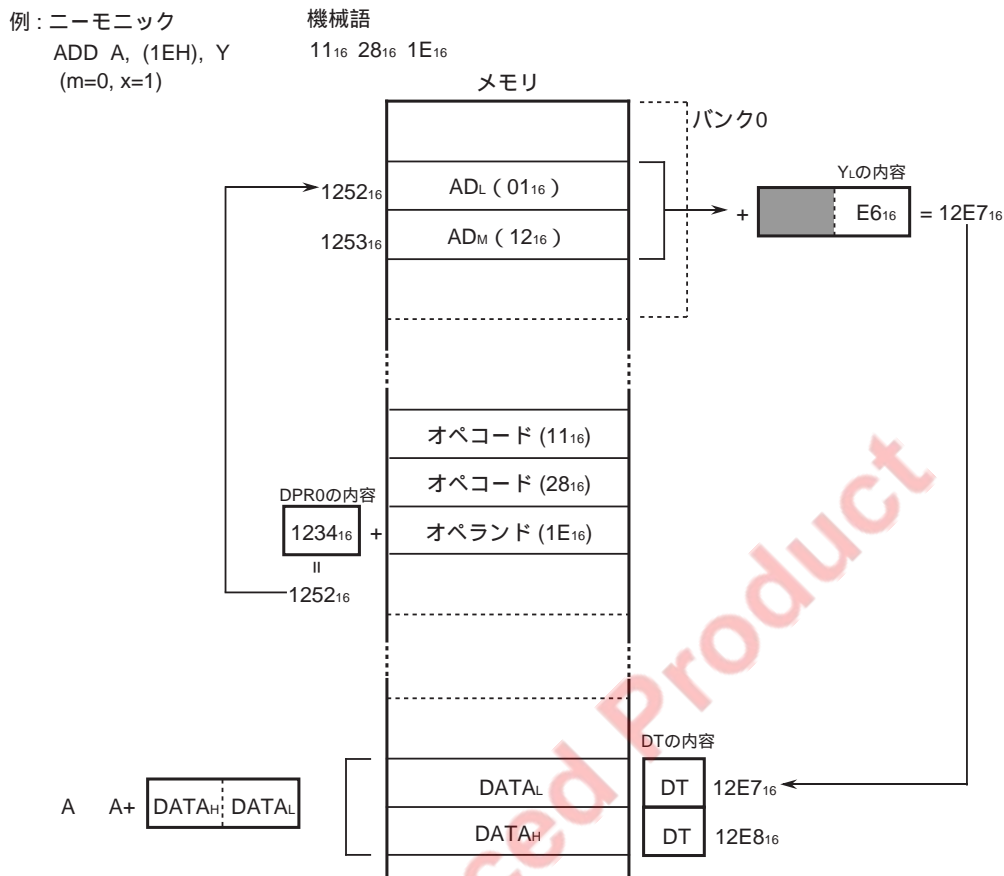
【名称】 ダイレクト・インダイレクト・インデクストY アドレッシングモード

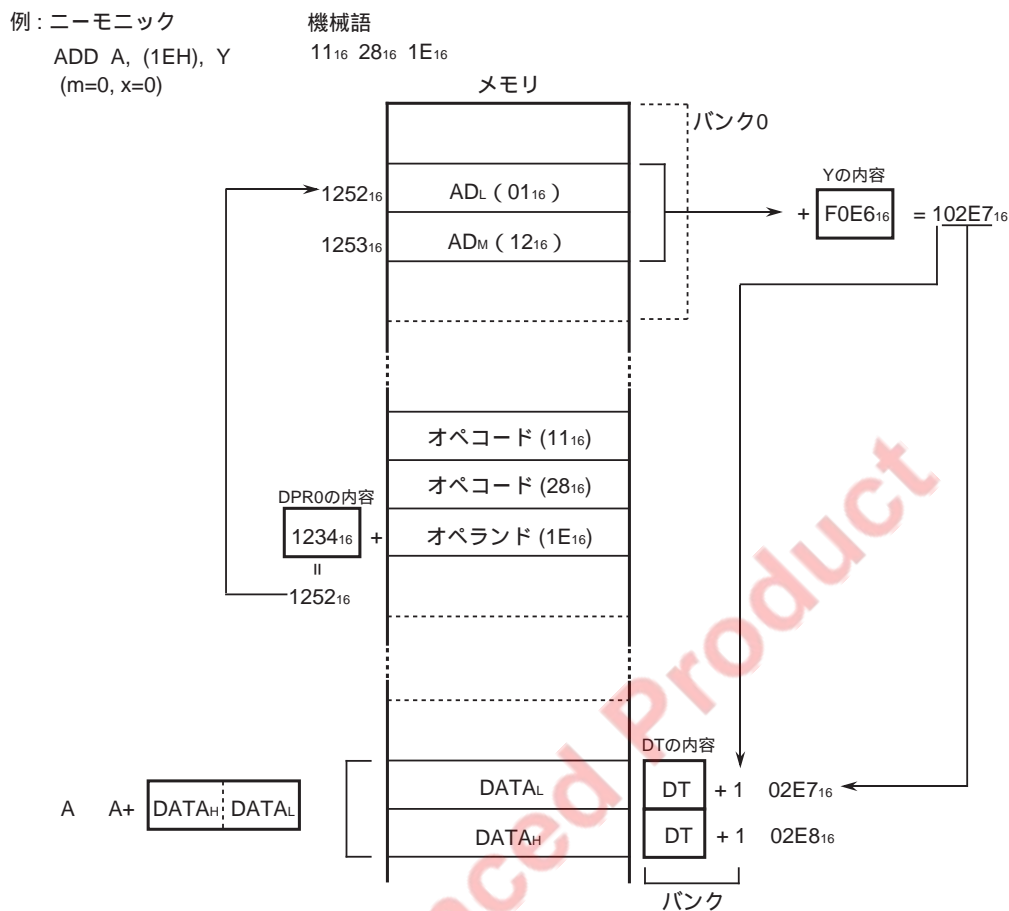
【機能】 命令のオペランドとダイレクトページレジスタの内容を加算した値で指定されるバンク0内の連続する2バイトのメモリを指定します。指定されたメモリの内容にインデックスレジスタYの内容を加算した値とデータバンクレジスタの内容で示される番地のメモリの内容が演算データとなります。ただし、命令のオペランドとダイレクトページレジスタの内容を加算した値がバンク0を越える場合はバンク1内のメモリが指定されます。また、メモリの内容にインデックスレジスタYの内容を加算したときキャリーが生じるとバンクは、データバンクレジスタの内容に1加算した値が使用されます。

<ダイレクトページレジスタ切り替えビットが“0”の場合>

例：ニーモニック 機械語
 ADD A, (1EH), Y 11₁₆ 28₁₆ 1E₁₆
 (m=1, x=1)







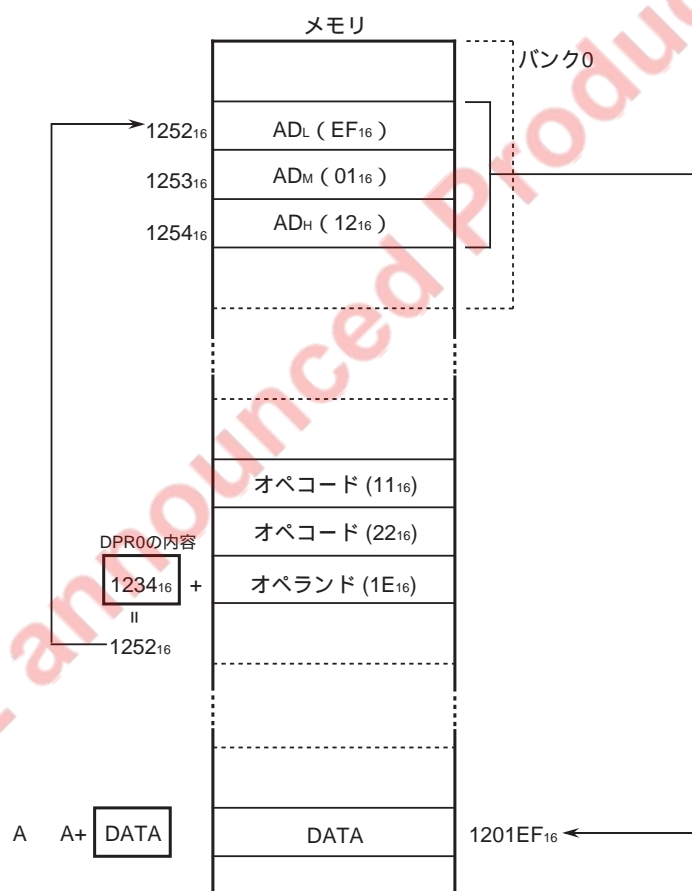
EOL announced Product

【名称】 ダイレクト・インダイレクトロング アドレッシングモード

【機能】 命令のオペランドとダイレクトページレジスタの内容を加算した値でバンク0内の連続する3バイトのメモリを指定します。指定されたメモリの内容で示される番地のメモリの内容が演算データとなります。ただし、命令のオペランドとダイレクトページレジスタの内容を加算した値がバンク0を越える場合はバンク1内のメモリが指定されます。3バイトのメモリはバンク間にまたがってもかまいません。

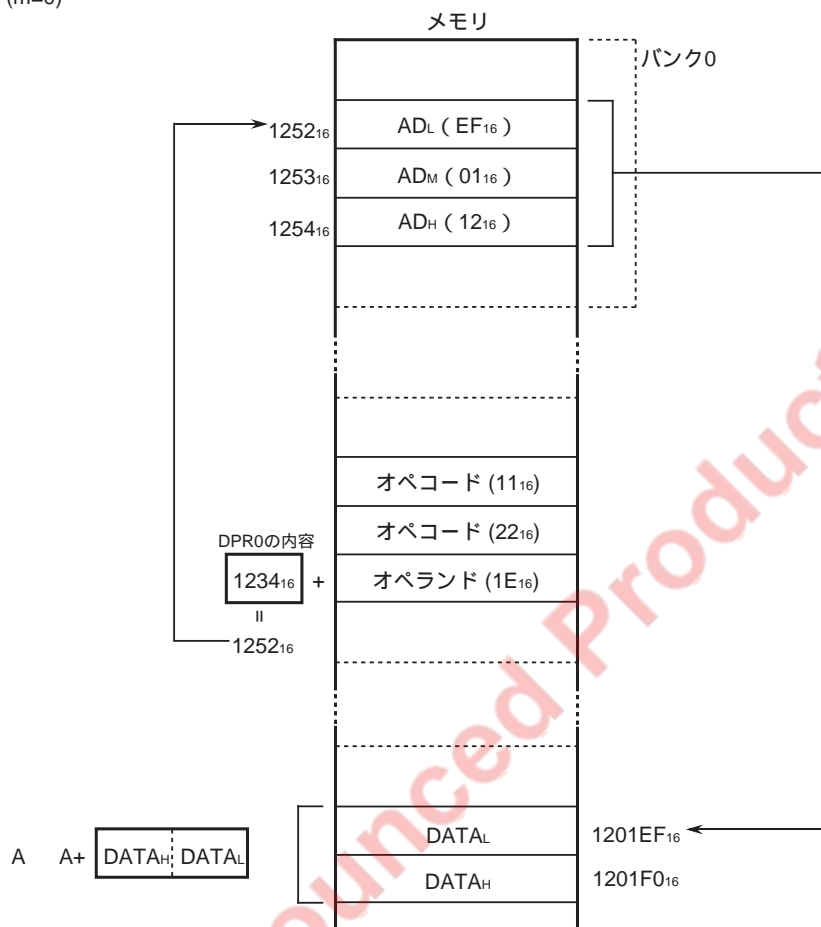
<ダイレクトページレジスタ切り替えビットが“0”の場合>

例：ニーモニック 機械語
 ADD A, L(1EH) 11₁₆ 22₁₆ 1E₁₆
 (m=1)



例：ニーモニック
ADD A, L (1EH)
(m=0)

機械語
11₁₆ 22₁₆ 1E₁₆



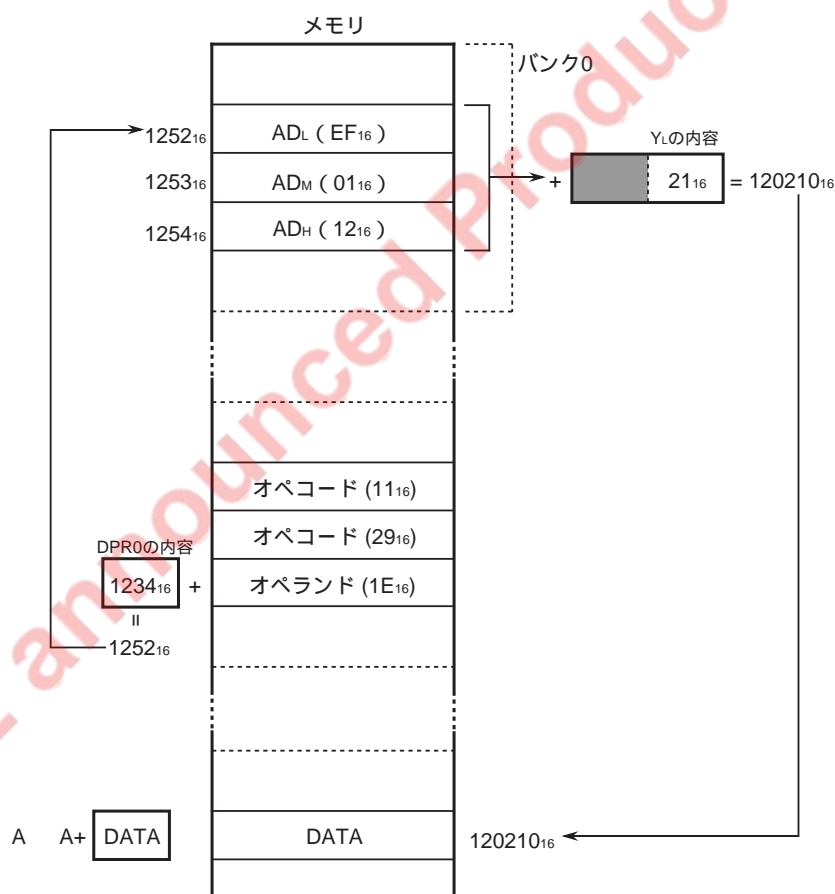
EOL announced Product

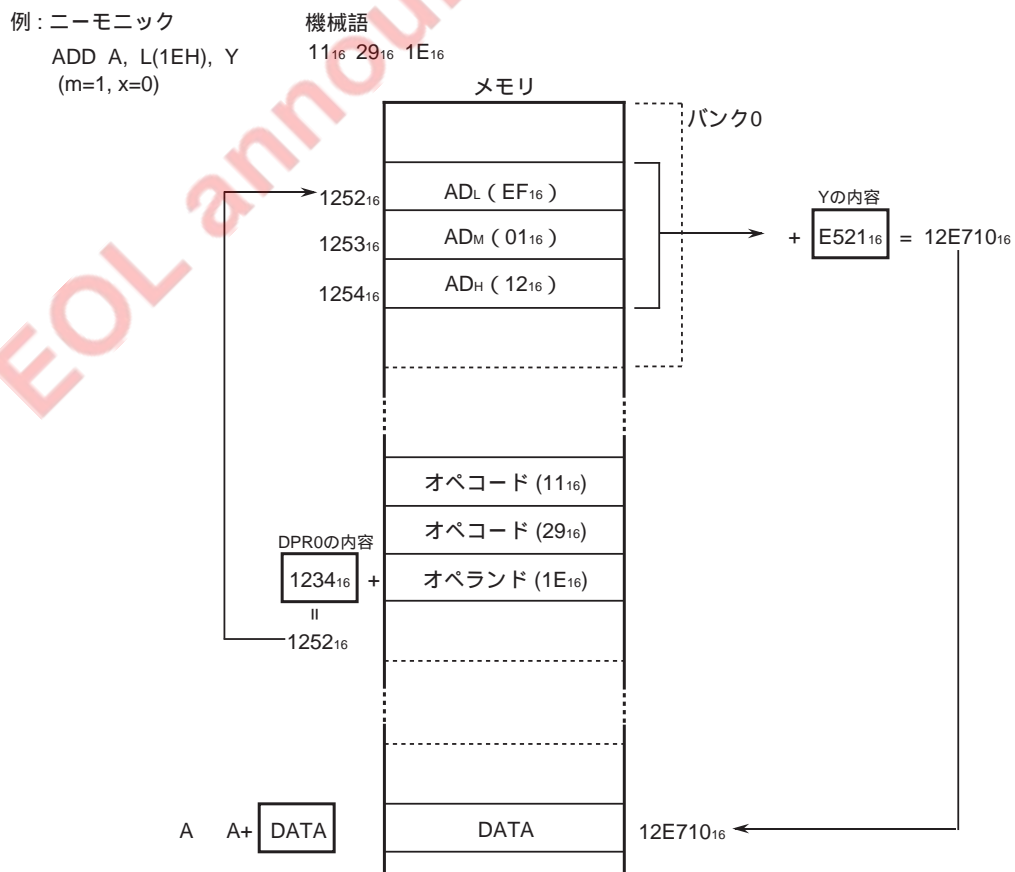
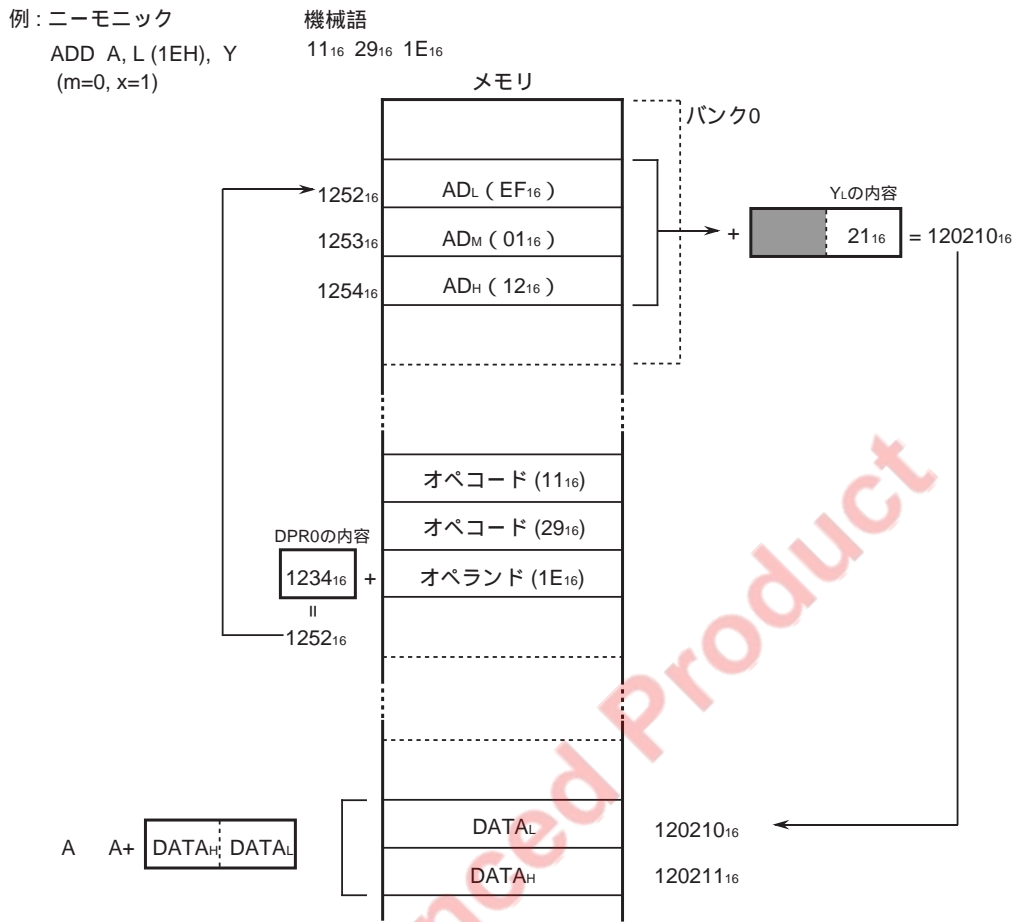
【名称】 ダイレクト・インダイレクトロング・インデクストY アドレッシングモード

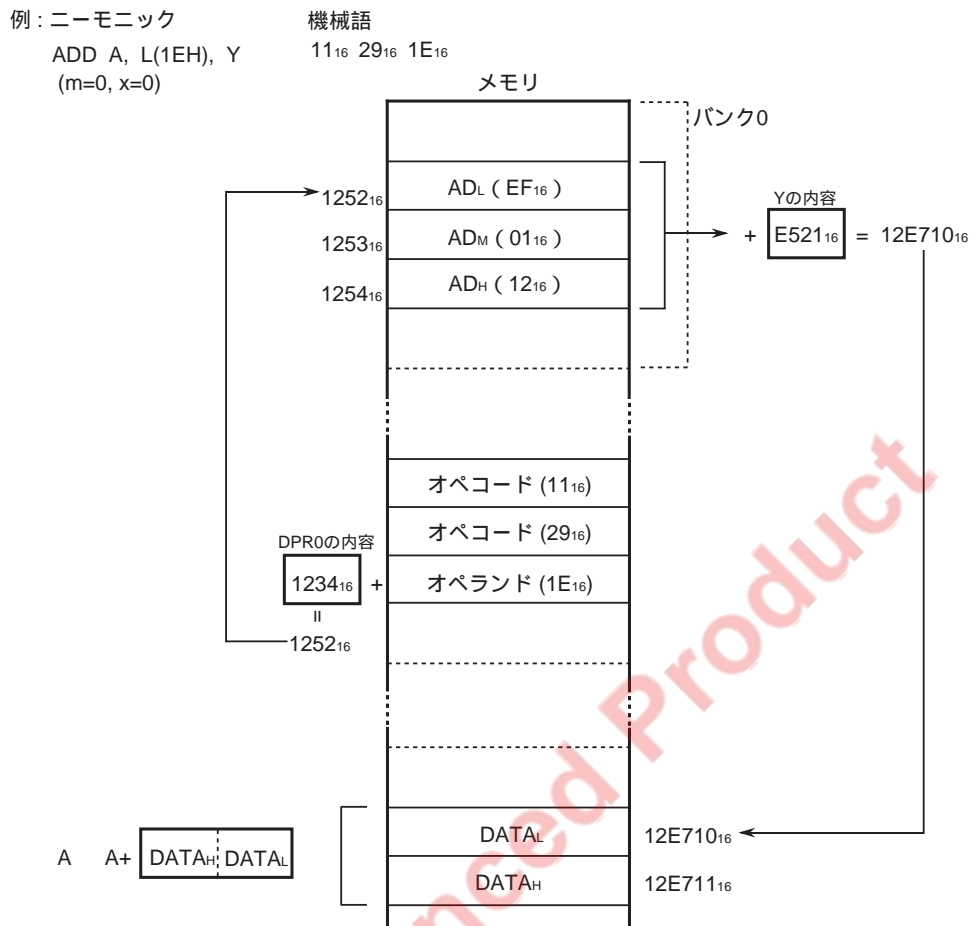
【機能】 命令のオペランドとダイレクトページレジスタの内容を加算した値でバンク0内の連続する3バイトのメモリを指定します。指定されたメモリの内容にインデックスレジスタYの内容を加算した値で示される番地のメモリの内容が演算データとなります。ただし、命令のオペランドとダイレクトページレジスタの内容を加算した値がバンク0を越える場合はバンク1内のメモリが指定されます。3バイトのメモリはバンク間にまたがってもかまいません。

<ダイレクトページレジスタ切り替えビットが“0”の場合>

例：ニーモニック 機械語
 ADD A, L (1EH), Y 11₁₆ 29₁₆ 1E₁₆
 (m=1, x=1)



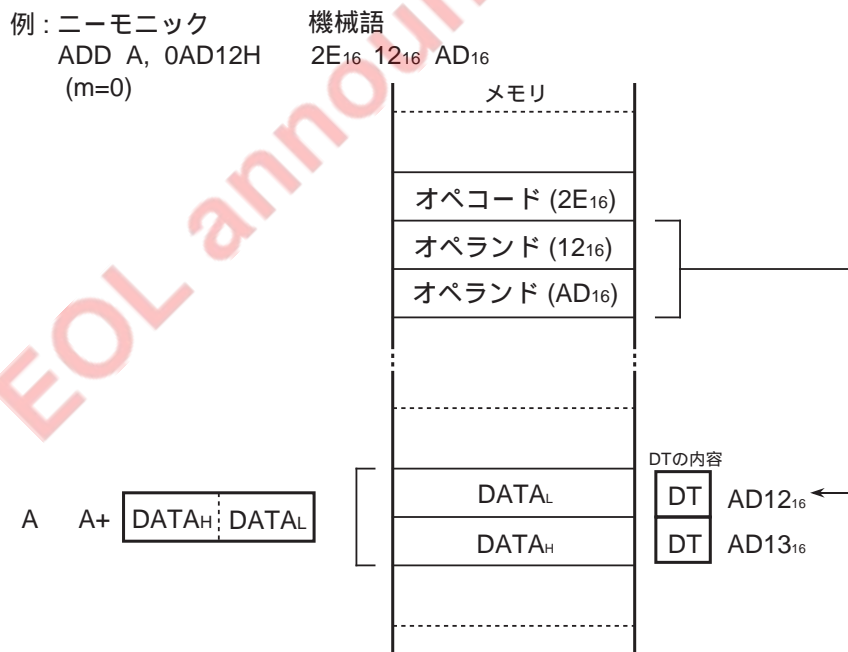
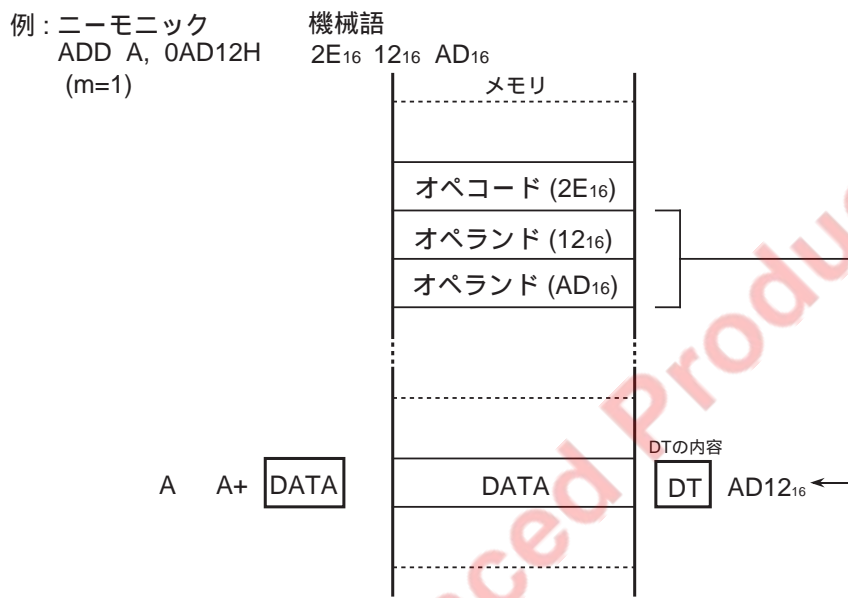




EOL announced Product

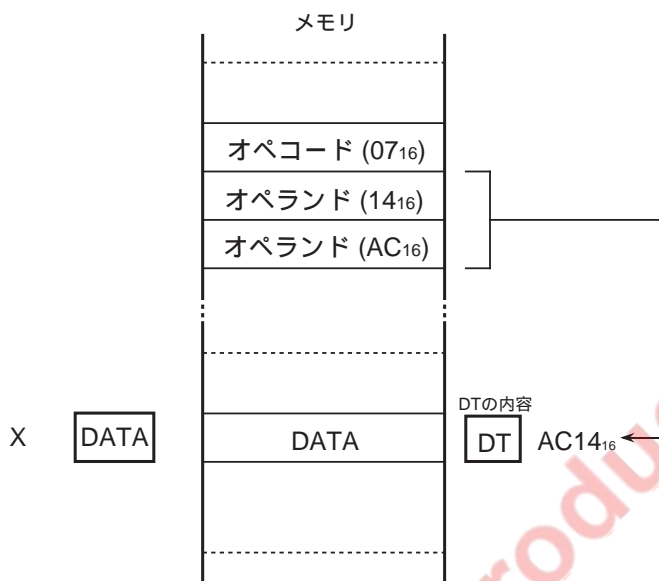
【名称】 アブソリュート アドレッシングモード

【機能】 命令のオペランドとデータバンクレジスタで指定されるメモリの内容が演算データとなります。ただし、JMP命令、JSR命令の場合は命令のオペランドがプログラムカウンタに入ります。



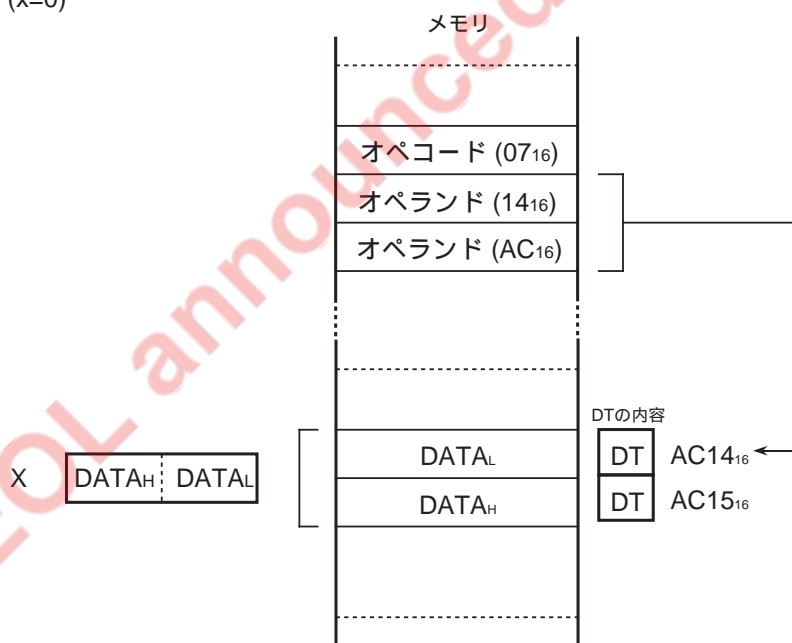
例：ニーモニック
LDX 0AC14H
(x=1)

機械語
07₁₆ 14₁₆ AC₁₆



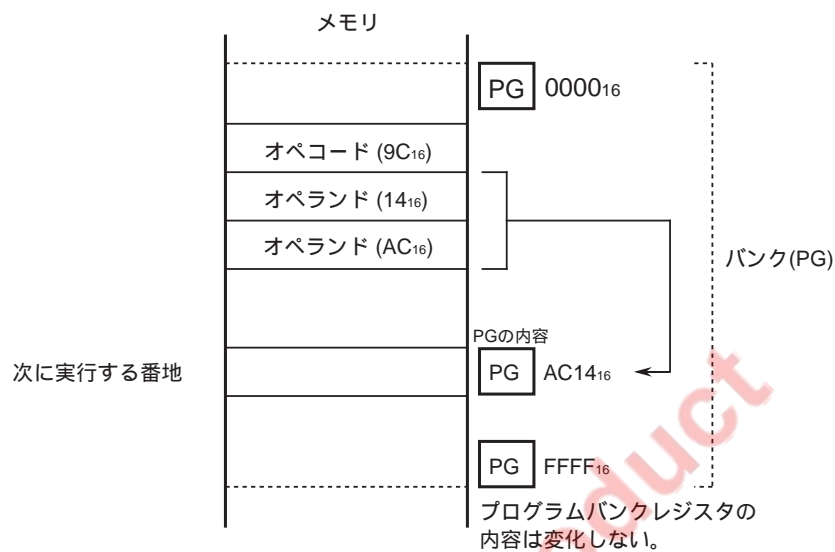
例：ニーモニック
LDX 0AC14H
(x=0)

機械語
07₁₆ 14₁₆ AC₁₆



例：ニーモニック
JMP 0AC14H

機械語
9C₁₆ 14₁₆ AC₁₆



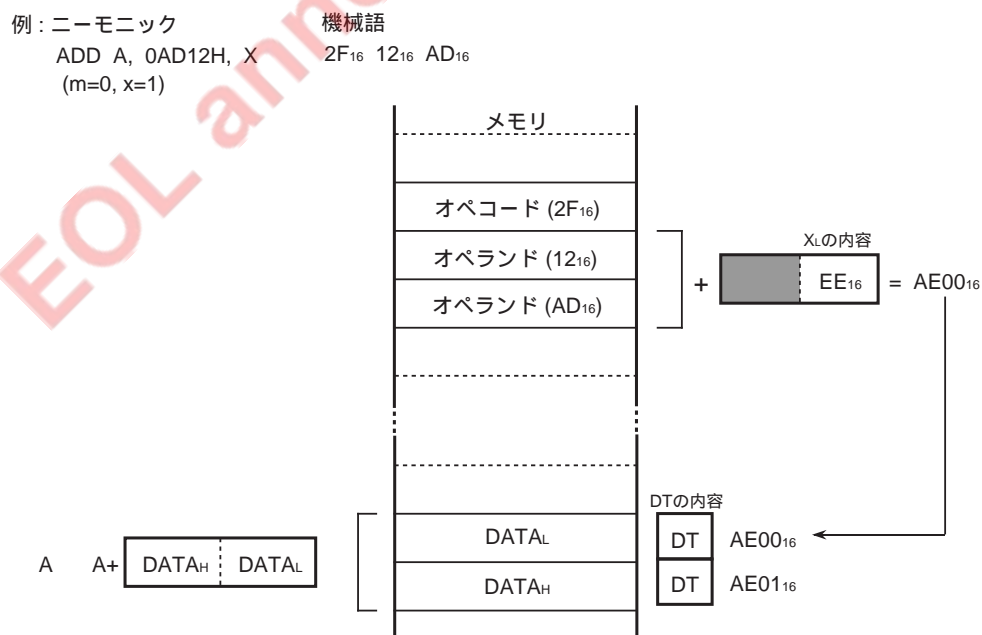
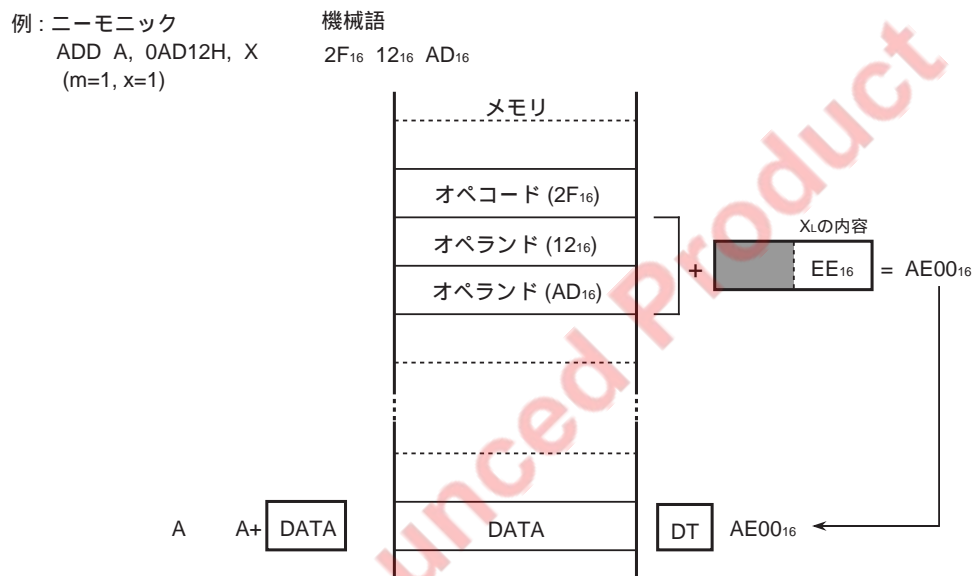
注 . JMP命令及びJSR命令では、命令がバンク境界付近に配置される場合は、飛び先のバンクに注意する必要があります。

JMP/JMPL命令の詳細説明P4-106を参照してください。

JSR/JSRL命令の詳細説明P4-107を参照してください。

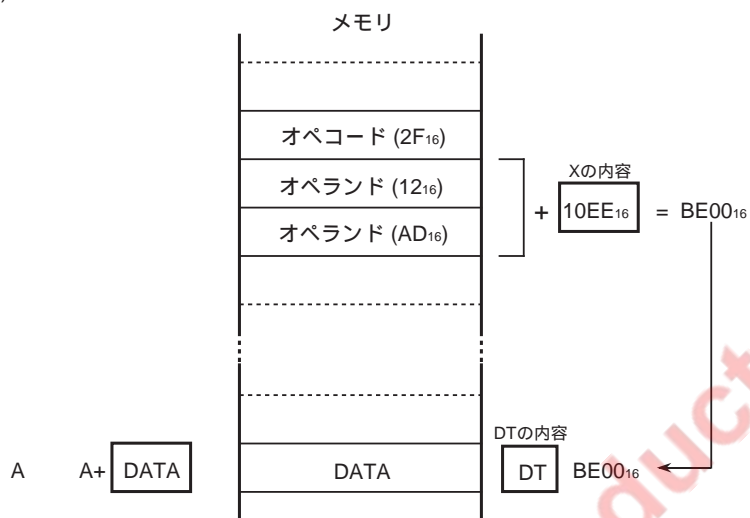
【名称】 アブソリュート・インデクストX アドレッシングモード

【機能】 命令のオペランドで表される16ビット長の数値にインデックスレジスタXの内容を加算した値とデータバンクレジスタの内容で指定される番地のメモリの内容が演算データとなります。ただし、命令のオペランドで表される数値にインデックスレジスタXの内容を加算したときキャリーが生じると、バンクはデータバンクレジスタの内容に1加算した値が使用されます。



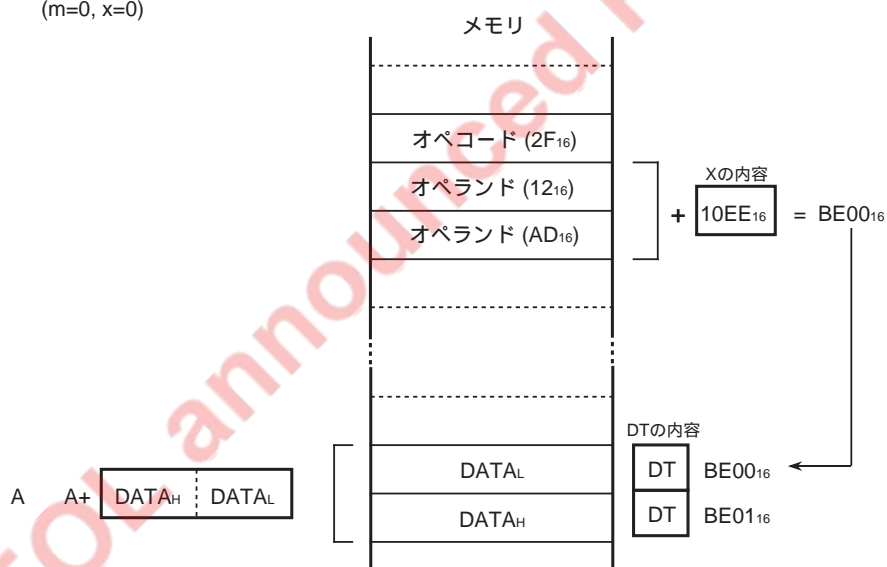
例：ニーモニック
ADD A, 0AD12H, X
(m=1, x=0)

機械語
2F₁₆ 12₁₆ AD₁₆



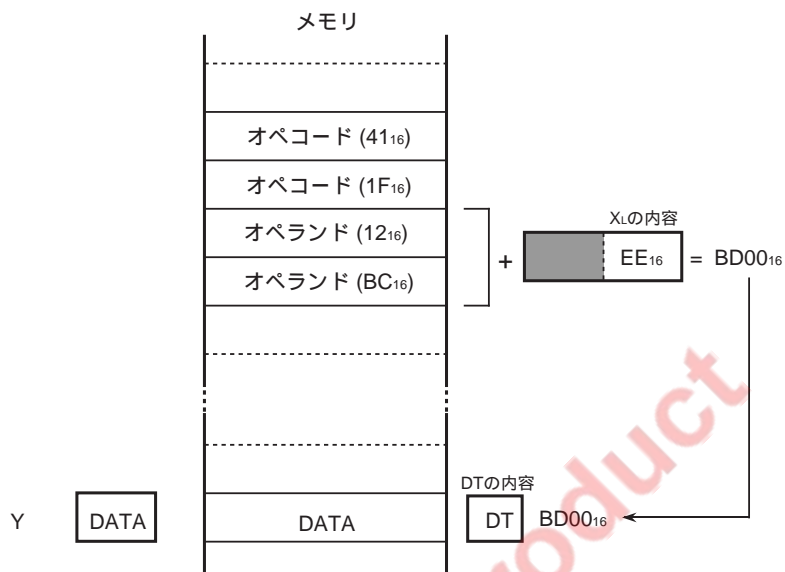
例：ニーモニック
ADD A, 0AD12H, X
(m=0, x=0)

機械語
2F₁₆ 12₁₆ AD₁₆



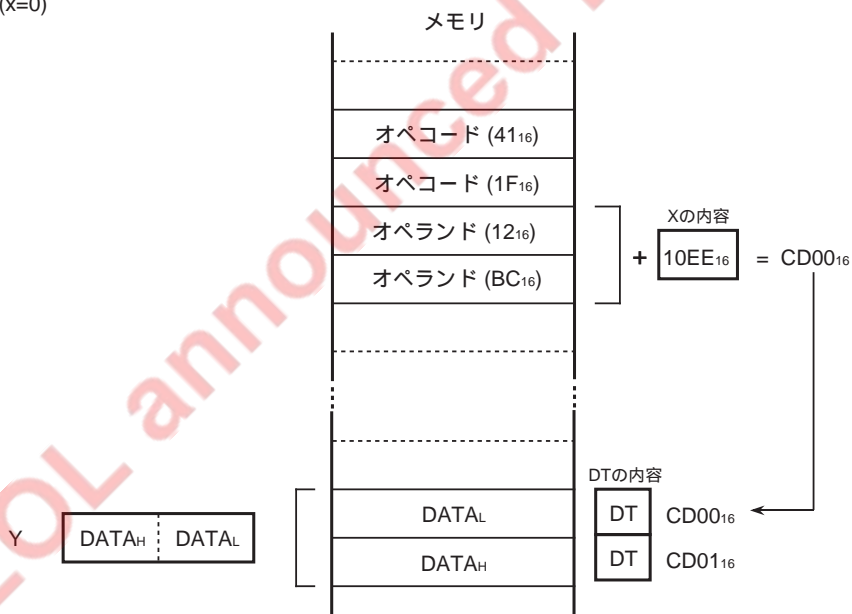
例: ニーモニック
LDY 0BC12H, X
(x=1)

機械語
41₁₆ 1F₁₆ 12₁₆ BC₁₆



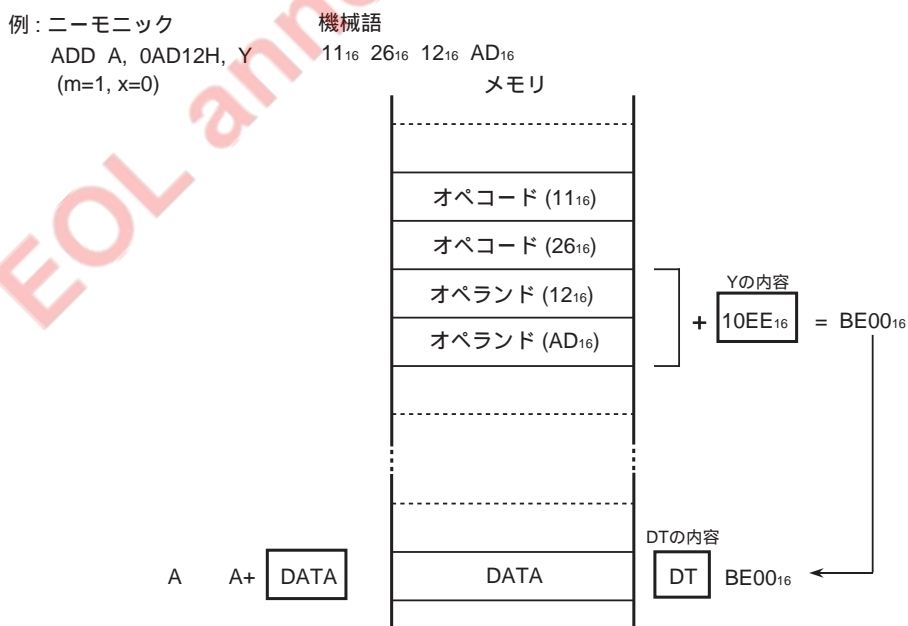
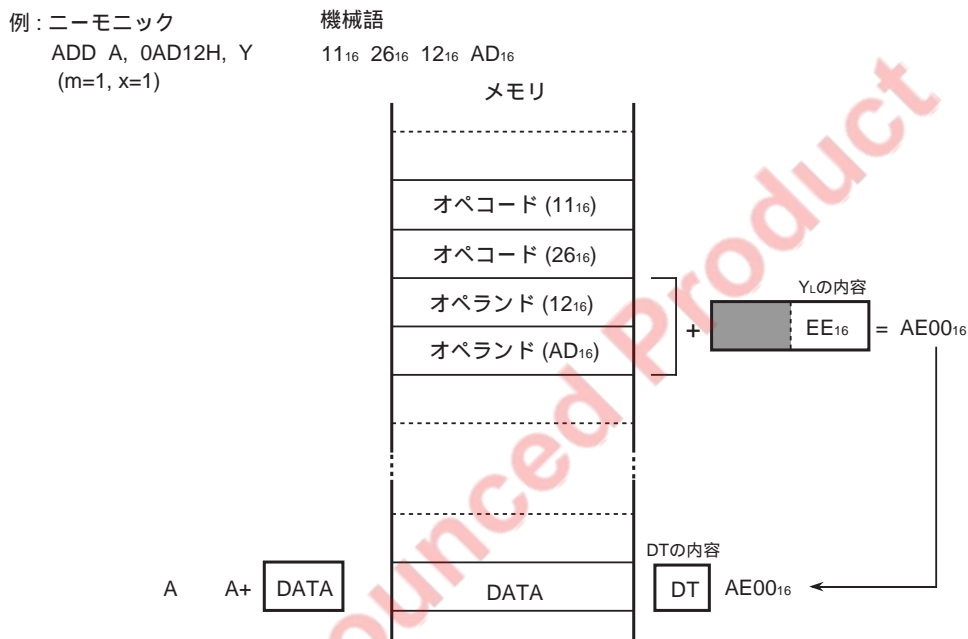
例: ニーモニック
LDY 0BC12H, X
(x=0)

機械語
41₁₆ 1F₁₆ 12₁₆ BC₁₆



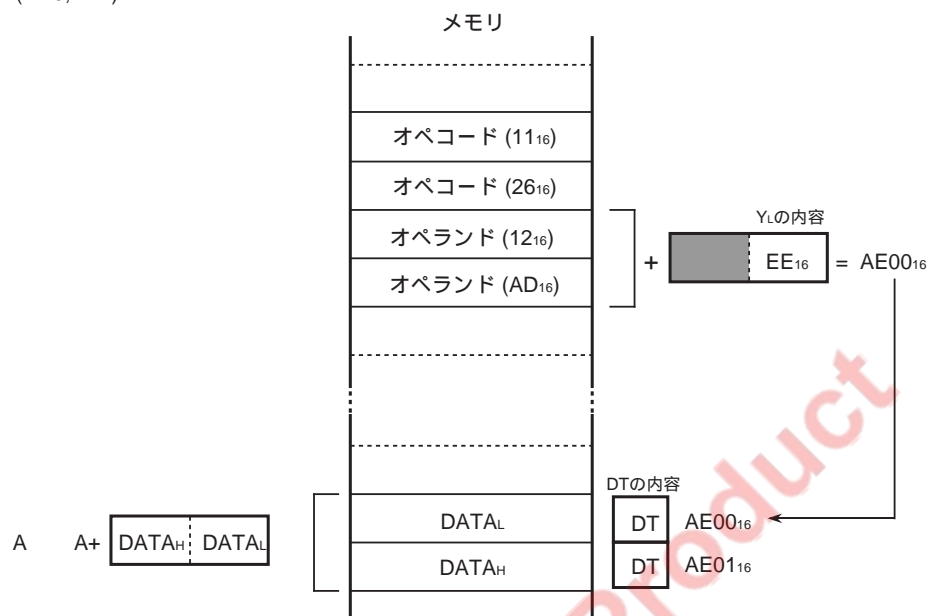
【名称】 アブソリュート・インデクストY アドレッシングモード

【機能】 命令のオペランドで表される16ビット長の数値にインデックスレジスタYの内容を加算した値とデータバンクレジスタの内容で指定される番地のメモリの内容が演算データとなります。ただし、命令のオペランドで表される数値にインデックスレジスタYの内容を加算したときキャリーが生じると、バンクはデータバンクレジスタの内容に1加算した値が使用されます。



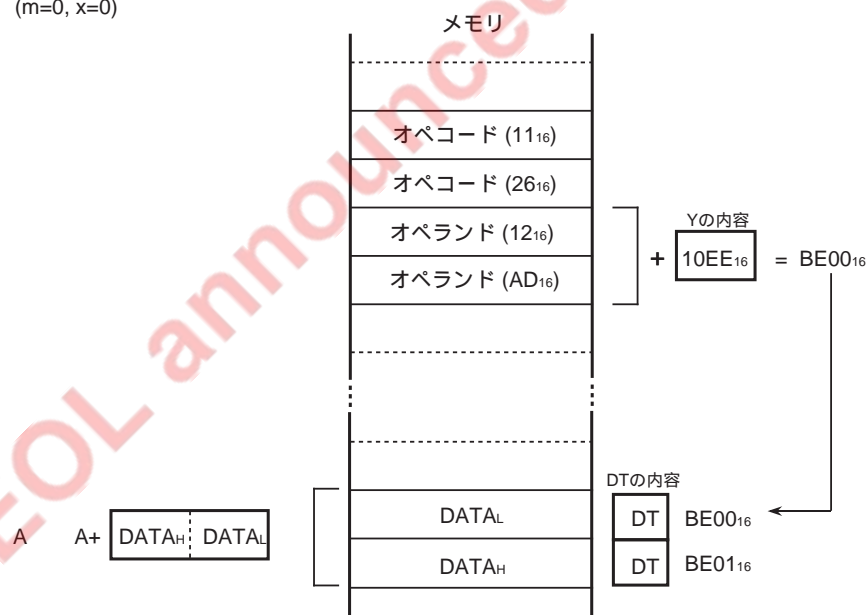
例：ニーモニック
ADD A, 0AD12H, Y
(m=0, x=1)

機械語
11₁₆ 26₁₆ 12₁₆ AD₁₆



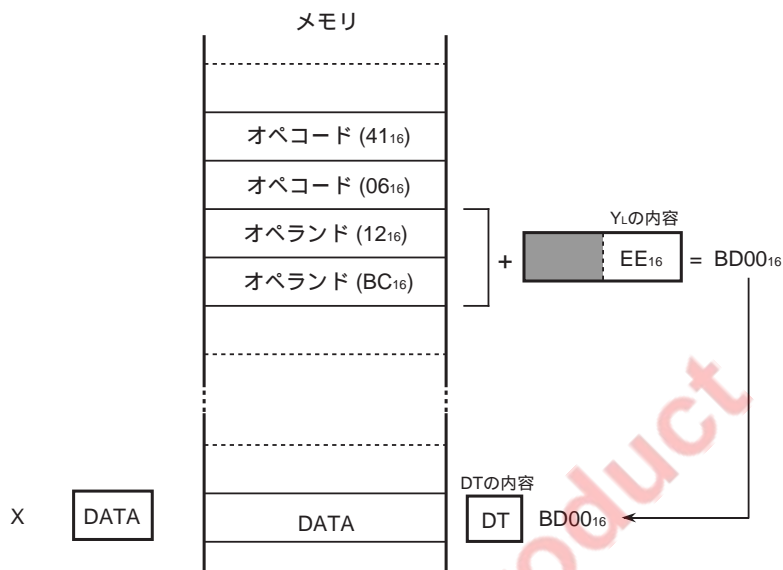
例：ニーモニック
ADD A, 0AD12H, Y
(m=0, x=0)

機械語
11₁₆ 26₁₆ 12₁₆ AD₁₆



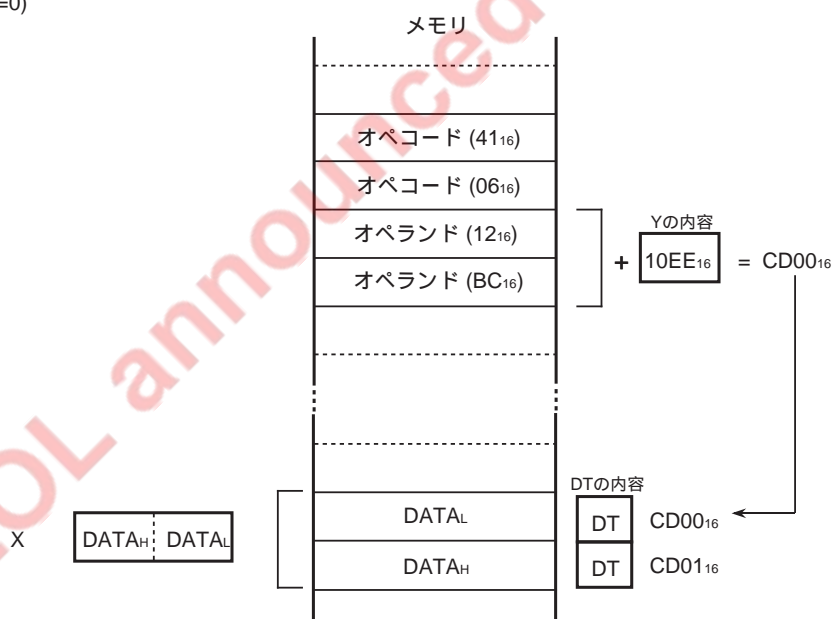
例：ニーモニック
LDX 0BC12H, Y
(x=1)

機械語
41₁₆ 06₁₆ 12₁₆ BC₁₆



例：ニーモニック
LDX 0BC12H, Y
(x=0)

機械語
41₁₆ 06₁₆ 12₁₆ BC₁₆

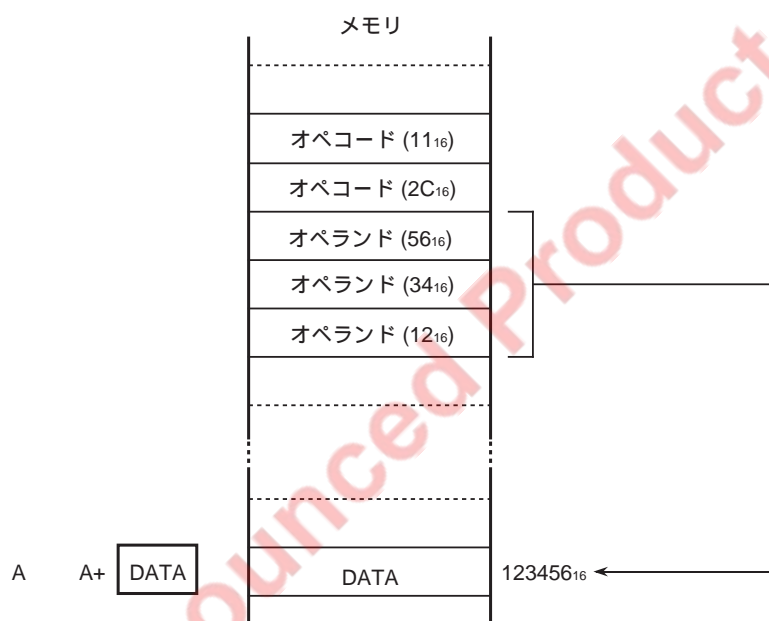


【名称】 アブソリュートロング アドレッシングモード

【機能】 命令のオペランドで指定される番地のメモリの内容が演算データとなります。ただし、JMPL命令、JSRL命令では命令の2バイト目と3バイト目がプログラムカウンタに、4バイト目がプログラムバンクレジスタに入ります。

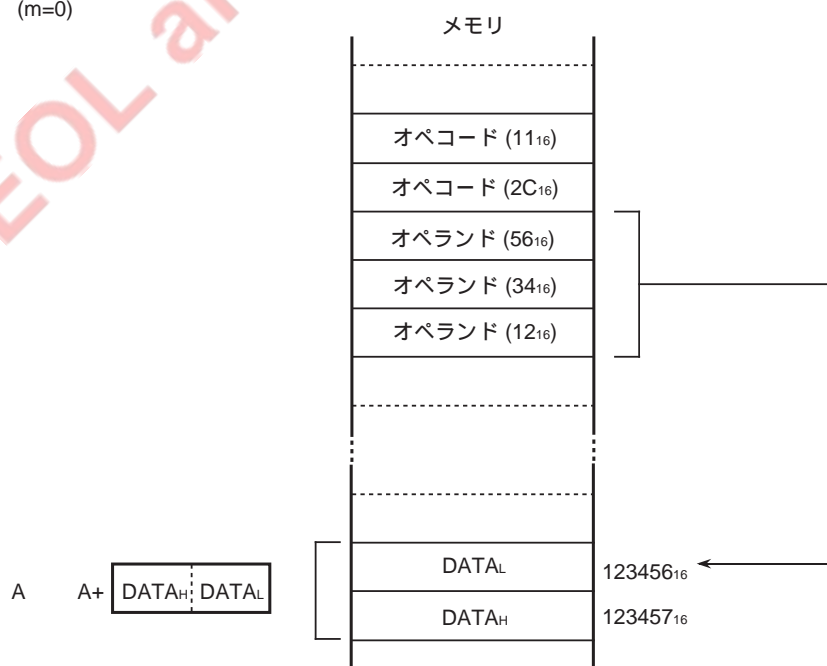
例：ニーモニック
ADD A, 123456H
(m=1)

機械語
 11_{16} $2C_{16}$ 56_{16} 34_{16} 12_{16}



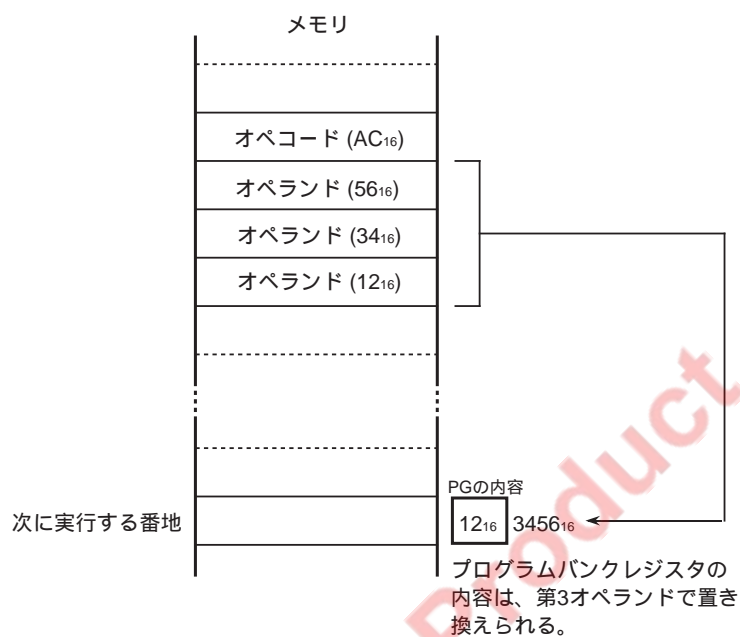
例：ニーモニック
ADD A, 123456H
(m=0)

機械語
 11_{16} $2C_{16}$ 56_{16} 34_{16} 12_{16}



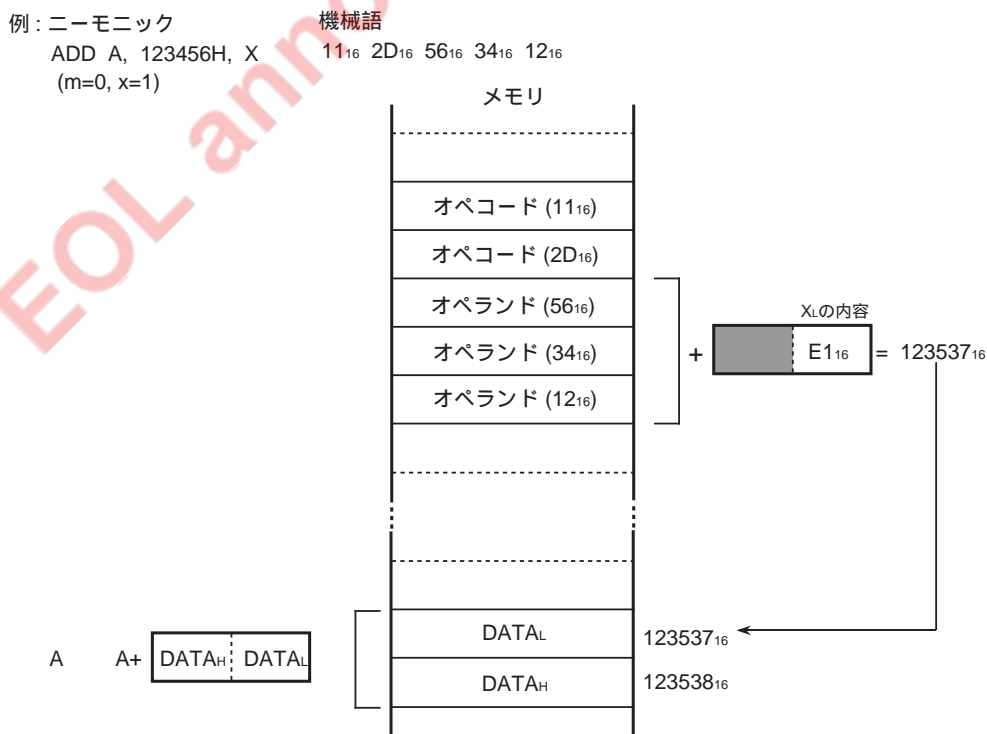
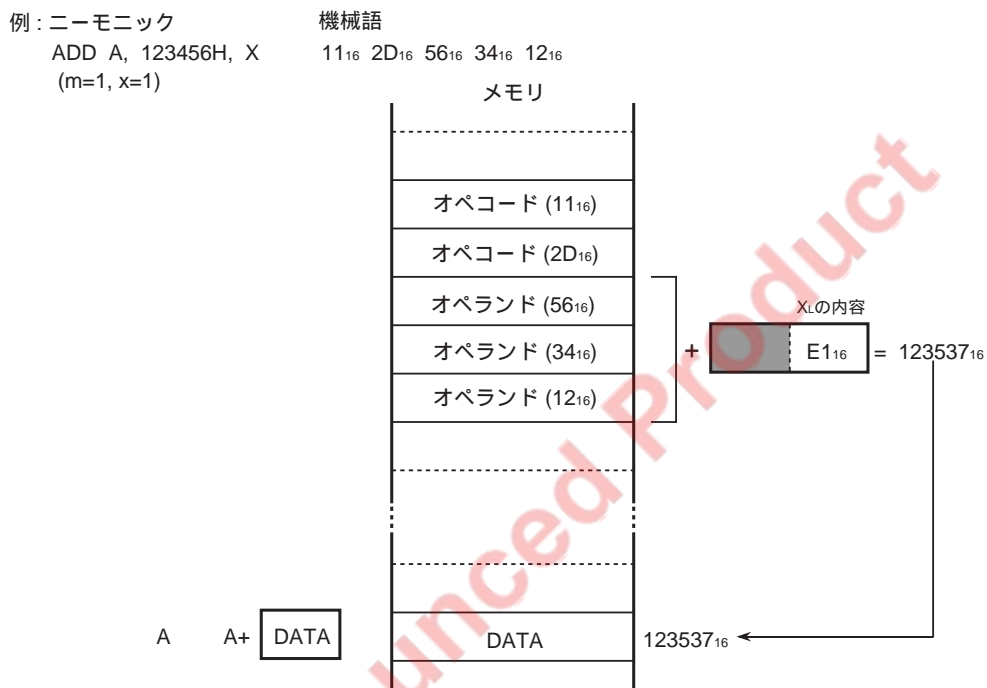
例：ニーモニック
JMPL 123456H

機械語
AC₁₆ 56₁₆ 34₁₆ 12₁₆



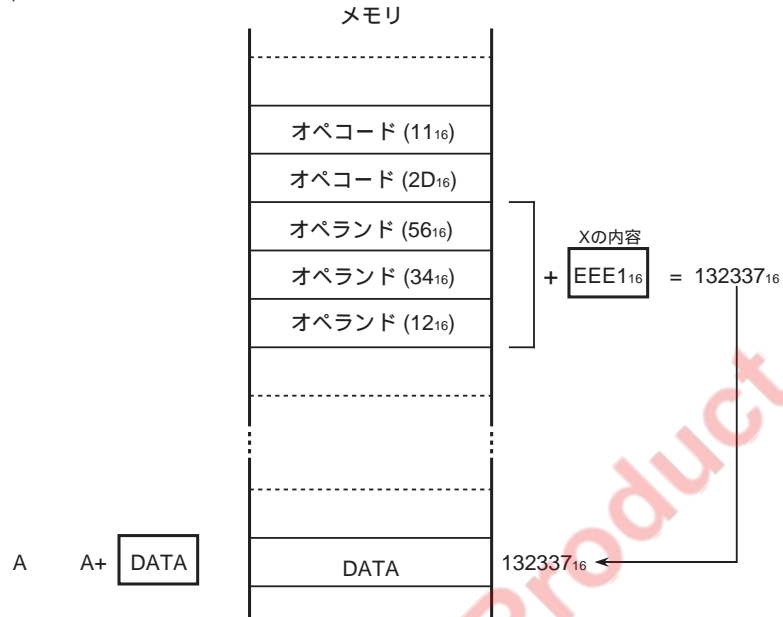
【名称】 アブソリュートロング・インデクストX アドレッシングモード

【機能】 命令のオペランドで表される数値にインデックスレジスタXの内容を加算した値で指定される番地のメモリの内容が演算データとなります。



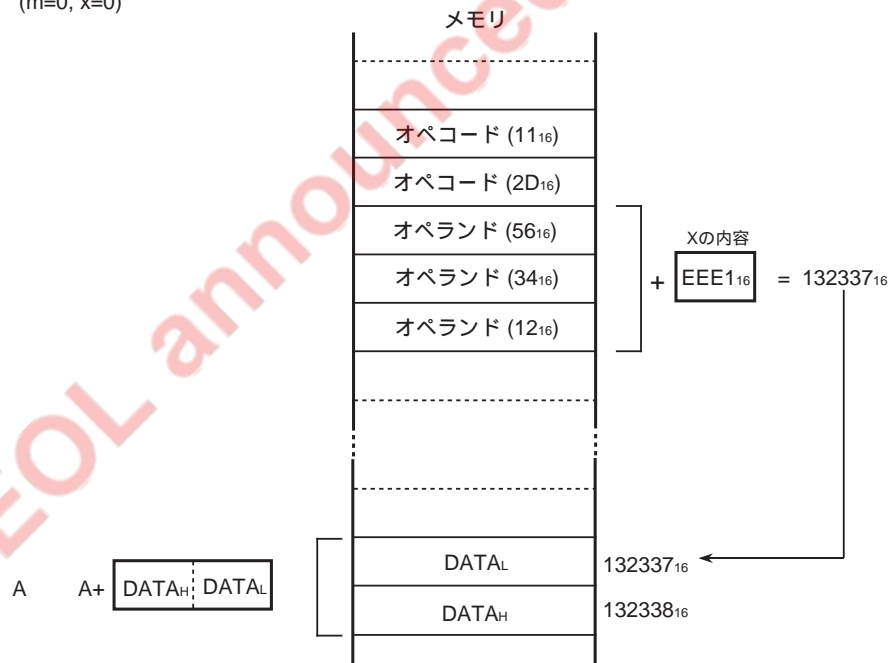
例：ニーモニック
ADD A, 123456H, X
(m=1, x=0)

機械語
11₁₆ 2D₁₆ 56₁₆ 34₁₆ 12₁₆



例：ニーモニック
ADD A, 123456H, X
(m=0, x=0)

機械語
11₁₆ 2D₁₆ 56₁₆ 34₁₆ 12₁₆

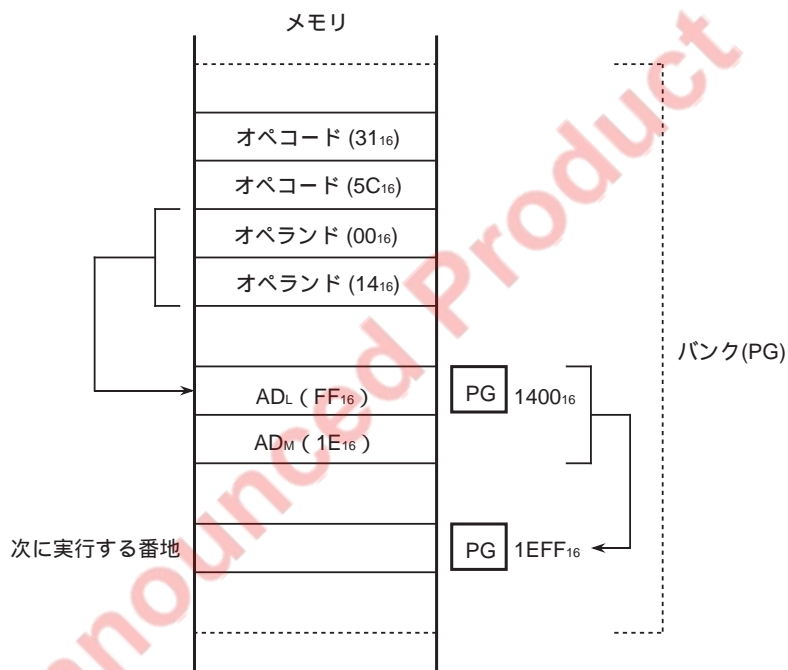


【名称】 アブソリュート・インダイレクト アドレッシングモード

【機能】 命令の3バイト目と4バイト目で同一プログラムバンク内の連続する2バイトのメモリを指定し、さらにこれらのメモリの内容で示される同一プログラムバンク内の番地へジャンプします。JMP命令で使用します。

例：ニーモニック
JMP (1400H)

機械語
31₁₆ 5C₁₆ 00₁₆ 14₁₆



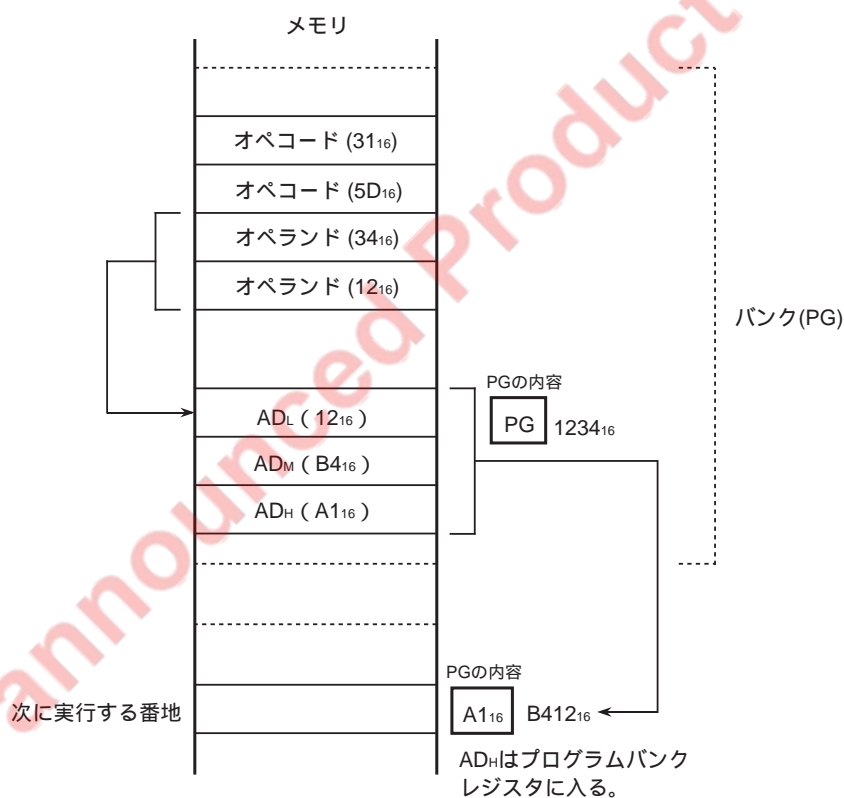
注．命令及び参照先の番地がバンク境界付近に配置される場合は、参照先及び飛び先のバンクに注意する必要があります。

JMP/JMPL命令の詳細説明P4-106を参照してください。

- 【名称】 アブソリュート・インダイレクトロング アドレッシングモード
- 【機能】 命令の3バイト目と4バイト目で連続する同一プログラムバンク内の3バイトのメモリを指定し、さらにこれらのメモリの内容で示される番地へジャンプします。
JMPL命令で使用します。

例：ニーモニック
JMPL L(1234H)

機械語
31₁₆ 5D₁₆ 34₁₆ 12₁₆



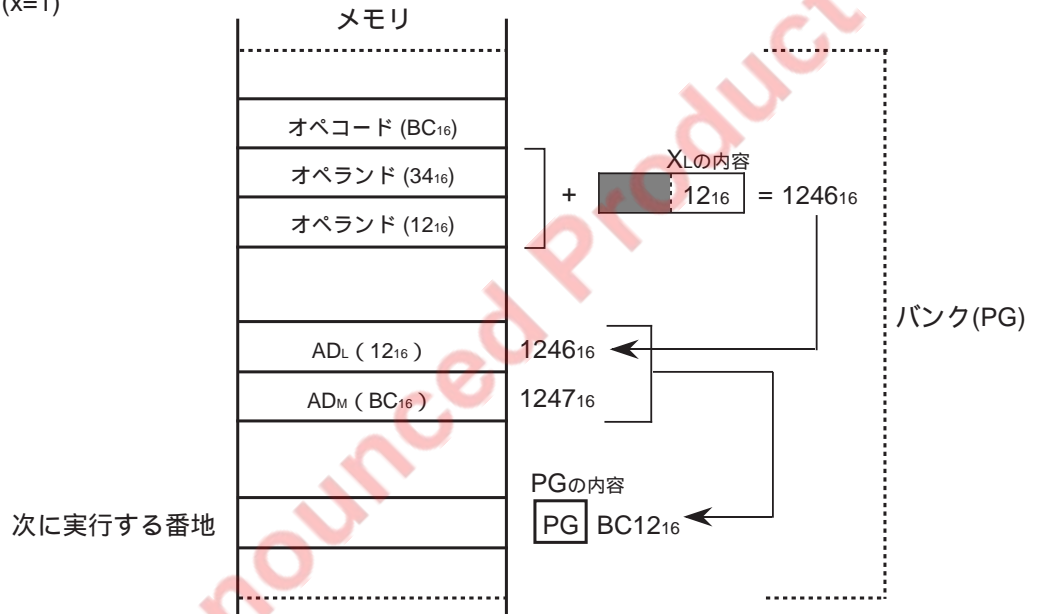
注．命令がバンク境界付近に配置される場合は、参照先のバンクに注意する必要があります。
JMP/JMPL命令の詳細説明P4-106を参照してください。

【名称】 アブソリュート・インデクストX・インダイレクト アドレッシングモード

【機能】 命令の2バイト目と3バイト目で表される数値にインデックスレジスタXの内容を加算した値で連続する2バイトのメモリを指定し(この時のメモリバンクはプログラムバンクレジスタが示す値です) そのメモリの内容が示す番地へジャンプします。
JMP命令、JSR命令で使用します。

例：ニーモニック
JMP (1234H, X)
(x=1)

機械語
BC₁₆ 34₁₆ 12₁₆



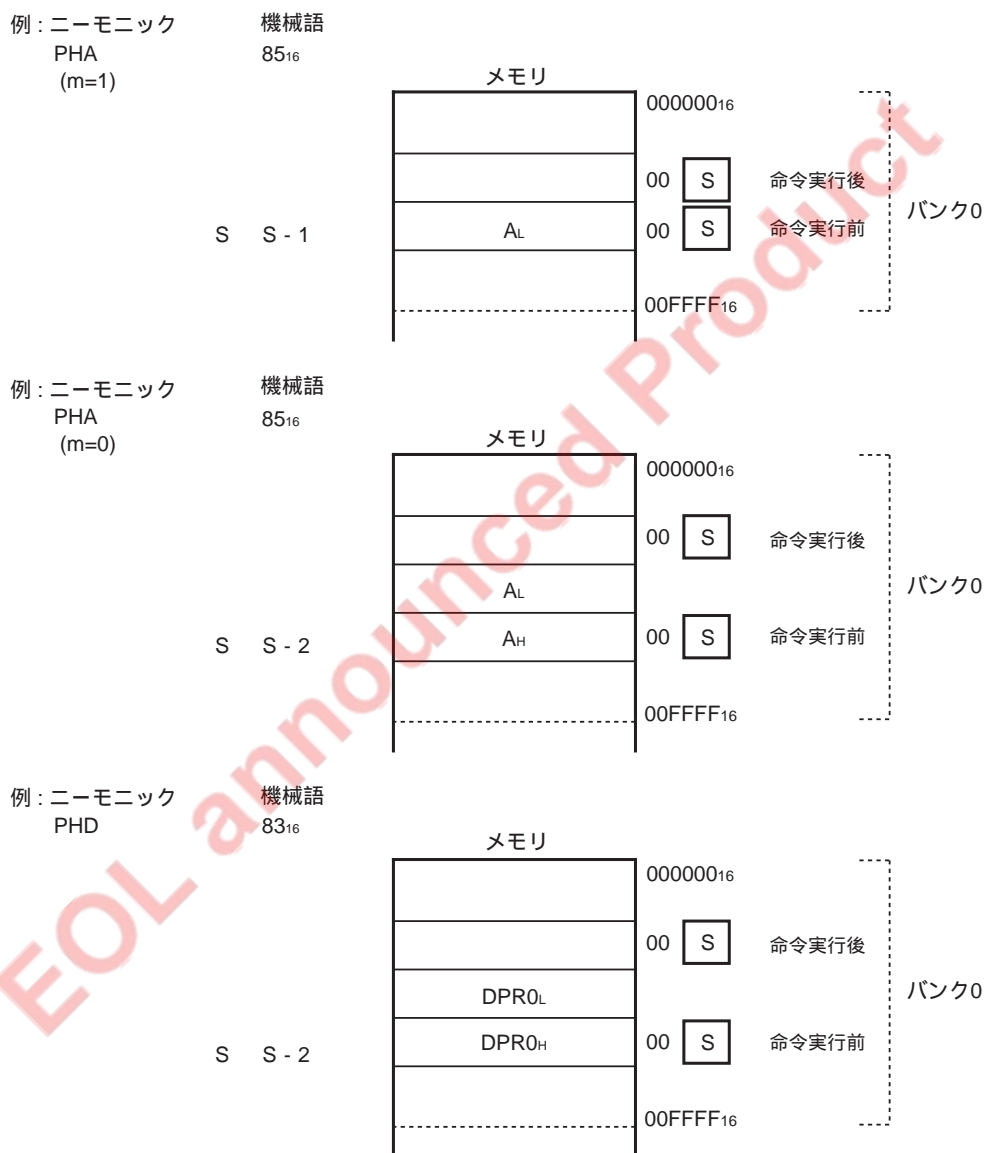
注． JMP命令及びJSR命令では、命令及び参照先の番地がバンク境界付近に配置される場合は、参照先及び飛び先のバンクに注意する必要があります。

JMP/JMPL命令の詳細説明P4-106を参照してください。

JSR/JSRL命令の詳細説明P4-107を参照してください。

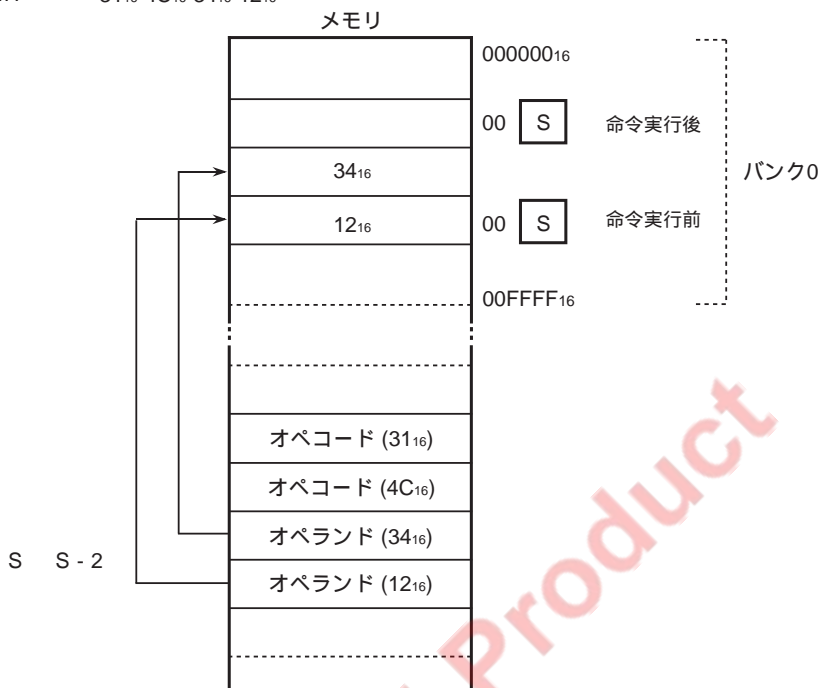
【名称】 スタック アドレッシングモード

【機能】 スタックポインタで指定されたメモリ(スタック領域)へレジスタ等の内容を退避したり、スタックポインタで指定されたメモリからレジスタ等の内容を復帰します。スタック領域はバンク0に設定されます。



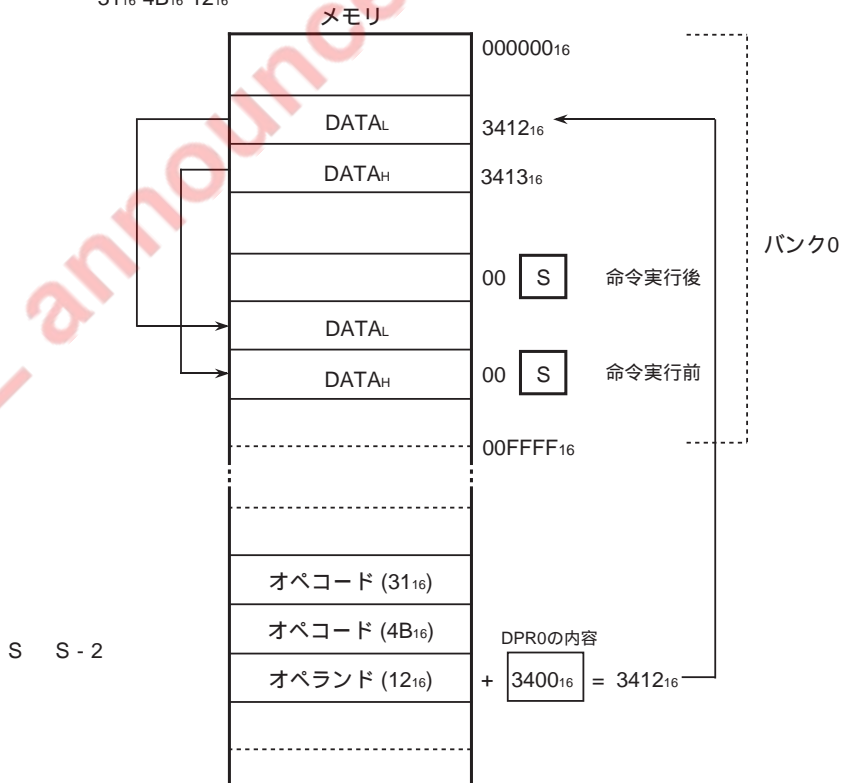
例：ニーモニック
PEA #1234H

機械語
31₁₆ 4C₁₆ 34₁₆ 12₁₆



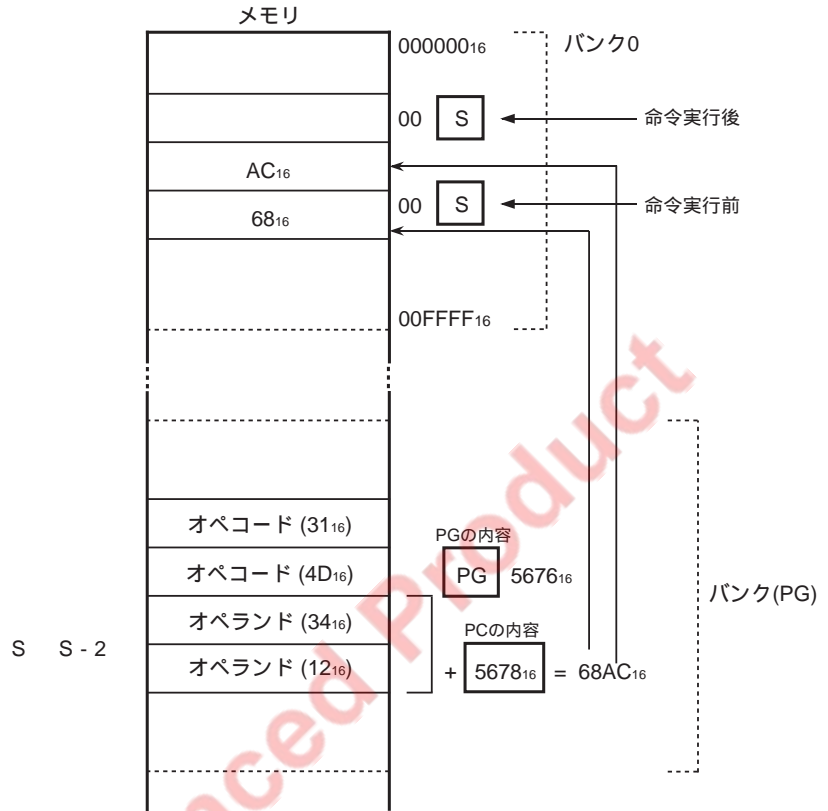
例：ニーモニック
PEI 12H

機械語
31₁₆ 4B₁₆ 12₁₆



例：ニーモニック
PER #1234H

機械語
31₁₆ 4D₁₆ 34₁₆ 12₁₆



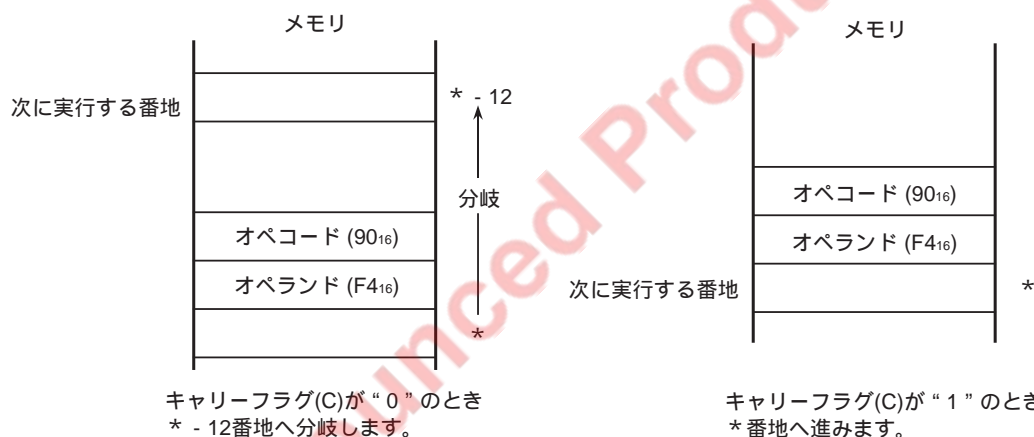
EOL announced Product

【名称】 レラティブ アドレッシングモード

【機能】 プログラムカウンタの内容に、命令の2バイト目を加算した値で示される番地へ分岐します。
 BRA命令でロングジャンプの場合は命令の2バイト目と3バイト目を15ビットの符号付き数値として加算します。
 BSR命令は命令の1バイト目のうちの3ビットと2バイト目を11ビットの符号付き数値として加算します。
 加算した結果、キャリー又はボローが生じたときプログラムバンクレジスタは+1又は-1されます。

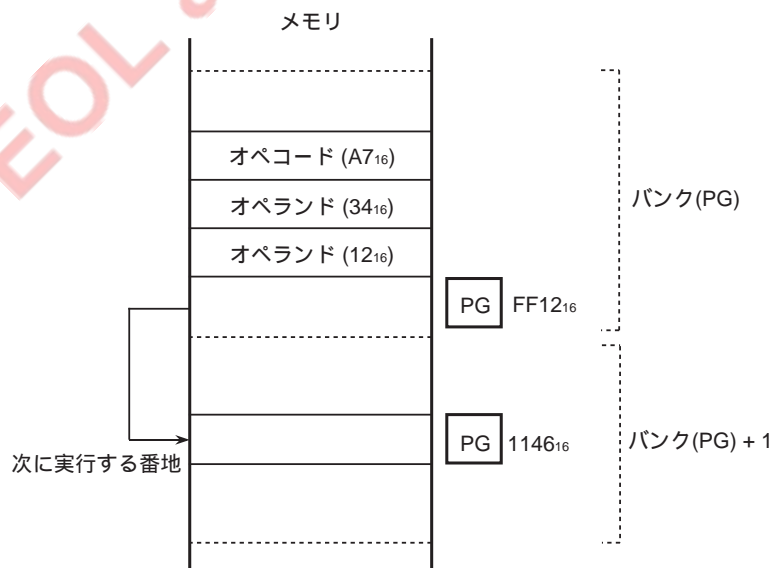
例：ニーモニック
BCC * - 12

機械語
90₁₆ F4₁₆



例：ニーモニック
BRAL 1234H

機械語
A7₁₆ 34₁₆ 12₁₆



【名称】 ダイレクト・ビット・レラティブ アドレッシングモード

【機能】 BBC, BBS命令

命令の3バイト目とダイレクトページレジスタの内容を加算した値でバンク0内のメモリを指定し、そのメモリ内の複数ビット位置を命令の4バイト目と5バイト目($m=1$ のときは4バイト目のみ)のビットパターンで指定します。指定されたビットがすべて条件に合えば命令の6バイト目($m=1$ のときは5バイト目)を符号付きでプログラムカウンタに加算し、加算結果の番地へ分岐します。ただし、命令の2バイト目とダイレクトページレジスタの内容を加算した値がバンク0を越える場合はバンク1内のメモリが指定されます。

BBCB, BBSB命令

命令の2バイト目とダイレクトページレジスタの内容を加算した値でバンク0内のメモリを指定し、そのメモリ内の複数ビット位置を命令の3バイト目のビットパターンで指定します。指定されたビットがすべて条件に合えば命令の4バイト目を符号付きでプログラムカウンタに加算し、加算結果の番地へ分岐します。ただし、命令の2バイト目とダイレクトページレジスタの内容を加算した値がバンク0を越える場合はバンク1内のメモリが指定されます。

<ダイレクトページレジスタ切り替えビットが“0”の場合>

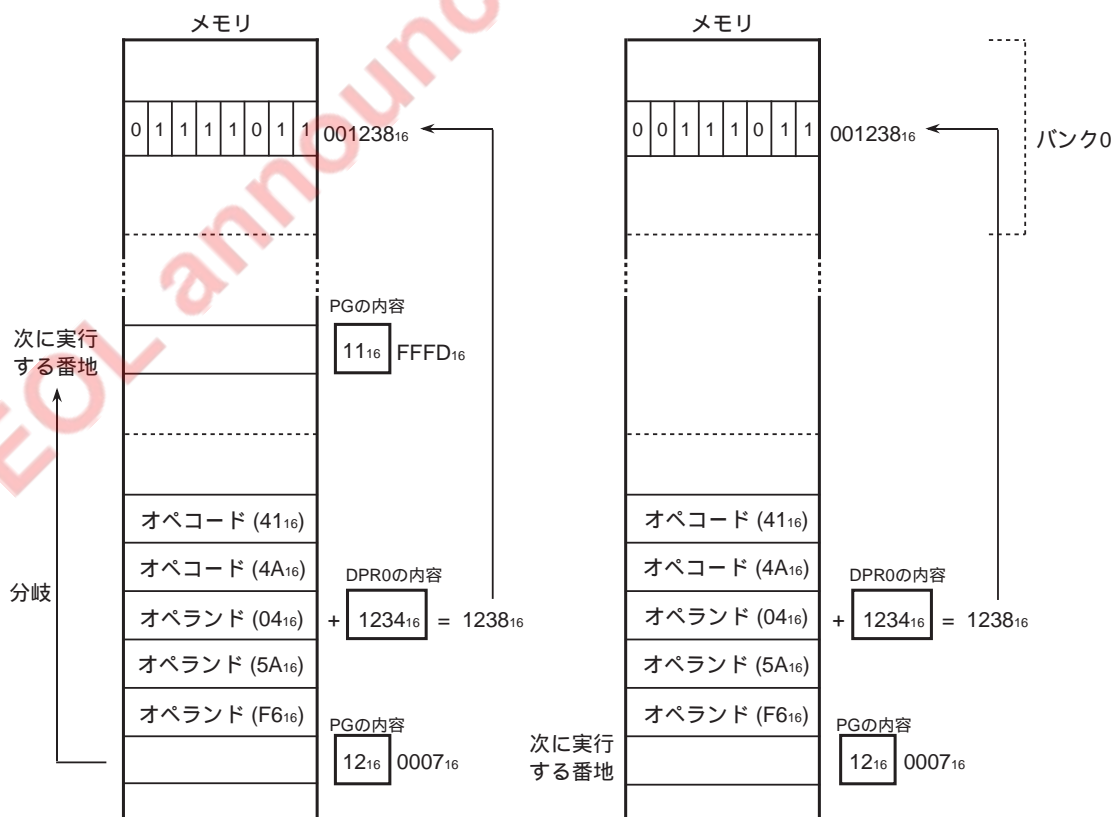
例：ニーモニック

機械語

BBS #5AH, 04H, 0F6H (m=1) 41₁₆ 4A₁₆ 04₁₆ 5A₁₆ F6₁₆

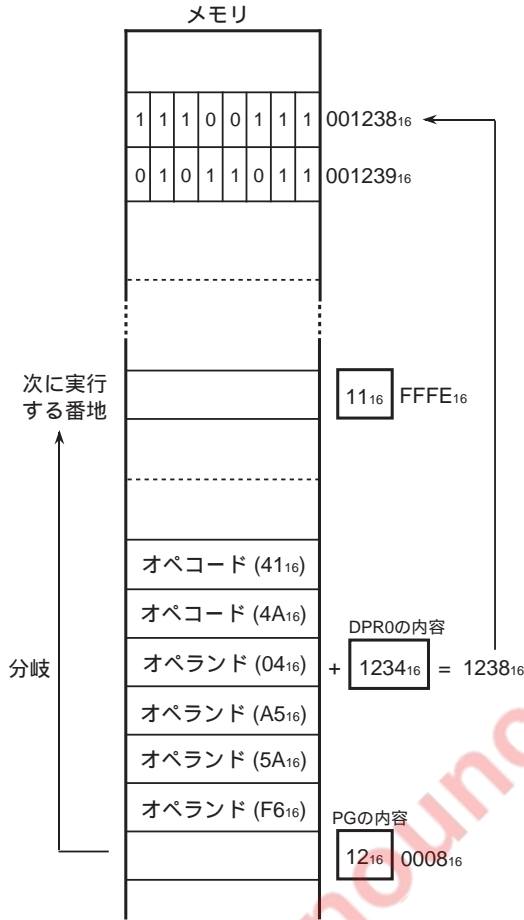
条件が合い分岐する場合

条件が合わないため分岐しない場合

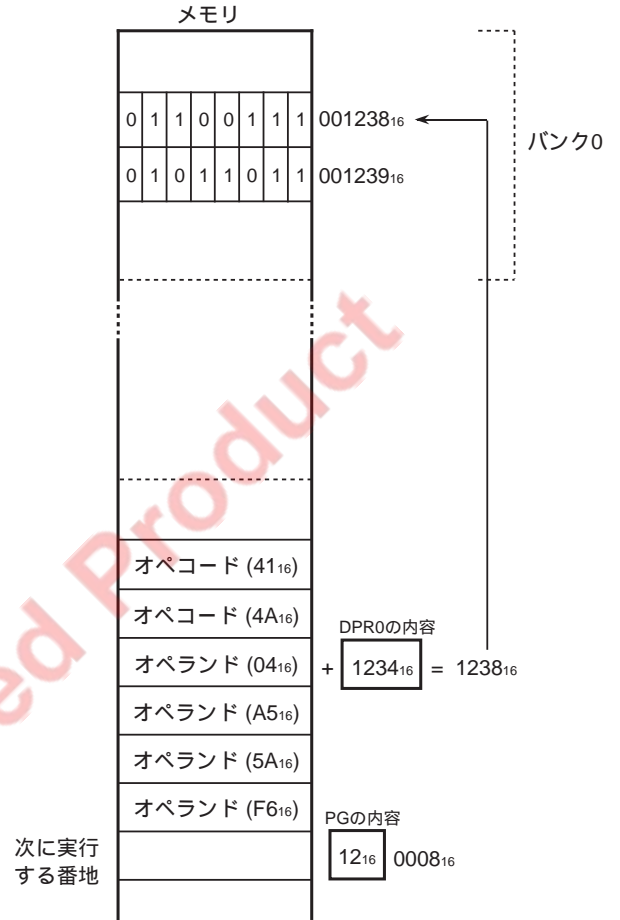


例：ニーモニック 機械語
 BBS #5AA5H, 04H, 0F6H 41₁₆ 4A₁₆ 04₁₆ A5₁₆ 5A₁₆ F6₁₆
 (m=0)

条件が合い分岐する場合



条件が合わないため分岐しない場合



EOL announced Product

【名称】 アブソリュート・ビット・レラティブ アドレッシングモード

【機能】 BBC, BBS命令

命令の3バイト目と4バイト目とデータバンクレジスタの内容でメモリ番地を指定し、そのメモリ内の複数ビット位置を命令の5バイト目と6バイト目($m=1$ のときは5バイト目のみ)のビットパターンで指定します。指定されたビットがすべて条件に合えば命令の7バイト目($m=1$ のときは6バイト目)を符号付きでプログラムカウンタに加算し、加算結果の番地へ分岐します。

BBCB, BBSB命令

命令の2バイト目と3バイト目とデータバンクレジスタの内容でメモリ番地を指定し、そのメモリ内の複数ビット位置を命令の4バイト目のビットパターンで指定します。指定されたビットがすべて条件に合えば命令の5バイト目を符号付きでプログラムカウンタに加算し、加算結果の番地へ分岐します。

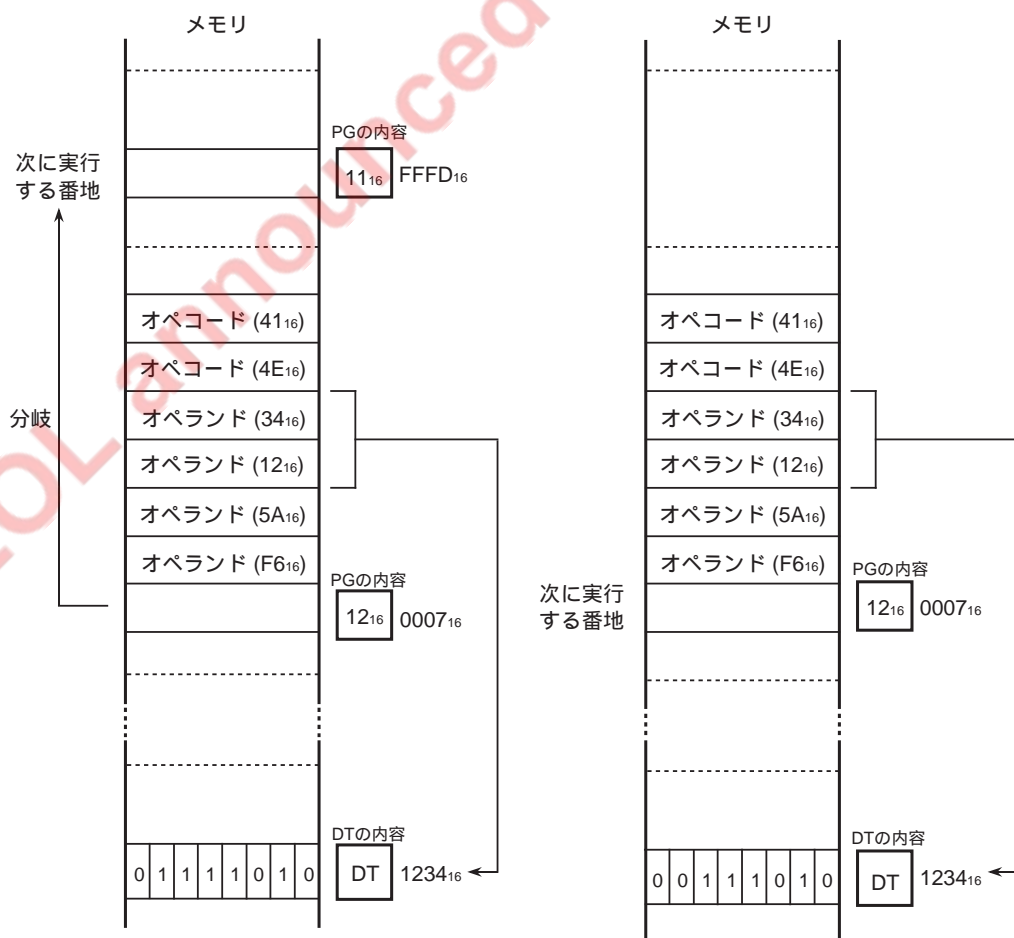
例：ニーモニック

機械語

BBS #5AH, 1234H, 0F6H 41₁₆ 4E₁₆ 34₁₆ 12₁₆ 5A₁₆ F6₁₆
($m=1$)

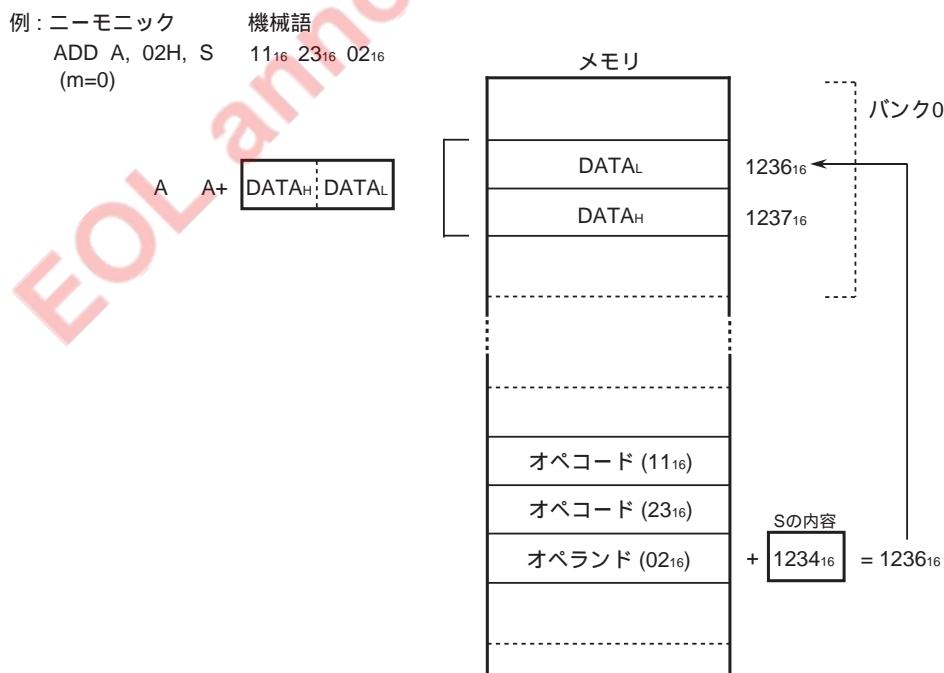
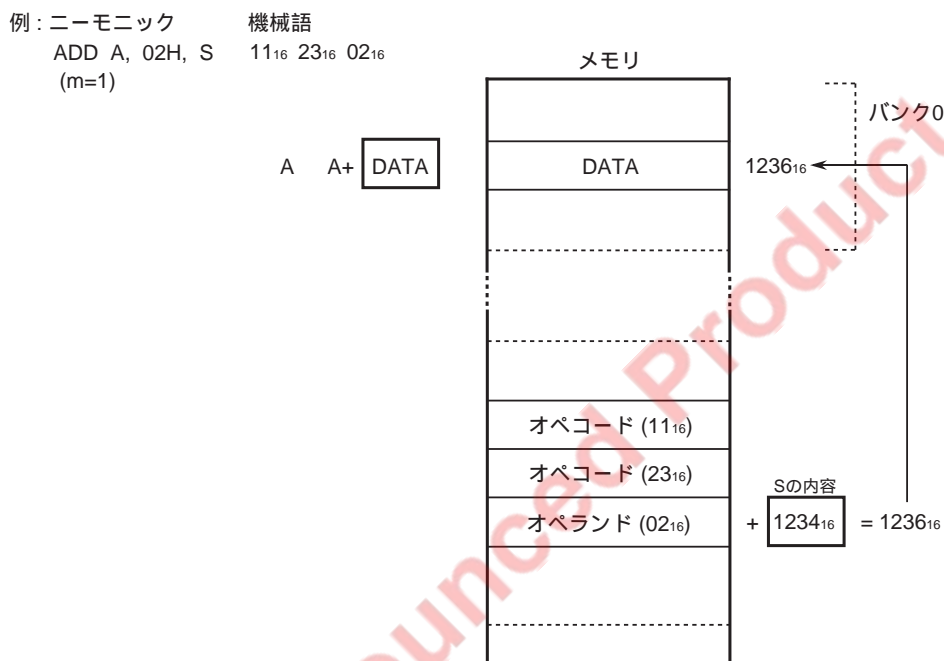
条件が合い分岐する場合

条件が合わないため分岐しない場合

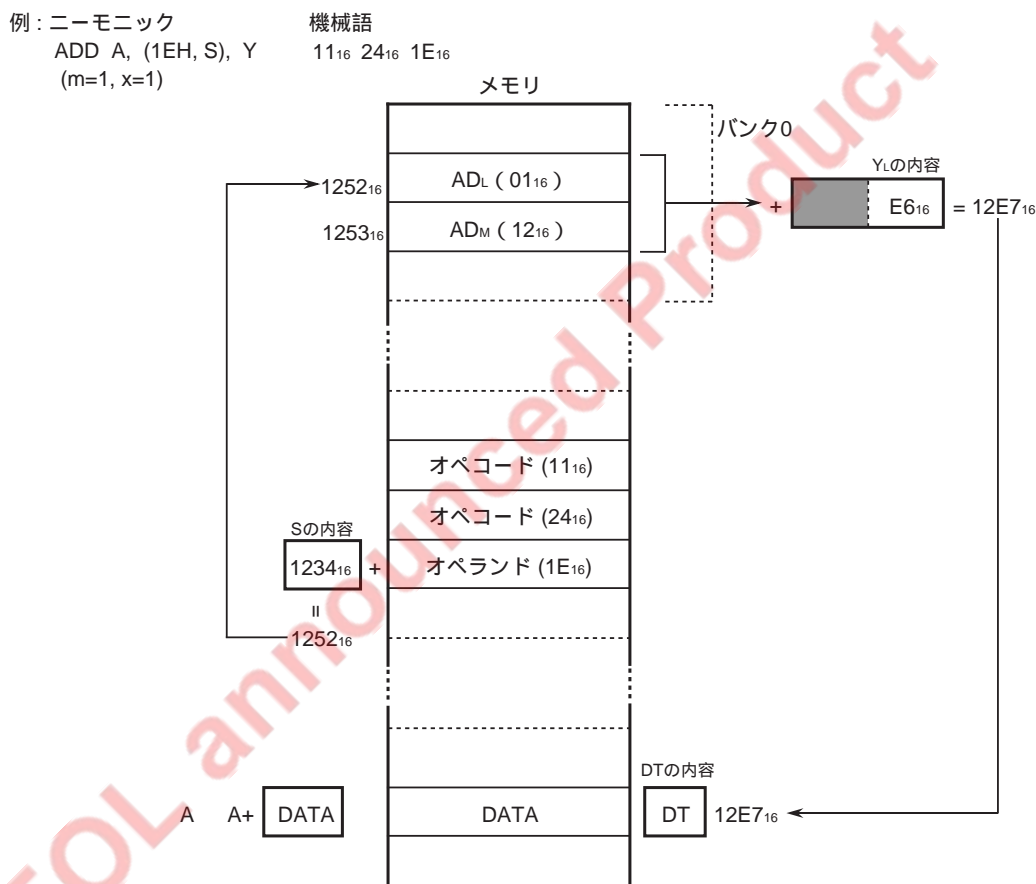


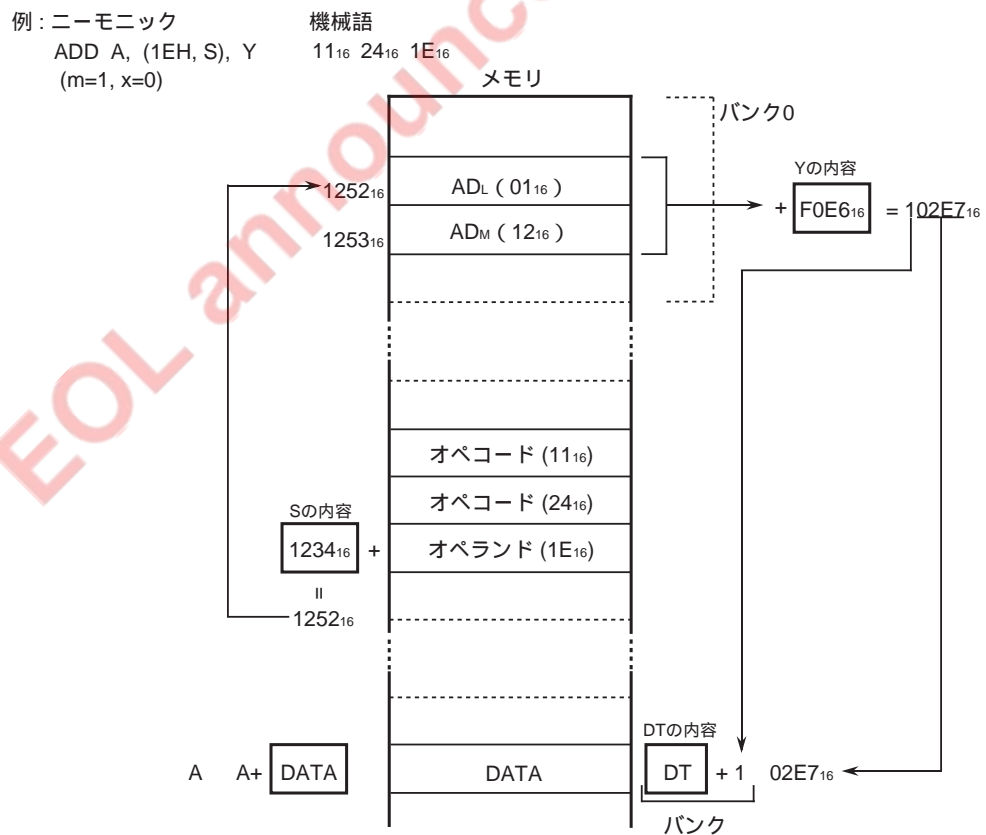
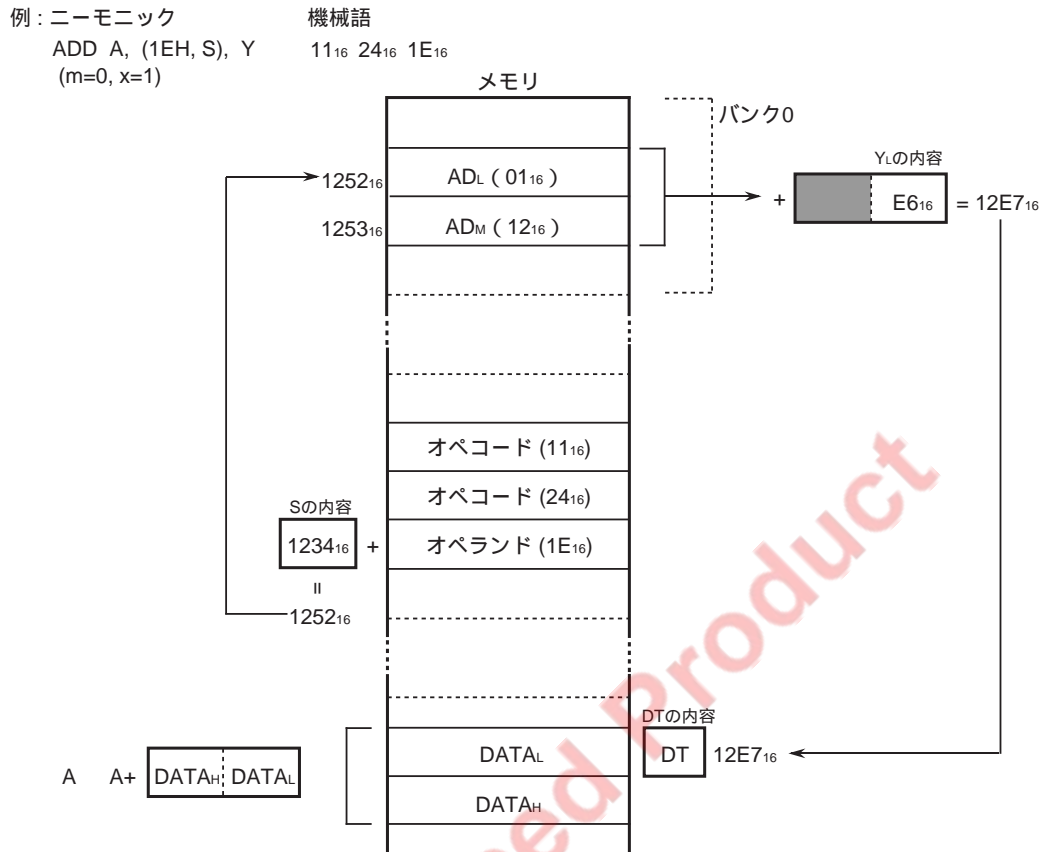
【名称】 スタックポインタ・レラティブ アドレッシングモード

【機能】 命令のオペランドとスタックポインタの内容を加算した値で指定されるバンク0内のメモリの内容が演算データとなります。ただし、命令のオペランドとスタックポインタの内容を加算した値がバンク0を越える場合はバンク1内のメモリが指定されます。

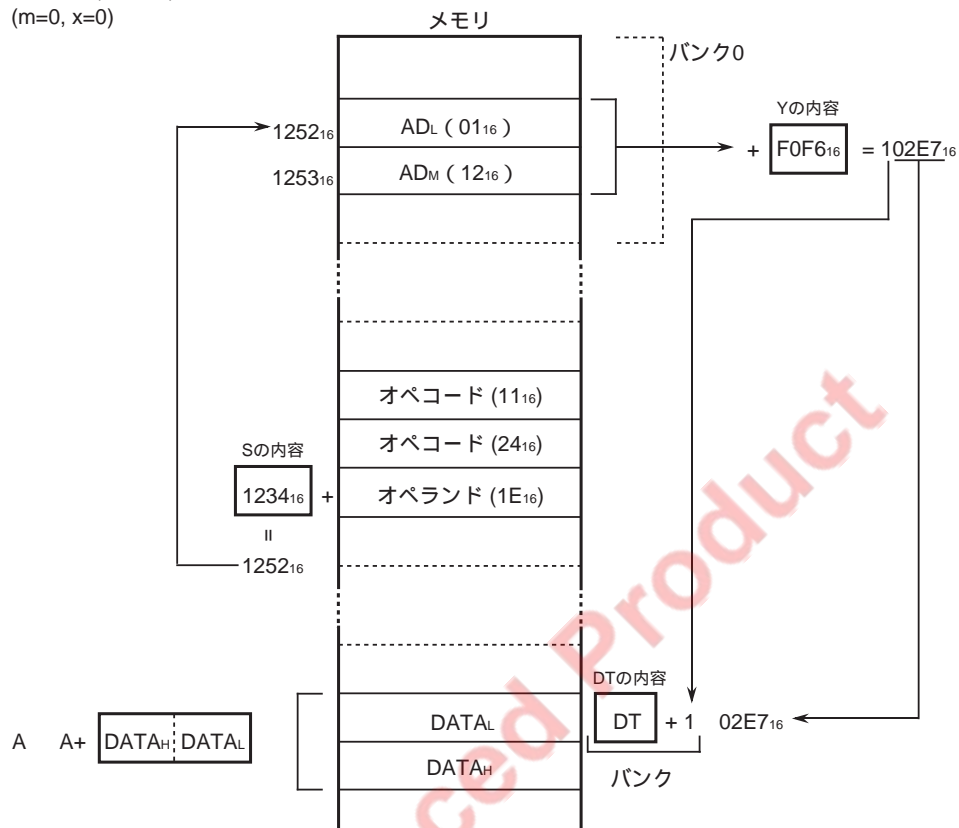


- 【名称】 スタックポインタ・レラティブ・インダイレクト・インデクストY アドレッシングモード
- 【機能】 命令のオペランドとスタックポインタの内容を加算した値で、連続する2バイトのメモリを指定します。指定されたメモリの内容にインデックスレジスタYの内容を加算した値とデータバンクレジスタの内容で指定される番地のメモリの内容が演算データとなります。ただし、連続する2バイトのメモリの内容とインデックスレジスタYの内容を加算したときキャリーが生じると、バンクはデータバンクレジスタの内容に1加算したものが使用されます。





例: ニーモニック 機械語
 ADD A, (1EH, S), Y 11₁₆ 24₁₆ 1E₁₆
 (m=0, x=0)



EOL announced Product

【名称】 ブロック転送 アドレッシングモード

【機能】 命令の3バイト目で転送先のデータバンクを示し、インデックスレジスタYの内容でそのデータバンク内の転送先番地を示します。命令の4バイト目で転送元のデータバンクを示し、インデックスレジスタXの内容でそのデータバンク内の転送データの番地を示します。アキュムレータAの内容が転送バイト数となります。転送終了時、データバンクレジスタの内容は転送先のデータバンクになっています。

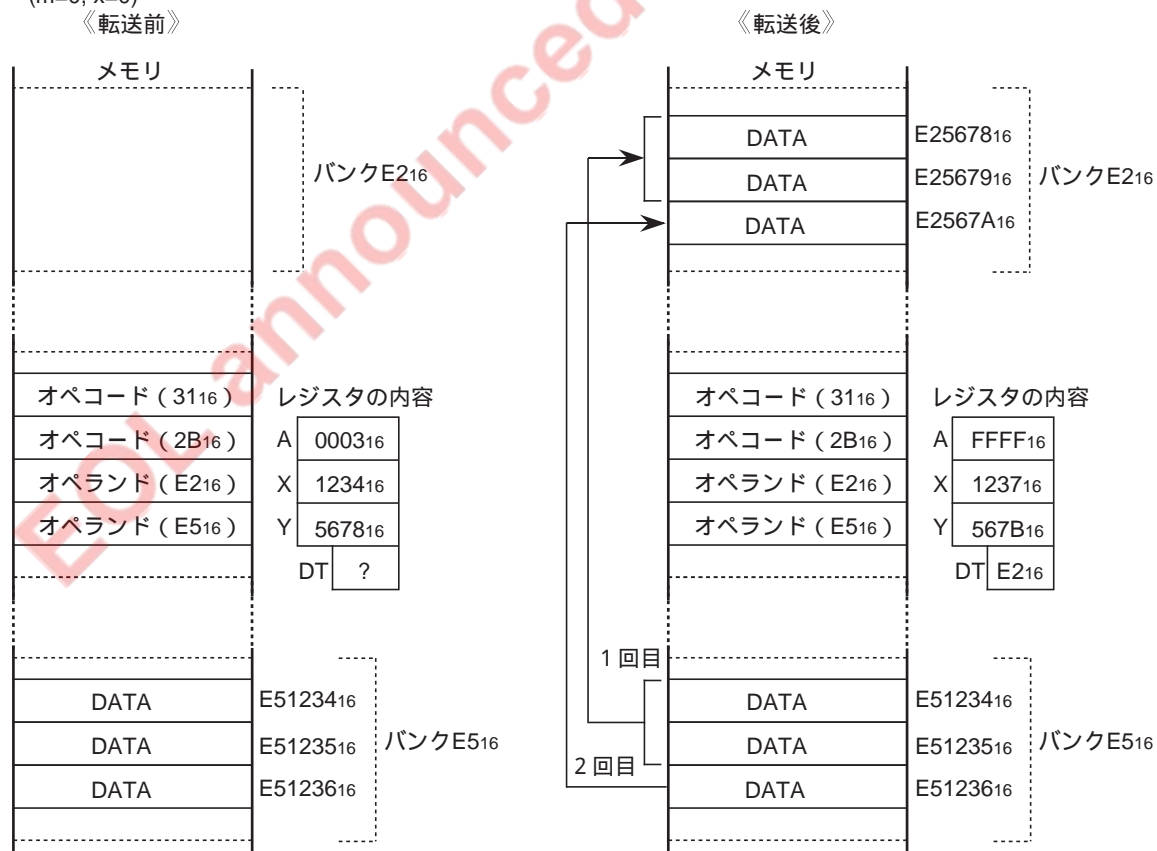
MVN命令

下位番地方向への転送ではMVN命令を使用します。インデックスレジスタX及びインデックスレジスタYの内容は転送の度にインクリメントされます。

MVP命令

上位番地方向への転送ではMVP命令を使用します。インデックスレジスタX及びインデックスレジスタYの内容は転送の度にデクリメントされます。転送データはバンク間にまたがって存在してもかまいません。

例：ニーモニック 機械語
 MVN 0E2H, 0E5H 31₁₆ 2B₁₆ E2₁₆ E5₁₆
 (m=0, x=0)
 《転送前》



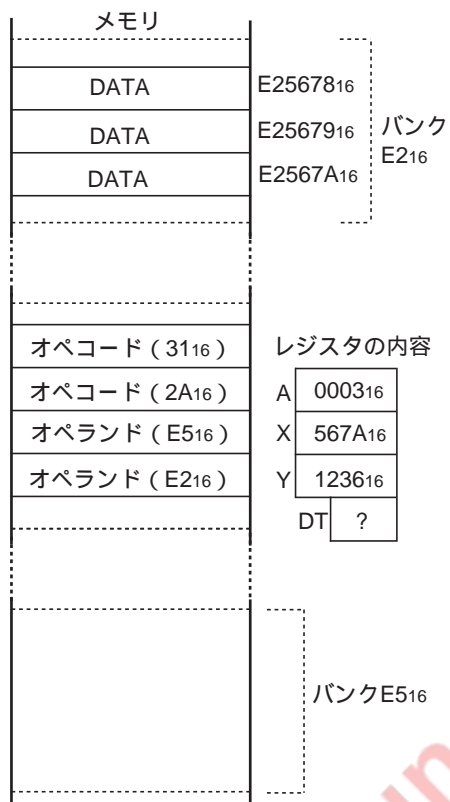
例：ニーモニック

MVP 0E5H, 0E2H
(m=0, x=0)

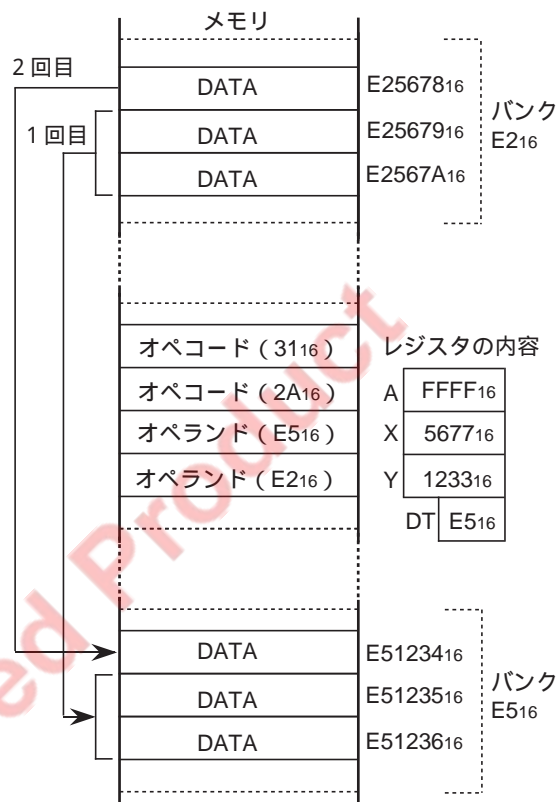
機械語

31₁₆ 2A₁₆ E5₁₆ E2₁₆

《転送前》



《転送後》



注．ブロック転送命令はm, xフラグの状態転送バイト数、転送元 / 先番地として設定可能な値の範囲が変化しますが、転送単位はこれらの影響を受けません。

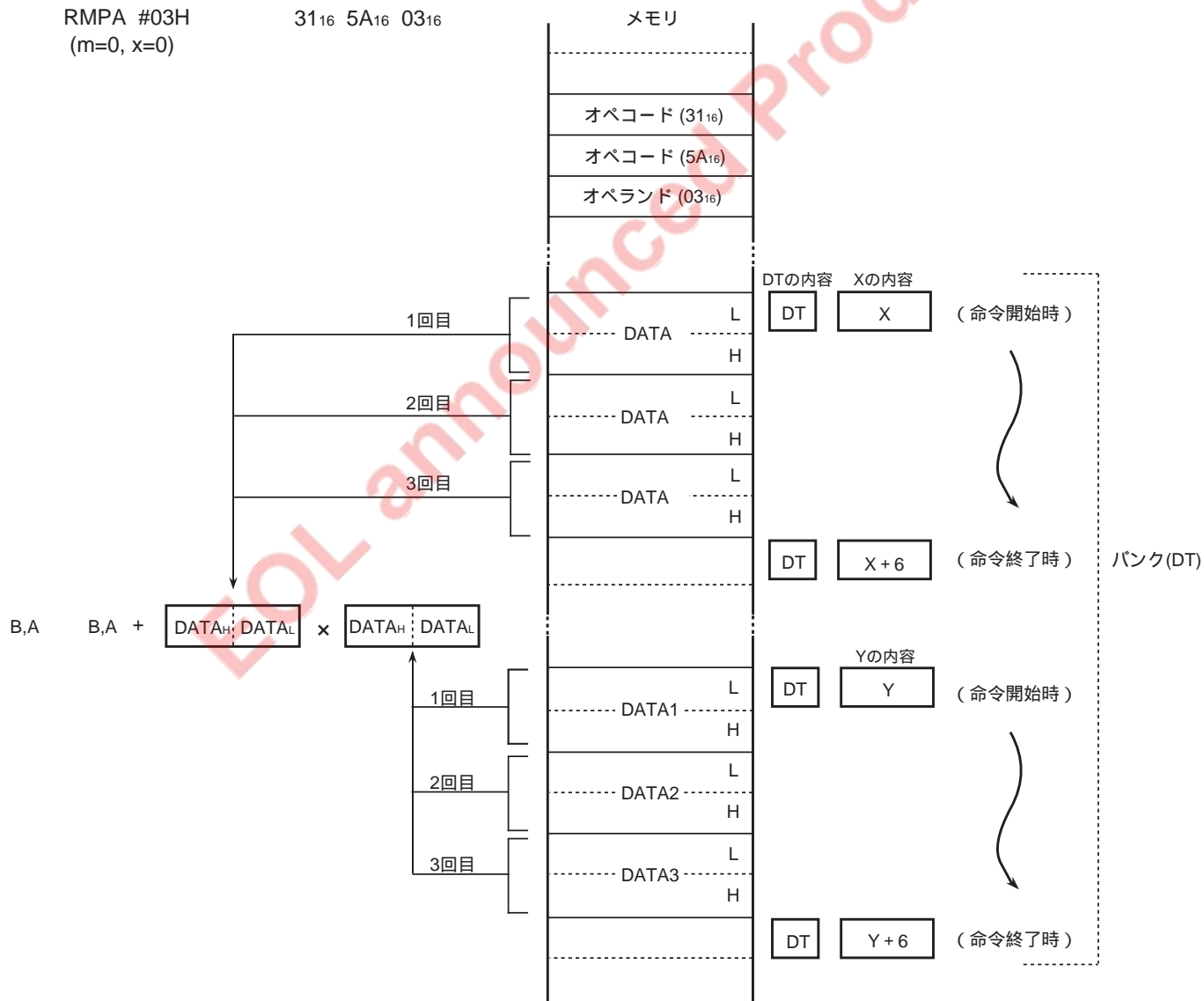
転送単位はワード(16ビット)幅ですが、奇数バイト転送時の最終バイト転送のみ1バイトのみの転送を行います。

【名称】 積和 アドレッシングモード

【機能】 インデックスレジスタX及びYの内容とデータバンクレジスタの内容で指定される番地のメモリの内容が、被乗数及び乗数になります。命令の3バイト目は繰り返し演算回数となります。インデックスレジスタX及びYの内容は、アキュムレータB、Aの内容と乗算結果の加算が終了するごとにインクリメントされます。したがって命令終了時のインデックスレジスタX及びYの内容は、最後に読み出した被乗数及び乗数の次の番地を示します。被乗数及び乗数は、同一バンク内に配置してください。また、次のバンクにまたがらないようにしてください。インデックスレジスタ長選択フラグxは“0”にして実行してください。RMPA命令で使用します。

例：ニーモニック
RMPA #03H
(m=0, x=0)

機械語
31₁₆ 5A₁₆ 03₁₆



第 3 章

7900命令の使用法

- 3.1 メモリアクセス
- 3.2 DPR0 ~ DPR3の使用
- 3.3 8/16ビット長データ操作
- 3.4 インデックスレジスタX, Y
- 3.5 分岐命令の使用法

3.1 メモリアクセス

メモリアクセスは基本的なものとして3種類に分類されます。

- ・ダイレクトアドレッシング
- ・アブソリュート、アブソリュートロングアドレッシング
- ・インダイレクト、インダイレクトロングアドレッシング

それぞれ以下の特長を持ちます。

3.1.1 ダイレクトアドレッシング

命令バイト長は2～3バイト。

命令実行サイクルが短い。

アドレッシング可能な範囲は、DPRnで指示する番地を基底とするバンク(000000₁₆～00FFFF₁₆)内のブロック。

- (i)ダイレクトページレジスタ切り替えビットが“0”の場合
ブロックサイズ：256バイト
- (ii)ダイレクトページレジスタ切り替えビットが“1”の場合
ブロックサイズ：64バイト

ただし、DPRnの内容とオフセット値の加算結果がバンク境界を越える場合は、境界を越えてバンク1をアクセス可能。

3.1.2 アブソリュート、アブソリュートロングアドレッシング

(1)アブソリュートアドレッシング

命令バイト長は3～4バイト。

アドレッシング可能な範囲は、24ビットのアドレスのうちの上位8ビットの値をDTで指示する64Kバイト空間(000000₁₆～FFFFFF₁₆の任意のバンク)。

ただし、JMP、JSR命令ではPGが上位8ビットの値を指示。

(2)アブソリュートロングアドレッシング

命令バイト長は4～5バイト。

アドレッシング可能な範囲は000000₁₆～FFFFFF₁₆。

24ビットのアドレスの全ビットを直接指定。

3.1.3 インダイレクト、インダイレクトロングアドレッシング

(1)ダイレクト・インダイレクトアドレッシング

命令バイト長は2～3バイト。

16ビット長のポインタデータをDPRnで指示する空間に配置し、メモリアクセス。

アドレッシング可能な範囲は、24ビットアドレスのうちの上位8ビットの値をDTで指示する64Kバイト空間(000000₁₆～FFFFFF₁₆の任意のバンク)。

(2)ダイレクト・インダイレクトロングアドレッシング

命令バイト長は2～3バイト。

24ビット長のポインタデータをDPRnで指示する空間に配置し、メモリアクセス。

アドレッシング可能な範囲は、16Mバイト空間の任意の番地(000000₁₆～FFFFFF₁₆)。

(3) アブソリュート・インダイレクトアドレッシング

このアドレッシングモードは間接分岐・間接サブルーチン呼び出し命令でのみ使用可能。

命令バイト長は3～4バイト。

16ビット長のポインタデータをPGで指示する空間に配置し、メモリアクセス。

アドレッシング可能な範囲は、24ビットアドレスのうちの上位8ビットの値をPGで指示する64Kバイト空間(000000₁₆～FFFFFF₁₆の任意のバンク)。

(4) アブソリュート・インダイレクトロングアドレッシング

このアドレッシングモードは間接分岐命令でのみ使用可能。

命令バイト長は3～4バイト。

24ビット長のポインタデータをPGで指示する空間に配置し、メモリアクセス。

アドレッシング可能な範囲は、16Mバイト空間の任意の番地(000000₁₆～FFFFFF₁₆)。

例として、データのブロック転送を図3.1.1に示します。

あるメモリ領域を指示するポインタをプログラム中で演算し、その値を実効アドレスとして参照しています。

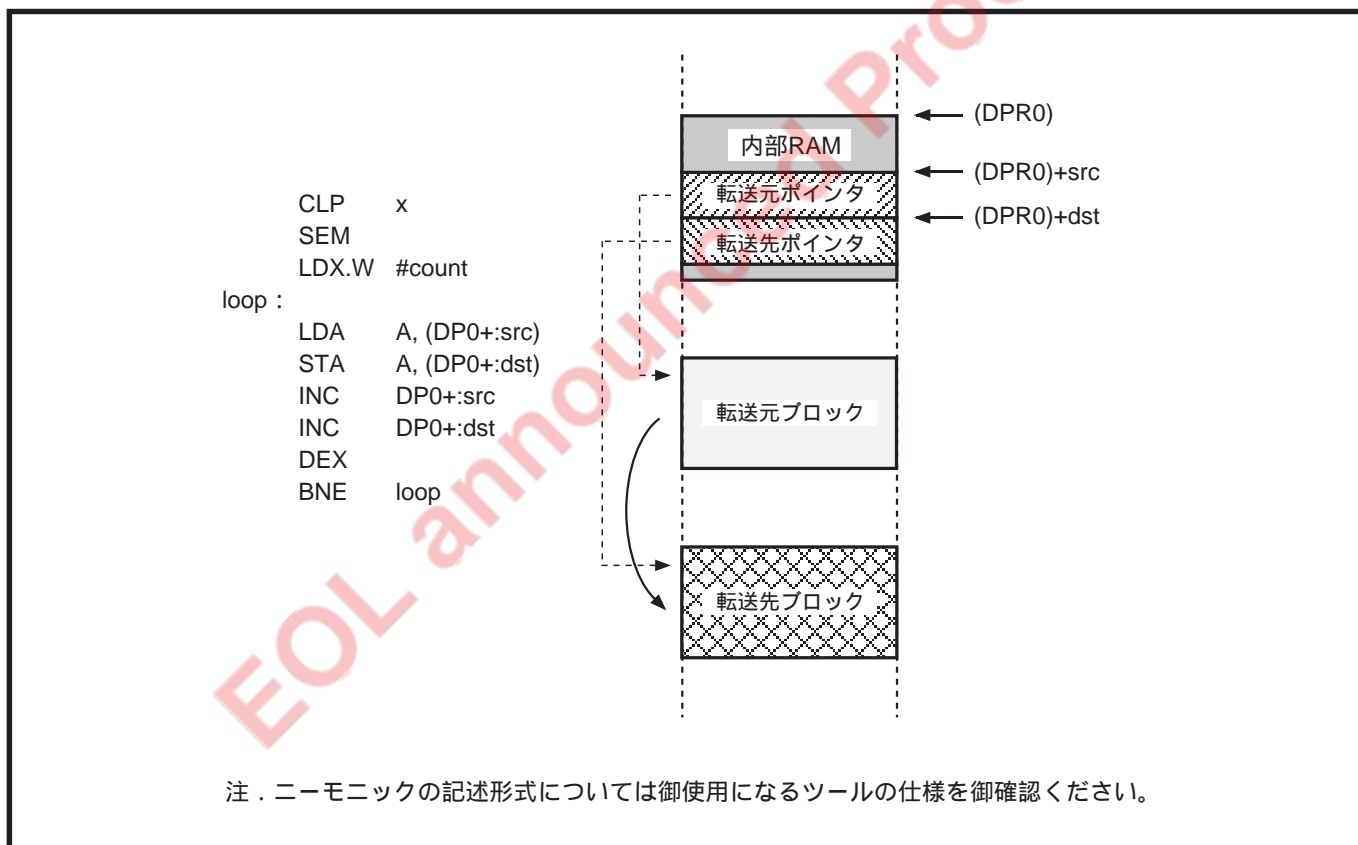


図3.1.1 間接アドレッシングモードの使用例

7900シリーズはこの他にも多くのアドレッシングモードを持ちます。詳しくは「2.3 アドレッシングモード」を御参照ください。

3.2 DPR0 ~ DPR3の使用

7900シリーズでは、従来の7700ファミリに対してダイレクトアドレッシングの機能が強化されています。これは000000₁₆ ~ 00FFFF₁₆の範囲でソフトウェアの効率をより向上させることに寄与します。

リセット直後、7900シリーズはDPR0のみが使用可能となっています。プロセッサモードレジスタ1のダイレクトページレジスタ切り替えビットを“1”にセットすると、4本のダイレクトページレジスタ(DPR0 ~ DPR3)が使用可能となります。

DPR0 ~ DPR3の使用例を図3.2.1に示します。

従来の7700ファミリはダイレクトページレジスタを1本しか持たないため、ダイレクトページアドレッシングを利用して効率的メモリアクセスを行うためにはダイレクトページレジスタの内容をその都度変更する処理が必要でした。7900シリーズでは異なるブロックの基点毎にダイレクトページレジスタを割り付けることができますので、このような処理が不要です。

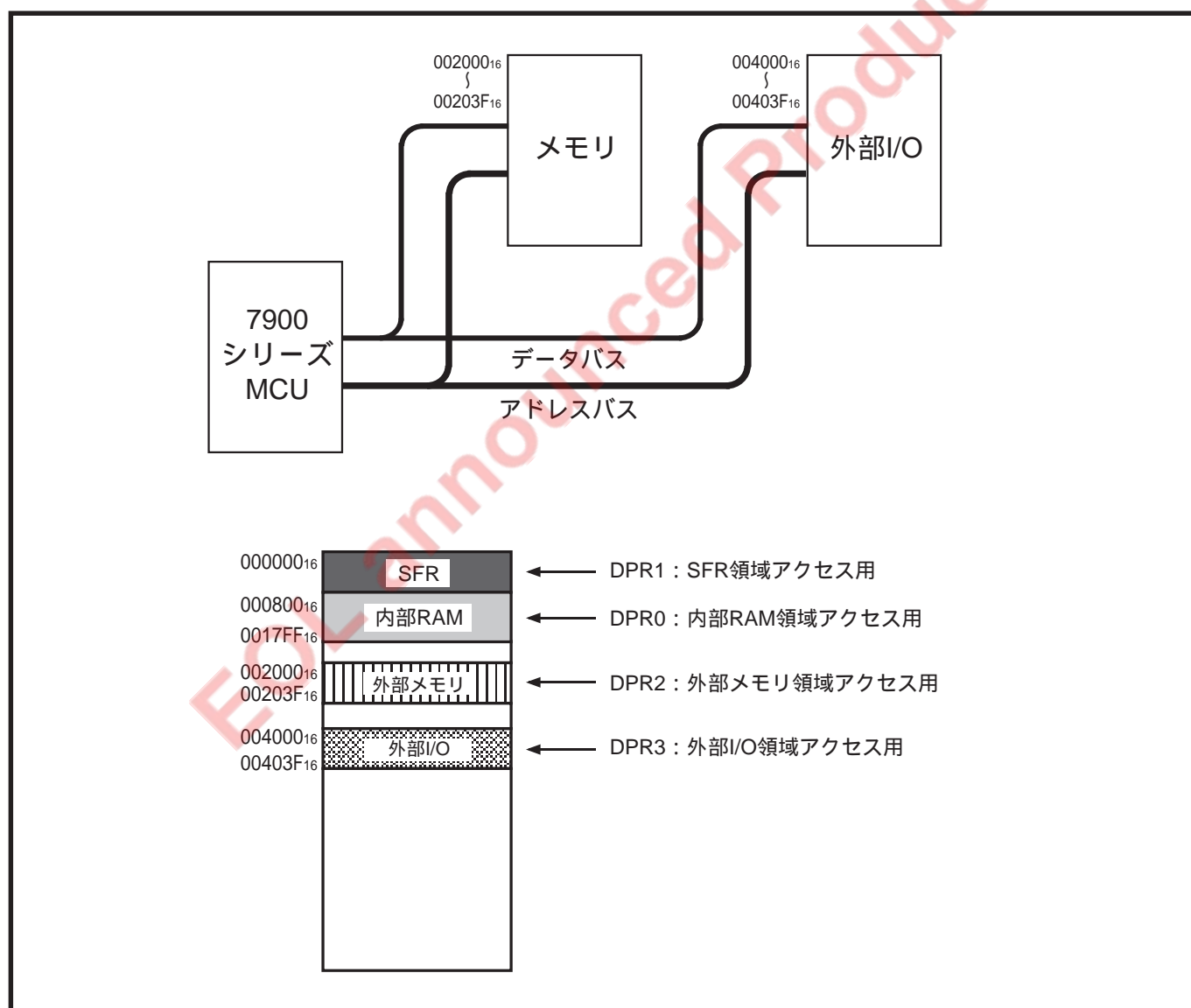


図3.2.1 DPR0 ~ DPR3の使用例

3.3 8/16ビット長データ操作

7700ファミリはプログラムサイズを小さくするための工夫として8/16ビット長命令に同一の機械語を割り付けています。このため、8ビット又は16ビットいずれのデータを扱うかをmフラグ又はxフラグで設する必要があります。

7900シリーズはこの方式に加え、データ長を設定するフラグの状態にかかわらず8ビット長演算を行う8ビット長データ専用命令を追加しました。

この命令を使用すれば、フラグの設定を16ビット長にしたまま8ビットデータを扱うことができます。これにより、フラグ切り替え操作のオーバーヘッドの発生を防ぐことができます。

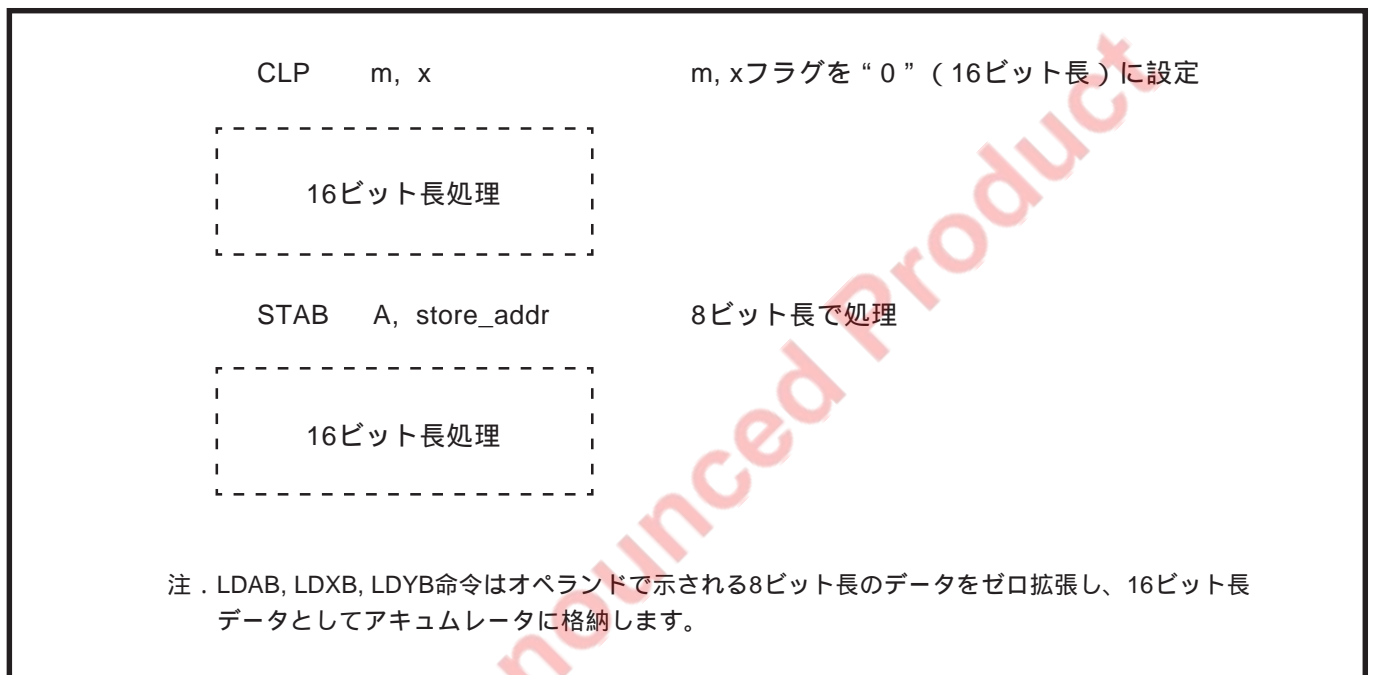


図3.3.1 8ビット演算の例

mフラグ又はxフラグで演算長を設定する命令は、バイト数及び実行サイクル数がその設定内容の影響を受けます。

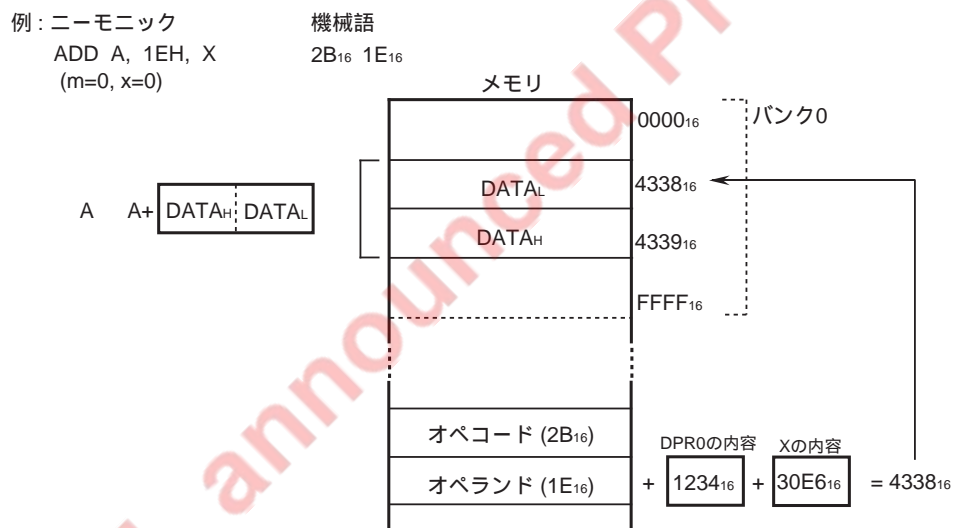
詳しくは「4.2 命令の説明」、又は「付録1 . 機械語命令一覧表」を参照してください。

3.4 インデックスレジスタ X, Y

実効アドレスを生成する際にインデックスレジスタ X 又は Y の内容を使用することができます。
例としてダイレクト・インデックス X アドレッシングモードを下記に示します。
詳しくは「2.3 アドレッシングモード」を参照してください。

< 例 > ダイレクト・インデックス X アドレッシングモード

命令のオペランドとダイレクトページレジスタの内容とインデックスレジスタ X の内容を加算した値で指定されるバンク 0 内のメモリの内容が演算データとなります。ただし、命令のオペランドとダイレクトページレジスタの内容とインデックスレジスタ X の内容を加算した値がバンク 0 あるいはバンク 1 を越える場合は、バンク 1 あるいはバンク 2 内のメモリが指定されます。



3.5 分岐命令の使用方式

分岐命令は以下の6種類に分類されます。

- (1) 相対(レラティブ)分岐
- (2) 絶対(アブソリュート、アブソリュートロング)分岐
- (3) 間接(アブソリュート・インダイレクト、アブソリュート・インダイレクトロング)分岐
- (4) 相対(レラティブ)サブルーチン呼び出し
- (5) 絶対(アブソリュート、アブソリュートロング)サブルーチン呼び出し
- (6) 間接(アブソリュート・インデクストX・インダイレクト)サブルーチン呼び出し

相対分岐及び相対サブルーチン呼び出し命令は、以下のような特長を持っています。

命令バイト長は2~3バイト。

プログラム動作中に動的にプログラム領域の配置を変更することが可能。

分岐可能な範囲は一定の範囲に限定。詳しくは「4.2 命令の説明」を参照。

- 例) (i) BRA命令 ... 命令実行後のPCから-128~+127の範囲。
 (ii) BRAL命令 ... 命令実行後のPCから-32768~+32767の範囲。
 (iii) BSR命令 ... 命令実行後のPCから-1024~+1023の範囲。

一方、絶対分岐、絶対サブルーチン呼び出し、間接分岐及び間接サブルーチン呼び出し命令は、以下のような特長を持っています。

16Mバイト空間の任意の番地を直接分岐先に指定し、その番地に分岐することが可能(アブソリュートロング)。

現在実行中のPCを含む64Kバイト空間(バンク)内に限定して分岐先番地を指定することも可能。

この場合、命令バイト長及び命令実行サイクル数を削減することが可能。

詳しくは「4.2 命令の説明」を参照。

- 例) (i) JMP命令 ... 命令語の最後のバイトが存在するPGで指示される64Kバイト空間に分岐します。
 (ii) JMPL命令 ... 16Mバイト空間の任意の番地に分岐します。
 (iii) JSR命令 ... 命令語の最後のバイトが存在するPGで指示される64Kバイト空間に分岐します。
 分岐先からのリターンにはRTS命令を使用します。
 (iv) JSRL命令 ... 16Mバイト空間の任意の番地に分岐します。
 分岐先からのリターンにはRTL命令を使用します。

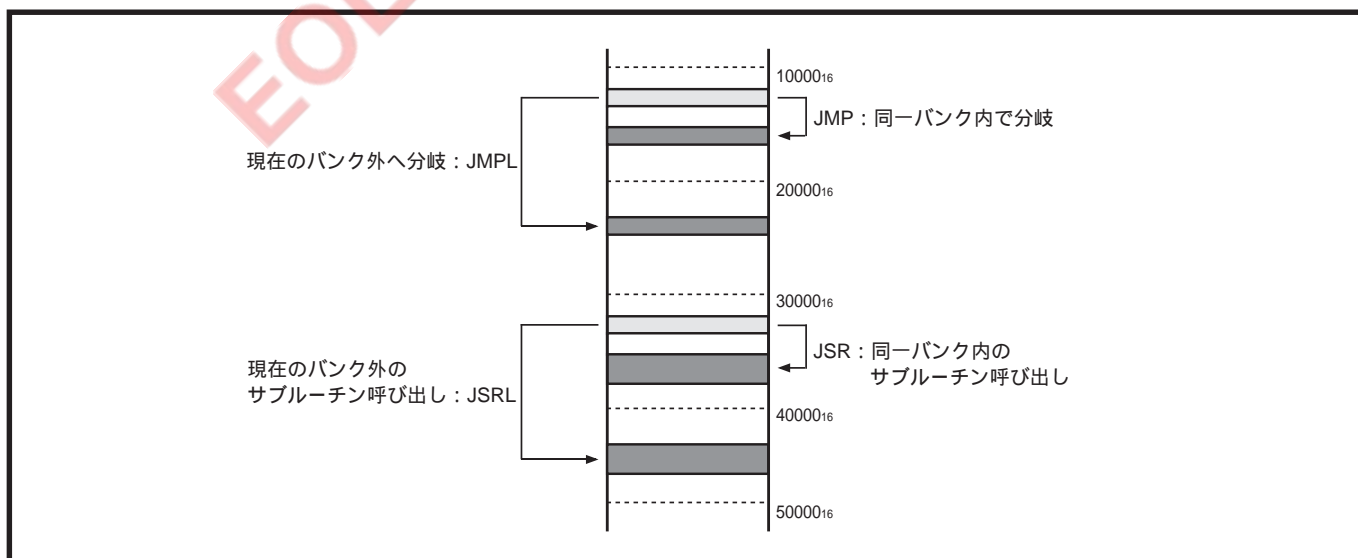


図3.5.1 JMP/JMPL命令、JSR/JSRL命令による分岐

Memo

EOL announced Product

第 4 章 命 令

- 4.1 命令の体系
- 4.2 命令の説明
- 4.3 ソフトウェア開発上の注意
- 4.4 32ビット演算命令実行時の注意

4.1 命令の体系

7900シリーズのCPUは、203個の命令セットをもちます。以下にその一覧を示します。

: 7751シリーズから追加された命令

備考欄には従来の7700ファミリの命令が当該命令に包含されることを示しています。

機能	命令	動作		備考
ロード	LDA	Acc	M	
	LDAB	Acc	M8 (ゼロ拡張)	
	LDAD	E	M32	
	LDD n	DPRn	IMM16 (n=0~3。複数指定可)	
	LDT	DT	IMM8	
	LDX	X	M	
	LDXB	X	IMM8 (ゼロ拡張)	
	LDY	Y	M	
	LDYB	Y	IMM8 (ゼロ拡張)	
ストア	STA	M	Acc	
	STAB	M8	AccL	
	STAD	M32	E	
	STX	M	X	
	STY	M	Y	
レジスタ間転送	TAD n	DPRn	A (n=0~3)	TAD命令包含
	TAS	S	A	
	TAX	X	A	
	TAY	Y	A	
	TBD n	DPRn	B (n=0~3)	TBD命令包含
	TBS	S	B	
	TBX	X	B	
	TBY	Y	B	
	TDA n	A	DPRn (n=0~3)	TDA命令包含
	TDB n	B	DPRn (n=0~3)	TDB命令包含
	TDS	S	DPR0	
	TSA	A	S	
	TSB	B	S	
	TSD	DPR0	S	
	TSX	X	S	
	TXA	A	X	
	TXB	B	X	
	TXS	S	X	
	TXY	Y	X	
	TYA	A	Y	
	TYB	B	Y	
TYX	X	Y		
XAB	A	B		

機能	命令	動作	備考
メモリ間転送	MOVM	M M	LDM命令包含
	MOVMB	M8 M8	
	MOVR	M(dest n) M(source n) (複数指定可) (n=0~15)	
	MOV RB	M8(dest n) M8(source n) (複数指定可) (n=0~15)	
ブロック転送	MVN	M(n ~ n + i - 1) M(m ~ m + i - 1) (i: 転送バイト数)	
	MVP	M(n - i + 1 ~ n) M(m - i + 1 ~ m) (i: 転送バイト数)	
スタック操作	PEA	スタック IMM16	
	PEI	スタック M16 (DPRn + dd) (n=0~3)	
	PER	スタック PC + IMM16	
	PHA	スタック A	
	PHB	スタック B	
	PHD	スタック DPR0	
	PHD n	スタック DPRn (n=0~3。複数指定可)	
	PHG	スタック PG	
	PHP	スタック PS	
	PHT	スタック DT	
	PHX	スタック X	
	PHY	スタック Y	
	PLA	A スタック	
	PLB	B スタック	
	PLD	DPR0 スタック	
	PLD n	DPRn スタック (n=0~3。複数指定可)	
	PLP	PS スタック	
	PLT	DT スタック	
	PLX	X スタック	
	PLY	Y スタック	
	PSH	スタック A, B, X, Y, DPR0, DT, PG, PS中の指定レジスタ (複数指定可) M (S ~ S - i + 1) A, B, X, Y, DPR0, DT, PG, PS S S - i (i: スタックに退避するレジスタに応じたバイト数)	
	PUL	A, B, X, Y, DPR0, DT, PS中の指定レジスタ スタック (複数指定可) A, B, X, Y, DPR0, DT, PS M (S + 1 ~ S + i) S S + i (i: スタックから復帰するレジスタに応じたバイト数)	
	スタック操作&ロード	PHLD n	スタック DPRn, DPRn IMM16 (n=0~3。複数指定可)
クリア	CLR	Acc 0	
	CLRB	AccL 0	
	CLRM	M 0	
	CLRMB	M8 0	
	CLR X	X 0	
	CLRY	Y 0	

機能	命令	動作	備考
加算	ADC	Acc Acc + M + C	
	ADCB	AccL AccL + IMM8 + C	
	ADCD	E E + M32 + C	
	ADD	Acc Acc + M	
	ADDB	AccL AccL + IMM8	
	ADDD	E E + M32	
	ADDM	M M + IMM	
	ADDMB	M8 M8 + IMM8	
	ADDMD	M32 M32 + IMM32	
	ADDS	S S + IMM8	
	ADDX	X X + IMM (IMM=0~31)	
	ADDY	Y Y + IMM (IMM=0~31)	
インクリメント	INC	Acc Acc + 1 又は M M + 1	
	INX	X X + 1	
	INY	Y Y + 1	
減算	SBC	Acc Acc - M - \bar{C}	
	SBCB	AccL AccL - IMM8 - \bar{C}	
	SBCD	E E - M32 - \bar{C}	
	SUB	Acc Acc - M	
	SUBB	AccL AccL - IMM8	
	SUBD	E E - M32	
	SUBM	M M - IMM	
	SUBMB	M8 M8 - IMM8	
	SUBMD	M32 M32 - IMM32	
	SUBS	S S - IMM8	
	SUBX	X X - IMM (IMM=0~31)	
	SUBY	Y Y - IMM (IMM=0~31)	
デクリメント	DEC	Acc Acc - 1 又は M M - 1	
	DEX	X X - 1	
	DEY	Y Y - 1	
乗算	MPY	(B, A) A (被乗数) × M (乗数) 符号なし	
	MPYS	(B, A) A (被乗数) × M (乗数) 符号付き	
除算	DIV	A (商), B (剰余) (B, A) ÷ M 符号なし	
	DIVS	A (商), B (剰余) (B, A) ÷ M 符号付き	
繰り返し積和	RMPA	(B, A) (B, A) + M (DT:X) × M (DT:Y) (0 ~ 255回繰り返し)	

機能	命令	動作	備考
論理和	ORA	Acc Acc M	
	ORAB	AccL AccL IMM8	
	ORAM	M M IMM	SEB命令包含
	ORAMB	M8 M8 IMM8	
	ORAMD	M32 M32 IMM32	
論理積	AND	Acc Acc M	
	ANDB	AccL AccL IMM8	
	ANDM	M M IMM	CLB命令包含
	ANDMB	M8 M8 IMM8	
	ANDMD	M32 M32 IMM32	
排他的論理和	EOR	Acc Acc∨M	
	EORB	AccL AccL∨IMM8	
	EORM	M M∨IMM	
	EORMB	M8 M8∨IMM8	
	EORMD	M32 M32∨IMM32	
比較	CMP	Acc - M	
	CMPB	AccL - IMM8	
	CMPD	E - M32	
	CMPM	M - IMM	
	CMPMB	M8 - IMM8	
	CMPMD	M32 - IMM32	
	CPX	X - M	
	CPY	Y - M	
算術左シフト	ASL	Acc又はMの内容を左へ1ビットシフト	
	ASL #n	Aの内容を左へnビットシフト(n=0~15)	
	ASLD #n	Eの内容を左へnビットシフト(n=0~31)	
算術右シフト	ASR	Acc又はMの内容を符号を残し、右へ1ビットシフト	
	ASR #n	Aの内容を符号を残し、右へnビットシフト(n=0~15)	
	ASRD #n	Eの内容を符号を残し、右へnビットシフト(n=0~31)	
論理右シフト	LSR	Acc又はMの内容を右へ1ビットシフト	
	LSR #n	Aの内容を右へnビットシフト(n=0~15)	
	LSRD #n	Eの内容を右へnビットシフト(n=0~31)	
左回転	RLA	Aの内容を左へnビット回転 (「m=0」のとき：n=0~65535、「m=1」のとき：n=0~255)	
	ROL	Acc又はMとCをつなぎ、その内容を左へ1ビット回転	
	ROL #n	AとCをつなぎ、その内容を左へnビット回転(n=0~15)	
	ROLD #n	EとCをつなぎ、その内容を左へnビット回転(n=0~31)	
右回転	ROR	Acc又はMとCをつなぎ、その内容を右へ1ビット回転	
	ROR #n	AとCをつなぎ、その内容を右へnビット回転(n=0~15)	
	RORD #n	EとCをつなぎ、その内容を右へnビット回転(n=0~31)	

機能	命令	動作		備考
符号拡張	EXTS	Acc	AccL (符号拡張)	
	EXTSD	E	EL (=A) (符号拡張)	
ゼロ拡張	EXTZ	Acc	AccL (ゼロ拡張)	
	EXTZD	E	EL (=A) (ゼロ拡張)	
符号反転	NEG	Acc	-Acc	
	NEGD	E	-E	
絶対値	ABS	Acc	Acc	
	ABSD	E	E	
フラグ操作	CLC	C	0	
	CLI	I	0	
	CLM	m	0	
	CLP	PSL(bit n)	0 (n=0~7。複数指定可)	
	CLV	V	0	
	SEC	C	1	
	SEI	I	1	
	SEM	m	1	
	SEP	PSL(bit n)	1 (n=0~7。複数指定可)	
無条件分岐	BRA/BRAL	PC	PC + cnt + REL (cnt : BRA/BRAL命令のバイト数)	
	JMP	PC	飛び先番地	
	JMPL	PG, PC	飛び先番地	
サブルーチン 呼び出し	BSR	スタック	PC + 2	
		PC	PC + 2 + REL	
	JSR	スタック	PC + 3	
		PC	飛び先番地	
JSRL	スタック	PG, PC + 4		
		PG, PC	飛び先番地	

機能	命令	動作	備考
条件分岐	BBC	Mの指定ビットの内容が全て0のとき相対分岐	
	BBCB	M8の指定ビットの内容が全て0のとき相対分岐	
	BBS	Mの指定ビットの内容が全て1のとき相対分岐	
	BBSB	M8の指定ビットの内容が全て1のとき相対分岐	
	BCC	C = 0のとき相対分岐	
	BCS	C = 1のとき相対分岐	
	BEQ	Z = 1のとき相対分岐	
	BGE	N \vee V = 0のとき相対分岐	
	BGT	Z = 0 かつ N \vee V = 0のとき相対分岐	
	BGTU	C = 1 かつ Z = 0のとき相対分岐	
	BLE	Z = 1 又は N \vee V = 1のとき相対分岐	
	BLEU	C = 0 又は Z = 1のとき相対分岐	
	BLT	N \vee V = 1のとき相対分岐	
	BMI	N = 1のとき相対分岐	
	BNE	Z = 0のとき相対分岐	
	BPL	N = 0のとき相対分岐	
	BSC	A又はMの指定1ビットの内容が0のとき相対分岐	
	BSS	A又はMの指定1ビットの内容が1のとき相対分岐	
	BVC	V = 0のとき相対分岐	
	BVS	V = 1のとき相対分岐	
比較&条件分岐	CBEQ	Acc = IMM又はM = IMMのとき相対分岐	
	CBEQB	AccL = IMM8又はM8 = IMM8のとき相対分岐	
	CBNE	Acc \neq IMM又はM \neq IMMのとき相対分岐	
	CBNEB	AccL \neq IMM8又はM8 \neq IMM8のとき相対分岐	
デクリメント &条件分岐	DEBNE	M M - IMM, M \neq 0のとき相対分岐 (IMM=0~31)	
	DXBNE	X X - IMM, X \neq 0のとき相対分岐 (IMM=0~31)	
	DYBNE	Y Y - IMM, Y \neq 0のとき相対分岐 (IMM=0~31)	
リターン	RTI	PG, PC, PS スタック	
	RTL	PG, PC スタック	
	RTS	PC スタック	
ロード&リターン	RTLd n	DPRn スタック、PG, PC スタック(n=0~3。複数指定可)	
	RTSD n	DPRn スタック、PC スタック(n=0~3。複数指定可)	
ソフトウェア割り込み 特殊	BRK	BRK割り込みを発生	
	STP	発振停止	
	WIT	CPUクロック停止	
ノーオペレーション	NOP	PC PC + 1	

4.2 命令の説明

この節では、個別に各命令を説明します。ここでは、命令のニーモニックを表題としてアルファベット順に並べ、原則として1ページ1命令の形式で記述してあります。各命令において、その命令の動作と説明(注1、2)、ステータスフラグの変化、また、アセンブリ言語記述形式(注3)、機械語、バイト数、サイクル数(注4)の一覧表を記載しました(図4.2.1参照)。

ADD
ADD
ADD

【機能】 加算

【演算長】 16ビット又は8ビット 注1、注2

【動作】 $Acc \leftarrow Acc + M$

・ m= " 0 " のとき

Acc

←

Acc

+

M16

・ m= " 1 " のとき

AccL

←

AccL

+

M8

このときAccHの内容は変化しません。

【説明】 Accの内容とメモリの内容を加算し、その結果をAccへ格納します。 注4

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
IMM	ADD A,#imm	26 ₁₆ ,imm (81 ₁₆ ,26 ₁₆ ,imm)	2 (3)	1 (2)
DIR	ADD A,dd	2A ₁₆ ,dd (81 ₁₆ ,2A ₁₆ ,dd)	2 (3)	3 (4)
⋮	⋮	⋮	⋮	⋮
ABS,Y	ADD A,mml,Y	11 ₁₆ ,26 ₁₆ ,ll,mm (91 ₁₆ ,26 ₁₆ ,ll,mm)	4 (4)	5 (5)
ABL	ADD A,hhmml	11 ₁₆ ,2C ₁₆ ,ll,mm,hh (91 ₁₆ ,2C ₁₆ ,ll,mm,hh)	5 (5)	5 (5)
ABL,X	ADD A,hhmml,X	11 ₁₆ ,2D ₁₆ ,ll,mm,hh (91 ₁₆ ,2D ₁₆ ,ll,mm,hh)	5 (5)	6 (6)

【記述例】 注3

```

CLM
ADD.W      A,#IMM16      ; A  A + IMM16
ADD       B,MEM16      ; B  B + MEM16
SEM
ADD.B     A,#IMM8       ; AL AL + IMM8
ADD      BL,MEM8       ; BL BL + MEM8

```

図4.2.1 命令の記述例

注1. 各命令の動作説明中、PQ(プログラムカウンタ)に関しては特に処理に影響を与える命令のみ、その動作を記述します。

PCの内容は命令実行時にその命令バイト数が加算され、次に実行すべき命令の格納番地となります。その加算の際にキャリーが発生した場合、PQ(プログラムバンクレジスタ)は1加算されます。

注2. 各命令説明中の「動作」は、命令実行後の各レジスタ、メモリの内容を示すもので動作シーケンスの詳細は省略しています。

注3. 命令の記述例

本マニュアルに掲げているニーモニック記述例はアセンブリ言語記述の一例です。

7900シリーズのニーモニックのうち、特にアドレッシングモードの指定については下記のような形式をはじめとする様々な記述方法があります。詳しくはご使用になるアセンブラのユーザーズマニュアルを参照してください。

三菱製アセンブラの場合のアドレッシングモード指定方法

アドレッシングモード	指定方法	命令記述例
ダイレクト	DP0+:Offset6/8 DP0:label	ADD A,DP0+:04H ADD A,DP0:WORK
ダイレクト・インダイレクト	(DP0+:Offset6/8) (DP0:label)	ADD A,(DP0+:04H) ADD A,(DP0:WORK)
ダイレクト・インダイレクトロング	L(DP0+:Offset6/8) L(DP0:label)	ADD A,L(DP0+:04H) ADD A,L(DP0:WORK)
スタックポインタ・レラティブ	Offset,S	ADD A,05H,S
スタックポインタ・レラティブ ・インダイレクト・インデクストY	(Offset,S),Y	ADD A,(05H,S),Y
アブソリュート	DT+:Offset16 DT:label	ADD A,DT +:1000H ADD A,DT:WORK
アブソリュート・インダイレクト	(Address) (label)	JMP (1000H) JMP (TABLE)
アブソリュートロング	LG:label	ADD A,LG:WORK
アブソリュート・インダイレクトロング	L(DT+:Offset16) L(DT:label)	ADD A,L(DT +:1000H) ADD A,L(DT:WORK)

備考1 . Offset6/8...6ビット(DPR0~DPR3使用時)又は8ビット(DPR0時)のオフセット値

- 2 . Offset ...8ビットのオフセット値
- 3 . Offset16 ...16ビットのオフセット値
- 4 . Address ...参照するメモリ番地
- 5 . label ...参照するメモリ番地を示すラベル

注4. 一覧表のサイクル数は、最短時の数を記載しています。サイクル数は以下の条件によって変化します。

ダイレクトページレジスタの下位バイトの値

一覧表に示すサイクル数は、ダイレクトページレジスタの下位バイト(DPRnL)が00₁₆の場合の数です。DPRnL “00₁₆”の際にダイレクトページレジスタを使用するアドレッシングモードを使用する場合は、表中のサイクル数に+1した数が実際のサイクル数になります。

命令キューバッファに取り込まれているバイト数

メモリの読み出し、書き込み時の番地(偶数か奇数か)

外部データバス幅8ビットで外部メモリ領域をアクセスした場合

バスサイクル

以下に、命令を機能毎に分類して示します。説明、及び一覧表に用いた記号を示します。

記号	内容
C	キャリーフラグ
Z	ゼロフラグ
I	割り込み禁止フラグ
D	10進演算モードフラグ
x	インデックスレジスタ長選択フラグ
m	データ長選択フラグ
V	オーバフローフラグ
N	ネガティブフラグ
IPL	プロセッサ割り込み優先レベル
+	加算
-	減算
× 又は *	乗算
÷ 又は /	除算
∧	論理積
∨	論理和
∇	排他的論理和
	絶対値
—	否定
→	矢印の方向に移動することを示す
←	矢印の方向に移動することを示す
↔	交換
Acc	アキュムレータ
AccH	アキュムレータの上位8ビット
AccL	アキュムレータの下位8ビット
A	アキュムレータA
AH	アキュムレータAの上位8ビット
AL	アキュムレータAの下位8ビット
B	アキュムレータB
BH	アキュムレータBの上位8ビット
BL	アキュムレータBの下位8ビット
E	アキュムレータE
EH	アキュムレータEの上位16ビット
EL	アキュムレータEの下位16ビット
X	インデックスレジスタX
XH	インデックスレジスタXの上位8ビット
XL	インデックスレジスタXの下位8ビット
Y	インデックスレジスタY
YH	インデックスレジスタYの上位8ビット
YL	インデックスレジスタYの下位8ビット
S	スタックポインタ
REL	相対アドレス

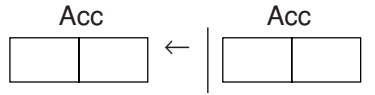
記号	内容
PC	プログラムカウンタ
PC _H	プログラムカウンタの上位8ビット
PC _L	プログラムカウンタの下位8ビット
PG	プログラムバンクレジスタ
DT	データバンクレジスタ
DPR0	ダイレクトページレジスタ0
DPR0 _H	ダイレクトページレジスタ0の上位8ビット
DPR0 _L	ダイレクトページレジスタ0の下位8ビット
DPR _n	ダイレクトページレジスタn
DPR _{nH}	ダイレクトページレジスタnの上位8ビット
DPR _{nL}	ダイレクトページレジスタnの下位8ビット
PS	プロセッサステータスレジスタ
PS _H	プロセッサステータスレジスタの上位8ビット
PS _L	プロセッサステータスレジスタの下位8ビット
PS(bit n)	プロセッサステータスレジスタ中のnビット目
M	メモリの内容
M _n , MEM _n	nビットのメモリの番地、又は内容
M(oprd)	()内のオペランド等で示されたメモリの内容
M(bit n)	メモリのnビット目
IMM	イミディエイト値(8ビット又は16ビット)
IMM _n	nビットのイミディエイト値
IMM _{nH}	nビットのイミディエイト値の上位データ
IMM _{nL}	nビットのイミディエイト値の下位データ
EAR	実効アドレス(16ビット)
EAR _H	実効アドレスの上位8ビット
EAR _L	実効アドレスの下位8ビット
MSB	最上位ビット
LSB	最下位ビット
dd	<ul style="list-style-type: none"> ダイレクトページレジスタ切り替えビットが0の場合、DPR0に対する変位(8ビット)を示す ダイレクトページレジスタ切り替えビットが1の場合、DPR_r(n=0~3)に対する変位(下位6ビット)及びDPR_r(上位2ビット)を示す
imm _{HH} imm _{HL} imm _{LH} imm _{LL}	32ビットイミディエイト値 (上位バイトから、imm _{HH} 、imm _{HL} 、imm _{LH} 、imm _{LL} の順に表す)
imm _H imm _L	16ビットイミディエイト値 (imm _H は上位8ビット、imm _L は下位8ビットを表す)
imm	8ビットイミディエイト値
imm _n	nビットイミディエイト値
hhmmll	24ビットアドレス値(hhは上位8ビット、mmは中位8ビット、llは下位8ビットを表す)
mml	16ビットアドレス値(mmは上位8ビット、llは下位8ビットを表す)
nn	Sに対する変位(8ビット)
n ₁ , n ₂	8ビットデータ値(2種類の8ビットデータ値を並記する場合に用いる)
rr	PCに対する変位(符号付き8ビット)
rr _{HH} rr _{LL}	PCに対する変位(符号付き16ビット) (rr _H は上位8ビット、rr _L は下位8ビットを表す)
hh ₁ , hh ₂	バンク指定(2種類の8ビットデータ)
source	転送元として指定するオペランド(ソース)
dest	転送先として指定するオペランド(ディスティネーション)

【機能】 絶対値

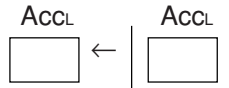
【演算長】 16ビット又は8ビット

【動作】 $Acc \leftarrow |Acc|$

・ $m = "0"$ のとき



・ $m = "1"$ のとき



このとき Acc_H の内容は変化しません。

【説明】 Acc の内容の絶対値をとり、その結果を Acc へ格納します。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	0	V					Z	0

N : 実行結果のMSBが“0”になるため、常に“0”になります。

V : 演算した結果が+32767（「 $m=1$ 」のときは+127）を越えると“1”になります。それ以外では“0”になります。

Z : 実行結果が“0”のとき、“1”になります。それ以外では“0”になります。

C : 常に“0”になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
A	ABS A	E_{16}	1	3
A	ABS B	$8_{16}, E_{16}$	2	4

【記述例】

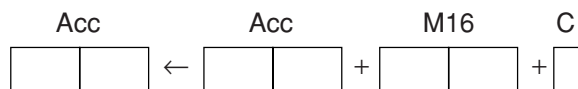
```
CLM
ABS A ; A |A|
SEM
ABS B ; BL |BL|
```


【機能】 加算（キャリー付き）

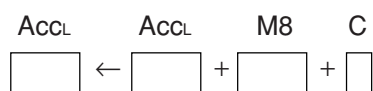
【演算長】 16ビット又は8ビット

【動作】 $Acc \leftarrow Acc + M + C$

・ $m = "0"$ のとき



・ $m = "1"$ のとき



このとき Acc_H の内容は変化しません。

【説明】 Acc 、メモリ、 C フラグの内容を加算し、その結果を Acc へ格納します。
「 $D=1$ 」のとき10進演算を行います。

【プロセッサステータスレジスタ】

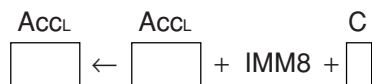
IPL	N	V	m	x	D	I	Z	C
	N	V					Z	C

- N : 実行結果のMSBが“1”のとき、“1”になります。それ以外では“0”になります。
「 $D=1$ 」のときは意味を持ちません。
- V : 符号付き演算と見なした結果が+32767、又は-32768（「 $m=1$ 」のときは+127又は-128）を越えると“1”になります。それ以外では“0”になります。
「 $D=1$ 」のときは意味を持ちません。
- Z : 実行結果が“0”のとき、“1”になります。それ以外では“0”になります。
「 $D=1$ 」のときは意味を持ちません。
- C : 「 $D=0$ 」のとき、符号なし演算と見なした結果が+65535（「 $m=1$ 」のときは+255）を越えると、“1”になります。それ以外では“0”になります。
「 $D=1$ 」のとき、符号なし演算と見なした結果が+9999（「 $m=1$ 」のときは+99）を越えると、“1”になります。それ以外では“0”になります。

【機能】 加算（キャリー付き）

【演算長】 8ビット

【動作】 $AcCL \leftarrow AcCL + IMM8 + C$



【説明】 AcCLの内容とイミディエイト値とCフラグの内容を8ビット長で加算し、その結果をAcCLへ格納します。
この命令はmフラグの影響を受けません。

AcCHの内容は変化しません。

「D=1」のとき10進演算を行います。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	N	V					Z	C

N : 実行結果のMSBが“1”のとき、“1”になります。それ以外では“0”になります。
「D=1」のときは意味を持ちません。

V : 符号付き演算と見なした結果が+127、又は-128を越えると“1”になります。
それ以外では“0”になります。
「D=1」のときは意味を持ちません。

Z : 実行結果が“0”のとき、“1”になります。それ以外では“0”になります。
「D=1」のときは意味を持ちません。

C : 符号なし演算と見なした結果が+255（「D=1」のときは+99）を越えると、“1”になります。
それ以外では“0”になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
IMM	ADCB A,#imm	31 ₁₆ ,1A ₁₆ ,imm	3	3
IMM	ADCB B,#imm	B1 ₁₆ ,1A ₁₆ ,imm	3	3

【記述例】

ADCB A,#IMM8 ; AL AL + IMM8 + C
ADCB B,#IMM8 ; BL BL + IMM8 + C

【機能】 加算（キャリー付き）

【演算長】 32ビット

「4.4 32ビット演算命令実行時の注意」
を参照してください。

【動作】 $E \leftarrow E + M32 + C$



【説明】 E、メモリ、Cフラグの内容を32ビット長で加算し、その結果をEへ格納します。
CPUはDフラグの内容にかかわらず、2進演算を行います。

この命令はmフラグの影響を受けません。

10進演算を行うことはできません。この命令を使用する際には「D=0」にしてください。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	N	V					Z	C

N : 実行結果のMSBが“1”のとき、“1”になります。それ以外では“0”になります。

V : 符号付き演算と見なした結果が+2147483647、又は-2147483648を越えると“1”になります。それ以外では“0”になります。

Z : 実行結果が“0”のとき、“1”になります。それ以外では“0”になります。

C : 符号なし演算と見なした結果が+4294967295を越えると、“1”になります。それ以外では“0”になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
IMM	ADCD E,#imm	31 ₁₆ ,1C ₁₆ ,immLL,immLH,immHL,immHH	6	4
DIR	ADCD E,dd	21 ₁₆ ,9A ₁₆ ,dd	3	7
DIR,X	ADCD E,dd,X	21 ₁₆ ,9B ₁₆ ,dd	3	8
(DIR)	ADCD E,(dd)	21 ₁₆ ,90 ₁₆ ,dd	3	9
(DIR,X)	ADCD E,(dd,X)	21 ₁₆ ,91 ₁₆ ,dd	3	10
(DIR),Y	ADCD E,(dd),Y	21 ₁₆ ,98 ₁₆ ,dd	3	10
L(DIR)	ADCD E,L(dd)	21 ₁₆ ,92 ₁₆ ,dd	3	11
L(DIR),Y	ADCD E,L(dd),Y	21 ₁₆ ,99 ₁₆ ,dd	3	12
SR	ADCD E,nn,S	21 ₁₆ ,93 ₁₆ ,nn	3	8
(SR),Y	ADCD E,(nn,S),Y	21 ₁₆ ,94 ₁₆ ,nn	3	11
ABS	ADCD E,mml	21 ₁₆ ,9E ₁₆ ,ll,mm	4	7
ABS,X	ADCD E,mml,X	21 ₁₆ ,9F ₁₆ ,ll,mm	4	8
ABS,Y	ADCD E,mml,Y	21 ₁₆ ,96 ₁₆ ,ll,mm	4	8
ABL	ADCD E,hhmml	21 ₁₆ ,9C ₁₆ ,ll,mm,hh	5	8
ABL,X	ADCD E,hhmml,X	21 ₁₆ ,9D ₁₆ ,ll,mm,hh	5	9

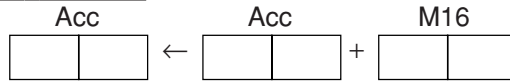
【記述例】

```
ADCD      E,#IMM32          ; E  E + IMM32 + C
                                ; (B,A  B,A + IMM32 + C)
ADCD      E,MEM32          ; E  E + MEM32 + C
                                ; (B,A  B,A + MEM32 + C)
```

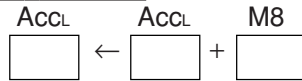

【機能】 加算

【演算長】 16ビット又は8ビット

【動作】 $Acc \leftarrow Acc + M$
 ・ $m = "0"$ のとき



・ $m = "1"$ のとき



このとき Acc_H の内容は変化しません。

【説明】 Acc の内容とメモリの内容を加算し、その結果を Acc へ格納します。
 10進演算を行うことはできません。この命令を使用する際には「 $D=0$ 」にしてください。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	N	V					Z	C

- N : 実行結果のMSBが“1”のとき、“1”になります。それ以外では“0”になります。
- V : 符号付き演算と見なした結果が+32767、又は-32768（「 $m=1$ 」のときは+127又は-128）を越えると、“1”になります。それ以外では“0”になります。
- Z : 実行結果が“0”のとき、“1”になります。それ以外では“0”になります。
- C : 符号なし演算と見なした結果が+65535（「 $m=1$ 」のときは+255）を越えると、“1”になります。それ以外では“0”になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
IMM	ADD A,#imm	26 ₁₆ ,imm (81 ₁₆ ,26 ₁₆ ,imm)	2 (3)	1 (2)
DIR	ADD A,dd	2A ₁₆ ,dd (81 ₁₆ ,2A ₁₆ ,dd)	2 (3)	3 (4)
DIR,X	ADD A,dd,X	2B ₁₆ ,dd (81 ₁₆ ,2B ₁₆ ,dd)	2 (3)	4 (5)
(DIR)	ADD A,(dd)	11 ₁₆ ,20 ₁₆ ,dd (91 ₁₆ ,20 ₁₆ ,dd)	3 (3)	6 (6)
(DIR),X	ADD A,(dd),X	11 ₁₆ ,21 ₁₆ ,dd (91 ₁₆ ,21 ₁₆ ,dd)	3 (3)	7 (7)
(DIR),Y	ADD A,(dd),Y	11 ₁₆ ,28 ₁₆ ,dd (91 ₁₆ ,28 ₁₆ ,dd)	3 (3)	7 (7)
L(DIR)	ADD A,L(dd)	11 ₁₆ ,22 ₁₆ ,dd (91 ₁₆ ,22 ₁₆ ,dd)	3 (3)	8 (8)
L(DIR),Y	ADD A,L(dd),Y	11 ₁₆ ,29 ₁₆ ,dd (91 ₁₆ ,29 ₁₆ ,dd)	3 (3)	9 (9)
SR	ADD A,nn,S	11 ₁₆ ,23 ₁₆ ,nn (91 ₁₆ ,23 ₁₆ ,nn)	3 (3)	5 (5)
(SR),Y	ADD A,(nn,S),Y	11 ₁₆ ,24 ₁₆ ,nn (91 ₁₆ ,24 ₁₆ ,nn)	3 (3)	8 (8)
ABS	ADD A,mml	2E ₁₆ ,ll,mm (81 ₁₆ ,2E ₁₆ ,ll,mm)	3 (4)	3 (4)
ABS,X	ADD A,mml,X	2F ₁₆ ,ll,mm (81 ₁₆ ,2F ₁₆ ,ll,mm)	3 (4)	4 (5)
ABS,Y	ADD A,mml,Y	11 ₁₆ ,26 ₁₆ ,ll,mm (91 ₁₆ ,26 ₁₆ ,ll,mm)	4 (4)	5 (5)
ABL	ADD A,hhmmll	11 ₁₆ ,2C ₁₆ ,ll,mm,hh (91 ₁₆ ,2C ₁₆ ,ll,mm,hh)	5 (5)	5 (5)
ABL,X	ADD A,hhmmll,X	11 ₁₆ ,2D ₁₆ ,ll,mm,hh (91 ₁₆ ,2D ₁₆ ,ll,mm,hh)	5 (5)	6 (6)

注1. この表はアキュムレータAを使用する場合について書かれています。アキュムレータBを使用する場合は記述形式の“ $A,$ ”の部分“ $B,$ ”になり、機械語、バイト数、サイクル数は、それぞれ（ ）内に示す値になります。
 2. IMMアドレッシングモードで、 m フラグが“0”の場合は、バイト数は1バイト増加します。

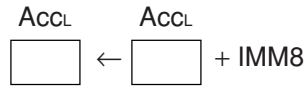
【記述例】

```
CLM
ADD.W      A,#IMM16          ; A   A + IMM16
ADD        B,MEM16          ; B   B + MEM16
SEM
ADD.B      A,#IMM8          ; AL  AL + IMM8
ADD        B,MEM8           ; BL  BL + MEM8
```

【機能】 加算

【演算長】 8ビット

【動作】 $AcCL \leftarrow AcCL + IMM8$



【説明】 AcCLの内容とイミディエイト値を8ビット長で加算し、その結果をAcCLへ格納します。

この命令はmフラグの影響を受けません。

AcCHの内容は変化しません。

10進演算を行うことはできません。この命令を使用する際には「D=0」にしてください。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	N	V					Z	C

N : 実行結果のMSBが“1”のとき、“1”になります。それ以外では“0”になります。

V : 符号付き演算と見なした結果が+127、又は-128を越えると“1”になります。それ以外では“0”になります。

Z : 実行結果が“0”のとき、“1”になります。それ以外では“0”になります。

C : 符号なし演算と見なした結果が+255を越えると、“1”になります。それ以外では“0”になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
IMM	ADDB A,#imm	29 ₁₆ ,imm	2	1
IMM	ADDB B,#imm	81 ₁₆ ,29 ₁₆ ,imm	3	2

【記述例】

ADDB A,#IMM8 ; AL AL + IMM8
ADDB B,#IMM8 ; BL BL + IMM8

【機能】 加算

【演算長】 32ビット

「4.4 32ビット演算命令実行時の注意」を参照してください。

【動作】 $E \leftarrow E + M32$



【説明】 Eの内容とメモリの内容を32ビット長で加算し、その結果をEへ格納します。

この命令はmフラグの影響を受けません。

10進演算を行うことはできません。この命令を使用する際には「D=0」にしてください。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	N	V					Z	C

N : 実行結果のMSBが“1”のとき、“1”になります。それ以外では“0”になります。

V : 符号付き演算と見なした結果が+2147483647、又は-2147483648を越えると“1”になります。それ以外では“0”になります。

Z : 実行結果が“0”のとき、“1”になります。それ以外では“0”になります。

C : 符号なし演算と見なした結果が+4294967295を越えると、“1”になります。それ以外では“0”になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
IMM	ADDD E,#imm	2D ₁₆ ,immLL,immLH,immHL,immHH	5	3
DIR	ADDD E,dd	9A ₁₆ ,dd	2	6
DIR,X	ADDD E,dd,X	9B ₁₆ ,dd	2	7
(DIR)	ADDD E,(dd)	11 ₁₆ ,90 ₁₆ ,dd	3	9
(DIR,X)	ADDD E,(dd,X)	11 ₁₆ ,91 ₁₆ ,dd	3	10
(DIR),Y	ADDD E,(dd),Y	11 ₁₆ ,98 ₁₆ ,dd	3	10
L(DIR)	ADDD E,L(dd)	11 ₁₆ ,92 ₁₆ ,dd	3	11
L(DIR),Y	ADDD E,L(dd),Y	11 ₁₆ ,99 ₁₆ ,dd	3	12
SR	ADDD E,nn,S	11 ₁₆ ,93 ₁₆ ,nn	3	8
(SR),Y	ADDD E,(nn,S),Y	11 ₁₆ ,94 ₁₆ ,nn	3	11
ABS	ADDD E,mml	9E ₁₆ ,ll,mm	3	6
ABS,X	ADDD E,mml,X	9F ₁₆ ,ll,mm	3	7
ABS,Y	ADDD E,mml,Y	11 ₁₆ ,96 ₁₆ ,ll,mm	4	8
ABL	ADDD E,hhmmll	11 ₁₆ ,9C ₁₆ ,ll,mm,hh	5	8
ABL,X	ADDD E,hhmmll,X	11 ₁₆ ,9D ₁₆ ,ll,mm,hh	5	9

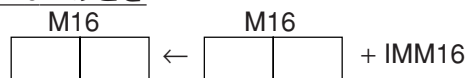
【記述例】

ADDD E,#IMM32 ; E E + IMM32 (B,A B,A + IMM32)
 ADDD E,MEM32 ; E E + MEM32 (B,A B,A + MEM32)

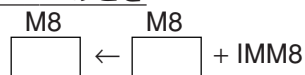
【機能】 加算

【演算長】 16ビット又は8ビット

【動作】 $M \leftarrow M + IMM$
 ・ $m = "0"$ のとき



・ $m = "1"$ のとき



【説明】 メモリの内容とイミディエイト値を加算し、その結果をメモリへ格納します。
 10進演算を行うことはできません。この命令を使用する際には「D=0」にしてください。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	N	V					Z	C

- N : 実行結果のMSBが“1”のとき、“1”になります。それ以外では“0”になります。
- V : 符号付き演算と見なした結果が+32767、又は-32768（「m=1」のときは+127又は-128）を越えると“1”になります。それ以外では“0”になります。
- Z : 実行結果が“0”のとき、“1”になります。それ以外では“0”になります。
- C : 符号なし演算と見なした結果が+65535（「m=1」のときは+255）を越えると、“1”になります。それ以外では“0”になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
DIR	ADDM dd,#imm	51 ₁₆ ,03 ₁₆ ,dd,imm	4	7
ABS	ADDM mml,#imm	51 ₁₆ ,07 ₁₆ ,ll,mm,imm	5	7

注 . mフラグが“0”の場合は、バイト数は1バイト増加します。

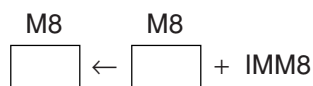
【記述例】

```
CLM
ADDM.W      MEM16,#IMM16          ; MEM16  MEM16 + IMM16
SEM
ADDM.B      MEM8,#IMM8           ; MEM8   MEM8 + IMM8
```

【機能】 加算

【演算長】 8ビット

【動作】 $M8 \leftarrow M8 + IMM8$



【説明】 メモリの内容とイミディエイト値を8ビット長で加算し、その結果をメモリへ格納します。
この命令はmフラグの影響を受けません。
10進演算を行うことはできません。この命令を使用する際には「D=0」にしてください。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	N	V					Z	C

- N : 実行結果のMSBが“1”のとき、“1”になります。それ以外では“0”になります。
- V : 符号付き演算と見なした結果が+127、又は-128を越えると“1”になります。それ以外では“0”になります。
- Z : 実行結果が“0”のとき、“1”になります。それ以外では“0”になります。
- C : 符号なし演算と見なした結果が+255を越えると、“1”になります。それ以外では“0”になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
DIR	ADDMB dd,#imm	51 ₁₆ ,02 ₁₆ ,dd,imm	4	7
ABS	ADDMB mmll,#imm	51 ₁₆ ,06 ₁₆ ,ll,mm,imm	5	7

【記述例】

ADDMB MEM8,#IMM8 ; MEM8 MEM8 + IMM8

【機能】 加算

【演算長】 32ビット

「4.4 32ビット演算命令実行時の注意」を参照してください。

【動作】 $M32 \leftarrow M32 + IMM32$



【説明】 メモリの内容とイミディエイト値を32ビット長で加算し、その結果をメモリへ格納します。この命令はmフラグの影響を受けません。10進演算を行うことはできません。この命令を使用する際には「D=0」にしてください。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	N	V					Z	C

- N : 実行結果のMSBが“1”のとき、“1”になります。それ以外では“0”になります。
- V : 符号付き演算と見なした結果が+2147483647、又は-2147483648を越えると“1”になります。それ以外では“0”になります。
- Z : 実行結果が“0”のとき、“1”になります。それ以外では“0”になります。
- C : 符号なし演算と見なした結果が+4294967295を越えると、“1”になります。それ以外では“0”になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
DIR	ADDMD dd,#imm	51 ₁₆ ,83 ₁₆ ,dd,immLL,immLH,immHL,immHH	7	10
ABS	ADDMD mml,#imm	51 ₁₆ ,87 ₁₆ ,ll,mm,immLL,immLH,immHL,immHH	8	10

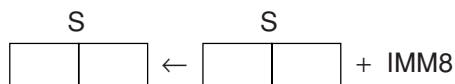
【記述例】

ADDMD MEM32,#IMM32 ; MEM32 MEM32 + IMM32

【機能】 加算

【演算長】 16ビット

【動作】 $S \leftarrow S + \text{IMM8}$



【説明】 Sの内容と8ビット長のイミディエイト値を16ビット長で加算し、その結果をSへ格納します。演算の際、イミディエイト値を16ビット長にゼロ拡張します。

この命令はmフラグの影響を受けません。

10進演算を行うことはできません。この命令を使用する際には「D=0」にしてください。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	N	V					Z	C

N : 実行結果のMSBが“1”のとき、“1”になります。それ以外では“0”になります。

V : 符号付き演算と見なした結果が+32767、又は-32768を越えると“1”になります。それ以外では“0”になります。

Z : 実行結果が“0”のとき、“1”になります。それ以外では“0”になります。

C : 符号なし演算と見なした結果が+65535を越えると、“1”になります。それ以外では“0”になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
IMM	ADDS #imm	31 ₁₆ ,0A ₁₆ ,imm	3	2

【記述例】

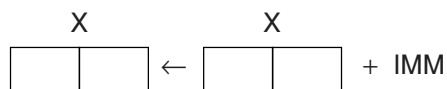
ADDS #IMM8 ; S S + IMM8

【機能】 加算

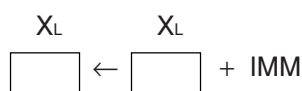
【演算長】 16ビット又は8ビット

【動作】 $X \leftarrow X + IMM$ (IMM=0~31)

・x=“0”のとき



・x=“1”のとき



このときX_Hの内容は変化しません。

【説明】 Xの内容とイミディエイト値(0~31)を加算し、その結果をXへ格納します。

この命令はmフラグの影響を受けません。

10進演算を行うことはできません。この命令を使用する際には「D=0」にしてください。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	N	V					Z	C

N : 実行結果のMSBが“1”のとき、“1”になります。それ以外では“0”になります。

V : 符号付き演算と見なした結果が+32767、又は-32768(「x=1」のときは+127又は-128)を越えると“1”になります。それ以外では“0”になります。

Z : 実行結果が“0”のとき、“1”になります。それ以外では“0”になります。

C : 符号なし演算と見なした結果が+65535(「x=1」のときは+255)を越えると、“1”になります。それ以外では“0”になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
IMM	ADDX #imm	01 ₁₆ ,imm	2	2

注．immには0~31の値が設定できます。

【記述例】

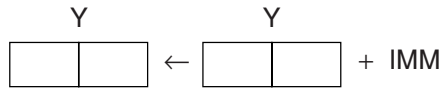
```
CLP          x
ADDX         #IMM          ; X   X + IMM
SEP          x
ADDX         #IMM          ; XL XL + IMM
```


【機能】 加算

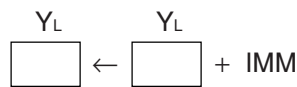
【演算長】 16ビット又は8ビット

【動作】 $Y \leftarrow Y + IMM$ (IMM=0~31)

・x=“0”のとき



・x=“1”のとき



このときY_Hの内容は変化しません。

【説明】 Yの内容とイミディエイト値(0~31)を加算し、その結果をYへ格納します。

この命令はmフラグの影響を受けません。

10進演算を行うことはできません。この命令を使用する際には「D=0」にしてください。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	N	V					Z	C

N : 実行結果のMSBが“1”のとき、“1”になります。それ以外では“0”になります。

V : 符号付き演算と見なした結果が+32767、又は-32768(「x=1」のときは+127又は-128)を越えると“1”になります。それ以外では“0”になります。

Z : 実行結果が“0”のとき、“1”になります。それ以外では“0”になります。

C : 符号なし演算と見なした結果が+65535(「x=1」のときは+255)を越えると、“1”になります。それ以外では“0”になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
IMM	ADDY #imm	01 ₁₆ ,imm+20 ₁₆	2	2

注 . immには0~31の値が設定できます。

【記述例】

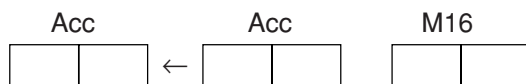
```
CLP          x
ADDY         #IMM          ; Y   Y + IMM
SEP          x
ADDY         #IMM          ; YL  YL + IMM
```

【機能】 論理積

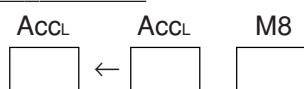
【演算長】 16ビット又は8ビット

【動作】 $Acc \leftarrow Acc \quad M$

・ $m = "0"$ のとき



・ $m = "1"$ のとき



このとき Acc_H の内容は変化しません。

【説明】 Acc の内容とメモリの内容の論理積をとり、結果を Acc へ格納します。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	N						Z	

N : 実行結果のMSBが“1”のとき、“1”になります。それ以外では“0”になります。

Z : 実行結果が“0”のとき、“1”になります。それ以外では“0”になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
IMM	AND A,#imm	66 ₁₆ ,imm (81 ₁₆ ,66 ₁₆ ,imm)	2 (3)	1 (2)
DIR	AND A,dd	6A ₁₆ ,dd (81 ₁₆ ,6A ₁₆ ,dd)	2 (3)	3 (4)
DIR,X	AND A,dd,X	6B ₁₆ ,dd (81 ₁₆ ,6B ₁₆ ,dd)	2 (3)	4 (5)
(DIR)	AND A,(dd)	11 ₁₆ ,60 ₁₆ ,dd (91 ₁₆ ,60 ₁₆ ,dd)	3 (3)	6 (6)
(DIR,X)	AND A,(dd,X)	11 ₁₆ ,61 ₁₆ ,dd (91 ₁₆ ,61 ₁₆ ,dd)	3 (3)	7 (7)
(DIR),Y	AND A,(dd),Y	11 ₁₆ ,68 ₁₆ ,dd (91 ₁₆ ,68 ₁₆ ,dd)	3 (3)	7 (7)
L(DIR)	AND A,L(dd)	11 ₁₆ ,62 ₁₆ ,dd (91 ₁₆ ,62 ₁₆ ,dd)	3 (3)	8 (8)
L(DIR),Y	AND A,L(dd),Y	11 ₁₆ ,69 ₁₆ ,dd (91 ₁₆ ,69 ₁₆ ,dd)	3 (3)	9 (9)
SR	AND A,nn,S	11 ₁₆ ,63 ₁₆ ,nn (91 ₁₆ ,63 ₁₆ ,nn)	3 (3)	5 (5)
(SR),Y	AND A,(nn,S),Y	11 ₁₆ ,64 ₁₆ ,nn (91 ₁₆ ,64 ₁₆ ,nn)	3 (3)	8 (8)
ABS	AND A,mll	6E ₁₆ ,ll,mm (81 ₁₆ ,6E ₁₆ ,ll,mm)	3 (4)	3 (4)
ABS,X	AND A,mll,X	6F ₁₆ ,ll,mm (81 ₁₆ ,6F ₁₆ ,ll,mm)	3 (4)	4 (5)
ABS,Y	AND A,mll,Y	11 ₁₆ ,66 ₁₆ ,ll,mm (91 ₁₆ ,66 ₁₆ ,ll,mm)	4 (4)	5 (5)
ABL	AND A,hhmll	11 ₁₆ ,6C ₁₆ ,ll,mm,hh (91 ₁₆ ,6C ₁₆ ,ll,mm,hh)	5 (5)	5 (5)
ABL,X	AND A,hhmll,X	11 ₁₆ ,6D ₁₆ ,ll,mm,hh (91 ₁₆ ,6D ₁₆ ,ll,mm,hh)	5 (5)	6 (6)

注1. この表はアキュムレータAを使用する場合について書かれています。アキュムレータBを使用する場合は記述形式の“A,”の部分“A,”になり、機械語、バイト数、サイクル数は()内の値になります。

2. IMMアドレッシングモードで、mフラグが“0”の場合は、バイト数は1バイト増加します。

【記述例】

```
CLM
AND.W      A,#IMM16          ; A   A   IMM16
AND        B,MEM16          ; B   B   MEM16
SEM
AND.B      A,#IMM8           ; AL  AL  IMM8
AND        B,MEM8            ; BL  BL  MEM8
```


【機能】 論理積

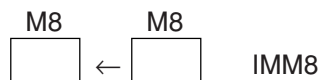
【演算長】 16ビット又は8ビット

【動作】 $M \leftarrow M \text{ AND } IMM$

・ $m = "0"$ のとき



・ $m = "1"$ のとき



【説明】 メモリの内容とイミディエイト値の論理積をとり、結果をメモリへ格納します。
従来の7700ファミリのCLB命令の機能はこの命令に含まれます。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	N						Z	

N : 実行結果のMSBが“1”のとき、“1”になります。それ以外では“0”になります。

Z : 実行結果が“0”のとき、“1”になります。それ以外では“0”になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
DIR	ANDM dd,#imm	51 ₁₆ ,63 ₁₆ ,dd,imm	4	7
ABS	ANDM mml,#imm	51 ₁₆ ,67 ₁₆ ,ll,mm,imm	5	7

注 . mフラグが“0”の場合は、バイト数は1バイト増加します。

【記述例】

```
CLM
ANDM.W    MEM16,#IMM16          ; MEM16    MEM16    IMM16
SEM
ANDM.B    MEM8,#IMM8           ; MEM8    MEM8    IMM8
```

【機能】 論理積

【演算長】 8ビット

【動作】 $M8 \leftarrow M8 \text{ AND } IMM8$

$\begin{array}{ccc} M8 & & IMM8 \\ \boxed{} & \leftarrow & \boxed{} \\ & & IMM8 \end{array}$

【説明】 8ビット長でメモリの内容とイミディエイト値の論理積をとり、結果をメモリへ格納します。
この命令はmフラグの影響を受けません。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	N						Z	

N : 実行結果のMSBが“1”のとき、“1”になります。それ以外では“0”になります。

Z : 実行結果が“0”のとき、“1”になります。それ以外では“0”になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
DIR	ANDMB dd,#imm	51 ₁₆ ,62 ₁₆ ,dd,imm	4	7
ABS	ANDMB mmlI,#imm	51 ₁₆ ,66 ₁₆ ,ll,mm,imm	5	7

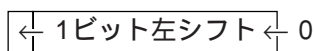
【記述例】

ANDMB MEM8,#IMM8 ; MEM8 MEM8 IMM8

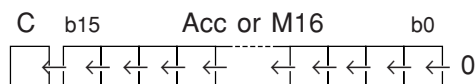
【機能】 算術左シフト

【演算長】 16ビット又は8ビット

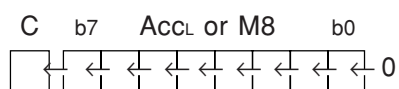
【動作】 C Acc or M



・ m=“0” のとき



・ m=“1” のとき



このときAccHの内容は変化しません。

【説明】 Acc又はメモリの内容を左へ1ビットシフトします。このとき、Acc又はメモリのLSBには“0”が入ります。また、シフトする前のMSBの内容はCフラグへ入ります。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	N						Z	C

N : 実行結果のMSBが“1”のとき、“1”になります。それ以外では“0”になります。

Z : 実行結果が“0”のとき、“1”になります。それ以外では“0”になります。

C : 実行前のMSBが“1”のとき、“1”になります。それ以外では“0”になります。


アドレッシングモード	記述形式	機械語	バイト数	サイクル数
A	ASL A	03 ₁₆	1	1
A	ASL B	81 ₁₆ ,03 ₁₆	2	2
DIR	ASL dd	21 ₁₆ ,0A ₁₆ ,dd	3	7
DIR,X	ASL dd,X	21 ₁₆ ,0B ₁₆ ,dd	3	8
ABS	ASL mml	21 ₁₆ ,0E ₁₆ ,ll,mm	4	7
ABS,X	ASL mml,X	21 ₁₆ ,0F ₁₆ ,ll,mm	4	8

【記述例】

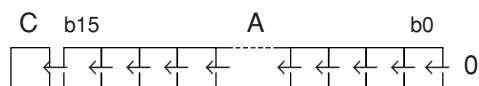
```
CLM
ASL      A          ; A   Aを1ビット算術左シフト
ASL     MEM16      ; MEM16 MEM16を1ビット算術左シフト
SEM
ASL     A          ; AL  ALを1ビット算術左シフト
ASL     MEM8      ; MEM8 MEM8を1ビット算術左シフト
```

【機能】 算術左シフト

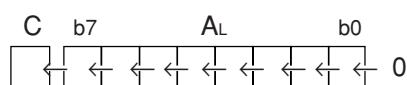
【演算長】 16ビット又は8ビット

【動作】 C A
 (n: シフト回数。 n=0~15)

・ m= “0” のとき



・ m= “1” のとき



このときA_Hの内容は変化しません。

【説明】 Aの内容を左へnビットシフトします。このとき、1ビットシフトごとに、Aのビット0には“0”が入ります。また、1ビットシフトごとに、MSBはCフラグへ入ります。
 この命令ではBは使用できません。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	N						Z	C

N : 実行結果のMSBが“1”のとき、“1”になります。それ以外では“0”になります。

Z : 実行結果が“0”のとき、“1”になります。それ以外では“0”になります。

C : (n-1)ビットシフトしたときのMSBが“1”のとき、“1”になります。それ以外では“0”になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
A	ASL A,#imm	C1 ₁₆ ,imm+40 ₁₆	2	imm+6


注 . immには0~15の値(シフト回数)が設定できます。

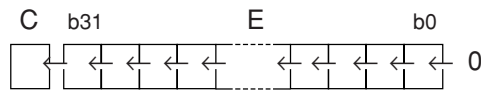
【記述例】

CLM
 ASL A,#15 ; A Aを15ビット算術左シフト
 SEM
 ASL A,#7 ; AL ALを7ビット算術左シフト

【機能】 算術左シフト

【演算長】 32ビット

【動作】 C E
 (n : シフト回数。n=0~31)



【説明】 Eの内容を32ビット長で左へnビットシフトします。このとき、1ビットシフトごとに、Eのビット0には“0”が入ります。また、1ビットシフトごとに、MSBはCフラグへ入ります。

この命令はmフラグの影響を受けません。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	N						Z	C

N : 実行結果のMSBが“1”のとき、“1”になります。それ以外では“0”になります。

Z : 実行結果が“0”のとき、“1”になります。それ以外では“0”になります。

C : (n-1)ビットシフトしたときのMSBが“1”のとき、“1”になります。それ以外では、“0”になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
A	ASLD E,#imm	D1 ₁₆ ,imm+40 ₁₆	2	imm+8

注 . immには0~31の値(シフト回数)が設定できます。

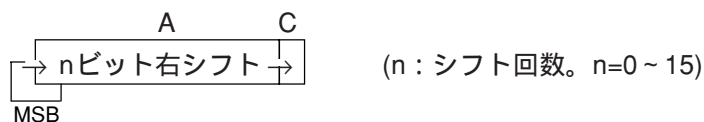
【記述例】

ASLD E,#16 ; E Eを16ビット算術左シフト

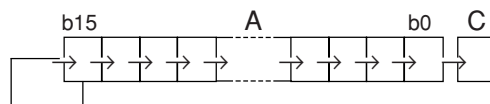
【機能】 算術右シフト

【演算長】 16ビット又は8ビット

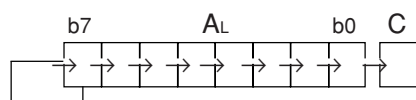
【動作】



・ m= "0" のとき



・ m= "1" のとき



このときAHの内容は変化しません。

【説明】 Aの内容を右へnビットシフトします。このとき、AのMSBにはシフトする前のMSBが入ります。また、1ビットシフトごとに、LSBはCフラグへ入ります。

この命令ではBは使用できません。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	N						Z	C

N : 実行結果のMSBが"1"のとき、"1"になります。それ以外では"0"になります。

Z : 実行結果が"0"のとき、"1"になります。それ以外では"0"になります。

C : (n-1)ビットシフトしたときのLSBが"1"のとき、"1"になります。それ以外では"0"になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
A	ASR A,#imm	C1 ₁₆ ,imm+80 ₁₆	2	imm+6

注．immには0~15の値(シフト回数)が設定できます。

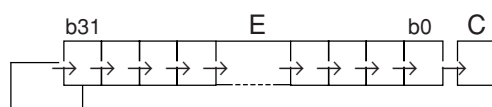
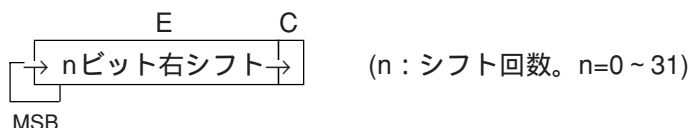
【記述例】

```
CLM
ASR      A,#15      ; A   Aを15ビット算術右シフト
SEM
ASR      A,#7       ; AL  ALを7ビット算術右シフト
```

【機能】 算術右シフト

【演算長】 32ビット

【動作】



【説明】 Eの内容を32ビット長で右へnビットシフトします。このとき、EのMSBにはシフトする前のMSBが入ります。また、1ビットシフトごとに、LSBはCフラグへ入ります。

この命令はmフラグの影響を受けません。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	N						Z	C

N : 実行結果のMSBが“1”のとき、“1”になります。それ以外では“0”になります。

Z : 実行結果が“0”のとき、“1”になります。それ以外では“0”になります。

C : (n-1)ビットシフトしたときのLSBが“1”のとき、“1”になります。

それ以外では“0”になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
A	ASRD E,#imm	D1 ₁₆ ,imm+80 ₁₆	2	imm+8

注: immには0~31の値(シフト回数)が設定できます。

【記述例】

ASRD E,#16 ; E Eを16ビット算術右シフト

【機能】 条件分岐

【演算長】 16ビット又は8ビット

【動作】 $M(\text{bit } n) = 0$ のとき指定番地へ相対分岐 (n はビット位置を指定し、複数ビット指定可)

【説明】 メモリ中の指定されたビット (複数指定可) の内容がすべて “0” のとき、指定された番地へ分岐します。PCに対する8ビットの相対値 (-128 ~ +127) で、分岐先の番地を指定します。テストするビット位置はイミディエイト値のビットパターンで示し、“1” にセットしたビットが対象のビットとなります。

「 $m=0$ 」のとき：16ビット長で演算します。

「 $m=1$ 」のとき：8ビット長で演算します。

テストするビットの指定がないときは分岐します。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
DIR,b,R	BBC #imm,dd,rr	$41_{16}, 5A_{16}, dd, imm, rr$	5	9
ABS,b,R	BBC #imm,mml,rr	$41_{16}, 5E_{16}, ll, mm, imm, rr$	6	9

注．mフラグが“0”の場合は、バイト数は1バイト増加します。

【記述例】

CLM

BBC.W #IMM16, MEM16, LABEL1 ; MEM16中の指定ビットがすべて“0”のとき、
; LABEL1へ分岐

SEM

BBC.B #IMM8, MEM8, LABEL2 ; MEM8中の指定ビットがすべて“0”のとき、
; LABEL2へ分岐

【機能】 条件分岐

【演算長】 16ビット又は8ビット

【動作】 $M(\text{bit } n) = 1$ のとき指定番地へ相対分岐 (n はビット位置を指定し、複数ビット指定可)

【説明】 メモリ中の指定されたビット (複数指定可) の内容がすべて “1” のとき、指定された番地へ分岐します。PCに対する8ビットの相対値 (-128 ~ +127) で、分岐先の番地を指定します。テストするビット位置はイミディエイト値のビットパターンで示し、“1” にセットしたビットが対象のビットとなります。

「 $m=0$ 」のとき：16ビット長で演算します。

「 $m=1$ 」のとき：8ビット長で演算します。

テストするビットの指定がないときは分岐します。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
DIR,b,R	BBS #imm,dd,rr	41 ₁₆ ,4A ₁₆ ,dd,imm,rr	5	9
ABS,b,R	BBS #imm,mml,rr	41 ₁₆ ,4E ₁₆ ,ll,mm,imm,rr	6	9

注 . mフラグが “0” の場合は、バイト数は1バイト増加します。

【記述例】

```
CLM
BBS.W      #IMM16,MEM16,LABEL1      ; MEM16中の指定ビットがすべて “1” のとき、
                                       ; LABEL1へ分岐

SEM
BBS.B      #IMM8,MEM8,LABEL2        ; MEM8中の指定ビットがすべて “1” のとき、
                                       ; LABEL2へ分岐
```


【機能】 条件分岐

【演算長】 -

【動作】 C=0のとき指定番地へ相対分岐

【説明】 Cフラグが“0”のとき、指定された番地へ分岐します。PCに対する8ビットの相対値（-128～+127）で、分岐先の番地を指定します。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
REL	BCC rr	90 ₁₆ ,rr	2	6

【記述例】

BCC LABEL ; C = 0のときLABELへ分岐

【機能】 条件分岐

【演算長】 -

【動作】 C=1のとき指定番地へ相対分岐

【説明】 Cフラグが“1”のとき、指定された番地へ分岐します。PCに対する8ビットの相対値（-128～+127）で、分岐先の番地を指定します。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
REL	BCS rr	B0 ₁₆ ,rr	2	6

【記述例】

BCS LABEL ; C = 1のときLABELへ分岐

【機能】 条件分岐

【演算長】 -

【動作】 Z=1のとき指定番地へ相対分岐

【説明】 Zフラグが“1”のとき、指定された番地へ分岐します。PCに対する8ビットの相対値（-128～+127）で、分岐先の番地を指定します。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
REL	BEQ rr	F0 ₁₆ ,rr	2	6

【記述例】

BEQ LABEL ; Z = 1のときLABELへ分岐

【機能】 条件分岐

【演算長】 -

【動作】 N = V=0のとき指定番地へ相対分岐

【説明】 NとVの内容が同じとき、指定された番地へ分岐します。PCに対する8ビットの相対値（-128～+127）で、分岐先の番地を指定します。

比較命令、減算命令の結果が「Greater or Equal」条件のとき分岐します。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
REL	BGE rr	C0 ₁₆ ,rr	2	6

【記述例】

BGE LABEL ; N = V = 0のときLABELへ分岐



【機能】 条件分岐

【演算長】 -

【動作】 Z=0 かつ N V=0のとき指定番地へ相対分岐

【説明】 Zの内容が“0”、かつ、NとVの内容が等しいとき、指定された番地へ分岐します。PCに対する8ビットの相対値(-128~+127)で、分岐先の番地を指定します。

比較命令、減算命令の結果が符号付き「Greater than >」条件のとき分岐します。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
REL	BGT rr	80 ₁₆ ,rr	2	6

【記述例】

BGT LABEL ; Z = 0 かつ N V = 0のときLABELへ分岐

【機能】 条件分岐

【演算長】 -

【動作】 C=1 かつ Z=0のとき指定番地へ相対分岐

【説明】 Cの内容が“1”、かつ、Zの内容が“0”のとき、指定された番地へ分岐します。PCに対する8ビットの相対値(-128~+127)で、分岐先の番地を指定します。

比較命令、減算命令の結果が符号なし「Greater than >」条件のとき分岐します。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
REL	BGTU rr	40 ₁₆ ,rr	2	6

【記述例】

BGTU LABEL ; C = 1 かつ Z = 0のときLABELへ分岐

【機能】 条件分岐

【演算長】 -

【動作】 Z=1 又は N V=1のとき指定番地へ相対分岐

【説明】 Zの内容が“1”、又はNとVの内容が異なるとき、指定された番地へ分岐します。PCに対する8ビットの相対値(-128~+127)で、分岐先の番地を指定します。

比較命令、減算命令の結果が符号付き「Less Equal」条件のとき分岐します。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
REL	BLE rr	A0 ₁₆ ,rr	2	6

【記述例】

BLE LABEL ; Z = 1 又は N V = 1のときLABELへ分岐

【機能】 条件分岐

【演算長】 -

【動作】 C=0 又は Z=1のとき指定番地へ相対分岐

【説明】 Cの内容が“0”、又はZの内容が“1”のとき、指定された番地へ分岐します。PCに対する8ビットの相対値(-128~+127)で、分岐先の番地を指定します。

比較命令、減算命令の結果が符号なし「Less or Equal」条件のとき分岐します。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
REL	BLEU rr	60 ₁₆ ,rr	2	6

【記述例】

BLEU LABEL ; C = 0 又は Z = 1のときLABELへ分岐

【機能】 条件分岐

【演算長】 -

【動作】 N = V=1のとき指定番地へ相対分岐

【説明】 NとVの内容が異なるとき、指定された番地へ分岐します。PCに対する8ビットの相対値（-128～+127）で、分岐先の番地を指定します。

比較命令、減算命令の結果が「Less than <」条件のとき分岐します。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
REL	BLT rr	E0 ₁₆ ,rr	2	6

【記述例】

BLT LABEL ; N = V = 1のときLABELへ分岐

【機能】 条件分岐

【演算長】 -

【動作】 N=1のとき指定番地へ相対分岐

【説明】 Nフラグが“1”のとき、指定された番地へ分岐します。PCに対する8ビットの相対値（-128～+127）で、分岐先の番地を指定します。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C

アドレッシングモード	記述形式	機械語	ビット数	サイクル数
REL	BMI rr	30 ₁₆ ,rr	2	6

【記述例】

BMI LABEL ; N = 1のときLABELへ分岐

【機能】 条件分岐

【演算長】 -

【動作】 Z=0のとき指定番地へ相対分岐

【説明】 Zフラグが“0”のとき、指定された番地へ分岐します。PCに対する8ビットの相対値（-128～+127）で、分岐先の番地を指定します。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C

アドレッシングモード	記述形式	機械語	ビット数	サイクル数
REL	BNE rr	D0 ₁₆ ,rr	2	6

【記述例】

BNE LABEL ; Z = 0のときLABELへ分岐

EOL announced Product

【機能】 条件分岐

【演算長】 -

【動作】 N=0のとき指定番地へ相対分岐

【説明】 Nフラグが“0”のとき、指定された番地へ分岐します。PCに対する8ビットの相対値（-128～+127）で、分岐先の番地を指定します。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
REL	BPL rr	10 ₁₆ ,rr	2	6

【記述例】

BPL LABEL ; N = 0のときLABELへ分岐

【機能】 無条件分岐

【演算長】 -

【動作】 PC PC + cnt + REL (cnt : BRA/BRAL命令のバイト数)

【説明】 指定された番地へ分岐します。分岐命令実行後のPCに対する8ビットの相対値 (BRA : -128 ~ +127)、又は16ビットの相対値 (BRAL : -32768 ~ +32767) で、分岐先の番地を指定します。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
REL	BRA rr	20 ₁₆ ,rr	2	5
	BRAL rrHrL	A7 ₁₆ ,rrL,rrH	3	5

【記述例】

BRA REL8 ; PC + 2 + REL8番地へ分岐
 BRAL REL16 ; PC + 3 + REL16番地へ分岐

【機能】 ソフトウェア割り込み

【演算長】 -

【動作】 BRK割り込みを発生

【説明】 BRK命令の次の命令が格納されている番地、PSの内容をPG、PC、PSの順にスタックに退避し、FFFA₁₆番地の内容を下位番地、FFFB₁₆番地の内容を上位番地とする番地へ分岐します。
この命令はデバッグツール用の予約命令です。エミュレータ使用時には使用できません。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
						1		

I : “1” になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
IMP	BRK	00 ₁₆ ,74 ₁₆	2	15

【記述例】

BRK ;

【機能】 条件分岐

【演算長】 16ビット又は8ビット

【動作】 A (bit n) 又はM (bit n) =0のとき指定番地へ相対分岐 (n=0~15。指定ビットで1ビットのみ指定)

【説明】 A又はメモリ中の指定されたビットの内容が“0”のとき、指定された番地へ分岐します。PCに対する8ビットの相対値 (-128~+127) で、分岐先の番地を指定します。テストするビット位置は、ビット番号で指定します。

「m=0」のとき：b0~b15中の任意の1ビットが指定できます。

「m=1」のとき：b0~b7中の任意の1ビットが指定できます。

この命令ではBは使用できません。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
A	BSC n,A,rr	01 ₁₆ ,n+A0 ₁₆ ,rr	3	7
DIR	BSC n,dd,rr	71 ₁₆ ,n+A0 ₁₆ ,dd,rr	4	11
ABS	BSC n,mml,rr	71 ₁₆ ,n+E0 ₁₆ ,ll,mm,rr	5	10

注．nには0~15の値が設定できます。

【記述例】

```
CLM
BSC      8,A,LABEL1          ; Aのb8が“0”のときLABEL1へ分岐
BSC      15,MEM16,LABEL2     ; MEM16のb15が“0”のときLABEL2へ分岐
SEM
BSC      7,A,LABEL3          ; Aのb7が“0”のときLABEL3へ分岐
BSC      7,MEM8,LABEL4       ; MEM8のb7が“0”のときLABEL4へ分岐
```

【機能】 サブルーチン呼び出し

【演算長】 -

【動作】 スタック PC + 2
PC PC + 2 + REL

【説明】 PCの内容をスタックに退避したのち、指定された番地へ分岐します。PCに対する11ビットの相対値（-1024～+1023）で、分岐先の番地を示します。

バンクをまたぐ分岐には使用できません。

この命令はバンクの境界に配置しないでください。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
REL	BSR rr	$(11111b_{10} b_9 b_8)_2, (b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0)_2$ b ₁₀ ~ b ₀ はrrの b ₁₀ ~ b ₀ を指す。	2	7

注．rrには-1023～1024（11ビット長）の値が設定できます。

【記述例】

BSR LABEL ; LABELへ分岐

【機能】 条件分岐

【演算長】 16ビット又は8ビット

【動作】 A (bit n) 又はM (bit n) =1のとき指定番地へ相対分岐 (n=0~15。指定ビットで1ビットのみ指定)

【説明】 A又はメモリ中の指定されたビットの内容が“1”のとき、指定された番地へ分岐します。PCに対する8ビットの相対値 (-128~+127) で、分岐先の番地を指定します。テストするビット位置はビット番号で指定します。

「m=0」のとき：b0~b15中の任意の1ビットが指定できます。

「m=1」のとき：b0~b7中の任意の1ビットが指定できます。

この命令ではBは使用できません。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
A	BSS n,A,rr	01 ₁₆ ,n+80 ₁₆ ,rr	3	7
DIR	BSS n,dd,rr	71 ₁₆ ,n+80 ₁₆ ,dd,rr	4	11
ABS	BSS n,mml,rr	71 ₁₆ ,n+C0 ₁₆ ,ll,mm,rr	5	10

注．nには0~15の値が設定できます。

【記述例】

```
CLM
BSS      8,A,LABEL1          ; Aのb8が“1”のときLABEL1へ分岐
BSS     15,MEM16,LABEL2     ; MEM16のb15が“1”のときLABEL2へ分岐
SEM
BSS      7,A,LABEL3         ; Aのb7が“1”のときLABEL3へ分岐
BSS      7,MEM8,LABEL4     ; MEM8のb7が“1”のときLABEL4へ分岐
```

【機能】 条件分岐

【演算長】 -

【動作】 V=0のとき指定番地へ相対分岐

【説明】 Vフラグが“0”のとき、指定された番地へ分岐します。PCに対する8ビットの相対値（-128～+127）で、分岐先の番地を指定します。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
REL	BVC rr	50 ₁₆ ,rr	2	6

【記述例】

BVC LABEL ; V = 0のときLABELへ分岐

【機能】 条件分岐

【演算長】 -

【動作】 V=1のとき指定番地へ相対分岐

【説明】 Vフラグが“1”のとき、指定された番地へ分岐します。PCに対する8ビットの相対値（-128～+127）で、分岐先の番地を指定します。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
REL	BVS rr	70 ₁₆ ,rr	2	6

【記述例】

BVS LABEL ; V = 1のときLABELへ分岐

【機能】 比較&条件分岐

【演算長】 16ビット又は8ビット

【動作】 Acc=IMM又はM=IMMのとき指定番地へ相対分岐

【説明】 Acc又はメモリの内容とイミディエイト値が等しいとき、指定された番地へ分岐します。PCに対する8ビットの相対値(-128~+127)で、分岐先の番地を指定します。

「m=0」のとき：16ビット長で演算します。

「m=1」のとき：8ビット長で演算します。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	N	V					Z	C

N : 実行結果のMSBが“1”のとき、“1”になります。それ以外では“0”になります。

V : 符号付き演算と見なした結果が+32767、又は-32768(「m=1」のときは+127又は-128)を越えると“1”になります。それ以外では“0”になります。

Z : 実行結果が“0”のとき、“1”になります。それ以外では“0”になります。

C : ボローが発生すると、“0”になります。それ以外では“1”になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
A	CBEQ A,#imm,rr	A6 ₁₆ ,imm,rr	3	6
A	CBEQ B,#imm,rr	81 ₁₆ ,A6 ₁₆ ,imm,rr	4	7
DIR	CBEQ dd,#imm,rr	41 ₁₆ ,6A ₁₆ ,dd,imm,rr	5	9

注．mフラグが“0”の場合は、バイト数は1バイト増加します。

【記述例】

CLM

CBEQ.W A,#IMM16,LABEL1 ; A=IMM16のときLABEL1へ分岐

CBEQ.W MEM16,#IMM16,LABEL2 ; MEM16=IMM16のときLABEL2へ分岐

SEM

CBEQ.B B,#IMM8,LABEL3 ; B_L=IMM8のときLABEL3へ分岐

【機能】 比較&条件分岐

【演算長】 8ビット

【動作】 $AccL=IMM8$ 又は $M8=IMM8$ のとき指定番地へ相対分岐

【説明】 $AccL$ 又はメモリの内容とイミディエイト値が8ビット長で比較して等しいとき、指定された番地へ分岐します。PCに対する8ビットの相対値(-128~+127)で、分岐先の番地を指定します。

この命令はmフラグの影響を受けません。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	N	V					Z	C

N : 実行結果のMSBが“1”のとき、“1”になります。それ以外では“0”になります。

V : 符号付き演算と見なした結果が+127、又は-128を越えると“1”になります。それ以外では“0”になります。

Z : 実行結果が“0”のとき、“1”になります。それ以外では“0”になります。

C : ポラリティーが発生すると、“0”になります。それ以外では“1”になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
A	CBEQB A,#imm,rr	$A2_{16},imm,rr$	3	6
A	CBEQB B,#imm,rr	$81_{16},A2_{16},imm,rr$	4	7
DIR	CBEQB dd,#imm,rr	$62_{16},dd,imm,rr$	4	8

【記述例】

```
CBEQB      A,#IMM8,LABEL1          ; AL=IMM8のときLABEL1へ分岐
CBEQB      MEM8,#IMM8,LABEL2       ; MEM8=IMM8のときLABEL2へ分岐
```

【機能】 比較&条件分岐

【演算長】 16ビット又は8ビット

【動作】 Acc IMM又はM IMMのとき指定番地へ相対分岐

【説明】 Acc又はメモリの内容とイミディエイト値を比較して等しくないとき、指定された番地へ分岐します。PCに対する8ビットの相対値(-128~+127)で、分岐先の番地を指定します。

「m=0」のとき：16ビット長で演算します。

「m=1」のとき：8ビット長で演算します。

このときAcc_Hの内容は変化しません。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	N	V					Z	C

N : 実行結果のMSBが“1”のとき、“1”になります。それ以外では“0”になります。

V : 符号付き演算と見なした結果が+32767、又は-32768(「m=1」のときは+127又は-128)を越えると“1”になります。それ以外では“0”になります。

Z : 実行結果が“0”のとき、“1”になります。それ以外では“0”になります。

C : ボローが発生すると、“0”になります。それ以外では“1”になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
A	CBNE A,#imm,rr	B6 ₁₆ ,imm,rr	3	6
A	CBNE B,#imm,rr	81 ₁₆ ,B6 ₁₆ ,imm,rr	4	7
DIR	CBNE dd,#imm,rr	41 ₁₆ ,7A ₁₆ ,dd,imm,rr	5	9

注. mフラグが“0”の場合は、バイト数は1バイト増加します。

【記述例】

CLM

CBNE.W A,#IMM16,LABEL1 ; A IMM16のときLABEL1へ分岐

CBNE.W MEM16,#IMM16,LABEL2 ; MEM16 IMM16のときLABEL2へ分岐

【機能】 比較&条件分岐

【演算長】 8ビット

【動作】 AccL IMM8又はM8 IMM8のとき指定番地へ相対分岐

【説明】 AccL又はメモリの内容とイミディエイト値が8ビット長で比較して等しくないとき、指定された番地へ分岐します。PCに対する8ビットの相対値(-128~+127)で、分岐先の番地を指定します。

この命令はmフラグの影響を受けません。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	N	V					Z	C

N : 実行結果のMSBが“1”のとき、“1”になります。それ以外では“0”になります。

V : 符号付き演算と見なした結果が+127、又は-128を越えると“1”になります。それ以外では“0”になります。

Z : 実行結果が“0”のとき、“1”になります。それ以外では“0”になります。

C : ポロアが発生すると、“0”になります。それ以外では“1”になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
A	CBNEB A,#imm,rr	B2 ₁₆ ,imm,rr	3	6
A	CBNEB B,#imm,rr	81 ₁₆ ,B2 ₁₆ ,imm,rr	4	7
DIR	CBNEB dd,#imm,rr	72 ₁₆ ,dd,imm,rr	4	8

【記述例】

```
CBNEB    A,#IMM8,LABEL1          ; AL IMM8のときLABEL1へ分岐
CBNEB    MEM8,#IMM8,LABEL2       ; MEM8 IMM8のときLABEL2へ分岐
```

【機能】 フラグ操作

【演算長】 -

【動作】 C 0

【説明】 Cフラグの内容を“0”にします。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
								0

C : “0”になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
IMP	CLC	14 ₁₆	1	1

【記述例】

CLC ; C 0

【機能】 フラグ操作

【演算長】 -

【動作】 I 0

【説明】 Iフラグの内容を“0”にします。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
						0		

I : “0”になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
IMP	CLI	15 ₁₆	1	3

【記述例】

CLI ; I 0

【機能】 フラグ操作

【演算長】 -

【動作】 m 0

【説明】 mフラグの内容を“0”にします。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
			0					

m : “0”になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
IMP	CLM	45 ₁₆	1	3

【記述例】

CLM ; m 0

【機能】 フラグ操作

【演算長】 -

【動作】 PSL (bit n) 0 (n=0~7。複数ビット指定可)

【説明】 PSL中の指定されたフラグ（複数指定可）を“0”にします。フラグの指定（PSLのビット位置）は8ビットのイミディエイト値で行います。“1”にセットしたビットが対象のフラグとなります。

この命令はmフラグの影響を受けません。

PSL							
b7	b6	b5	b4	b3	b2	b1	b0
N	V	m	x	D	I	Z	C

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	N	V	m	x	D	I	Z	C

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
IMM	CLP #imm	98 ₁₆ ,imm	2	4

【記述例】

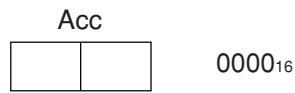
CLP #IMM8 ; PSL の指定ビット 0

【機能】 クリア

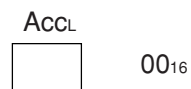
【演算長】 16ビット又は8ビット

【動作】 Acc 0

・ m=“0”のとき



・ m=“1”のとき



このときAccHの内容は変化しません。

【説明】 Accの内容を“0”にクリアします。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	0						1	

N : 実行結果のMSBが“0”のため、常に“0”になります。

Z : 実行結果が“0”のため、常に“1”になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
A	CLR A	54 ₁₆	1	1
A	CLR B	81 ₁₆ ,54 ₁₆	2	2

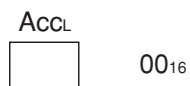
【記述例】

```
CLM
CLR      A           ; A  000016
CLR      B           ; B  000016
SEM
CLR      A           ; AL 0016
CLR      B           ; BL 0016
```

【機能】 クリア

【演算長】 8ビット

【動作】 AcCL 00₁₆



【説明】 AcCLの内容を“00₁₆”にクリアします。
AcCHの内容は変化しません。
この命令はmフラグの影響を受けません。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	0						1	

N : 実行結果のMSBが“0”のため、常に“0”になります。

Z : 実行結果が“0”のため、常に“1”になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
A	CLRB A	44 ₁₆	1	1
A	CLRB B	81 ₁₆ , 44 ₁₆	2	2

【記述例】

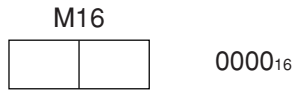
```
CLRB    A           ; AL  0016
CLRB    B           ; BL  0016
```

【機能】 クリア

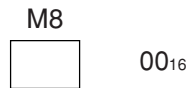
【演算長】 16ビット又は8ビット

【動作】 M 0

・ m=“0” のとき



・ m=“1” のとき



【説明】 メモリの内容を“0”にクリアします。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C

アドレスモード	記述形式	機械語	バイト数	サイクル数
DIR	CLRM dd	D2 ₁₆ ,dd	2	5
ABS	CLRM mml	D7 ₁₆ ,ll,mm	3	5

【記述例】

```
CLM
CLRM      MEM16          ; MEM16  000016
SEM
CLRM      MEM8           ; MEM8   0016
```

【機能】 クリア

【演算長】 8ビット

【動作】 M8 00₁₆
 M8
 00₁₆

【説明】 8ビット長でメモリの内容を“00₁₆”にクリアします。
 この命令はmフラグの影響を受けません。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
DIR	CLRMB dd	C2 ₁₆ ,dd	2	5
ABS	CLRMB mml	C7 ₁₆ ,ll,mm	3	5

【記述例】

CLRMB MEM8 ; MEM8 00₁₆

【機能】 クリア

【演算長】 16ビット又は8ビット

【動作】 X 0

・ x= " 0 " のとき



・ x= " 1 " のとき



このときX_Hの内容は変化しません。

【説明】 Xの内容を " 0 " にクリアします。

この命令はmフラグの影響を受けません。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	0						1	

N : 実行結果のMSBが " 0 " のため、常に " 0 " になります。

Z : 実行結果が " 0 " のため、常に " 1 " になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
IMP	CLR X	E4 ₁₆	1	1

【記述例】

```
CLP      x
CLR X    ; X  000016
SEP      x
CLR X    ; XL  0016
```


【機能】 クリア

【演算長】 16ビット又は8ビット

【動作】 Y 0

・ x= " 0 " のとき



・ x= " 1 " のとき



このときY_Hの内容は変化しません。

【説明】 Yの内容を " 0 " にクリアします。

この命令はmフラグの影響を受けません。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	0						1	

N : 実行結果のMSBが " 0 " のため、常に " 0 " になります。

Z : 実行結果が " 0 " のため、常に " 1 " になります。

アドレスモード	記述形式	機械語	バイト数	サイクル数
IMP	CLRY	F4 ₁₆	1	1

【記述例】

```
CLP      x
CLRY                      ; Y  000016
SEP      x
CLRY                      ; YL 0016
```

【機能】 フラグ操作

【演算長】 -

【動作】 $V \leftarrow 0$

【説明】 Vフラグの内容を“0”にします。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
		0						

V : “0”になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
IMP	CLV	65 ₁₆	1	1

【記述例】

CLV ; V ← 0

【機能】 比較

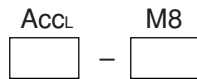
【演算長】 16ビット又は8ビット

【動作】 Acc – M

・ m= “ 0 ” のとき



・ m= “ 1 ” のとき



【説明】 Accの内容からメモリの内容を減算します。結果はどこにも格納しません。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	N	V					Z	C

- N : 実行結果のMSBが “ 1 ” のとき、 “ 1 ” になります。それ以外では “ 0 ” になります。
- V : 符号付き演算と見なした結果が+32767、又は-32768 (「 m=1 」 のときは+127又は-128) を越えると “ 1 ” になります。それ以外では “ 0 ” になります。
- Z : 実行結果が “ 0 ” のとき、 “ 1 ” になります。それ以外では “ 0 ” になります。
- C : ボローが発生すると、 “ 0 ” になります。それ以外では “ 1 ” になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
IMM	CMP A,#imm	46 ₁₆ ,imm (81 ₁₆ ,46 ₁₆ ,imm)	2 (3)	1 (2)
DIR	CMP A,dd	4A ₁₆ ,dd (81 ₁₆ ,4A ₁₆ ,dd)	2 (3)	3 (4)
DIR,X	CMP A,dd,X	4B ₁₆ ,dd (81 ₁₆ ,4B ₁₆ ,dd)	2 (3)	4 (5)
(DIR)	CMP A,(dd)	11 ₁₆ ,40 ₁₆ ,dd (91 ₁₆ ,40 ₁₆ ,dd)	3 (3)	6 (6)
(DIR,X)	CMP A,(dd,X)	11 ₁₆ ,41 ₁₆ ,dd (91 ₁₆ ,41 ₁₆ ,dd)	3 (3)	7 (7)
(DIR),Y	CMP A,(dd),Y	11 ₁₆ ,48 ₁₆ ,dd (91 ₁₆ ,48 ₁₆ ,dd)	3 (3)	7 (7)
L(DIR)	CMP A,L(dd)	11 ₁₆ ,42 ₁₆ ,dd (91 ₁₆ ,42 ₁₆ ,dd)	3 (3)	8 (8)
L(DIR),Y	CMP A,L(dd),Y	11 ₁₆ ,49 ₁₆ ,dd (91 ₁₆ ,49 ₁₆ ,dd)	3 (3)	9 (9)
SR	CMP A,nn,S	11 ₁₆ ,43 ₁₆ ,nn (91 ₁₆ ,43 ₁₆ ,nn)	3 (3)	5 (5)
(SR),Y	CMP A,(nn,S),Y	11 ₁₆ ,44 ₁₆ ,nn (91 ₁₆ ,44 ₁₆ ,nn)	3 (3)	8 (8)
ABS	CMP A,mml	4E ₁₆ ,ll,mm (81 ₁₆ ,4E ₁₆ ,ll,mm)	3 (4)	3 (4)
ABS,X	CMP A,mml,X	4F ₁₆ ,ll,mm (81 ₁₆ ,4F ₁₆ ,ll,mm)	3 (4)	4 (5)
ABS,Y	CMP A,mml,Y	11 ₁₆ ,46 ₁₆ ,ll,mm (91 ₁₆ ,46 ₁₆ ,ll,mm)	4 (4)	5 (5)
ABL	CMP A,hhmmll	11 ₁₆ ,4C ₁₆ ,ll,mm,hh (91 ₁₆ ,4C ₁₆ ,ll,mm,hh)	5 (5)	5 (5)
ABL,X	CMP A,hhmmll,X	11 ₁₆ ,4D ₁₆ ,ll,mm,hh (91 ₁₆ ,4D ₁₆ ,ll,mm,hh)	5 (5)	6 (6)

注1 . この表はアキュムレータAを使用する場合について書かれています。アキュムレータBを使用する場合は記述形式の “ A, ” の部分が “ B, ” になり、機械語、バイト数、サイクル数は () 内の値になります。
 2 . IMMアドレッシングモードで、mフラグが “ 0 ” の場合は、バイト数は1バイト増加します。

【記述例】

```

CLM
CMP.W      A,#IMM16          ; A - IMM16
CMP        B,MEM16          ; B - MEM16
SEM
CMP.B     A,#IMM8           ; AL - IMM8
CMP        B,MEM8           ; BL - MEM8
    
```

【機能】 比較

【演算長】 8ビット

【動作】 AccL - IMM8

AccL
 - IMM8

【説明】 AccLの内容からイミディエイト値を8ビット長で減算します。結果はどこにも格納しません。
 この命令はmフラグの影響を受けません。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	N	V					Z	C

- N : 実行結果のMSBが“1”のとき、“1”になります。それ以外では“0”になります。
 V : 符号付き演算と見なした結果が+127、又は-128を越えると“1”になります。それ以外では“0”になります。
 Z : 実行結果が“0”のとき、“1”になります。それ以外では“0”になります。
 C : ボローが発生すると、“0”になります。それ以外では“1”になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
IMM	CMPB A,#imm	38 ₁₆ ,imm	2	1
IMM	CMPB B,#imm	81 ₁₆ ,38 ₁₆ ,imm	3	2

【記述例】

CMPB A,#IMM8 ; AL - IMM8
 CMPB B,#IMM8 ; BL - IMM8

【機能】 比較

【演算長】 32ビット

【動作】 E - M32



「4.4 32ビット演算命令実行時の注意」を参照してください。

【説明】 Eの内容からメモリの内容を32ビット長で減算します。結果はどこにも格納しません。
この命令はmフラグの影響を受けません。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	N	V					Z	C

- N : 実行結果のMSBが“1”のとき、“1”になります。それ以外では“0”になります。
- V : 符号付き演算と見なした結果が+2147483647、又は-2147483648を越えると“1”になります。それ以外では“0”になります。
- Z : 実行結果が“0”のとき、“1”になります。それ以外では“0”になります。
- C : ボローが発生すると、“0”になります。それ以外では“1”になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
IMM	CMPD E,#imm	3C ₁₆ ,imm _{LL} ,imm _{LH} ,imm _{HL} ,imm _{HH}	5	3
DIR	CMPD E,dd	BA ₁₆ ,dd	2	6
DIR,X	CMPD E,dd,X	BB ₁₆ ,dd	2	7
(DIR)	CMPD E,(dd)	11 ₁₆ ,B0 ₁₆ ,dd	3	9
(DIR),X	CMPD E,(dd),X	11 ₁₆ ,B1 ₁₆ ,dd	3	10
(DIR),Y	CMPD E,(dd),Y	11 ₁₆ ,B8 ₁₆ ,dd	3	10
L(DIR)	CMPD E,L(dd)	11 ₁₆ ,B2 ₁₆ ,dd	3	11
L(DIR),Y	CMPD E,L(dd),Y	11 ₁₆ ,B9 ₁₆ ,dd	3	12
SR	CMPD E,nn,S	11 ₁₆ ,B3 ₁₆ ,nn	3	8
(SR),Y	CMPD E,(nn),S,Y	11 ₁₆ ,B4 ₁₆ ,nn	3	11
ABS	CMPD E,mml	BE ₁₆ ,ll,mm	3	6
ABS,X	CMPD E,mml,X	BF ₁₆ ,ll,mm	3	7
ABS,Y	CMPD E,mml,Y	11 ₁₆ ,B6 ₁₆ ,ll,mm	4	8
ABL	CMPD E,hhmml	11 ₁₆ ,BC ₁₆ ,ll,mm,hh	5	8
ABL,X	CMPD E,hhmml,X	11 ₁₆ ,BD ₁₆ ,ll,mm,hh	5	9

【記述例】

CMPD E,#IMM32 ; E - IMM32

【機能】 比較

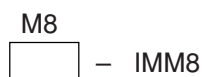
【演算長】 16ビット又は8ビット

【動作】 M - IMM

・ m = "0" のとき



・ m = "1" のとき



【説明】 メモリの内容からイミディエイト値を減算します。結果はどこにも格納しません。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	N	V					Z	C

- N : 実行結果のMSBが "1" のとき、"1" になります。それ以外では "0" になります。
- V : 符号付き演算と見なした結果が +32767、又は -32768 (「m=1」のときは +127 又は -128) を越えると "1" になります。それ以外では "0" になります。
- Z : 実行結果が "0" のとき、"1" になります。それ以外では "0" になります。
- C : ポロラが発生すると、"0" になります。それ以外では "1" になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
DIR	CMPM dd,#imm	51 ₁₆ ,23 ₁₆ ,dd,imm	4	5
ABS	CMPM mmlI,#imm	51 ₁₆ ,27 ₁₆ ,ll,mm,imm	5	5

注 . mフラグが "0" の場合は、バイト数は1バイト増加します。

【記述例】

```
CLM
CMPM.W    MEM16,#IMM16          ; MEM16 - IMM16
SEM
CMPM.B    MEM8,#IMM8           ; MEM8 - IMM8
```

【機能】 比較

【演算長】 8ビット

【動作】 M8 – IMM8

M8

– IMM8

【説明】 メモリの内容からイミディエイト値を8ビット長で減算します。結果はどこにも格納しません。
この命令はmフラグの影響を受けません。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	N	V					Z	C

- N : 実行結果のMSBが“1”のとき、“1”になります。それ以外では“0”になります。
- V : 符号付き演算と見なした結果が+127、又は-128を越えると“1”になります。それ以外では“0”になります。
- Z : 実行結果が“0”のとき、“1”になります。それ以外では“0”になります。
- C : ボローが発生すると、“0”になります。それ以外では“1”になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
DIR	CMPMB dd,#imm	5 ₁₆ ,22 ₁₆ ,dd,imm	4	5
ABS	CMPMB mmlI,#imm	5 ₁₆ ,26 ₁₆ ,ll,mm,imm	5	5

【記述例】

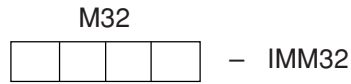
CMPMB MEM8,#IMM8 ; MEM8 - IMM8

【機能】 比較

【演算長】 32ビット

「4.4 32ビット演算命令実行時の注意」を参照してください。

【動作】 M32 – IMM32



【説明】 メモリの内容からイミディエイト値を32ビット長で減算します。結果はどこにも格納しません。この命令はmフラグの影響を受けません。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	N	V					Z	C

- N : 実行結果のMSBが“1”のとき、“1”になります。それ以外では“0”になります。
- V : 符号付き演算と見なした結果が+2147483647、又は-2147483648を越えると“1”になります。それ以外では“0”になります。
- Z : 実行結果が“0”のとき、“1”になります。それ以外では“0”になります。
- C : ボローが発生すると、“0”になります。それ以外では“1”になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
DIR	CMPMD dd,#imm	51 ₁₆ ,A3 ₁₆ ,dd,immLL,immLH,immHL,immHH	7	7
ABS	CMPMD mmlI,#imm	51 ₁₆ ,A7 ₁₆ ,ll,mm,immLL,immLH,immHL,immHH	8	7

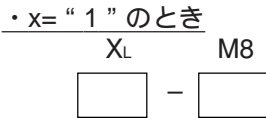
【記述例】

CMPMD MEM32,#IMM32 ; MEM32 - IMM32

【機能】 比較

【演算長】 16ビット又は8ビット

【動作】 X - M
 ・x=“0”のとき



【説明】 Xの内容からメモリの内容を減算します。結果はどこにも格納しません。
 この命令はmフラグの影響を受けません。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	N	V					Z	C

- N : 実行結果のMSBが“1”のとき、“1”になります。それ以外では“0”になります。
- V : 符号付き演算と見なした結果が+32767、又は-32768（「x=1」のときは+127又は-128）を越えると“1”になります。それ以外では“0”になります。
- Z : 実行結果が“0”のとき、“1”になります。それ以外では“0”になります。
- C : ボローが発生すると、“0”になります。それ以外では“1”になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
IMM	CPX #imm	E6 ₁₆ ,imm	2	1
DIR	CPX dd	22 ₁₆ ,dd	2	3
ABS	CPX mml	41 ₁₆ ,2E ₁₆ ,ll,mm	4	4

注：IMMアドレッシングモードで、xフラグが“0”の場合は、バイト数は1バイト増加します。

【記述例】

```
CLP          x
CPX.W       #IMM16          ; X - IMM16
CPX         MEM16          ; X - MEM16
SEP         x
CPX.B      #IMM8           ; XL - IMM8
CPX         MEM8           ; XL - MEM8
```

【機能】 比較

【演算長】 16ビット又は8ビット

【動作】 Y - M

・x=“0”のとき



・x=“1”のとき



【説明】 Yの内容からメモリの内容を減算します。結果はどこにも格納しません。
この命令はmフラグの影響を受けません。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	N	V					Z	C

- N : 実行結果のMSBが“1”のとき、“1”になります。それ以外では“0”になります。
- V : 符号付き演算と見なした結果が+32767、又は-32768（「x=1」のときは+127又は-128）を越えると“1”になります。それ以外では“0”になります。
- Z : 実行結果が“0”のとき、“1”になります。それ以外では“0”になります。
- C : ポローが発生すると、“0”になります。それ以外では“1”になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
IMM	CPY #imm	F6 ₁₆ ,imm	2	1
DIR	CPY dd	32 ₁₆ ,dd	2	3
ABS	CPY mml	41 ₁₆ ,3E ₁₆ ,ll,mm	4	4

注：IMMアドレッシングモードで、xフラグが“0”の場合は、バイト数は1バイト増加します。

【記述例】

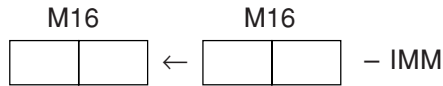
```
CLP          x
CPY.W       #IMM16          ; Y - IMM16
CPY        MEM16          ; Y - MEM16
SEP          x
CPY.B      #IMM8          ; YL - IMM8
CPY        MEM8          ; YL - MEM8
```

【機能】 デクリメント&条件分岐

【演算長】 16ビット又は8ビット

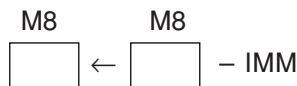
【動作】 $M \leftarrow M - IMM$ (IMM=0~31)

・ m=“0”のとき



M16 (演算結果) = 0のとき、次の命令を実行
M16 (演算結果) 0のとき、指定アドレスへ分岐

・ m=“1”のとき



M8 (演算結果) = 0のとき、次の命令を実行
M8 (演算結果) 0のとき、指定アドレスへ分岐

【説明】 メモリの内容からイミディエイト値 (0~31) を減算し、その結果をメモリへ格納します。このとき、結果が“0”でなければ指定された番地へ分岐します。PCに対する8ビットの相対値 (-128~+127) で、分岐先の番地を示します。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
DIR	DEBNE dd,#imm,rr	C1 ₁₆ ,imm+A0 ₁₆ ,dd,rr	4	12
ABS	DEBNE mmlI,#imm,rr	D1 ₁₆ ,imm+E0 ₁₆ ,ll,mm,rr	5	11

注 . immには0~31の値が設定できます。

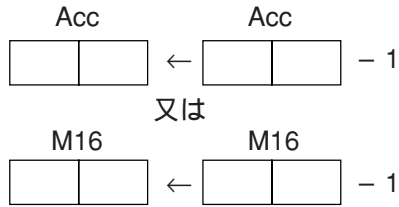
【記述例】

CLM
DEBNE MEM16,#IMM,LABEL1 ; MEM16 - IMM(0~31)が0以外るときLABEL1へ分岐
SEM
DEBNE MEM8,#IMM,LABEL2 ; MEM8 - IMM(0~31)が0以外るときLABEL2へ分岐

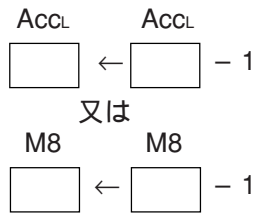
【機能】 デクリメント

【演算長】 16ビット又は8ビット

【動作】 $Acc \leftarrow Acc - 1$ 又は $M \leftarrow M - 1$
 ・ $m = "0"$ のとき



・ $m = "1"$ のとき



このとき $AccH$ の内容は変化しません。

【説明】 Acc の内容又はメモリの内容から1減算します。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	N						Z	

N : 実行結果のMSBが“1”のとき、“1”になります。それ以外では“0”になります。

Z : 実行結果が“0”のとき、“1”になります。それ以外では“0”になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
A	DEC A	B3 ₁₆	1	1
A	DEC B	81 ₁₆ , B3 ₁₆	2	2
DIR	DEC dd	92 ₁₆ , dd	2	6
DIR,X	DEC dd,X	41 ₁₆ , 9B ₁₆ , dd	3	8
ABS	DEC mml	97 ₁₆ , ll, mm	3	6
ABS,X	DEC mml,X	41 ₁₆ , 9F ₁₆ , ll, mm	4	8

【記述例】

```

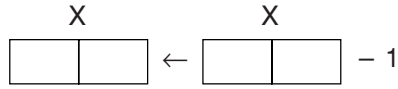
CLM
DEC      A          ; A  A - 1
SEM
DEC      A          ; AL AL - 1
    
```

【機能】 デクリメント

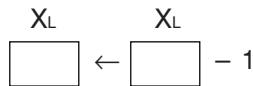
【演算長】 16ビット又は8ビット

【動作】 $X \leftarrow X - 1$

・x=“0”のとき



・x=“1”のとき



このときX_Hの内容は変化しません。

【説明】 Xの内容から1減算します。

この命令はmフラグの影響を受けません。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	N						Z	

- N : 実行結果のMSBが“1”のとき、“1”になります。それ以外では“0”になります。
- Z : 実行結果が“0”のとき、“1”になります。それ以外では“0”になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
IMP	DEX	E3 ₁₆	1	1

【記述例】

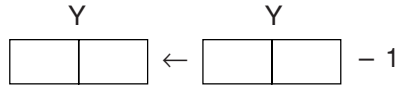
```
CLP      x
DEX                      ; X    X - 1
SEP      x
DEX                      ; XL  XL - 1
```

【機能】 デクリメント

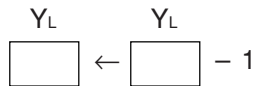
【演算長】 16ビット又は8ビット

【動作】 $Y \leftarrow Y - 1$

・x=“0”のとき



・x=“1”のとき



このときY_Hの内容は変化しません。

【説明】 Yの内容から1減算します。
この命令はmフラグの影響を受けません。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	N						Z	

N : 実行結果のMSBが“1”のとき、“1”になります。それ以外では“0”になります。

Z : 実行結果が“0”のとき、“1”になります。それ以外では“0”になります。

ア・レシ・ン・グ・モ・ド・	記述形式	機 械 語	バ イ ト 数	サイ クル 数
IMP	DEY	F3 ₁₆	1	1

【記述例】:

```

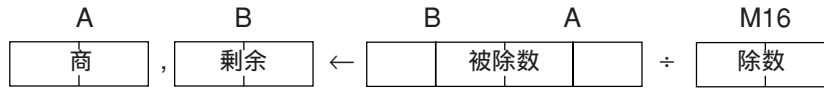
CLP      x
DEY                                ; Y   Y - 1
SEP      x
DEY                                ; YL  YL - 1
    
```

【機能】 除算（符号なし）

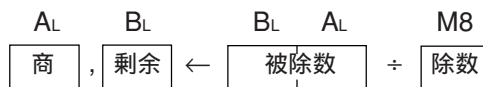
【演算長】 16ビット又は8ビット

【動作】 A （商）, B （剰余） $\leftarrow (B, A) \div M$

・ $m = "0"$ のとき



・ $m = "1"$ のとき



このとき A_H, B_H の内容は変化しません。

【説明】 B の内容を上位、 A の内容を下位とするデータを、メモリの内容で除算します。

商は A に、剰余は B に格納されます。

演算の結果オーバーフローが発生した場合は、 V フラグが "1" になり、 A 、 B の内容は不定になります。

除数が "0" の場合は、0 除算割り込みが発生します。このとき A 、 B の内容は変化しません。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	N	V				I	Z	C

N : 実行結果の A （商）の MSB が "1" のとき、"1" になります。オーバーフロー発生時、及び除数が "0" の場合は変化しません。それ以外では "0" になります。

V : オーバーフロー発生時は、"1" になります。除数が "0" の場合は、変化しません。それ以外では "0" になります。

I : 除数が "0" の場合は、"1" になります。それ以外では変化しません。

Z : 実行結果の A （商）が "0" のとき、"1" になります。オーバーフロー発生時、及び除数が "0" の場合は変化しません。それ以外では "0" になります。

C : オーバーフロー発生時は、"1" になります。除数が "0" の場合は、変化しません。それ以外では "0" になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
IMM	DIV #imm	31 ₁₆ ,E7 ₁₆ ,imm	3	15
DIR	DIV dd	21 ₁₆ ,EA ₁₆ ,dd	3	16
DIR,X	DIV dd,X	21 ₁₆ ,EB ₁₆ ,dd	3	17
(DIR)	DIV (dd)	21 ₁₆ ,E0 ₁₆ ,dd	3	18
(DIR,X)	DIV (dd,X)	21 ₁₆ ,E1 ₁₆ ,dd	3	19
(DIR),Y	DIV (dd),Y	21 ₁₆ ,E8 ₁₆ ,dd	3	19
L(DIR)	DIV L(dd)	21 ₁₆ ,E2 ₁₆ ,dd	3	20
L(DIR),Y	DIV L(dd),Y	21 ₁₆ ,E9 ₁₆ ,dd	3	21
SR	DIV nn,S	21 ₁₆ ,E3 ₁₆ ,nn	3	17
(SR),Y	DIV (nn,S),Y	21 ₁₆ ,E4 ₁₆ ,nn	3	20
ABS	DIV mmll	21 ₁₆ ,EE ₁₆ ,ll,mm	4	16
ABS,X	DIV mmll,X	21 ₁₆ ,EF ₁₆ ,ll,mm	4	17
ABS,Y	DIV mmll,Y	21 ₁₆ ,E6 ₁₆ ,ll,mm	4	17
ABL	DIV hhmmll	21 ₁₆ ,EC ₁₆ ,ll,mm,hh	5	17
ABL,X	DIV hhmmll,X	21 ₁₆ ,ED ₁₆ ,ll,mm,hh	5	18

注1. IMMアドレッシングモードで、mフラグが“0”の場合は、バイト数は1バイト増加します。

2. この表のサイクル数は、16ビット÷8ビット演算の場合を示しています。

32ビット÷16ビット演算の場合は、サイクル数を8サイクル増加してください。

3. この表及び注2のサイクル数は、演算が正常に行われた（割り込みが発生しなかった）場合を示しています。0除算割り込みが発生した場合は、演算のデータ長に関わらず、16サイクルになります。

【記述例】

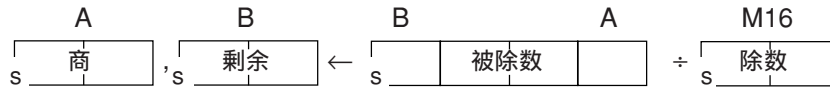
```
CLM
DIV          MEM16          ; A,B    (B,A) / MEM16
DIV.W       #IMM16        ; A,B    (B,A) / IMM16
SEM
DIV          MEM8          ; AL,BL  (BL,AL) / MEM8
DIV.B       #IMM8         ; AL,BL  (BL,AL) / IMM8
```


【機能】 除算（符号付き）

【演算長】 16ビット又は8ビット

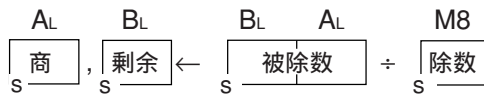
【動作】 $A(商), B(剰余) \leftarrow (B, A) \div M$

・ $m = "0"$ のとき



Sは符号ビットでデータのMSBを示します。

・ $m = "1"$ のとき



Sは符号ビットでデータのMSBを示します。

このとき A_H, B_H の内容は変化しません。

【説明】 Bの内容を上位、Aの内容を下位とする符号付きデータを、メモリの内容（符号付き）で除算します。商はAに、剰余はBにそれぞれ符号付きで格納されます。

剰余の符号は被除数と同じになります。

演算の結果オーバーフローが発生した（商が「 $m=0$ 」のとき $-32767 \sim +32767$ 、「 $m=1$ 」のとき $-127 \sim +127$ の範囲を越えた）場合は演算が途中で終了し、Vフラグが“1”になります。このときA、Bの内容は不定になります。

除数が“0”の場合は0除算割り込みが発生します。このときA、Bの内容は変化しません。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	N	V				I	Z	C

- N : 実行結果のA（商）のMSBが“1”のとき、“1”になります。オーバーフロー発生時、及び除数が“0”の場合は変化しません。それ以外では“0”になります。
- V : オーバーフロー発生時は、“1”になります。除数が“0”の場合は、変化しません。それ以外では“0”になります。
- I : 除数が“0”の場合は、“1”になります。それ以外では変化しません。
- Z : 実行結果のA（商）が“0”のとき、“1”になります。オーバーフロー発生時、及び除数が“0”の場合は変化しません。それ以外では“0”になります。
- C : オーバーフロー発生時は、“1”になります。除数が“0”の場合は、変化しません。それ以外では“0”になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
IMM	DIVS #imm	31 ₁₆ ,F7 ₁₆ ,imm	3	22
DIR	DIVS dd	21 ₁₆ ,FA ₁₆ ,dd	3	23
DIR,X	DIVS dd,X	21 ₁₆ ,FB ₁₆ ,dd	3	24
(DIR)	DIVS (dd)	21 ₁₆ ,F0 ₁₆ ,dd	3	25
(DIR,X)	DIVS (dd,X)	21 ₁₆ ,F1 ₁₆ ,dd	3	26
(DIR),Y	DIVS (dd),Y	21 ₁₆ ,F8 ₁₆ ,dd	3	26
L(DIR)	DIVS L(dd)	21 ₁₆ ,F2 ₁₆ ,dd	3	27
L(DIR),Y	DIVS L(dd),Y	21 ₁₆ ,F9 ₁₆ ,dd	3	28
SR	DIVS nn,S	21 ₁₆ ,F3 ₁₆ ,nn	3	24
(SR),Y	DIVS (nn,S),Y	21 ₁₆ ,F4 ₁₆ ,nn	3	27
ABS	DIVS mml	21 ₁₆ ,FE ₁₆ ,ll,mm	4	23
ABS,X	DIVS mml,X	21 ₁₆ ,FF ₁₆ ,ll,mm	4	24
ABS,Y	DIVS mml,Y	21 ₁₆ ,F6 ₁₆ ,ll,mm	4	24
ABL	DIVS hhmml	21 ₁₆ ,FC ₁₆ ,ll,mm,hh	5	24
ABL,X	DIVS hhmml,X	21 ₁₆ ,FD ₁₆ ,ll,mm,hh	5	25

注1. IMMアドレッシングモードで、mフラグが“0”の場合は、バイト数は1バイト増加します。

2. この表のサイクル数は、16ビット÷8ビット演算の場合を示しています。

また、32ビット÷16ビット演算の場合は、さらに8サイクル増加します。

3. この表及び注2のサイクル数は、演算が正常に行われた（割り込みが発生しなかった）場合を示しています。
0除算割り込みが発生した場合は、演算のデータ長に関わらず、16サイクルになります。

【記述例】

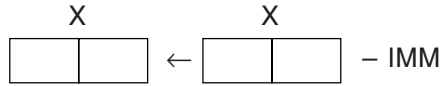
CLM
 DIVS MEM16 ; A,B (B,A) / MEM16
 SEM
 DIVS.B #IMM8 ; AL,BL (BL,AL) / IMM8

【機能】 デクリメント&条件分岐

【演算長】 16ビット又は8ビット

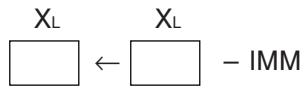
【動作】 $X \leftarrow X - IMM$ (IMM=0~31)

・x=“0”のとき



X (演算結果) = 0のとき、次の命令を実行
 X (演算結果) 0のとき、指定アドレスへ分岐

・x=“1”のとき



XL (演算結果) = 0のとき、次の命令を実行
 XL (演算結果) 0のとき、指定アドレスへ分岐
 このときXHの内容は変化しません。

【説明】 Xの内容からイミディエイト値(0~31)を減算し、その結果をXへ格納します。
 このとき、結果が“0”でなければ指定された番地へ分岐します。PCに対する8ビットの相対値(-128~+127)で、分岐先の番地を示します。
 この命令はmフラグの影響を受けません。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
IMM	DXBNE #imm,rr	01 ₁₆ ,imm+C0 ₁₆ ,rr	3	7

注 . immには0~31の値が設定できます。

【記述例】

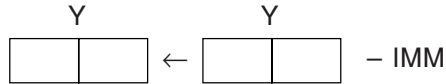
```
CLP          x
DXBNE        #IMM,LABEL1          ; X - IMM(0~31)が0以外るときLABEL1へ分岐
SEP          x
DXBNE        #IMM,LABEL2          ; XL - IMM(0~31)が0以外るときLABEL2へ分岐
```

【機能】 デクリメント&条件分岐

【演算長】 16ビット又は8ビット

【動作】 $Y \leftarrow Y - IMM$ (IMM=0~31)

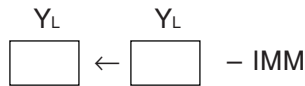
・x=“0”のとき



Y (演算結果) = 0のとき、次の命令を実行

Y (演算結果) 0のとき、指定アドレスへ分岐

・x=“1”のとき



YL (演算結果) = 0のとき、次の命令を実行

YL (演算結果) 0のとき、指定アドレスへ分岐

このときYHの内容は変化しません。

【説明】 Yの内容からイミディエイト値(0~31)を減算し、その結果をYに格納します。

このとき、結果が“0”でなければ指定された番地へ分岐します。PCに対する8ビットの相対値(-128~+127)で、分岐先の番地を示します。

この命令はmフラグの影響を受けません。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
IMM	DYBNE #imm,rr	01 ₁₆ ,imm+E0 ₁₆ ,rr	3	7

注．immには0~31の値が設定できます。

【記述例】

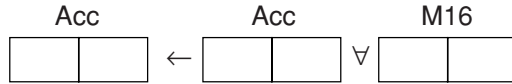
```

CLP          x
DYBNE       #IMM,LABEL1          ; Y - IMM(0~31)が0以外るときLABEL1へ分岐
SEP        x
DYBNE       #IMM,LABEL2          ; YL - IMM(0~31)が0以外るときLABEL2へ分岐
    
```

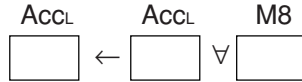
【機能】 排他的論理和

【演算長】 16ビット又は8ビット

【動作】 $Acc \leftarrow Acc \vee M$
 ・ m=“0”のとき



・ m=“1”のとき



このときAccHの内容は変化しません。

【説明】 Accの内容とメモリの内容の排他的論理和をとり、結果をAccへ格納します。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	N						Z	

N : 実行結果のMSBが“1”のとき、“1”になります。それ以外では“0”になります。

Z : 実行結果が“0”のとき、“1”になります。それ以外では“0”になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
IMM	EOR A,#imm	76 ₁₆ ,imm (81 ₁₆ ,76 ₁₆ ,imm)	2 (3)	1 (2)
DIR	EOR A,dd	7A ₁₆ ,dd (81 ₁₆ ,7A ₁₆ ,dd)	2 (3)	3 (4)
DIR,X	EOR A,dd,X	7B ₁₆ ,dd (81 ₁₆ ,7B ₁₆ ,dd)	2 (3)	4 (5)
(DIR)	EOR A,(dd)	11 ₁₆ ,70 ₁₆ ,dd (91 ₁₆ ,70 ₁₆ ,dd)	3 (3)	6 (6)
(DIR,X)	EOR A,(dd,X)	11 ₁₆ ,71 ₁₆ ,dd (91 ₁₆ ,71 ₁₆ ,dd)	3 (3)	7 (7)
(DIR),Y	EOR A,(dd),Y	11 ₁₆ ,78 ₁₆ ,dd (91 ₁₆ ,78 ₁₆ ,dd)	3 (3)	7 (7)
L(DIR)	EOR A,L(dd)	11 ₁₆ ,72 ₁₆ ,dd (91 ₁₆ ,72 ₁₆ ,dd)	3 (3)	8 (8)
L(DIR),Y	EOR A,L(dd),Y	11 ₁₆ ,79 ₁₆ ,dd (91 ₁₆ ,79 ₁₆ ,dd)	3 (3)	9 (9)
SR	EOR A,nn,S	11 ₁₆ ,73 ₁₆ ,nn (91 ₁₆ ,73 ₁₆ ,nn)	3 (3)	5 (5)
(SR),Y	EOR A,(nn,S),Y	11 ₁₆ ,74 ₁₆ ,nn (91 ₁₆ ,74 ₁₆ ,nn)	3 (3)	8 (8)
ABS	EOR A,mml	7E ₁₆ ,ll,mm (81 ₁₆ ,7E ₁₆ ,ll,mm)	3 (4)	3 (4)
ABS,X	EOR A,mml,X	7F ₁₆ ,ll,mm (81 ₁₆ ,7F ₁₆ ,ll,mm)	3 (4)	4 (5)
ABS,Y	EOR A,mml,Y	11 ₁₆ ,76 ₁₆ ,ll,mm (91 ₁₆ ,76 ₁₆ ,ll,mm)	4 (4)	5 (5)
ABL	EOR A,hhmmll	11 ₁₆ ,7C ₁₆ ,ll,mm,hh (91 ₁₆ ,7C ₁₆ ,ll,mm,hh)	5 (5)	5 (5)
ABL,X	EOR A,hhmmll,X	11 ₁₆ ,7D ₁₆ ,ll,mm,hh (91 ₁₆ ,7D ₁₆ ,ll,mm,hh)	5 (5)	6 (6)

注1. この表はアキュムレータAを使用する場合について、書かれています。アキュムレータBを使用する場合は記述形式の“A,”の部分か“B,”になり、機械語、バイト数、サイクル数は()内の値になります。

2. IMMアドレッシングモードで、mフラグが“0”の場合は、バイト数は1バイト増加します。

【記述例】

```
CLM
EOR.W      A,#IMM16      ; A  A  IMM16
EOR        B,MEM16      ; B  B  MEM16
SEM
EOR.B      A,#IMM8       ; AL AL  IMM8
EOR        B,MEM8       ; BL BL  MEM8
```

【機能】 排他的論理和

【演算長】 8ビット

【動作】 $AccL \leftarrow AccL \vee IMM8$

$\begin{array}{c} AccL \\ \square \end{array} \leftarrow \begin{array}{c} AccL \\ \square \end{array} \vee IMM8$

【説明】 AccLの内容とイミディエイト値の排他的論理和を8ビット長でとり、結果をAccLへ格納します。
この命令はmフラグの影響を受けません。
AccHの内容は変化しません。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	N						Z	

N : 実行結果のMSBが“1”のとき、“1”になります。それ以外では“0”になります。

Z : 実行結果が“0”のとき、“1”になります。それ以外では“0”になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
IMM	EORB A,#imm	33 ₁₆ ,imm	2	1
IMM	EORB B,#imm	81 ₁₆ ,33 ₁₆ ,imm	3	2

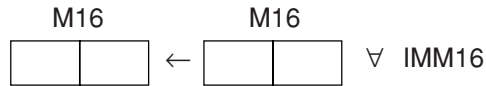
【記述例】

EORB A,#IMM8 ; AL AL IMM8
EORB B,#IMM8 ; BL BL IMM8

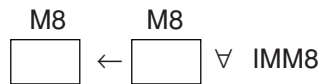
【機能】 排他的論理和

【演算長】 16ビット又は8ビット

【動作】 $M \leftarrow M \vee IMM$
 ・ m = “0” のとき



・ m = “1” のとき



【説明】 メモリの内容とイミディエイト値の排他的論理和をとり、結果をメモリへ格納します。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	N						Z	

N : 実行結果のMSBが“1”のとき、“1”になります。それ以外では“0”になります。

Z : 実行結果が“0”のとき、“1”になります。それ以外では“0”になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
DIR	EORM dd,#imm	5 ₁₆ ,73 ₁₆ ,dd,imm	4	7
ABS	EORM mmlI,#imm	5 ₁₆ ,77 ₁₆ ,ll,mm,imm	5	7

注 . mフラグが“0”の場合は、バイト数は1バイト増加します。

【記述例】

CLM
 EORM.W MEM16,#IMM16 ; MEM16 MEM16 IMM16
 SEM
 EORM.B MEM8,#IMM8 ; MEM8 MEM8 IMM8

【機能】 排他的論理和

【演算長】 8ビット

【動作】 $M8 \leftarrow M8 \vee IMM8$

$$\begin{array}{ccc} M8 & & M8 \\ \square & \leftarrow & \square \vee IMM8 \end{array}$$

【説明】 メモリの内容とイミディエイト値の排他的論理和を8ビット長でとり、結果をメモリへ格納します。
この命令はmフラグの影響を受けません。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	N						Z	

N : 実行結果のMSBが“1”のとき、“1”になります。それ以外では“0”になります。

Z : 実行結果が“0”のとき、“1”になります。それ以外では“0”になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
DIR	EORMB dd,#imm	51 ₁₆ ,72 ₁₆ ,dd,imm	4	7
ABS	EORMB mml,#imm	51 ₁₆ ,76 ₁₆ ,ll,mm,imm	5	7

【記述例】

EORMB MEM8,#IMM8 ; MEM8 MEM8 IMM8

【機能】 排他的論理和

【演算長】 32ビット

「4.4 32ビット演算命令実行時の注意」を参照してください。

【動作】 $M32 \leftarrow M32 \vee IMM32$



【説明】 メモリの内容とイミディエイト値の排他的論理和を32ビット長でとり、結果をメモリへ格納します。この命令はmフラグの影響を受けません。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	N						Z	

N : 実行結果のMSBが“1”のとき、“1”になります。それ以外では“0”になります。

Z : 実行結果が“0”のとき、“1”になります。それ以外では“0”になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
DIR	EORMD dd,#imm	51 ₁₆ ,F3 ₁₆ ,dd,immLL,immLH,immHL,immHH	7	10
ABS	EORMD mll,#imm	51 ₁₆ ,F7 ₁₆ ,ll,mm,immLL,immLH,immHL,immHH	8	10

【記述例】

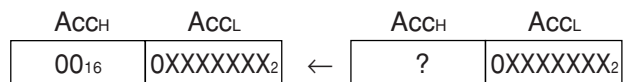
EORMD MEM32,#IMM32 ; MEM32 MEM32 IMM32

【機能】 符号拡張

【演算長】 16ビット

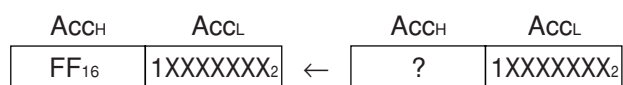
【動作】 $Acc \leftarrow AccL$ (符号拡張)
・ $AccL$ のビット7=“0”のとき

$AcCH \leftarrow 00_{16}$



・ $AccL$ のビット7=“1”のとき

$AcCH \leftarrow FF_{16}$



$AcCH$ の内容はmフラグに関係なく変化します。

【説明】 $AccL$ の内容を Acc へ符号拡張します。
この命令はmフラグの影響を受けません。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	N						Z	

N : 実行結果のビット15が“1”のとき、“1”になります。それ以外では“0”になります。

Z : 実行結果が“0”のとき、“1”になります。それ以外では“0”になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
A	EXTS A	35 ₁₆	1	1
A	EXTS B	81 ₁₆ , 35 ₁₆	2	2

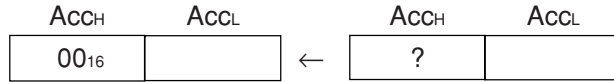
【記述例】

EXTS A ; AH 00₁₆ or FF₁₆
EXTS B ; BH 00₁₆ or FF₁₆

【機能】 ゼロ拡張

【演算長】 16ビット

【動作】 $Acc \leftarrow AccL$ (ゼロ拡張)



ACCHの内容はmフラグに関係なく変化します。

【説明】 AccLの内容をAccへゼロ拡張します。
この命令はmフラグの影響を受けません。
ACCHの内容は常に“00₁₆”になります。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	0						Z	

N : 実行結果のMSBが“0”のため、常に“0”になります。

Z : 実行結果が“0”のとき、“1”になります。それ以外では“0”になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
A	EXTZ A	34 ₁₆	1	1
A	EXTZ B	81 ₁₆ , 34 ₁₆	2	2

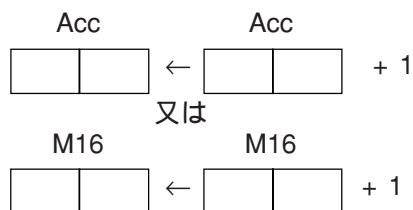
【記述例】

```
EXTZ      A          ; A  AL (AH  0016 , AL  AL)
EXTZ      B          ; B  BL (BH  0016 , BL  BL)
```

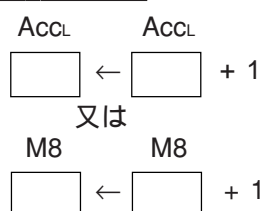

【機能】 インクリメント

【演算長】 16ビット又は8ビット

【動作】 $Acc \leftarrow Acc + 1$ 又は $M \leftarrow M + 1$
 ・ $m = "0"$ のとき



・ $m = "1"$ のとき



このとき $AccH$ の内容は変化しません。

【説明】 Acc の内容又はメモリの内容に1加算します。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	N						Z	

N : 実行結果のMSBが“1”のとき、“1”になります。それ以外では“0”になります。

Z : 実行結果が“0”のとき、“1”になります。それ以外では“0”になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
A	INC A	A3 ₁₆	1	1
A	INC B	81 ₁₆ ,A3 ₁₆	2	2
DIR	INC dd	82 ₁₆ ,dd	2	6
DIR,X	INC dd,X	41 ₁₆ ,8B ₁₆ ,dd	3	8
ABS	INC mml	87 ₁₆ ,ll,mm	3	6
ABS,X	INC mml,X	41 ₁₆ ,8F ₁₆ ,ll,mm	4	8

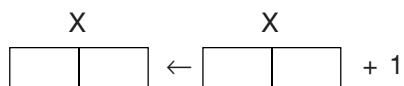
【記述例】

```
CLM
INC      A                ; A  A + 1
INC      MEM16            ; MEM16  MEM16 + 1
SEM
INC      B                ; BL  BL + 1
INC      MEM8             ; MEM8  MEM8 + 1
```

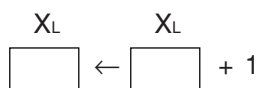
【機能】 インクリメント

【演算長】 16ビット又は8ビット

【動作】 $X \leftarrow X + 1$
 ・x=“0”のとき



・x=“1”のとき



このときX_Hの内容は変化しません。

【説明】 Xの内容に1加算します。
 この命令はmフラグの影響を受けません。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	N						Z	

N : 実行結果のMSBが“1”のとき、“1”になります。それ以外では“0”になります。

Z : 実行結果が“0”のとき、“1”になります。それ以外では“0”になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
IMP	INX	C3 ₁₆	1	1

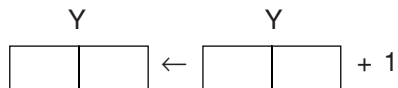
【記述例】

```
CLP      x
INX                      ; X  X + 1
SEP      x
INX                      ; XL XL + 1
```

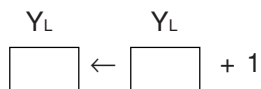
【機能】 インクリメント

【演算長】 16ビット又は8ビット

【動作】 $Y \leftarrow Y + 1$
 ・x=“0”のとき



・x=“1”のとき



このときY_Hの内容は変化しません。

【説明】 Yの内容に1加算します。
 この命令はmフラグの影響を受けません。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	N						Z	

N : 実行結果のMSBが“1”のとき、“1”になります。それ以外では“0”になります。

Z : 実行結果が“0”のとき、“1”になります。それ以外では“0”になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
IMP	INY	D3 ₁₆	1	1

【記述例】

```
CLP      x
INY      ; Y   Y + 1
SEP      x
INY      ; YL  YL + 1
```


【機能】 無条件分岐

【演算長】 -

【動作】 ・JMP命令

PC ← 飛び先番地

PC ← mml^{*1}

^{*1}: (ABS)のときM(mml + 1, mml)

(ABS,X)のときM(mml + X + 1, mml + X)

・JMPL命令

PG, PC ← 飛び先番地

PC ← mml^{*2}

PG ← hh^{*3}

^{*2}: L(ABS)のときM(mml + 1, mml)

^{*3}: L(ABS)のときM(mml + 2)

【説明】 指定された番地へジャンプします。16ビット（JMP）、又は24ビット（JMPL）の番地で、分岐先の番地を指定します。

JMP命令は、命令の最後のバイトをバンクの最上位（ $XXFFFF_{16}$ ）番地に配置したり、命令をバンクを横切るように配置すると、PGの内容が1加算され、飛び先が次のバンクになります。

インダイレクトアドレッシングを使用する場合、参照するメモリは同一プログラムバンク（PGで指すバンク）になります。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
ABS	JMP mml	9C ₁₆ ,ll,mm	3	4
ABL	JMPL hhmmll	AC ₁₆ ,ll,mm,hh	4	5
(ABS)	JMP (mml)	31 ₁₆ ,5C ₁₆ ,ll,mm	4	7
L(ABS)	JMPL L(mml)	31 ₁₆ ,5D ₁₆ ,ll,mm	4	9
(ABS,X)	JMP (mml,X)	BC ₁₆ ,ll,mm	3	7

【記述例】

JMP ADDR16 ; ADDR16番地へ分岐
 JMPL ADDR24 ; ADDR24番地へ分岐

【機能】 サブルーチン呼び出し

【演算長】 -

【動作】 ・ JSR命令

スタック ← PC + 3

PC ← 飛び先番地

PC ← PC + 3

M(S, S - 1) ← PC

S ← S - 2

PC ← mmlI*

* : (ABS,X)のときM(mmlI + X + 1, mmlI + X)

スタック領域

命令実行直後の (S)

命令実行直前の (S)

PCL
PCH

・ JSRL命令

スタック ← PG, PC + 4

PG, PC ← 飛び先番地

PC ← PC + 4

M(S ~ S - 2) ← PG, PC

S ← S - 3

PC ← mmlI

PG ← hh

スタック領域

命令実行直後の (S)

命令実行直前の (S)

PCL
PCH
PG

【説明】 PG、PCの内容をスタックに退避した後、指定された番地へジャンプします。16ビット (JSR)、又は24ビット (JSRL) の番地で、分岐先の番地を指定します。

JSR命令は、命令の最後のバイトをバンクの最上位 (XXXXFF₁₆) 番地に配置したり、命令をバンクを横切るように配置すると、PGの内容が1加算され、飛び先が次のバンクになります。

インダイレクトアドレッシングを使用する場合、参照するメモリは同一プログラムバンク (PGで指すバンク) になります。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
ABS	JSR mmlI	9D ₁₆ ,ll,mm	3	6
ABL	JSRL hhmmI	AD ₁₆ ,ll,mm,hh	4	7
(ABS,X)	JSR (mmlI,X)	BD ₁₆ ,ll,mm	3	8

【記述例】

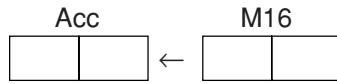
JSR ADDR16 ; ADDR16番地へ分岐
JSRL ADDR24 ; ADDR24番地へ分岐

【機能】 ロード

【演算長】 16ビット又は8ビット

【動作】 $Acc \leftarrow M$

・ $m = "0"$ のとき



・ $m = "1"$ のとき



このとき Acc_H の内容は変化しません。

【説明】 メモリの内容を Acc へロードします。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	N						Z	

N : 実行結果のMSBが“1”のとき、“1”になります。それ以外では“0”になります。

Z : 実行結果が“0”のとき、“1”になります。それ以外では“0”になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
IMM	LDA A,#imm	16 ₁₆ ,imm (81 ₁₆ ,16 ₁₆ ,imm)	2 (3)	1 (2)
DIR	LDA A,dd	1A ₁₆ ,dd (81 ₁₆ ,1A ₁₆ ,dd)	2 (3)	3 (4)
DIR,X	LDA A,dd,X	1B ₁₆ ,dd (81 ₁₆ ,1B ₁₆ ,dd)	2 (3)	4 (5)
(DIR)	LDA A,(dd)	11 ₁₆ ,10 ₁₆ ,dd (91 ₁₆ ,10 ₁₆ ,dd)	3 (3)	6 (6)
(DIR,X)	LDA A,(dd,X)	11 ₁₆ ,11 ₁₆ ,dd (91 ₁₆ ,11 ₁₆ ,dd)	3 (3)	7 (7)
(DIR),Y	LDA A,(dd),Y	18 ₁₆ ,dd (81 ₁₆ ,18 ₁₆ ,dd)	2 (3)	6 (7)
L(DIR)	LDA A,L(dd)	11 ₁₆ ,12 ₁₆ ,dd (91 ₁₆ ,12 ₁₆ ,dd)	3 (3)	8 (8)
L(DIR),Y	LDA A,L(dd),Y	19 ₁₆ ,dd (81 ₁₆ ,19 ₁₆ ,dd)	2 (3)	8 (9)
SR	LDA A,nn,S	11 ₁₆ ,13 ₁₆ ,nn (91 ₁₆ ,13 ₁₆ ,nn)	3 (3)	5 (5)
(SR),Y	LDA A,(nn,S),Y	11 ₁₆ ,14 ₁₆ ,nn (91 ₁₆ ,14 ₁₆ ,nn)	3 (3)	8 (8)
ABS	LDA A,mml	1E ₁₆ ,ll,mm (81 ₁₆ ,1E ₁₆ ,ll,mm)	3 (4)	3 (4)
ABS,X	LDA A,mml,X	1F ₁₆ ,ll,mm (81 ₁₆ ,1F ₁₆ ,ll,mm)	3 (4)	4 (5)
ABS,Y	LDA A,mml,Y	11 ₁₆ ,16 ₁₆ ,ll,mm (91 ₁₆ ,16 ₁₆ ,ll,mm)	4 (4)	5 (5)
ABL	LDA A,hhmml	1C ₁₆ ,ll,mm,hh (81 ₁₆ ,1C ₁₆ ,ll,mm,hh)	4 (5)	4 (5)
ABL,X	LDA A,hhmml,X	1D ₁₆ ,ll,mm,hh (81 ₁₆ ,1D ₁₆ ,ll,mm,hh)	4 (5)	5 (6)

注1. この表はアキュムレータAを使用する場合について、書かれています。アキュムレータBを使用する場合は記述形式の“ A, ”の部分が“ B, ”になり、機械語、バイト数、サイクル数は()内の値になります。

2. IMMアドレッシングモードで、mフラグが“0”の場合は、バイト数は1バイト増加します。

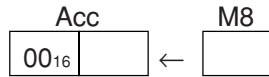
【記述例】

```
CLM
LDA.W    A,#IMM16          ; A  IMM16
LDA      B,MEM16          ; B  MEM16
SEM
LDA.B    A,#IMM8          ; AL IMM8
LDA      B,MEM8           ; BL MEM8
```

【機能】 ロード

【演算長】 16ビット

【動作】 $Acc \leftarrow M8$ (ゼロ拡張)



【説明】 メモリ中の8ビットデータを16ビット長にゼロ拡張してAccへ転送します。

この命令はmフラグの影響を受けません。

AccHの内容は常に“00₁₆”になります。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	0						Z	

N : 実行結果のMSBが“0”のため、常に“0”になります。

Z : 実行結果が“0”のとき、“1”になります。それ以外では“0”になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
IMM	LDAB A,#imm	28 ₁₆ ,imm (81 ₁₆ ,28 ₁₆ ,imm)	2 (3)	1 (2)
DIR	LDAB A,dd	0A ₁₆ ,dd (81 ₁₆ ,0A ₁₆ ,dd)	2 (3)	3 (4)
DIR,X	LDAB A,dd,X	0B ₁₆ ,dd (81 ₁₆ ,0B ₁₆ ,dd)	2 (3)	4 (5)
(DIR)	LDAB A,(dd)	11 ₁₆ ,00 ₁₆ ,dd (91 ₁₆ ,00 ₁₆ ,dd)	3 (3)	6 (6)
(DIR,X)	LDAB A,(dd,X)	11 ₁₆ ,01 ₁₆ ,dd (91 ₁₆ ,01 ₁₆ ,dd)	3 (3)	7 (7)
(DIR),Y	LDAB A,(dd),Y	08 ₁₆ ,dd (81 ₁₆ ,08 ₁₆ ,dd)	2 (3)	6 (7)
L(DIR)	LDAB A,L(dd)	11 ₁₆ ,02 ₁₆ ,dd (91 ₁₆ ,02 ₁₆ ,dd)	3 (3)	8 (8)
L(DIR),Y	LDAB A,L(dd),Y	09 ₁₆ ,dd (81 ₁₆ ,09 ₁₆ ,dd)	2 (3)	8 (9)
SR	LDAB A,nn,S	11 ₁₆ ,03 ₁₆ ,nn (91 ₁₆ ,03 ₁₆ ,nn)	3 (3)	5 (5)
(SR),Y	LDAB A,(nn,S),Y	11 ₁₆ ,04 ₁₆ ,nn (91 ₁₆ ,04 ₁₆ ,nn)	3 (3)	8 (8)
ABS	LDAB A,mml	0E ₁₆ ,ll,mm (81 ₁₆ ,0E ₁₆ ,ll,mm)	3 (4)	3 (4)
ABS,X	LDAB A,mml,X	0F ₁₆ ,ll,mm (81 ₁₆ ,0F ₁₆ ,ll,mm)	3 (4)	4 (5)
ABS,Y	LDAB A,mml,Y	11 ₁₆ ,06 ₁₆ ,ll,mm (91 ₁₆ ,06 ₁₆ ,ll,mm)	4 (4)	5 (5)
ABL	LDAB A,hhmml	0C ₁₆ ,ll,mm,hh (81 ₁₆ ,0C ₁₆ ,ll,mm,hh)	4 (5)	4 (5)
ABL,X	LDAB A,hhmml,X	0D ₁₆ ,ll,mm,hh (81 ₁₆ ,0D ₁₆ ,ll,mm,hh)	4 (5)	5 (6)

注. この表はアキュムレータAを使用する場合について、書かれています。アキュムレータBを使用する場合は記述形式の“ A, ”の部分が“ B, ”になり、機械語、バイト数、サイクル数は()内に示す値になります。

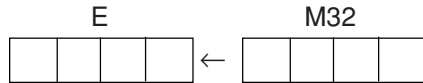
【記述例】

LDAB A,#IMM8 ; A IMM8 (AH 00₁₆, AL IMM8)
 LDAB B,MEM8 ; B MEM8 (BH 00₁₆, BL MEM8)

【機能】 ロード

【演算長】 32ビット

【動作】 $E \leftarrow M32$



「4.4 32ビット演算命令実行時の注意」を参照してください。

【説明】 メモリ中の32ビットデータをEへ転送します。
この命令はmフラグの影響を受けません。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	N						Z	

- N : 実行結果のMSBが“1”のとき、“1”になります。それ以外では“0”になります。
- Z : 実行結果が“0”のとき、“1”になります。それ以外では“0”になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
IMM	LDAD E,#imm	2C ₁₆ ,imm _{LL} ,imm _{LH} ,imm _{HL} ,imm _{HH}	5	3
DIR	LDAD E,dd	8A ₁₆ ,dd	2	6
DIR,X	LDAD E,dd,X	8B ₁₆ ,dd	2	7
(DIR)	LDAD E,(dd)	11 ₁₆ ,80 ₁₆ ,dd	3	9
(DIR,X)	LDAD E,(dd,X)	11 ₁₆ ,81 ₁₆ ,dd	3	10
(DIR),Y	LDAD E,(dd),Y	88 ₁₆ ,dd	2	9
L(DIR)	LDAD E,L(dd)	11 ₁₆ ,82 ₁₆ ,dd	3	11
L(DIR),Y	LDAD E,L(dd),Y	89 ₁₆ ,dd	2	11
SR	LDAD E,nn,S	11 ₁₆ ,83 ₁₆ ,nn	3	8
(SR),Y	LDAD E,(nn,S),Y	11 ₁₆ ,84 ₁₆ ,nn	3	11
ABS	LDAD E,mml	8E ₁₆ ,ll,mm	3	6
ABS,X	LDAD E,mml,X	8F ₁₆ ,ll,mm	3	7
ABS,Y	LDAD E,mml,Y	11 ₁₆ ,86 ₁₆ ,ll,mm	4	8
ABL	LDAD E,hhmml	8C ₁₆ ,ll,mm,hh	4	7
ABL,X	LDAD E,hhmml,X	8D ₁₆ ,ll,mm,hh	4	8

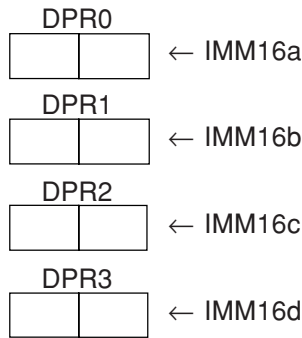
【記述例】

```
LDAD    E,#IMM32                ; E    IMM32
                                ; (B    IMM32H, A    IMM32L)
LDAD    E,MEM32                 ; E    MEM32
                                ; (B    MEM32H, A    MEM32L)
```

【機能】 ロード

【演算長】 16ビット

【動作】 DPR0 ← IMM16a (複数DPRへの設定可)
 DPR1 ← IMM16b
 DPR2 ← IMM16c
 DPR3 ← IMM16d



【説明】 16ビット長のイミディエイト値をDPR0～DPR3へ転送します。

この命令はmフラグの影響を受けません。

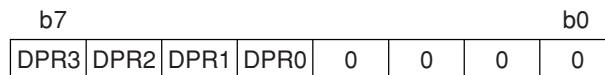
複数のDPRへの値の設定が一命令で可能です。複数のDPRを指定したとき、転送はDPR0、DPR1、DPR2、DPR3の順で行われます。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
IMM	LDD n,#imm	B8 ₁₆ ,?0 ₁₆ ,imm _L ,imm _H	4	13
	LDD (n ₁ ,...,n _i),#imm ₁ ,...,#imm _i	B8 ₁₆ ,?0 ₁₆ ,imm _{L1} ,imm _{H1} ,...,imm _{Li} ,imm _{Hi}	2 × i + 2	2 × i + 11

- 注1. nには0～3の値が設定できます。
 2. 記述形式2行目は、複数のDPRへの設定を一命令で行う場合です。
 3. (n₁,...,n_i)の括弧内は0～3 (DPRnの番号) を指定します。
 4. i : 指定するDPRnの数 (1～4) を示します。
 5. ? : 指定したDPRnに対応するビットが“1”になります。
 ビットとDPRnの対応は下図のとおりです。



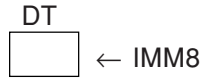
【記述例】

```
LDD      0,#IMM16          ; DPR0  IMM16
LDD      (0,3),#IMM16a,#IMM16b ; DPR0  IMM16a
                                           ; DPR3  IMM16b
```

【機能】 ロード

【演算長】 8ビット

【動作】 DT ← IMM8



【説明】 イミディエイト値をDTへロードします。
この命令はmフラグの影響を受けません。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
IMM	LDT #imm	31 ₁₆ ,4A ₁₆ ,imm	3	4

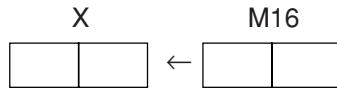
【記述例】 LDT #IMM8 ; DT IMM8

【機能】 ロード

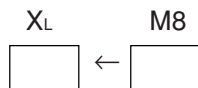
【演算長】 16ビット又は8ビット

【動作】 $X \leftarrow M$

・x=“0”のとき



・x=“1”のとき



このときX_Hの内容は変化しません。

【説明】 メモリの内容をXへロードします。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	N						Z	

N : 実行結果のMSBが“1”のとき、“1”になります。それ以外では“0”になります。

Z : 実行結果が“0”のとき、“1”になります。それ以外では“0”になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
IMM	LDX #imm	C6 ₁₆ ,imm	2	1
DIR	LDX dd	02 ₁₆ ,dd	2	3
DIR,Y	LDX dd,Y	41 ₁₆ ,05 ₁₆ ,dd	3	5
ABS	LDX mml	07 ₁₆ ,ll,mm	3	3
ABS,Y	LDX mml,Y	41 ₁₆ ,06 ₁₆ ,ll,mm	4	5

注 . IMMアドレッシングモードで、xフラグが“0”の場合は、バイト数は1バイト増加します。

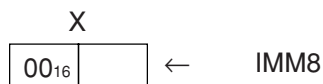
【記述例】

```
CLP      x
LDX.W    #IMM16          ; X  IMM16
LDX      MEM16          ; X  MEM16
SEP      x
LDX.B    #IMM8          ; XL IMM8
LDX      MEM8           ; XL MEM8
```


【機能】 ロード

【演算長】 16ビット

【動作】 $X \leftarrow \text{IMM8}$ (ゼロ拡張)



【説明】 8ビット長のイミディエイト値を16ビット長にゼロ拡張してXへ転送します。

この命令はxフラグの影響を受けません。

X_Hの内容は常に“00₁₆”になります。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	0						Z	

N : 実行結果のMSBが“0”のため、常に“0”になります。

Z : 実行結果が“0”のとき、“1”になります。それ以外では“0”になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
IMM	LDXB #imm	27 ₁₆ ,imm	2	1

【記述例】

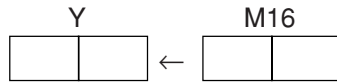
LDXB #IMM8 ; X ← IMM8 (X_H ← 00₁₆, X_L ← IMM8)

【機能】 ロード

【演算長】 16ビット又は8ビット

【動作】 $Y \leftarrow M$

・x="0" のとき



・x="1" のとき



このときY_Hの内容は変化しません。

【説明】 メモリの内容をYへロードします。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	N						Z	

N : 実行結果のMSBが“1”のとき、“1”になります。それ以外では“0”になります。

Z : 実行結果が“0”のとき、“1”になります。それ以外では“0”になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
IMM	LDY #imm	D6 ₁₆ ,imm	2	1
DIR	LDY dd	12 ₁₆ ,dd	2	3
DIR,X	LDY dd,X	41 ₁₆ ,1B ₁₆ ,dd	3	5
ABS	LDY mml	17 ₁₆ ,ll,mm	3	3
ABS,X	LDY mml,X	41 ₁₆ ,1F ₁₆ ,ll,mm	4	5

注：IMMアドレッシングモードで、xフラグが“0”の場合は、バイト数は1バイト増加します。

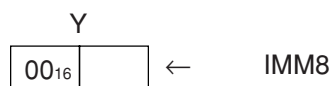
【記述例】

```
CLP      x
LDY.W    #IMM16          ; Y  IMM16
LDY      MEM16          ; Y  MEM16
SEP      x
LDY.B    #IMM8          ; YL IMM8
LDY      MEM8           ; YL MEM8
```

【機能】 ロード

【演算長】 16ビット

【動作】 $Y \leftarrow \text{IMM8}$ (ゼロ拡張)



【説明】 8ビット長のイミディエイト値を16ビット長にゼロ拡張してYへ転送します。

この命令はxフラグの影響を受けません。

Y_Hの内容は常に“00₁₆”になります。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	0						Z	

N : 実行結果のMSBが“0”のため、常に“0”になります。

Z : 実行結果が“0”のとき、“1”になります。それ以外では“0”になります。

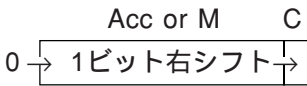
アドレスモード	記述形式	機械語	バイト数	サイクル数
IMM	LDYB #imm	37 ₁₆ ,imm	2	1

【記述例】

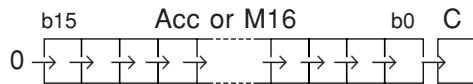
LDYB #IMM8 ; Y IMM8 (Y_H 00₁₆, Y_L IMM8)

【機能】 論理右シフト

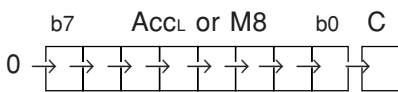
【演算長】 16ビット又は8ビット

【動作】 

・ m = “ 0 ” のとき



・ m = “ 1 ” のとき



このときAccHの内容は変化しません。

【説明】 Acc又はメモリの内容を右へ1ビットシフトします。このとき、Acc又はメモリのMSBには“0”が入ります。また、シフトする前のLSBの内容はCフラグへ入ります。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	0						Z	C

N : “ 0 ” になります。

Z : 実行結果が“ 0 ” のとき、“ 1 ” になります。それ以外では“ 0 ” になります。

C : 実行前のLSBが“ 1 ” のとき、“ 1 ” になります。それ以外では“ 0 ” になります。

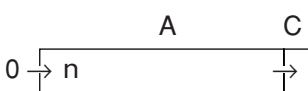
アドレッシングモード	記述形式	機械語	バイト数	サイクル数
A	LSR A	43 ₁₆	1	1
A	LSR B	81 ₁₆ ,43 ₁₆	2	2
DIR	LSR dd	21 ₁₆ ,2A ₁₆ ,dd	3	7
DIR,X	LSR dd,X	21 ₁₆ ,2B ₁₆ ,dd	3	8
ABS	LSR mml	21 ₁₆ ,2E ₁₆ ,ll,mm	4	7
ABS,X	LSR mml,X	21 ₁₆ ,2F ₁₆ ,ll,mm	4	8

【記述例】

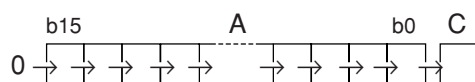
```
CLM
LSR      A                ; A   Aを1ビット論理右シフト
LSR     MEM16            ; MEM16 MEM16を1ビット論理右シフト
SEM
LSR     A                ; AL  ALを1ビット論理右シフト
LSR     MEM8             ; MEM8 MEM8を1ビット論理右シフト
```

【機能】 論理右シフト

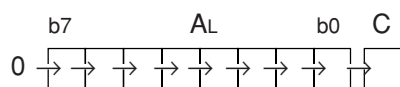
【演算長】 16ビット又は8ビット

【動作】  (n : シフト回数。n=0~15)

・ m = “ 0 ” のとき



・ m = “ 1 ” のとき



このときAHの内容は変化しません。

【説明】 Aの内容を右へnビットシフトします。1ビットシフトごとに、AのMSBには“0”が入ります。また、1ビットシフトごとに、LSBはCフラグへ入ります。この命令では、Bは使用できません。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	0						Z	C

N : 実行結果のMSBが“0”のため、常に“0”になります。

Z : 実行結果が“0”のとき、“1”になります。それ以外では“0”になります。

C : (n-1)ビットシフトしたときのLSBが“1”のとき、“1”になります。それ以外では“0”になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
A	LSR A,#imm	C1 ₁₆ ,imm	2	imm + 6

注 . immには0~15の値(シフト回数)が設定できます。


【記述例】

```
CLM
LSR      A,#15      ; A   Aを15ビット論理右シフト
SEM
LSR      A,#7       ; AL  ALを7ビット論理右シフト
```

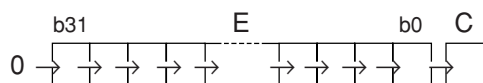
【機能】 論理右シフト

【演算長】 32ビット

【動作】



(n : シフト回数。n=0~31)



【説明】 Eの内容を32ビット長で右へnビットシフトします。1ビットシフトごとに、EのMSBには“0”が入ります。また、1ビットシフトごとに、LSBはCフラグへ入ります。
この命令はmフラグの影響を受けません。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	0						Z	C

- N : 実行結果のMSBが“0”のため、常に“0”になります。
Z : 実行結果が“0”のとき、“1”になります。それ以外では“0”になります。
C : (n-1)ビットシフトしたときのLSBが“1”のとき、“1”になります。それ以外では“0”になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
A	LSRD E,#imm	D1 ₁₆ ,imm	2	imm + 8

注 . immには0~31の値(シフト回数)が設定できます。

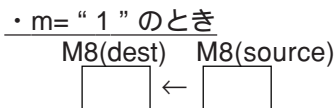
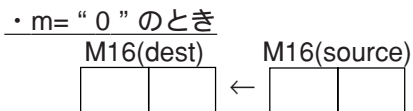
【記述例】

LSRD E,#16 ; E Eを16ビット論理右シフト

【機能】 メモリ間転送

【演算長】 16ビット又は8ビット

【動作】 $M \leftarrow M$



【説明】 ソース側メモリの内容をデスティネーション側メモリへ転送します。
従来の7700ファミリのLDM命令の機能はこの命令に含まれます。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C

アドレッシングモード		記述形式	機械語	バイト数	サイクル数
dest	source				
DIR	IMM	MOVM dd,#imm	86 ₁₆ ,imm,dd	3	5
DIR	ABS	MOVM dd,mml	5C ₁₆ ,ll,mm,dd	4	6
DIR	ABS,X	MOVM dd,mml,X	5D ₁₆ ,ll,mm,dd	4	7
ABS	IMM	MOVM mml,#imm	96 ₁₆ ,imm,ll,mm	4	4
ABS	DIR	MOVM mml,dd	78 ₁₆ ,dd,ll,mm	4	5
ABS	DIR,X	MOVM mml,dd,X	79 ₁₆ ,dd,ll,mm	4	6
ABS,X	IMM	MOVM mml,X,#imm	31 ₁₆ ,57 ₁₆ ,imm,ll,mm	5	6
ABS	ABS	MOVM mml ₁ ,mml ₂	7C ₁₆ ,ll ₂ ,mm ₂ ,ll ₁ ,mm ₁	5	5
DIR,X	IMM	MOVM dd,X,#imm	31 ₁₆ ,47 ₁₆ ,imm,dd	4	7
DIR	DIR	MOVM dd ₁ ,dd ₂	58 ₁₆ ,dd ₂ ,dd ₁	3	6

注 . IMMアドレッシングモードで、mフラグが“0”の場合は、バイト数は1バイト増加します。

【記述例】

```

CLM
MOVM.W    MEM16,#IMM16                ; MEM16  IMM16
MOVM      MEM16(dest),MEM16(source)    ; MEM16(dest) MEM16(source)
SEM
MOVM.B    MEM8,#IMM8                  ; MEM8  IMM8
MOVM      MEM8(dest),MEM8(source)      ; MEM8(dest) MEM8(source)
    
```

【機能】 メモリ間転送

【演算長】 8ビット

【動作】 M8 ← M8



【説明】 ソース側メモリの内容を8ビット長でデスティネーション側メモリへ転送します。

ソース側メモリの内容は変化しません。

この命令はmフラグの影響を受けません。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C

アドレッシングモード		記述形式	機械語	バイト数	サイクル数
dest	source				
DIR	IMM	MOVMB dd,#imm	A9 ₁₆ ,imm,dd	3	5
DIR	ABS	MOVMB dd,mml	4C ₁₆ ,ll,mm,dd	4	6
DIR	ABS,X	MOVMB dd,mml,X	4D ₁₆ ,ll,mm,dd	4	7
ABS	IMM	MOVMB mml,#imm	B9 ₁₆ ,imm,ll,mm	4	4
ABS	DIR	MOVMB mml,dd	68 ₁₆ ,dd,ll,mm	4	5
ABS	DIR,X	MOVMB mml,dd,X	69 ₁₆ ,dd,ll,mm	4	6
ABS,X	IMM	MOVMB mml,X,#imm	31 ₁₆ ,3B ₁₆ ,imm,ll,mm	5	6
ABS	ABS	MOVMB mml ₁ ,mml ₂	6C ₁₆ ,ll ₂ ,mm ₂ ,ll ₁ ,mm ₁	5	5
DIR,X	IMM	MOVMB dd,X,#imm	31 ₁₆ ,3A ₁₆ ,imm,dd	4	7
DIR	DIR	MOVMB dd ₁ ,dd ₂	48 ₁₆ ,dd ₂ ,dd ₁	3	6

【記述例】

```

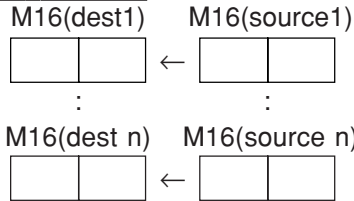
MOVMB     MEM8,#IMM8                ; MEM8 IMM8
MOVMB     MEM8(dest),MEM8(source)   ; MEM8(dest) MEM8(source)
  
```


【機能】 メモリ間転送

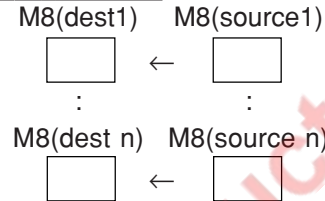
【演算長】 16ビット又は8ビット

【動作】 $M(\text{dest}1) \leftarrow M(\text{source}1)$ (n : 繰り返し転送回数。n=0~15)
 $M(\text{dest}2) \leftarrow M(\text{source}2)$
 ⋮
 $M(\text{dest} n) \leftarrow M(\text{source} n)$

・ m=“0” のとき



・ m=“1” のとき



【説明】 複数のメモリ間転送を一命令で行います。命令の3バイト目以降で指定されたアドレスに従い、転送します。最大15回の転送が行えます。

ソース側メモリの内容は変化しません。

転送回数に“0”を指定したときは転送を行いません。

ソース及びデスティネーションは各々すべて同一のアドレッシングモードで転送します。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C

アドレッシングモード		記述形式	機械語	バイト数	サイクル数
dest	source				
DIR	IMM	MOVR #n,dd ₁ ,#imm ₁ ...,dd _n ,#imm _n	61 ₁₆ ,n+10 ₁₆ ,imm ₁ ,dd ₁ ,...,imm _n ,dd _n	2 × n+2 (注 2)	5 × n+3
DIR	DIR	MOVR #n,dd ₁ ,dd _{s1} ...,dd _{dn} ,dd _{sn}	61 ₁₆ ,n+50 ₁₆ ,dd _{s1} ,dd _{d1} ,...,dd _{sn} ,dd _{dn}	2 × n+2	6 × n+3
DIR	ABS	MOVR #n,dd ₁ ,mml ₁ ...,dd _n ,mml _n	61 ₁₆ ,n+90 ₁₆ ,ll ₁ ,mm ₁ ,dd ₁ ...,ll _n ,mm _n ,dd _n	3 × n+2	6 × n+3
DIR	ABS,X	MOVR #n,dd ₁ ,mml ₁ ,X ...,dd _n ,mml _n ,X	71 ₁₆ ,n+10 ₁₆ ,ll ₁ ,mm ₁ ,dd ₁ ...,ll _n ,mm _n ,dd _n	3 × n+2	6 × n+3
ABS	IMM	MOVR #n,mml ₁ ,#imm ₁ ...,mml _n ,#imm _n	61 ₁₆ ,n+30 ₁₆ ,imm ₁ ,ll ₁ ,mm ₁ ...,imm _n ,ll _n ,mm _n	3 × n+2 (注 2)	4 × n+3
ABS	DIR	MOVR #n,mml ₁ ,dd ₁ ...,mml _n ,dd _n	61 ₁₆ ,n+70 ₁₆ ,dd ₁ ,ll ₁ ,mm ₁ ...,dd _n ,ll _n ,mm _n	3 × n+2	5 × n+3
ABS	DIR,X	MOVR #n,mml ₁ ,dd ₁ ,X ...,mml _n ,dd _n ,X	71 ₁₆ ,n+70 ₁₆ ,dd ₁ ,ll ₁ ,mm ₁ ...,dd _n ,ll _n ,mm _n	3 × n+2	6 × n+3
ABS	ABS	MOVR #n,mml ₁ ,mml _{s1} ...,mml _{dn} ,mml _{sn}	61 ₁₆ ,n+B0 ₁₆ ,ll _{s1} ,mm _{s1} ,ll _{d1} ,mm _{d1} ...,ll _{sn} ,mm _{sn} ,ll _{dn} ,mm _{dn}	4 × n+2	5 × n+3

注1 . nには0~15の値が設定できます。
 2 . m=0の場合、nバイト増加します。

【記述例】

CLM
 MOVR.W 2,MEM16(dest1),#IMM16a,MEM16(dest2),#IMM16b ; MEM16(dest1) IMM16a
 ; MEM16(dest2) IMM16b
 MOVR 2,MEM16(dest1),MEM16(source1),MEM16(dest2),MEM16(source2)
 ; MEM16(dest1) MEM16(source1)
 ; MEM16(dest2) MEM16(source2)
 SEM
 MOVR.B 2,MEM8(dest1),#IMM8a,MEM8(dest2),#IMM8b ; MEM8(dest1) IMM8a
 ; MEM8(dest2) IMM8b
 MOVR 2,MEM8(dest1),MEM8(source1),MEM8(dest2),MEM8(source2)
 ; MEM8(dest1) MEM8(source1)
 ; MEM8(dest2) MEM8(source2)

【機能】 メモリ間転送

【演算長】 8ビット

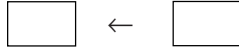
【動作】 M8(dest1) ← M8(source1) (n : 繰り返し転送回数。n=0 ~ 15)

M8(dest2) ← M8(source2)

⋮

M8(dest n) ← M8(source n)

M8(dest1) M8(source1)



⋮

M8(dest n) M8(source n)



【説明】 複数のメモリ間転送を一命令で行います。命令の3バイト目以降で指定されたアドレスに従い、バイト長で転送します。最大15回の転送が行えます。

ソース側メモリの内容は変化しません。

転送回数に“0”を指定したときは転送を行いません。

ソース及びデスティネーションは各々すべて同一のアドレッシングモードで転送します。

この命令はmフラグの影響を受けません。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C

アドレッシングモード		記述形式	機械語	バイト数	サイクル数
dest	source				
DIR	IMM	MOVRB #n,dd1,#imm1 ...,ddn,#immn	61 ₁₆ ,n+00 ₁₆ ,imm1,dd1,...,immn,ddn	2 × n+2	5 × n+3
DIR	DIR	MOVRB #n,dd1,dds1 ...,dddn,dds1	61 ₁₆ ,n+40 ₁₆ ,dds1,dd1,...,ddsn,dddn	2 × n+2	6 × n+3
DIR	ABS	MOVRB #n,dd1,mml1 ...,ddn,mmln	61 ₁₆ ,n+80 ₁₆ ,ll1,mm1,dd1 ...,lln,mmn,ddn	3 × n+2	6 × n+3
DIR	ABS,X	MOVRB #n,dd1,mml1,X ...,ddn,mmln,X	71 ₁₆ ,n+00 ₁₆ ,ll1,mm1,dd1 ...,lln,mmn,ddn	3 × n+2	6 × n+3
ABS	IMM	MOVRB #n,mml1,#imm1 ...,mmln,#immn	61 ₁₆ ,n+20 ₁₆ ,imm1,ll1,mm1 ...,immn,lln,mmn	3 × n+2	4 × n+3
ABS	DIR	MOVRB #n,mml1,dd1 ...,mmln,ddn	61 ₁₆ ,n+60 ₁₆ ,dd1,ll1,mm1 ...,ddn,lln,mmn	3 × n+2	5 × n+3
ABS	DIR,X	MOVRB #n,mml1,dd1,X ...,mmln,ddn,X	71 ₁₆ ,n+60 ₁₆ ,dd1,ll1,mm1 ...,ddn,lln,mmn	3 × n+2	6 × n+3
ABS	ABS	MOVRB #n,mml1,mm1s1 ...,mml1dn,mm1s1n	61 ₁₆ ,n+A0 ₁₆ ,lls1,mm1s1,ll1,mm1 ...,llsn,mm1sn,ll1dn,mm1dn	4 × n+2	5 × n+3

注 . nには0 ~ 15の値が設定できます。

【記述例】

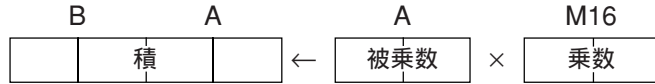
```
MOVRB 2, MEM8(dest1), #IMM8a, MEM8(dest2), #IMM8b ; MEM8(dest1) IMM8a
; MEM8(dest2) IMM8b
MOVRB 2, MEM8(dest1), MEM8(source1), MEM8(dest2), MEM8(source2)
; MEM8(dest1) MEM8(source1)
; MEM8(dest2) MEM8(source2)
```

【機能】 乗算（符号なし）

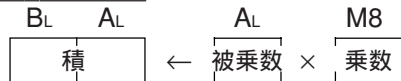
【演算長】 16ビット又は8ビット

【動作】 (B, A) ← A (被乗数) × M (乗数)

・ m = “0” のとき



・ m = “1” のとき



このときAH、BHの内容は変化しません。

【説明】 Aの内容とメモリの内容を乗算します。積の上位はBに、下位はAに格納されます。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	N						Z	0

N : 実行結果のMSB (BのMSB) が “1” のとき、 “1” になります。それ以外では “0” になります。

Z : 実行結果が “0” のとき、 “1” になります。それ以外では “0” になります。

C : “0” になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
IMM	MPY #imm	31 ₁₆ ,C7 ₁₆ ,imm	3	8
DIR	MPY dd	21 ₁₆ ,CA ₁₆ ,dd	3	9
DIR,X	MPY dd,X	21 ₁₆ ,CB ₁₆ ,dd	3	10
(DIR)	MPY (dd)	21 ₁₆ ,C0 ₁₆ ,dd	3	11
(DIR,X)	MPY (dd),X	21 ₁₆ ,C1 ₁₆ ,dd	3	12
(DIR),Y	MPY (dd),Y	21 ₁₆ ,C8 ₁₆ ,dd	3	12
L(DIR)	MPY L(dd)	21 ₁₆ ,C2 ₁₆ ,dd	3	13
L(DIR),Y	MPY L(dd),Y	21 ₁₆ ,C9 ₁₆ ,dd	3	14
SR	MPY nn,S	21 ₁₆ ,C3 ₁₆ ,nn	3	10
(SR),Y	MPY (nn,S),Y	21 ₁₆ ,C4 ₁₆ ,nn	3	13
ABS	MPY mml	21 ₁₆ ,CE ₁₆ ,ll,mm	4	9
ABS,X	MPY mml,X	21 ₁₆ ,CF ₁₆ ,ll,mm	4	10
ABS,Y	MPY mml,Y	21 ₁₆ ,C6 ₁₆ ,ll,mm	4	10
ABL	MPY hhmml	21 ₁₆ ,CC ₁₆ ,ll,mm,hh	5	10
ABL,X	MPY hhmml,X	21 ₁₆ ,CD ₁₆ ,ll,mm,hh	5	11

注1 . IMMアドレッシングモードで、mフラグが “0” の場合は、バイト数は1バイト増加します。

2 . この表でのサイクル数は、8ビット×8ビット演算の場合を示しています。

16ビット×16ビット演算の場合のサイクル数は、さらに4サイクル増加します。

【記述例】

```

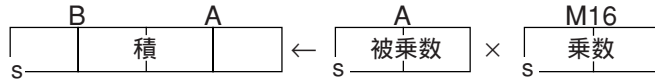
CLM
MPY.W      #IMM16      ; B,A   A × IMM16
MPY        MEM16      ; B,A   A × MEM16
SEM
MPY.B      #IMM8       ; BL,AL  AL × IMM8
MPY        MEM8       ; BL,AL  AL × MEM8
    
```

【機能】 乗算（符号付き）

【演算長】 16ビット又は8ビット

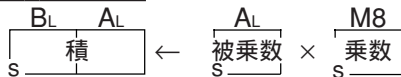
【動作】 (B, A) ← A (被乗数) × M (乗数)

・ m= “0” のとき



Sは符号ビットでデータのMSBを示します。

・ m= “1” のとき



Sは符号ビットでデータのMSBを示します。

このときA_H、B_Hの内容は変化しません。

【説明】 Aの内容とメモリの内容を符号付きで乗算します。積の上位はBに、下位はAに格納されます。BのMSBが符号ビットになります。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	N						Z	0

- N : 実行結果のMSB (BのMSB) が “1” のとき、 “1” になります。それ以外では “0” になります。
- Z : 実行結果が “0” のとき、 “1” になります。それ以外では “0” になります。
- C : “0” になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
IMM	MPYS #imm	31 ₁₆ ,D7 ₁₆ ,imm	3	8
DIR	MPYS dd	21 ₁₆ ,DA ₁₆ ,dd	3	9
DIR,X	MPYS dd,X	21 ₁₆ ,DB ₁₆ ,dd	3	10
(DIR)	MPYS (dd)	21 ₁₆ ,D0 ₁₆ ,dd	3	11
(DIR,X)	MPYS (dd,X)	21 ₁₆ ,D1 ₁₆ ,dd	3	12
(DIR),Y	MPYS (dd),Y	21 ₁₆ ,D8 ₁₆ ,dd	3	12
L(DIR)	MPYS L(dd)	21 ₁₆ ,D2 ₁₆ ,dd	3	13
L(DIR),Y	MPYS L(dd),Y	21 ₁₆ ,D9 ₁₆ ,dd	3	14
SR	MPYS nn,S	21 ₁₆ ,D3 ₁₆ ,nn	3	10
(SR),Y	MPYS (nn,S),Y	21 ₁₆ ,D4 ₁₆ ,nn	3	13
ABS	MPYS mmll	21 ₁₆ ,DE ₁₆ ,ll,mm	4	9
ABS,X	MPYS mmll,X	21 ₁₆ ,DF ₁₆ ,ll,mm	4	10
ABS,Y	MPYS mmll,Y	21 ₁₆ ,D6 ₁₆ ,ll,mm	4	10
ABL	MPYS hhmmll	21 ₁₆ ,DC ₁₆ ,ll,mm,hh	5	10
ABL,X	MPYS hhmmll,X	21 ₁₆ ,DD ₁₆ ,ll,mm,hh	5	11

- 注1. IMMアドレッシングモードで、mフラグが “0” の場合は、バイト数は1バイト増加します。
- 2. この表でのサイクル数は、乗数と被乗数が同符号の8ビット×8ビット演算の場合を示しています。また、16ビット×16ビット演算の場合のサイクル数は、さらに4サイクル増加します。

【記述例】

```

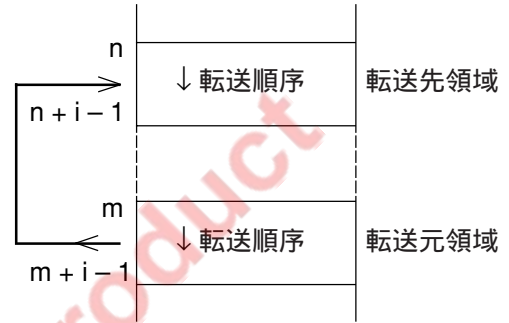
CLM
MPYS.W      #IMM16          ; B,A  A × IMM16
MPYS        MEM16          ; B,A  A × MEM16
SEM
MPYS.B      #IMM8          ; BL,AL AL × IMM8
MPYS        MEM8           ; BL,AL AL × MEM8
    
```

【機能】 ブロック転送

【演算長】 16ビット又は8ビット

【動作】 $M(n \sim n+i-1) \leftarrow M(m \sim m+i-1)$ (i: 転送バイト数)

【説明】 通常、上位番地にあるデータブロックを下位番地へ転送します。
転送は転送ブロックの下位番地から上位番地方向の順で行います。



命令の3バイト目 : 転送先のバンク
命令の4バイト目 : 転送元のバンク

X : 転送元番地
Y : 転送先番地
A : 転送するデータブロックのバイト数

をそれぞれ指定 (X、Y、Aは本命令実行前に指定)

「m=0」のとき : 0 ~ 65535バイトまでのデータを転送可

「m=1」のとき : 0 ~ 255バイトまでのデータを転送可

「x=0」のとき : 転送元及び転送先の番地を0 ~ 65535 (FFFF₁₆) 番地まで設定可

「x=1」のとき : 転送元及び転送先の番地を0 ~ 255 (FF₁₆) 番地まで設定可

実行後の各レジスタ内容

X : 転送元最終 (最上位) 番地 + 1
Y : 転送先最終 (最上位) 番地 + 1
A : 「m=0」のときFFFF₁₆、 「m=1」のときFF₁₆
DT : 転送先バンク

「m=1」のときA_Hの内容は変化しません。また「x=1」のときX_H、Y_Hの内容は変化しません。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
BLK	MVN hh ₁ ,hh ₂	31 ₁₆ ,2B ₁₆ ,hh ₁ ,hh ₂	4	5×i+5

注 . この表でのサイクル数は、転送するバイト数 i が偶数の場合を示しています。

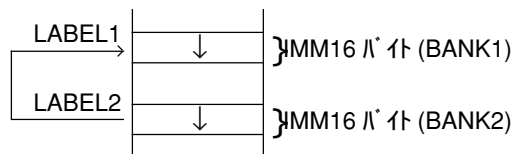
転送するバイト数が奇数の場合のサイクル数は、

5 × i + 10 サイクル

となります。

【記述例】

```
CLM ;
LDA.W #IMM16 ;
LDX LABEL2 ;
LDY LABEL1 ;
MVN BANK1,BANK2 ;
```

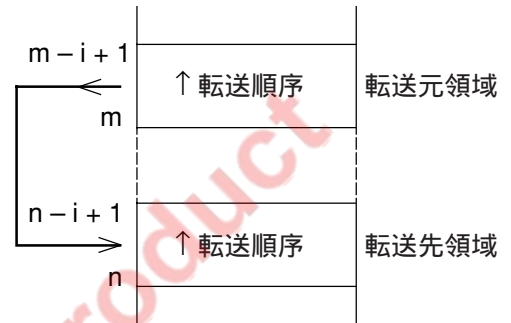


【機能】 ブロック転送

【演算長】 16ビット又は8ビット

【動作】 $M(n-i+1 \sim n) \leftarrow M(m-i+1 \sim m)$ (i: 転送バイト数)

【説明】 通常、下位番地にあるデータブロックを上位番地へ転送します。
転送は転送ブロックの上位番地から下位番地方向の順で行います。



命令の3バイト目 : 転送先のバンク
命令の4バイト目 : 転送元のバンク

X : 転送元番地
Y : 転送先番地
A : 転送するデータブロックのバイト数

をそれぞれ指定 (X、Y、Aは本命令実行前に指定)

「m=0」のとき : 0 ~ 65535バイトまでのデータを転送可

「m=1」のとき : 0 ~ 255バイトまでのデータを転送可

「x=0」のとき : 転送元及び転送先の番地を0 ~ 65535 (FFFF₁₆) 番地まで設定可

「x=1」のとき : 転送元及び転送先の番地を0 ~ 255 (FF₁₆) 番地まで設定可

実行後の各レジスタ内容

X : 転送元最終 (最下位) 番地 - 1
Y : 転送先最終 (最下位) 番地 - 1
A : 「m=0」のときFFFF₁₆、 「m=1」のときFF₁₆
DT : 転送先バンク

「m=1」のときA_Hの内容は変化しません。また「x=1」のときX_H、Y_Hの内容は変化しません。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
BLK	MVP hh1, hh2	31 ₁₆ , 2A ₁₆ , hh1, hh2	4	5 × i + 9

注 . この表でのサイクル数は、転送するバイト数 i が偶数の場合を示しています。

転送するバイト数が奇数の場合のサイクル数は、

5 × i + 14 サイクル (ただし、1バイト転送時は10サイクル)

となります。

【記述例】

```

CLM ;
LDA.W #IMM16 ;
LDX LABEL1 ;
LDY LABEL2 ;
MVP BANK2, BANK1 ;
    
```

【機能】 符号反転

【演算長】 16ビット又は8ビット

【動作】 $Acc \leftarrow -Acc$

・ $m = "0"$ のとき



・ $m = "1"$ のとき



このとき Acc_H の内容は変化しません。

【説明】 Acc の内容を符号反転し、その結果を Acc へ格納します。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	N	V					Z	C

N : 実行結果のMSBが“1”のとき、“1”になります。それ以外では“0”になります。

V : 符号付き演算と見なした結果が+32767、又は-32768（「 $m=1$ 」のときは+127又は-128）を越えると“1”になります。それ以外では“0”になります。

Z : 実行結果が“0”のとき、“1”になります。それ以外では“0”になります。

C : 符号なし演算と見なした結果が+65535（「 $m=1$ 」のときは+255）を越えると、“1”になります。それ以外では“0”になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
A	NEG A	24_{16}	1	1
A	NEG B	$81_{16}, 24_{16}$	2	2

【記述例】

```
CLM
NEG      A          ; A  -A
SEM
NEG      B          ; BL -BL
```

【機能】 符号反転

【演算長】 32ビット

【動作】 $E \leftarrow -E$



【説明】 Eの内容を符号反転し、その結果をEへ格納します。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	N	V					Z	C

- N : 実行結果のMSBが“1”のとき、“1”になります。それ以外では“0”になります。
- V : 符号付き演算と見なした結果が+2147483647、又は-2147483648を越えると“1”になります。それ以外では“0”になります。
- Z : 実行結果が“0”のとき、“1”になります。それ以外では“0”になります。
- C : 符号なし演算と見なした結果が+4294967295を越えると、“1”になります。それ以外では“0”になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
A	NEGD E	31 ₁₆ ,80 ₁₆	2	4

【記述例】

NEGD E ; E -E

【機能】 ノーオペレーション

【演算長】 -

【動作】 $PC \leftarrow PC + 1$
(PCにキャリーが生じた場合、 $PG \leftarrow PG + 1$)

【説明】 PCの内容を“1”増加させます。その他は何もしません。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
IMP	NOP	74 ₁₆	1	1

【記述例】

NOP ;

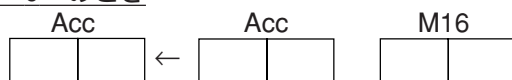
EOL announced Product

【機能】 論理和

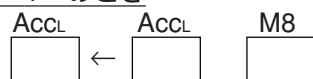
【演算長】 16ビット又は8ビット

【動作】 $Acc \leftarrow Acc \quad M$

・ $m = "0"$ のとき



・ $m = "1"$ のとき



このとき Acc_H の内容は変化しません。

【説明】 Acc の内容とメモリの内容の論理和をとり、結果を Acc へ格納します。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	N						Z	

N : 実行結果のMSBが“1”のとき、“1”になります。それ以外では“0”になります。

Z : 実行結果が“0”のとき、“1”になります。それ以外では“0”になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
IMM	ORA A,#imm	56 ₁₆ ,imm (81 ₁₆ ,56 ₁₆ ,imm)	2 (3)	1 (2)
DIR	ORA A,dd	5A ₁₆ ,dd (81 ₁₆ ,5A ₁₆ ,dd)	2 (3)	3 (4)
DIR,X	ORA A,dd,X	5B ₁₆ ,dd (81 ₁₆ ,5B ₁₆ ,dd)	2 (3)	4 (5)
(DIR)	ORA A,(dd)	11 ₁₆ ,50 ₁₆ ,dd (91 ₁₆ ,50 ₁₆ ,dd)	3 (3)	6 (6)
(DIR,X)	ORA A,(dd,X)	11 ₁₆ ,51 ₁₆ ,dd (91 ₁₆ ,51 ₁₆ ,dd)	3 (3)	7 (7)
(DIR),Y	ORA A,(dd),Y	11 ₁₆ ,58 ₁₆ ,dd (91 ₁₆ ,58 ₁₆ ,dd)	3 (3)	7 (7)
L(DIR)	ORA A,L(dd)	11 ₁₆ ,52 ₁₆ ,dd (91 ₁₆ ,52 ₁₆ ,dd)	3 (3)	8 (8)
L(DIR),Y	ORA A,L(dd),Y	11 ₁₆ ,59 ₁₆ ,dd (91 ₁₆ ,59 ₁₆ ,dd)	3 (3)	9 (9)
SR	ORA A,nn,S	11 ₁₆ ,53 ₁₆ ,nn (91 ₁₆ ,53 ₁₆ ,nn)	3 (3)	5 (5)
(SR),Y	ORA A,(nn,S),Y	11 ₁₆ ,54 ₁₆ ,nn (91 ₁₆ ,54 ₁₆ ,nn)	3 (3)	8 (8)
ABS	ORA A,mml	5E ₁₆ ,ll,mm (81 ₁₆ ,5E ₁₆ ,ll,mm)	3 (4)	3 (4)
ABS,X	ORA A,mml,X	5F ₁₆ ,ll,mm (81 ₁₆ ,5F ₁₆ ,ll,mm)	3 (4)	4 (5)
ABS,Y	ORA A,mml,Y	11 ₁₆ ,56 ₁₆ ,ll,mm (91 ₁₆ ,56 ₁₆ ,ll,mm)	4 (4)	5 (5)
ABL	ORA A,hhmmll	11 ₁₆ ,5C ₁₆ ,ll,mm,hh (91 ₁₆ ,5C ₁₆ ,ll,mm,hh)	5 (5)	5 (5)
ABL,X	ORA A,hhmmll,X	11 ₁₆ ,5D ₁₆ ,ll,mm,hh (91 ₁₆ ,5D ₁₆ ,ll,mm,hh)	5 (5)	6 (6)

注1. この表はアキュムレータAを使用する場合について、書かれています。アキュムレータBを使用する場合は記述形式の“ A , ” の部分が“ B , ” になり、機械語、バイト数、サイクル数は () 内の値になります。

2. IMMアドレッシングモードで、mフラグが“0”の場合は、バイト数は1バイト増加します。

【記述例】

```
CLM
ORA.W      A,#IMM16          ; A  A  IMM16
ORA        B,MEM16          ; B  B  MEM16
SEM
ORA.B      A,#IMM8           ; AL AL  IMM8
ORA        B,MEM8            ; BL BL  MEM8
```

【機能】 論理和

【演算長】 8ビット

【動作】 $AccL \leftarrow AccL \text{ } IMM8$
 $AccL \leftarrow AccL \text{ } IMM8$

【説明】 AccLの内容とイミディエイト値の論理和を8ビット長でとり、結果をAccLへ格納します。
 この命令はmフラグの影響を受けません。
 AccHの内容は変化しません。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	N						Z	

N : 実行結果のMSBが“1”のとき、“1”になります。それ以外では“0”になります。

Z : 実行結果が“0”のとき、“1”になります。それ以外では“0”になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
IMM	ORAB A,#imm	63 ₁₆ ,imm	2	1
IMM	ORAB B,#imm	81 ₁₆ ,63 ₁₆ ,imm	3	2

【記述例】

ORAB A,#IMM8 ; AL AL IMM8
 ORAB B,#IMM8 ; BL BL IMM8

【機能】 論理和

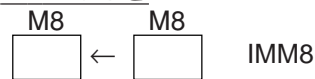
【演算長】 16ビット又は8ビット

【動作】 $M \leftarrow M \text{ OR } IMM$

・ m = “0” のとき



・ m = “1” のとき



【説明】 メモリの内容とイミディエイト値の論理和をとり、結果をメモリへ格納します。
従来の7700ファミリのSEB命令の機能はこの命令に含まれます。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	N						Z	

N : 実行結果のMSBが“1”のとき、“1”になります。それ以外では“0”になります。

Z : 実行結果が“0”のとき、“1”になります。それ以外では“0”になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
DIR	ORAM dd,#imm	51 ₁₆ ,33 ₁₆ ,dd,imm	4	7
ABS	ORAM mmll,#imm	51 ₁₆ ,37 ₁₆ ,ll,mm,imm	5	7

注 . mフラグが“0”の場合は、バイト数は1バイト増加します。

【記述例】

```
CLM
ORAM.W  MEM16,#IMM16          ; MEM16  MEM16  IMM16
SEM
ORAM.B  MEM8,#IMM8           ; MEM8   MEM8   IMM8
```

【機能】 論理和

【演算長】 8ビット

【動作】 $M8 \leftarrow M8 \text{ } \text{OR} \text{ } IMM8$

$\begin{matrix} M8 & & M8 \\ \square & \leftarrow & \square \end{matrix} \text{ } IMM8$

【説明】 メモリの内容とイミディエイト値の論理和を8ビット長でとり、結果をメモリへ格納します。
この命令はmフラグの影響を受けません。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	N						Z	

N : 実行結果のMSBが“1”のとき、“1”になります。それ以外では“0”になります。

Z : 実行結果が“0”のとき、“1”になります。それ以外では“0”になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
DIR	ORAMB dd,#imm	5 ₁₆ ,32 ₁₆ ,dd,imm	4	7
ABS	ORAMB mll,#imm	5 ₁₆ ,36 ₁₆ ,ll,mm,imm	5	7

【記述例】

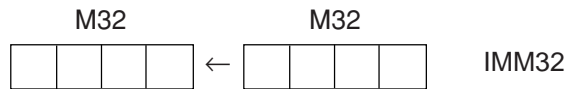
ORAMB MEM8,#IMM8 ; MEM8 MEM8 IMM8

【機能】 論理和

【演算長】 32ビット

「4.4 32ビット演算命令実行時の注意」を参照してください。

【動作】 $M32 \leftarrow M32 \text{ } IMM32$



【説明】 メモリの内容とイミディエイト値の論理和を32ビット長でとり、結果をメモリへ格納します。
この命令はmフラグの影響を受けません。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	N						Z	

N : 実行結果のMSBが“1”のとき、“1”になります。それ以外では“0”になります。

Z : 実行結果が“0”のとき、“1”になります。それ以外では“0”になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
DIR	ORAMD dd,#imm	51 ₁₆ ,B3 ₁₆ ,dd,immLL,immLH,immHL,immHH	7	10
ABS	ORAMD mml,#imm	51 ₁₆ ,B7 ₁₆ ,ll,mm,immLL,immLH,immHL,immHH	8	10

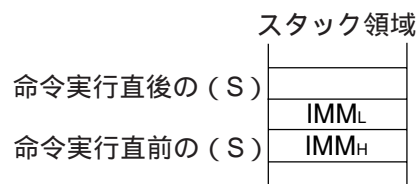
【記述例】

ORAMD MEM32,#IMM32 ; MEM32 MEM32 IMM32

【機能】 スタック操作（退避）

【演算長】 16ビット

【動作】 スタック ← IMM16



【説明】 16ビットのイミディエイト値をスタックへ退避します。
この命令はmフラグの影響を受けません。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
STK	PEA #imm _H imm _L	31 ₁₆ ,4C ₁₆ ,imm _L ,imm _H	4	5

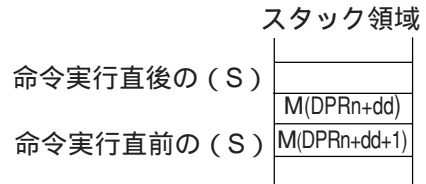
【記述例】

PEA #IMM16 ; (S) IMM16_H
 ; (S - 1) IMM16_L

【機能】 スタック操作（退避）

【演算長】 16ビット

【動作】 スタック ← M16(DPRn + dd) (n=0~3)



【説明】 DPRnの内容とオフセット値を加算した値が示す番地の内容を、16ビット長でスタックへ退避します。
 ダイレクトページレジスタ切り替えビットが“0”の場合
 DPR0の内容とオペランドddの値（8ビット）を加算した値をスタックへ退避します。
 ダイレクトページレジスタ切り替えビットが“1”の場合
 オペランドddの上位2ビットでDPRn（n=0~3）を指定し、指定されたDPRnの内容とオペランドddの下位6ビットの値を加算した値をスタックへ退避します。
 この命令はmフラグの影響を受けません。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
STK	PEI dd	31 ₁₆ ,4B ₁₆ ,dd	3	7

【記述例】

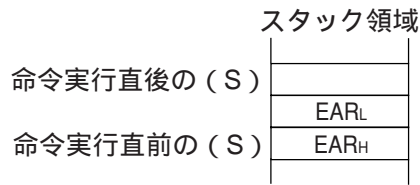
PEI DP0+: offset ; (S) (DPR0 + offset + 1)
 ; (S - 1) (DPR0 + offset)



【機能】 スタック操作（退避）

【演算長】 16ビット

【動作】 スタック ← PC + IMM16



EAR = PC + IMM16

【説明】 PCの内容と16ビットのイミディエイト値を加算した結果を、16ビット長でスタックへ退避します。
この命令はmフラグの影響を受けません。

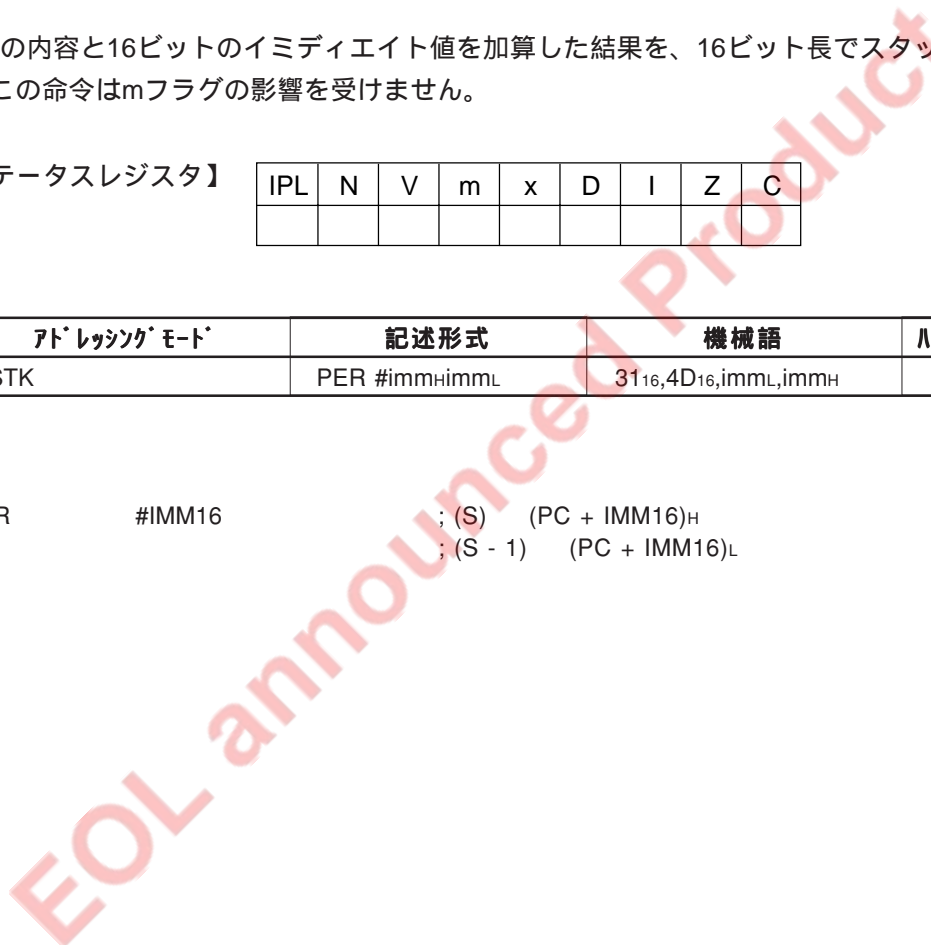
【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
STK	PER #immHimmL	31 ₁₆ ,4D ₁₆ ,immL,immH	4	6

【記述例】

PER #IMM16 ; (S) (PC + IMM16)_H
 ; (S - 1) (PC + IMM16)_L

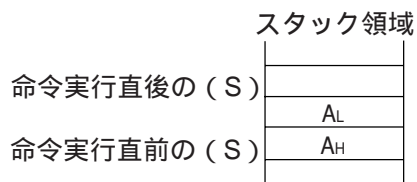


【機能】 スタック操作（退避）

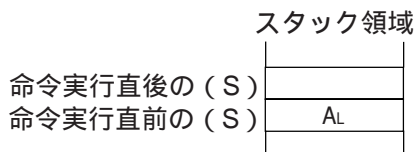
【演算長】 16ビット又は8ビット

【動作】 スタック ← A

・ m = “0” のとき



・ m = “1” のとき



【説明】 Aの内容をスタックへ退避します。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
STK	PHA	85 ₁₆	1	4

【記述例】

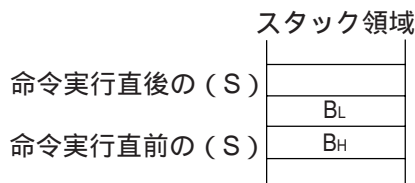
CLM
 PHA ; (S) AH, (S - 1) AL
 SEM
 PHA ; (S) AL

【機能】 スタック操作（退避）

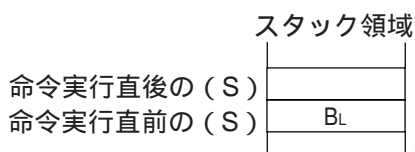
【演算長】 16ビット又は8ビット

【動作】 スタック ← B

・ m= “ 0 ” のとき



・ m= “ 1 ” のとき



【説明】 Bの内容をスタックへ退避します。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
STK	PHB	81 ₁₆ ,85 ₁₆	2	5

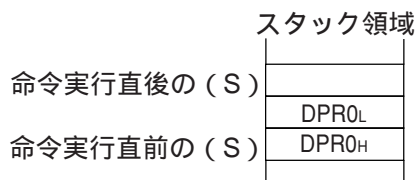
【記述例】

CLM
 PHB ; (S) BH, (S - 1) BL
 SEM
 PHB ; (S) BL

【機能】 スタック操作（退避）

【演算長】 16ビット

【動作】 スタック ← DPR0



【説明】 DPR0の内容を16ビット長でスタックへ退避します。
この命令はmフラグの影響を受けません。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
STK	PHD	83 ₁₆	1	4

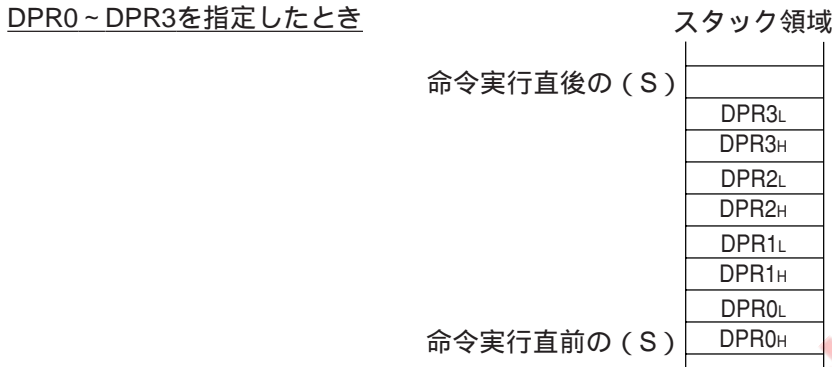
【記述例】

PHD ; (S, S - 1) DPR0

【機能】 スタック操作

【演算長】 16ビット

【動作】 スタック ← DPRn (n=0~3。複数のDPRをスタックに退避可)



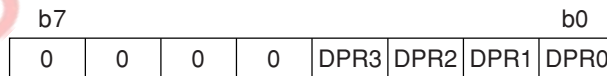
【説明】 指定されたDPRn (DPR0~DPR3) の内容を16ビット長でスタックへ退避します。
 複数のDPRの退避が一命令で可能です。複数のDPRを指定したとき、退避はDPR0、DPR1、DPR2、DPR3の順で行われます。
 この命令はmフラグの影響を受けません。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
STK	PHD n	B8 _{16,0} ? ₁₆	2	12
	PHD (n ₁ ,...,n _i)	B8 _{16,0} ? ₁₆	2	i + 11

- 注1. nには0~3の値が設定できます。
 2. 記述形式2行目は、複数のDPRの退避を一命令で行う場合です。
 3. (n₁,...,n_i)の括弧内は0~3 (DPRnの番号) を指定します。
 4. i : 指定するDPRnの数 (1~4) を示します。
 5. ? : 指定したDPRnに対応するビットが "1" になります。
 ビットとDPRnの対応は下図のとおりです。



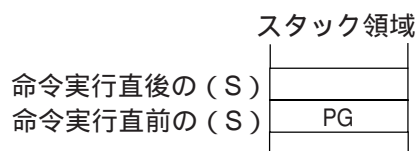
【記述例】

PHD 1 ; (S, S - 1) DPR1
 PHD (0,3) ; (S, S - 1) DPR0
 ; (S - 2, S - 3) DPR3

【機能】 スタック操作（退避）

【演算長】 8ビット

【動作】 スタック ← PG



【説明】 PGの内容を8ビット長でスタックへ退避します。
この命令はmフラグの影響を受けません。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C

アドレスモード	記述形式	機械語	バイト数	サイクル数
STK	PHG	31 ₁₆ ,60 ₁₆	2	4

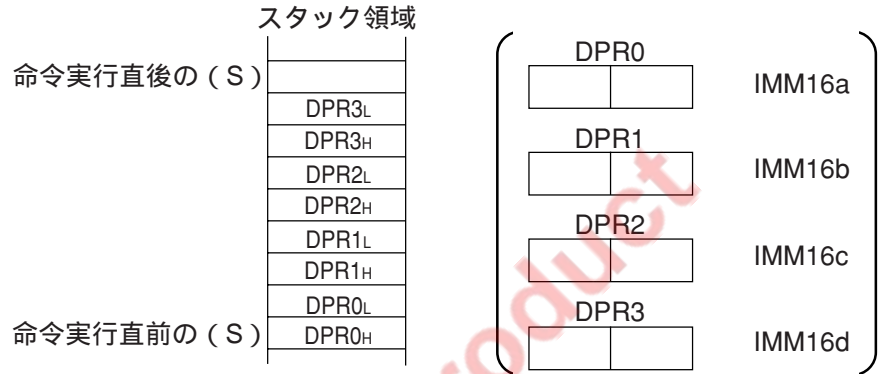
【記述例】

PHG ; (S) PG

【機能】 スタック操作&ロード

【演算長】 16ビット

【動作】 スタック ← DPRn (n=0~3。複数のDPRを指定可)
 DPRn ← IMM16
 DPR0~DPR3を指定したとき



【説明】 指定されたDPRn (DPR0~DPR3) の内容を16ビット長でスタックへ退避したのち、16ビット長のイミディエイト値をDPRnへ転送します。
 対象となるDPRは複数指定が可能です。複数のDPRを指定したとき、DPR0、DPR1、DPR2、DPR3の順で退避を行った後、同順でイミディエイト値をロードします。
 この命令はmフラグの影響を受けません。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
STK	PHLD n,#imm	B8 ₁₆ ,?? ₁₆ ,imm _L ,imm _H	4	14
	PHLD (n ₁ ,...,n _i),#imm ₁ ,...,#imm _i	B8 ₁₆ ,?? ₁₆ ,imm _{L1} ,imm _{H1} ,...,imm _{Li} ,imm _{Hi}	2 × i + 2	3 × i + 11

- 注1. nには0~3の値が設定できます。
- 2. 記述形式2行目は、複数のDPRの退避を一命令で行う場合です。
- 3. (n₁,...,n_i)の括弧内は0~3 (DPRnの番号) を指定します。
- 4. i : 指定するDPRnの数 (1~4) を示します。
- 5. ? : 指定したDPRnに対応するビットが“1”になります。
 ビットとDPRnの対応は下図のとおりです。



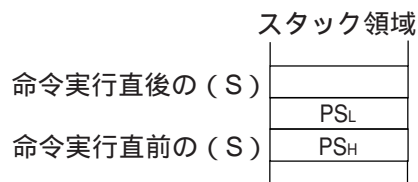
【記述例】

```
PHLD    0,#IMM16                ; (S, S - 1)  DPR0
                                ; DPR0  IMM16
PHLD    (0,3),#IMM16a,#IMM16b  ; (S, S - 1)  DPR0
                                ; (S - 2, S - 3)  DPR3
                                ; DPR0  IMM16a
                                ; DPR3  IMM16b
```

【機能】 スタック操作（退避）

【演算長】 16ビット

【動作】 スタック ← PS



【説明】 PSの内容を16ビット長でスタックへ退避します。
この命令はmフラグの影響を受けません。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
STK	PHP	A5 ₁₆	1	4

【記述例】

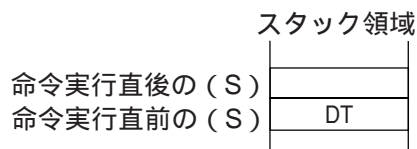
PHP ; (S, S - 1) PS



【機能】 スタック操作（退避）

【演算長】 8ビット

【動作】 スタック ← DT



【説明】 DTの内容を8ビット長でスタックへ退避します。
この命令はmフラグの影響を受けません。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
STK	PHT	31 ₁₆ ,40 ₁₆	2	4

【記述例】

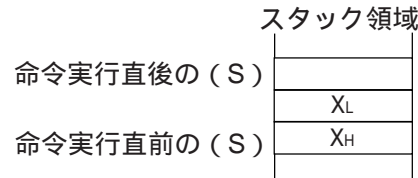
PHT ; (S) DT

【機能】 スタック操作（退避）

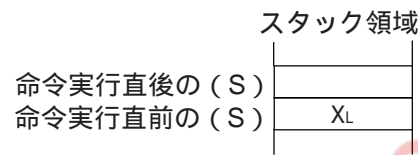
【演算長】 16ビット又は8ビット

【動作】 スタック ← X

・x=“0”のとき



・x=“1”のとき



【説明】 Xの内容をスタックへ退避します。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
STK	PHX	C516	1	4

【記述例】

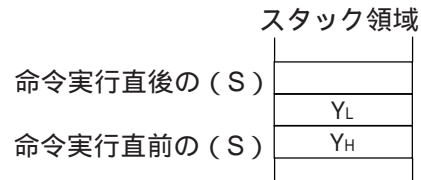
```
CLP      x
PHX                      ; (S, S - 1)  X
SEP      x
PHX                      ; (S)    XL
```

【機能】 スタック操作（退避）

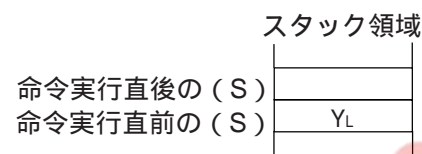
【演算長】 16ビット又は8ビット

【動作】 スタック ← Y

・ x=“0” のとき



・ x=“1” のとき



【説明】 Yの内容をスタックへ退避します。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C

アドレスモード	記述形式	機械語	ビット数	サイクル数
STK	PHY	E5 ₁₆	1	4

【記述例】

```

CLP          x
PHY                               ; (S, S - 1)  Y
SEP          x
PHY                               ; (S)    YL

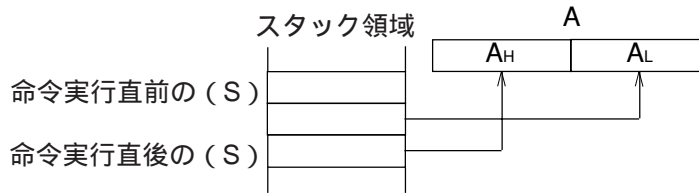
```

【機能】 スタック操作

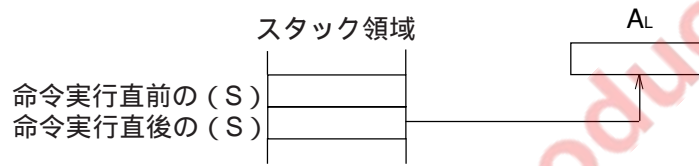
【演算長】 16ビット又は8ビット

【動作】 $A \leftarrow$ スタック

・ $m = "0"$ のとき



・ $m = "1"$ のとき



このときAHの内容は変化しません。

【説明】 スタックの内容をAへ復帰します。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	N						Z	

N : 実行結果のMSBが“1”のとき、“1”になります。それ以外では“0”になります。

Z : 実行結果が“0”のとき、“1”になります。それ以外では“0”になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
STK	PLA	95 ₁₆	1	4

【記述例】

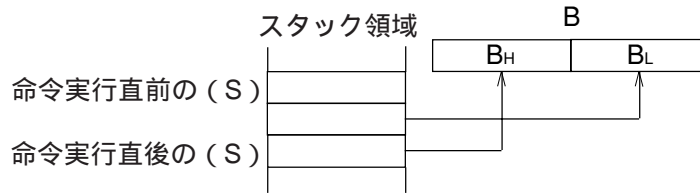
CLB
 PLA ; AL (S + 1), AH (S + 2)
 SEB
 PLA ; AL (S + 1)

【機能】 スタック操作

【演算長】 16ビット又は8ビット

【動作】 $B \leftarrow$ スタック

・ $m = "0"$ のとき



・ $m = "1"$ のとき



このときBHの内容は変化しません。

【説明】 スタックの内容をBへ復帰します。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	N						Z	

N : 実行結果のMSBが“1”のとき、“1”になります。それ以外では“0”になります。

Z : 実行結果が“0”のとき、“1”になります。それ以外では“0”になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
STK	PLB	81 ₁₆ ,95 ₁₆	2	5

【記述例】

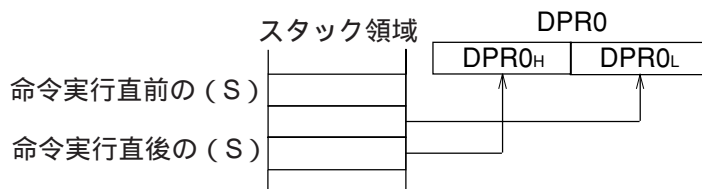
```

CLB
PLB                ; BL (S + 1) , BH (S + 2)
SEB
PLB                ; BL (S + 1)
    
```

【機能】 スタック操作

【演算長】 16ビット

【動作】 DPR0 ← スタック



【説明】 スタックの内容を16ビット長でDPR0へ復帰します。
この命令はmフラグの影響を受けません。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
STK	PLD	93 ₁₆	1	5

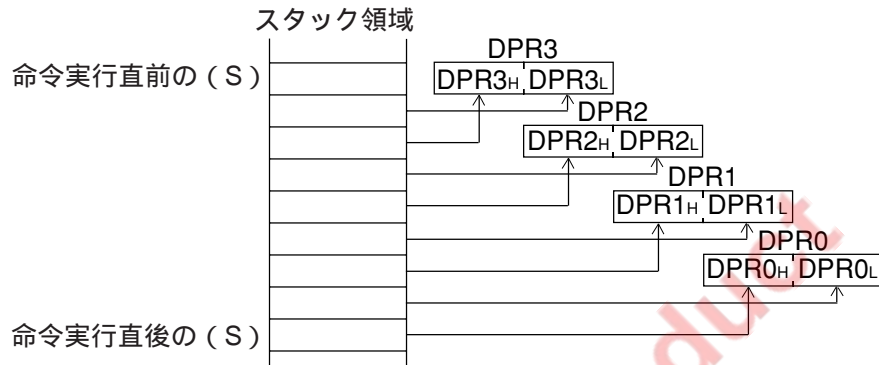
【記述例】

PLD ; DPR0L (S + 1)
; DPR0H (S + 2)

【機能】 スタック操作

【演算長】 16ビット

【動作】 $DPRn \leftarrow \text{スタック}$ (n=0~3。複数DPRの復帰可)
 ・ DPR0 ~ DPR3を指定したとき



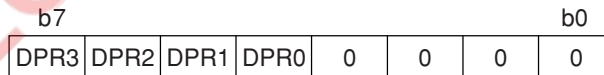
【説明】 指定されたDPRn (DPR0~DPR3) へスタックの内容を16ビット長で復帰します。
 複数のDPRへの復帰が一命令で可能です。複数のDPRを指定したとき、復帰はDPR3、DPR2、DPR1、DPR0の順で行われます。
 この命令はmフラグの影響を受けません。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
STK	PLD n	77 ₁₆ , ?0 ₁₆	2	11
	PLD (n ₁ , ..., n _i)	77 ₁₆ , ?0 ₁₆	2	3 × i + 8

- 注1. nには0~3の値が設定できます。
 2. 記述形式2行目は、複数のDPRの復帰を一命令で行う場合です。
 3. (n₁, ..., n_i)の括弧内は0~3 (DPRnの番号) を指定します。
 4. i : 指定するDPRnの数 (1~4) を示します。
 5. ? : 指定したDPRnに対応するビットが "1" になります。
 ビットとDPRnの対応は下図のとおりです。



【記述例】

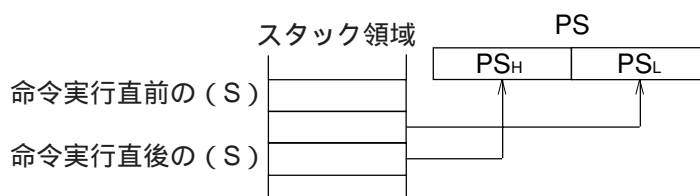
```

PLD      1           ; DPR1 (S + 1, S + 2)
PLD      (0,3)      ; DPR3 (S + 1, S + 2)
                    ; DPR0 (S + 3, S + 4)
    
```

【機能】 スタック操作

【演算長】 16ビット

【動作】 PS ← スタック



【説明】 スタックの内容を16ビット長でPSへ復帰します。
この命令はmフラグの影響を受けません。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
IPL	N	V	m	x	D	I	Z	C

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
STK	PLP	B5 ₁₆	1	5

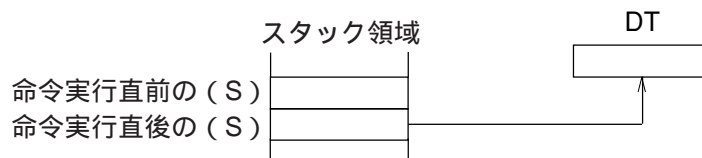
【記述例】

PLP ; PSL (S + 1)
; PSH (S + 2)

【機能】 スタック操作

【演算長】 8ビット

【動作】 DT ← スタック



【説明】 スタックの内容を8ビット長でDTへ復帰します。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	N						Z	

N : 実行結果のMSBが“1”のとき、“1”になります。それ以外では“0”になります。

Z : 実行結果が“0”のとき、“1”になります。それ以外では“0”になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
STK	PLT	31 ₁₆ ,50 ₁₆	2	6

【記述例】

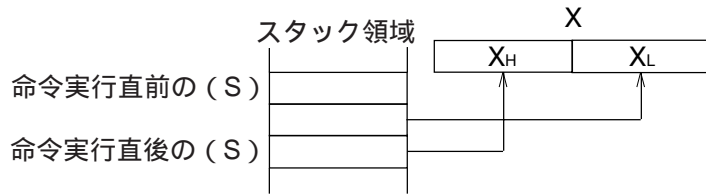
PLT ; DT (S + 1)

【機能】 スタック操作

【演算長】 16ビット又は8ビット

【動作】 $X \leftarrow$ スタック

・ $x = "0"$ のとき



・ $x = "1"$ のとき



このとき X_H の内容は変化しません。

【説明】 スタックの内容をXへ復帰します。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	N						Z	

N : 実行結果のMSBが“1”のとき、“1”になります。それ以外では“0”になります。

Z : 実行結果が“0”のとき、“1”になります。それ以外では“0”になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
STK	PLX	D5 ₁₆	1	4

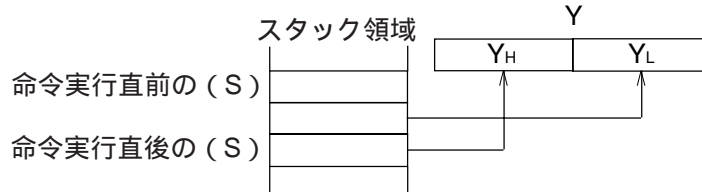
【記述例】

```
CLP      x
PLX                      ; XL (S + 1), XH (S + 2)
SEP      x
PLX                      ; XL (S + 1)
```

【機能】 スタック操作

【演算長】 16ビット又は8ビット

【動作】 $Y \leftarrow \text{スタック}$
 ・ $x = "0"$ のとき



・ $x = "1"$ のとき



このときY_Hの内容は変化しません。

【説明】 スタックの内容をYへ復帰します。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	N						Z	

- N : 実行結果のMSBが“1”のとき、“1”になります。それ以外では“0”になります。
- Z : 実行結果が“0”のとき、“1”になります。それ以外では“0”になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
STK	PLY	F5 ₁₆	1	4

【記述例】

```
CLP      x
PLY                      ; YL (S + 1), YH (S + 2)
SEP      x
PLY                      ; YL (S + 1)
```

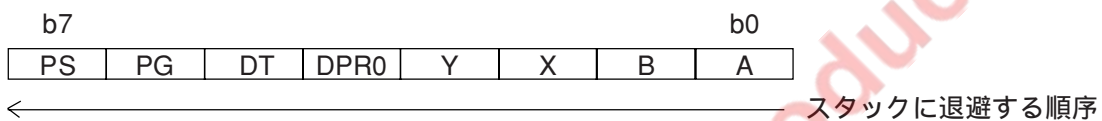
【機能】 スタック操作

【演算長】 16ビット又は8ビット

【動作】 スタック ← A, B, X, Y, DPR0, DT, PG, PS中の指定レジスタ（複数指定可）
 $M(S \sim S - i + 1) \leftarrow A, B, X, Y, DPR0, DT, PG, PS$
 $S \leftarrow S - i$

i : スタックに退避するレジスタに応じたバイト数。

【説明】 指定したレジスタの内容をスタックへ退避します。8ビットのイミディエイト値のビットパターンで退避するレジスタを示します。“1”にセットしたビットに対応するレジスタの内容がスタックへ退避されます。



「m=0」のとき：A、Bが16ビット長で退避されます。

「m=1」のとき：A_L、B_Lが8ビット長で退避されます。

「x=0」のとき：X、Yが16ビット長で退避されます。

「x=1」のとき：X_L、Y_Lが8ビット長で退避されます。

PS、PG、DT、DPR0の内容の退避に関しては、m及びxフラグの影響を受けません。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C

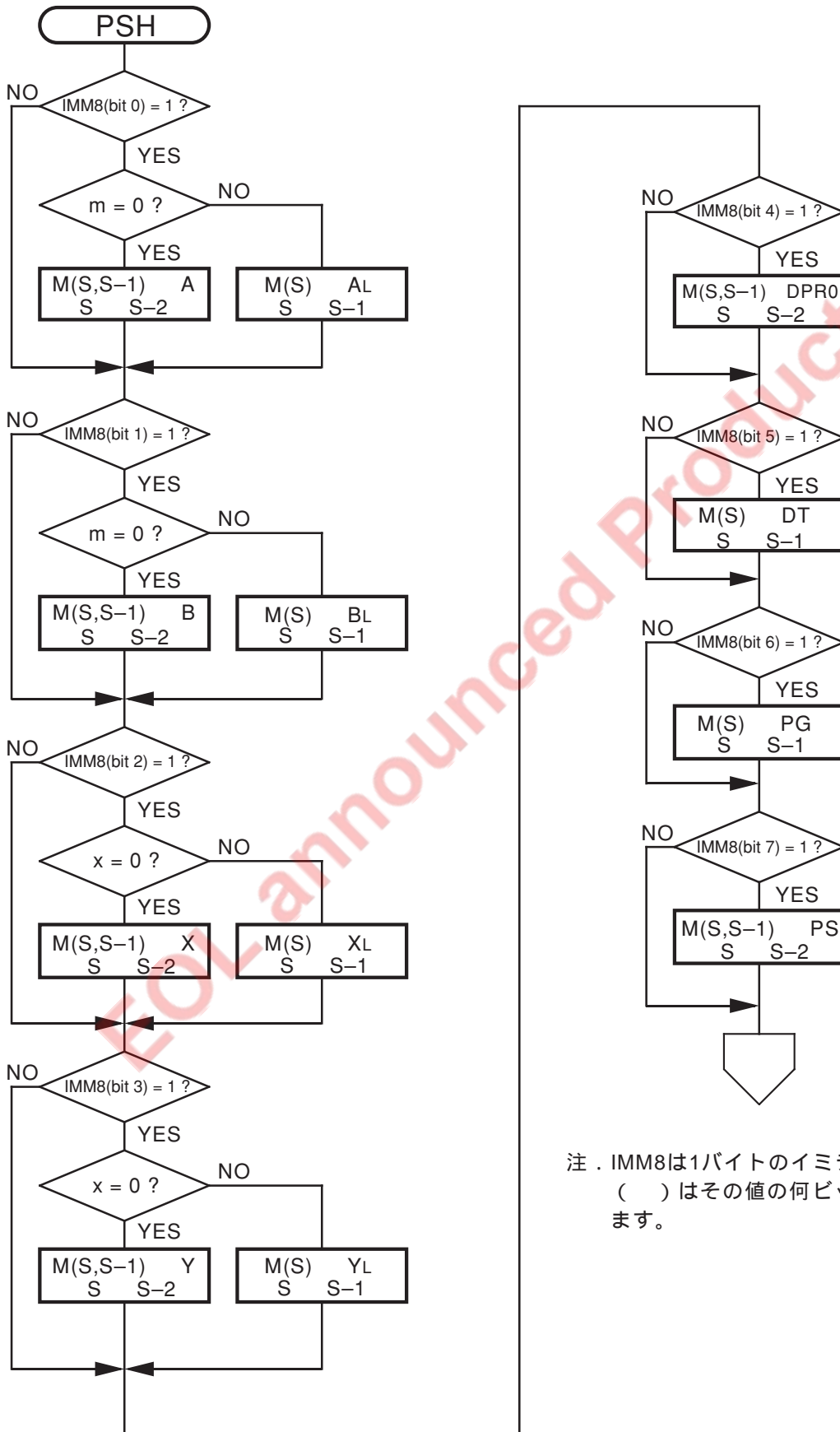
アドレッシングモード	記述形式	機械語	バイト数	サイクル数
STK	PSH #imm	A8 ₁₆ ,imm	2	2×i ₁ +i ₂ +11

注 . i₁ : A、B、X、Y、DPR0、PSのうち退避するレジスタの個数を示します。

i₂ : DT、PGのうち退避するレジスタの個数を示します。

【記述例】

PSH #IMM8 ; (S) 指定レジスタの内容



注 . IMM8は1バイトのイミディエイト値で、
()はその値の何ビット目かを示し
ます。

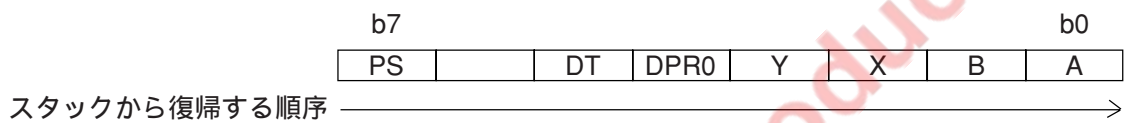
【機能】 スタック操作

【演算長】 16ビット又は8ビット

【動作】 A, B, X, Y, DPR0, DT, PS中の指定レジスタ (複数指定可) ← スタック
 A, B, X, Y, DPR0, DT, PS ← M(S + 1 ~ S + i)
 S ← S + i

i : スタックから復帰するレジスタに応じたバイト数。

【説明】 指定したレジスタにスタックの内容を復帰します。8ビットのイミディエイト値のビットパターンで復帰するレジスタを示します。“1”にセットしたビットに対応するレジスタにスタックの内容を復帰します。



- 「復帰したPS中のm=0」のとき : A、Bへ16ビット長で復帰されます。
- 「復帰したPS中のm=1」のとき : AL、BLへ8ビット長で復帰されます。
このときAH、BHの内容は変化しません。
- 「復帰したPS中のx=0」のとき : X、Yへ16ビット長で復帰されます。
- 「復帰したPS中のx=1」のとき : XL、YLへ8ビット長で復帰されます。
このときXH、YHの内容は変化しません。

【プロセッサステータスレジスタ】

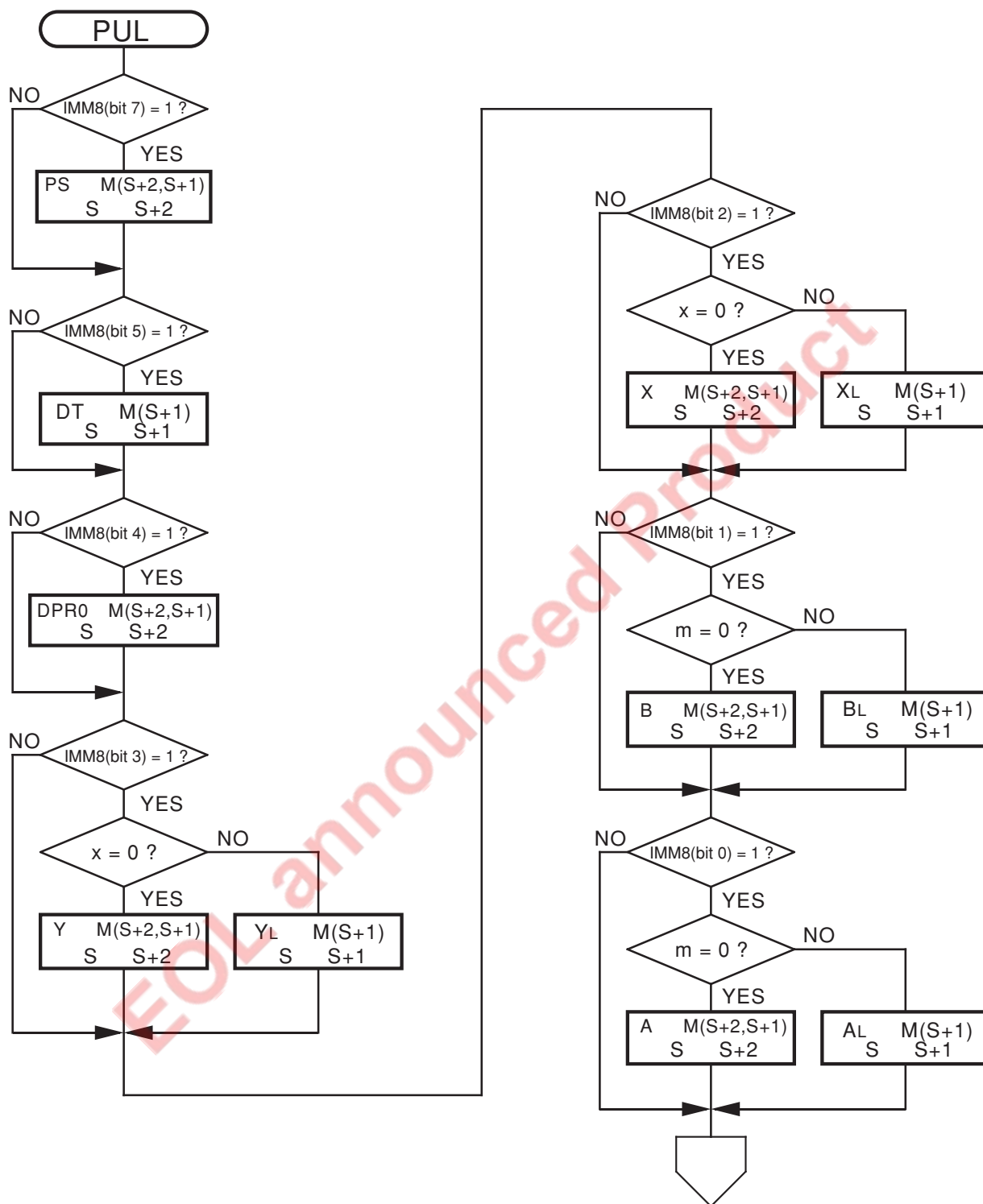
IPL	N	V	m	x	D	I	Z	C
IPL	N	V	m	x	D	I	Z	C

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
STK	PUL #imm	67 ₁₆ ,imm	2	3 × i + 13

注 . i : 復帰するレジスタの個数を示します。

【記述例】

PUL #IMM8 ; 指定レジスタの内容 (S + 1)

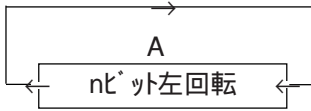


注 . IMM8は1バイトのイミディエイト値で、
 ()内はその値の何ビット目かを示します。

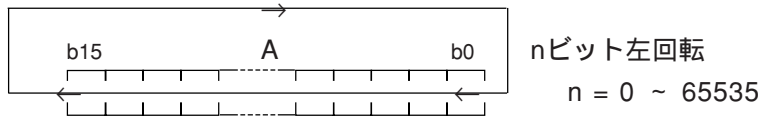
【機能】 左回転

【演算長】 16ビット又は8ビット

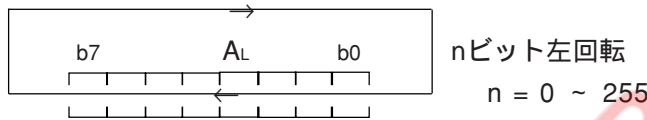
【動作】



・ m = “ 0 ” のとき



・ m = “ 1 ” のとき



このときAHの内容は変化しません。

【説明】 Aの内容を左へnビット回転します。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
IMM	RLA #imm	31 ₁₆ ,07 ₁₆ ,imm	3	n + 5

注1. n : immで指定する回転数を示します。

2. mフラグが“ 0 ”の場合は、バイト数は1バイト増加します。

【記述例】

CLM
 RLA #IMM16 ; A Aを左へIMM16で指定した回数だけ回転
 SEM
 RLA #IMM8 ; AL ALを左へIMM8で指定した回数だけ回転

【機能】 繰り返し積和

【演算長】 16ビット又は8ビット

【動作】 (B, A) (B, A) + M (DT:X) × M (DT:Y) (0 ~ 255回繰り返し)

【説明】 DTで示されるバンク内のXとYの内容で指定される番地の内容を符号付きで乗算します。その結果とB、Aの内容を加算します。加算結果はB、Aに格納され、X、Yの内容はそれぞれ加算されます。命令の3バイト目の8ビットのイミディエイト値で指定された回数(0 ~ 255回)だけこの処理を繰り返します。

「m=0」のとき：16ビット長で演算し、結果は32ビットになります。

$$E \quad E + M16 (DT:X) \times M16 (DT:Y)$$

また、加算後、X、Yの内容はそれぞれ2インクリメントされます。

「m=1」のとき：8ビット長で演算し、結果は16ビットになります。

$$(B_L, A_L) \quad (B_L, A_L) + M8 (DT:X) \times M8 (DT:Y)$$

このときA_H、B_Hの内容は変化しません。

また、加算後、X、Yの内容はそれぞれ1インクリメントされます。

実行後のX、Yの内容：最後に読み出した被乗数の次の番地、及び乗数の次の番地になります。

また、「x=1」のときX_H、Y_Hの内容は変化しません。

加算時にオーバーフローが発生した場合は、Vフラグを“1”にして実行を途中で終了します。

このときA、Bの内容は不定になります。X、Yの内容は最後に読み出した被乗数、及び乗数の次の番地を示します。

繰り返しを0に指定した場合は、演算を実行せずに終了します。このときA、B、X、Yの内容は変化しません。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	N	V					Z	C

N：加算の度にチェックされ、加算結果のMSB (BのMSB) が“1”のとき、“1”になります。それ以外では“0”になります。

V：加算の度にチェックされ、加算結果が+2147483647、又は-2147483648 (「m=1」のときは+32767又は-32768) を越えると“1”になります。それ以外では“0”になります。命令終了時、「V=0」であれば正常終了、「V=1」であればオーバーフローが判断できます。

Z：加算の度にチェックされ、加算結果が“0”のとき、“1”になります。それ以外では“0”になります。

C：加算の度にチェックされ、加算結果が符号無しデータと見なして+4294967295 (「m=1」のとき+65536) を越えたとき、“1”になります。それ以外では“0”になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
繰り返し積和	RMPA #imm	31 ₁₆ , 5A ₁₆ , imm	3	14 × imm + 5

注1. imm：繰り返し回数を示します。

2. この表でのサイクル数はm=1の場合です。m=0の場合は18 × imm + 5サイクルになります。

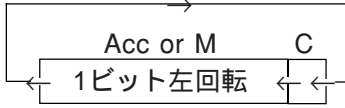
【記述例】

RMPA #IMM8 ; 演算をIMM8回繰り返す

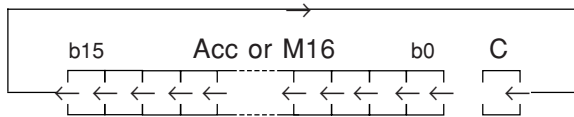
【機能】 左回転

【演算長】 16ビット又は8ビット

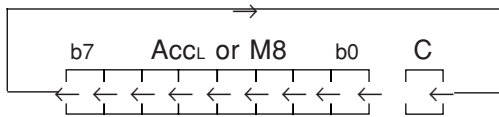
【動作】



・ m=“0” のとき



・ m=“1” のとき



このときAccHの内容は変化しません。

【説明】 Acc又はメモリをCフラグとつなぎ、その内容を左へ1ビット回転します。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	N						Z	C

N : 実行結果のMSBが“1”のとき、“1”になります。それ以外では“0”になります。

Z : 実行結果が“0”のとき、“1”になります。それ以外では“0”になります。

C : 実行前のMSBが“1”のとき、“1”になります。それ以外では“0”になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
A	ROL A	13 ₁₆	1	1
A	ROL B	81 ₁₆ ,13 ₁₆	2	2
DIR	ROL dd	21 ₁₆ ,1A ₁₆ ,dd	3	7
DIR,X	ROL dd,X	21 ₁₆ ,1B ₁₆ ,dd	3	8
ABS	ROL mmlI	21 ₁₆ ,1E ₁₆ ,Il,mm	4	7
ABS,X	ROL mmlI,X	21 ₁₆ ,1F ₁₆ ,Il,mm	4	8

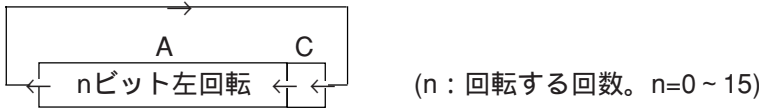
【記述例】

```
CLM
ROL    A                ; Aを左へ1ビット回転
ROL    MEM16           ; MEM16を左へ1ビット回転
SEM
ROL    B                ; BLを左へ1ビット回転
ROL    MEM8            ; MEM8を左へ1ビット回転
```

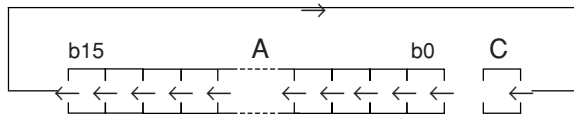
【機能】 左回転

【演算長】 16ビット又は8ビット

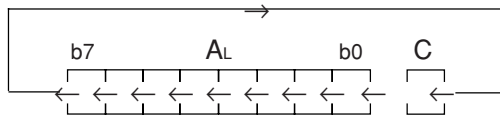
【動作】



・ m=“0” のとき



・ m=“1” のとき



このときAHの内容は変化しません。

【説明】 AとCフラグをつなぎ、その内容を左へnビット回転します。
この命令では、Bは使用できません。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	N						Z	C

- N : 実行結果のMSBが“1”のとき、“1”になります。それ以外では“0”になります。
- Z : 実行結果が“0”のとき、“1”になります。それ以外では“0”になります。
- C : (n-1)ビット回転したときのMSBが“1”のとき、“1”になります。それ以外では“0”になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
A	ROL A,#imm	C1 ₁₆ ,imm+60 ₁₆	2	imm + 6

注 . immには0~15の値(回転する回数)が設定できます。

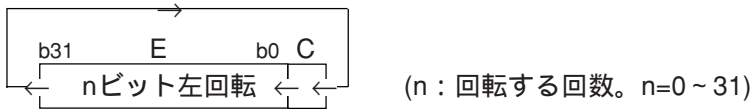
【記述例】

```
CLM
ROL    A,#15          ; A   AをCを含めて左へ15ビット回転
SEM
ROL    A,#7           ; AL  ALをCを含めて左へ7ビット回転
```

【機能】 左回転

【演算長】 32ビット

【動作】



【説明】 EとCフラグをつなぎ、その内容を32ビット長で左へnビット回転します。
この命令はmフラグの影響を受けません。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	N						Z	C

- N : 実行結果のMSBが“1”のとき、“1”になります。それ以外では“0”になります。
- Z : 実行結果が“0”のとき、“1”になります。それ以外では“0”になります。
- C : (n-1)ビット回転したときのMSBが“1”のとき、“1”になります。それ以外では“0”になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
A	ROLD E,#imm	D1 ₁₆ ,imm+60 ₁₆	2	imm + 8

注 . immには0 ~ 31の値 (回転する回数) が設定できます。

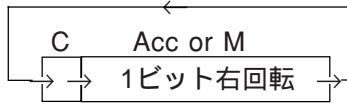
【記述例】

ROLD E,#16 ; E EをCを含めて左へ16ビット回転

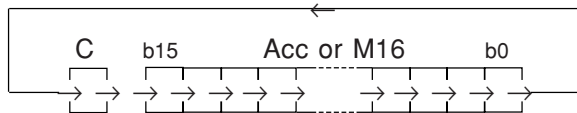
【機能】 右回転

【演算長】 16ビット又は8ビット

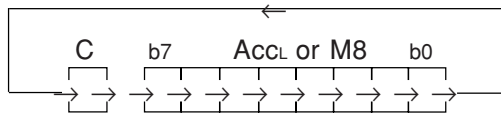
【動作】



・ m= " 0 " のとき



・ m= " 1 " のとき



このときAccHの内容は変化しません。

【説明】 Acc又はメモリをCフラグとつなぎ、その内容を右へ1ビット回転します。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	N						Z	C

- N : 実行結果のMSBが " 1 " のとき、 " 1 " になります。それ以外では " 0 " になります。
- Z : 実行結果が " 0 " のとき、 " 1 " になります。それ以外では " 0 " になります。
- C : 実行前のLSBが " 1 " のとき、 " 1 " になります。それ以外では " 0 " になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
A	ROR A	53 ₁₆	1	1
A	ROR B	81 ₁₆ , 53 ₁₆	2	2
DIR	ROR dd	21 ₁₆ , 3A ₁₆ , dd	3	7
DIR,X	ROR dd,X	21 ₁₆ , 3B ₁₆ , dd	3	8
ABS	ROR mml	21 ₁₆ , 3E ₁₆ , ll, mm	4	7
ABS,X	ROR mml,X	21 ₁₆ , 3F ₁₆ , ll, mm	4	8

【記述例】

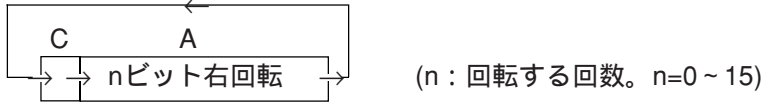
```

CLM
ROR    A                ; Aを右へ1ビット回転
ROR    MEM16           ; MEM16を右へ1ビット回転
SEM
ROR    B                ; BLを右へ1ビット回転
ROR    MEM8            ; MEM8を右へ1ビット回転
    
```

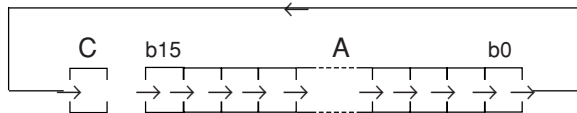
【機能】 右回転

【演算長】 16ビット又は8ビット

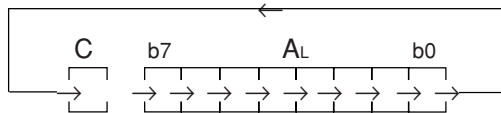
【動作】



・ m = "0" のとき



・ m = "1" のとき



このときAHの内容は変化しません。

【説明】 AとCフラグをつなぎ、その内容を右へnビット回転します。
この命令ではBは使用できません。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	N						Z	C

N : 実行結果のMSBが"1"のとき、"1"になります。それ以外では"0"になります。

Z : 実行結果が"0"のとき、"1"になります。それ以外では"0"になります。

C : (n-1)ビット回転したときのLSBが"1"のとき、"1"になります。それ以外では"0"になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
A	ROR A,#imm	C1 ₁₆ ,imm+20 ₁₆	2	imm + 6

注 . immには0~15の値(回転する回数)が設定できます。

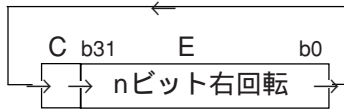
【記述例】

```
CLM
ROR    A,#15    ; A   AをCを含めて右へ15ビット回転
SEM
ROR    A,#7     ; AL  ALをCを含めて右へ7ビット回転
```

【機能】 右回転

【演算長】 32ビット

【動作】



(n : 回転する回数。n=0 ~ 31)

【説明】 EとCフラグをつなぎ、その内容を32ビット長で右へnビット回転します。
この命令はmフラグの影響を受けません。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	N						Z	C

N : 実行結果のMSBが“1”のとき、“1”になります。それ以外では“0”になります。

Z : 実行結果が“0”のとき、“1”になります。それ以外では“0”になります。

C : (n-1)ビット回転したときのLSBが“1”のとき、“1”になります。それ以外では“0”になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
A	RORD E,#imm	D1 ₁₆ ,imm+20 ₁₆	2	imm + 8

注 . immには0 ~ 31の値 (回転する回数) が設定できます。

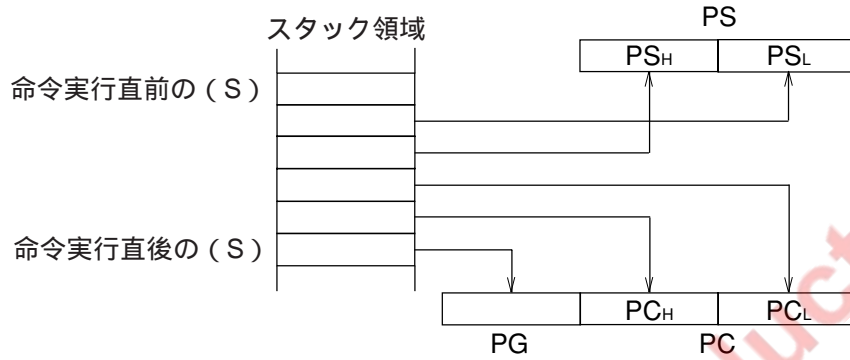
【記述例】

RORD E,#16 ; E EをCを含めて右へ16ビット回転

【機能】 リターン

【演算長】 -

【動作】 PG, PC, PS ← スタック



【説明】 スタックの内容をPS、PC、PGの順に復帰します。
割り込みルーチンからの復帰に使用します。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
IPL	N	V	m	x	D	I	Z	C

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
IMP	RTI	F1 ₁₆	1	12

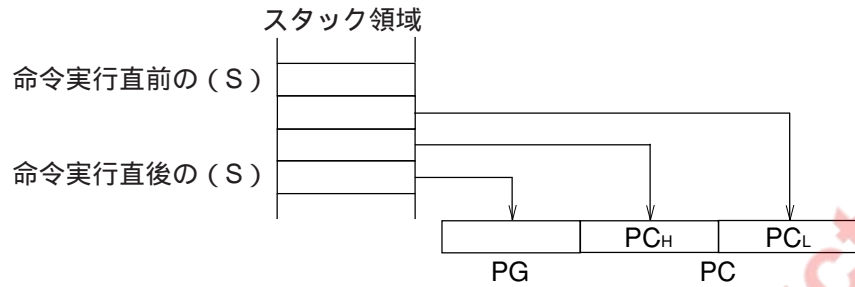
【記述例】

```
RTI                ; PS (S + 2, S + 1)
                   ; PC (S + 4, S + 3)
                   ; PG (S + 5)
```


【機能】 リターン

【演算長】 -

【動作】 PG, PC ← スタック



【説明】 スタックの内容をPC、PGの順に復帰します。
JSRLで呼び出したサブルーチンからの復帰に使用します。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
IMP	RTL	94 ₁₆	1	10

【記述例】

RTL ; PC (S + 2, S + 1)
; PG (S + 3)

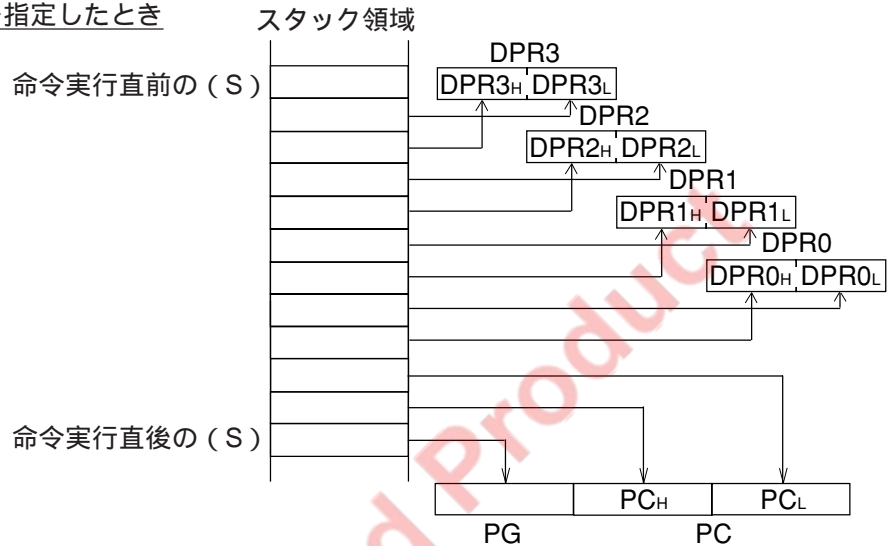
【機能】 ロード&リターン

【演算長】 16ビット

【動作】 DPRn ← スタック (n=0~3.複数指定可)

PG, PC ← スタック

・DPR0 ~ DPR3を指定したとき



【説明】 指定されたDPRn (DPR0~DPR3) の内容を16ビット長でスタックから復帰した後、RTL命令を実行 (スタックの内容をPC、PGの順で復帰) します。

復帰するDPRは複数指定が可能です。複数のDPRを指定したとき、復帰はDPR3、DPR2、DPR1、DPR0の順で行われます。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
STK	RTL D n	77 ₁₆ , ?C ₁₆	2	15
	RTL D (n ₁ , ..., n _i)	77 ₁₆ , ?C ₁₆	2	3 × i + 12

- 注1. nには0~3の値が設定できます。
 2. 記述形式2行目は、複数のDPRの指定を一命令で行う場合です。
 3. (n₁, ..., n_i)の括弧内は0~3 (DPRnの番号) を指定します。
 4. i : 指定するDPRnの数 (1~4) を示します。
 5. ? : 指定したDPRnに対応するビットが“1”になります。
 ビットとDPRnの対応は下図のとおりです。

b7				b0			
DPR3	DPR2	DPR1	DPR0	1	1	0	0

【記述例】

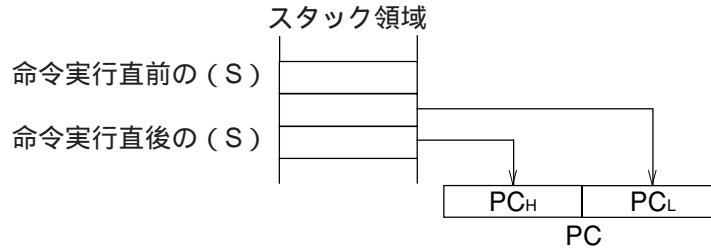
```
RTL D      1                ; DPR1  (S + 1)
                    ; RTL

RTL D      (0,3)           ; DPR3  (S + 1)
                    ; DPR0  (S + 3)
                    ; RTL
```

【機能】 リターン

【演算長】 -

【動作】 PC ← スタック



【説明】 スタックの内容をPCへ復帰します。

JSR、BSR命令で呼び出したサブルーチンからの復帰に使用します。

この命令をバンクの最上位 (XFFFFF₁₆) 番地に配置すると、PGの内容が1加算されます。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
IMP	RTS	84 ₁₆	1	7

【記述例】

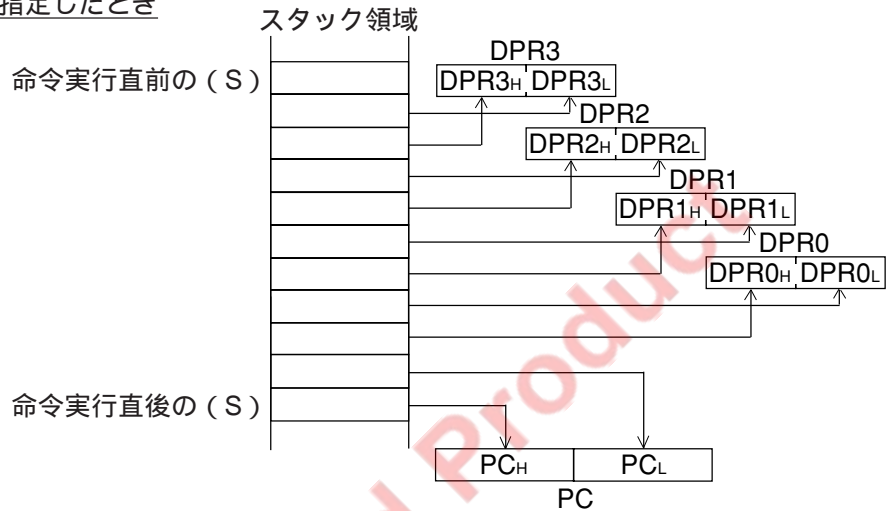
RTS ; PC (S + 2, S + 1)

EOL announced Product

【機能】 ロード&リターン

【演算長】 16ビット

【動作】 DPRn ← スタック (n=0~3。複数指定可)
 PC ← スタック
 ・DPR0 ~ DPR3を指定したとき



【説明】 指定されたDPRn (DPR0~DPR3) の内容を16ビット長でスタックから復帰した後、RTS命令を実行 (スタックの内容をPCに復帰) します。

復帰するDPRは複数指定が可能です。複数のDPRを指定したとき、復帰はDPR3、DPR2、DPR1、DPR0の順で行われます。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
STK	RTSD n	77 ₁₆ , ?8 ₁₆	2	14
	RTSD (n ₁ , ..., n _i)	77 ₁₆ , ?8 ₁₆	2	3 × i + 11

- 注1. nには0~3の値が設定できます。
- 2. 記述形式2行目は、複数のDPRの指定を一命令で行う場合です。
- 3. (n₁, ..., n_i)の括弧内は0~3 (DPRnの番号) を指定します。
- 4. i : 指定するDPRnの数 (1~4) を示します。
- 5. ? : 指定したDPRnに対応するビットが“1”になります。
 ビットとDPRnの対応は下図のとおりです。

b7				b0			
DPR3	DPR2	DPR1	DPR0	1	0	0	0

【記述例】

```
RTSD    1                ; DPR1 (S + 1)
                ; RTS

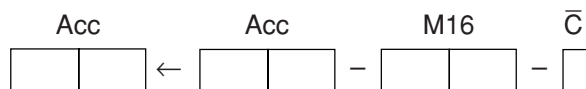
RTSD    (0,3)           ; DPR3 (S + 1)
                ; DPR0 (S + 3)
                ; RTS
```

【機能】 減算（キャリー付き）

【演算長】 16ビット又は8ビット

【動作】 $Acc \leftarrow Acc - M - \bar{C}$

・ $m = "0"$ のとき



・ $m = "1"$ のとき



このときAcc_Hの内容は変化しません。

【説明】 Accの内容からメモリの内容とCフラグの補数を減算し、その結果をAccへ格納します。
「D=1」のとき10進演算を行います。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	N	V					Z	C

- N : 実行結果のMSBが“1”のとき、“1”になります。それ以外では“0”になります。
「D=1」のときは、意味を持ちません。
- V : 符号付き演算と見なした結果が+32767、又は-32768（「m=1」のときは+127又は-128）を越えると“1”になります。それ以外では“0”になります。
「D=1」のときは、意味を持ちません。
- Z : 実行結果が“0”のとき、“1”になります。それ以外では“0”になります。
「D=1」のときは、意味を持ちません。
- C : ボローが発生すると、“0”になります。それ以外では“1”になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
IMM	SBC A,#imm	31 ₁₆ ,A7 ₁₆ ,imm (B1 ₁₆ ,A7 ₁₆ ,imm)	3	3 (3)
DIR	SBC A,dd	21 ₁₆ ,AA ₁₆ ,dd (A1 ₁₆ ,AA ₁₆ ,dd)	3	5 (7)
DIR,X	SBC A,dd,X	21 ₁₆ ,AB ₁₆ ,dd (A1 ₁₆ ,AB ₁₆ ,dd)	3	6 (8)
(DIR)	SBC A,(dd)	21 ₁₆ ,A0 ₁₆ ,dd (A1 ₁₆ ,A0 ₁₆ ,dd)	3	7 (9)
(DIR,X)	SBC A,(dd,X)	21 ₁₆ ,A1 ₁₆ ,dd (A1 ₁₆ ,A1 ₁₆ ,dd)	3	8 (10)
(DIR),Y	SBC A,(dd),Y	21 ₁₆ ,A8 ₁₆ ,dd (A1 ₁₆ ,A8 ₁₆ ,dd)	3	8 (10)
L(DIR)	SBC A,L(dd)	21 ₁₆ ,A2 ₁₆ ,dd (A1 ₁₆ ,A2 ₁₆ ,dd)	3	9 (11)
L(DIR),Y	SBC A,L(dd),Y	21 ₁₆ ,A9 ₁₆ ,dd (A1 ₁₆ ,A9 ₁₆ ,dd)	3	10(12)
SR	SBC A,nn,S	21 ₁₆ ,A3 ₁₆ ,nn (A1 ₁₆ ,A3 ₁₆ ,nn)	3	6 (8)
(SR),Y	SBC A,(nn,S),Y	21 ₁₆ ,A4 ₁₆ ,nn (A1 ₁₆ ,A4 ₁₆ ,nn)	3	9 (11)
ABS	SBC A,mml	21 ₁₆ ,AE ₁₆ ,ll,mm (A1 ₁₆ ,AE ₁₆ ,ll,mm)	4	5 (7)
ABS,X	SBC A,mml,X	21 ₁₆ ,AF ₁₆ ,ll,mm (A1 ₁₆ ,AF ₁₆ ,ll,mm)	4	6 (8)
ABS,Y	SBC A,mml,Y	21 ₁₆ ,A6 ₁₆ ,ll,mm (A1 ₁₆ ,A6 ₁₆ ,ll,mm)	4	6 (8)
ABL	SBC A,hhmml	21 ₁₆ ,AC ₁₆ ,ll,mm,hh (A1 ₁₆ ,AC ₁₆ ,ll,mm,hh)	5	6 (8)
ABL,X	SBC A,hhmml,X	21 ₁₆ ,AD ₁₆ ,ll,mm,hh (A1 ₁₆ ,AD ₁₆ ,ll,mm,hh)	5	7 (9)

注1. この表はアキュムレータAを使用する場合について、書かれています。アキュムレータBを使用する場合は記述形式の“A,”の部分が“B,”になり、機械語、及びサイクル数は()内の値になります。

2. IMMアドレッシングモードで、mフラグが“0”の場合は、バイト数は1バイト増加します。

【記述例】

CLM
 SBC.W A,#IMM16 ; A A - IMM16 - \bar{C}
 SBC B,MEM16 ; B B - MEM16 - \bar{C}
 SEB
 SBC.B A,#IMM8 ; AL AL - IMM8 - \bar{C}
 SBC B,MEM8 ; BL BL - MEM8 - \bar{C}

【機能】 減算（キャリー付き）

【演算長】 8ビット

【動作】 $AcCL \leftarrow AcCL - IMM8 - \bar{C}$

$$\begin{array}{c} AcCL \\ \square \end{array} \leftarrow \begin{array}{c} AcCL \\ \square \end{array} - IMM8 - \begin{array}{c} \bar{C} \\ \square \end{array}$$

【説明】 $AcCL$ の内容からイミディエイト値とCフラグの補数を8ビット長で減算し、その結果を $AcCL$ へ格納します。
この命令はmフラグの影響を受けません。

$AcCH$ の内容は変化しません。

「D=1」のとき10進演算を行います。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	N	V					Z	C

N : 実行結果のMSBが“1”のとき、“1”になります。それ以外では“0”になります。
「D=1」のときは、意味を持ちません。

V : 符号付き演算と見なした結果が+127、又は-128を越えると“1”になります。
それ以外では、“0”になります。
「D=1」のときは、意味を持ちません。

Z : 実行結果が“0”のとき、“1”になります。それ以外では“0”になります。
「D=1」のときは、意味を持ちません。

C : ボローが発生すると、“0”になります。それ以外では“1”になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
IMM	SBCB A,#imm	31 ₁₆ ,1B ₁₆ ,imm	3	3
IMM	SBCB B,#imm	B1 ₁₆ ,1B ₁₆ ,imm	3	3

【記述例】

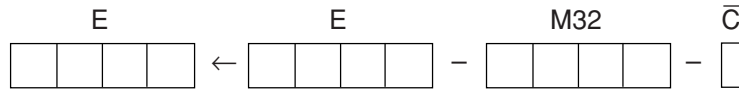
SBCB A,#IMM8 ; AL AL - IMM8 - \bar{C}
SBCB B,#IMM8 ; BL BL - IMM8 - \bar{C}

【機能】 減算（キャリー付き）

【演算長】 32ビット

「4.4 32ビット演算命令実行時の注意」を参照してください。

【動作】 $E \leftarrow E - M32 - \bar{C}$



【説明】 Eの内容からメモリの内容とCフラグの補数を32ビット長で減算し、その結果をEへ格納します。CPUはDフラグの内容にかかわらず、2進演算を行います。

この命令はmフラグの影響を受けません。

10進演算を行うことはできません。この命令を使用する際には「D=0」にしてください。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	N	V					Z	C

- N : 実行結果のMSBが“1”のとき、“1”になります。それ以外では“0”になります。
- V : 符号付き演算と見なした結果が+2147483647、又は-2147483648を越えると“1”になります。それ以外では、“0”になります。
- Z : 実行結果が“0”のとき、“1”になります。それ以外では“0”になります。
- C : ボローが発生すると、“0”になります。それ以外では“1”になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
IMM	SBCD E,#imm	31 ₁₆ ,1D ₁₆ ,immLL,immLH,immHL,immHH	6	4
DIR	SBCD E,dd	21 ₁₆ ,BA ₁₆ ,dd	3	7
DIR,X	SBCD E,dd,X	21 ₁₆ ,BB ₁₆ ,dd	3	8
(DIR)	SBCD E,(dd)	21 ₁₆ ,B0 ₁₆ ,dd	3	9
(DIR,X)	SBCD E,(dd,X)	21 ₁₆ ,B1 ₁₆ ,dd	3	10
(DIR),Y	SBCD E,(dd),Y	21 ₁₆ ,B8 ₁₆ ,dd	3	10
L(DIR)	SBCD E,L(dd)	21 ₁₆ ,B2 ₁₆ ,dd	3	11
L(DIR),Y	SBCD E,L(dd),Y	21 ₁₆ ,B9 ₁₆ ,dd	3	12
SR	SBCD E,nn,S	21 ₁₆ ,B3 ₁₆ ,nn	3	8
(SR),Y	SBCD E,(nn,S),Y	21 ₁₆ ,B4 ₁₆ ,nn	3	11
ABS	SBCD E,mml	21 ₁₆ ,BE ₁₆ ,ll,mm	4	7
ABS,X	SBCD E,mml,X	21 ₁₆ ,BF ₁₆ ,ll,mm	4	8
ABS,Y	SBCD E,mml,Y	21 ₁₆ ,B6 ₁₆ ,ll,mm	4	8
ABL	SBCD E,hhmml	21 ₁₆ ,BC ₁₆ ,ll,mm,hh	5	8
ABL,X	SBCD E,hhmml,X	21 ₁₆ ,BD ₁₆ ,ll,mm,hh	5	9

【記述例】

SBCD E,#IMM32 ; E E - IMM32 - \bar{C} (B,A B,A - IMM32 - \bar{C})
 SBCD E,MEM32 ; E E - MEM32 - \bar{C} (B,A B,A - MEM32 - \bar{C})

【機能】 フラグ操作

【演算長】 -

【動作】 $C \leftarrow 1$

【説明】 Cフラグの内容を“1”にします。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
								1

C : “1”になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
IMP	SEC	04 ₁₆	1	1

【記述例】

SEC ; C 1

【機能】 フラグ操作

【演算長】 -

【動作】 $I \leftarrow 1$

【説明】 Iフラグの内容を“1”にします。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
						1		

I : “1”になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
IMP	SEI	05 ₁₆	1	4

【記述例】

SEI ; I 1

【機能】 フラグ操作

【演算長】 -

【動作】 $m \leftarrow 1$

【説明】 mフラグの内容を“1”にします。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
			1					

m : “1”になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
IMP	SEM	25 ₁₆	1	3

【記述例】

SEM ; m 1

【機能】 フラグ操作

【演算長】 -

【動作】 $PSL(\text{bit } n) \leftarrow 1$ ($n=0 \sim 7$ 。複数ビット指定可)

【説明】 PSL中の指定されたフラグ（複数指定可）を“1”にします。フラグの指定（PSLのビット位置）は8ビットのイミディエイト値で行います。“1”にセットしたビットが対象のフラグとなります。
この命令はmフラグの影響を受けません。

PSL							
b7	b6	b5	b4	b3	b2	b1	b0
N	V	m	x	D	I	Z	C

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	N	V	m	x	D	I	Z	C

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
IMM	SEP #imm	99 ₁₆ ,imm	2	3

【記述例】

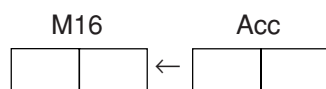
SEP #IMM8 ; 指定したPSL中のビット 1

【機能】 ストア

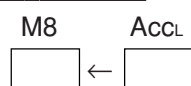
【演算長】 16ビット又は8ビット

【動作】 $M \leftarrow \text{Acc}$

・ $m = "0"$ のとき



・ $m = "1"$ のとき



【説明】 Accの内容をメモリへ格納します。Accの内容は変化しません。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
DIR	STA A,dd	DA ₁₆ ,dd (81 ₁₆ ,DA ₁₆ ,dd)	2 (3)	4 (5)
DIR,X	STA A,dd,X	DB ₁₆ ,dd (81 ₁₆ ,DB ₁₆ ,dd)	2 (3)	5 (6)
(DIR)	STA A,(dd)	11 ₁₆ ,D0 ₁₆ ,dd (91 ₁₆ ,D0 ₁₆ ,dd)	3 (3)	7 (7)
(DIR),X	STA A,(dd),X	11 ₁₆ ,D1 ₁₆ ,dd (91 ₁₆ ,D1 ₁₆ ,dd)	3 (3)	8 (8)
(DIR),Y	STA A,(dd),Y	D8 ₁₆ ,dd (81 ₁₆ ,D8 ₁₆ ,dd)	2 (3)	7 (8)
L(DIR)	STA A,L(dd)	11 ₁₆ ,D2 ₁₆ ,dd (91 ₁₆ ,D2 ₁₆ ,dd)	3 (3)	9 (9)
L(DIR),Y	STA A,L(dd),Y	D9 ₁₆ ,dd (81 ₁₆ ,D9 ₁₆ ,dd)	2 (3)	9 (10)
SR	STA A,nn,S	11 ₁₆ ,D3 ₁₆ ,nn (91 ₁₆ ,D3 ₁₆ ,nn)	3 (3)	6 (6)
(SR),Y	STA A,(nn,S),Y	11 ₁₆ ,D4 ₁₆ ,nn (91 ₁₆ ,D4 ₁₆ ,nn)	3 (3)	9 (9)
ABS	STA A,mml	DE ₁₆ ,ll,mm (81 ₁₆ ,DE ₁₆ ,ll,mm)	3 (4)	4 (5)
ABS,X	STA A,mml,X	DF ₁₆ ,ll,mm (81 ₁₆ ,DF ₁₆ ,ll,mm)	3 (4)	5 (6)
ABS,Y	STA A,mml,Y	11 ₁₆ ,D6 ₁₆ ,ll,mm (91 ₁₆ ,D6 ₁₆ ,ll,mm)	4 (4)	6 (6)
ABL	STA A,hhmml	DC ₁₆ ,ll,mm,hh (81 ₁₆ ,DC ₁₆ ,ll,mm,hh)	4 (5)	5 (6)
ABL,X	STA A,hhmml,X	DD ₁₆ ,ll,mm,hh (81 ₁₆ ,DD ₁₆ ,ll,mm,hh)	4 (5)	6 (7)

注．この表はアキュムレータAを使用する場合について、書かれています。アキュムレータBを使用する場合は記述形式の“A,”の部分“A,”になり、機械語、バイト数、サイクル数は()内の値になります。

【記述例】

```
CLM
STA      A,MEM16          ; MEM16  A
SEM
STA      B,MEM8           ; MEM8   BL
```

【機能】 ストア

【演算長】 8ビット

【動作】 $M8 \leftarrow \text{AccL}$



【説明】 AccLの内容を8ビット長でメモリへ格納します。

Acc (AccH, AccL)の内容は変化しません。

この命令はmフラグの影響を受けません。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
DIR	STAB A,dd	CA ₁₆ ,dd (81 ₁₆ ,CA ₁₆ ,dd)	2 (3)	4 (5)
DIR,X	STAB A,dd,X	CB ₁₆ ,dd (81 ₁₆ ,CB ₁₆ ,dd)	2 (3)	5 (6)
(DIR)	STAB A,(dd)	11 ₁₆ ,C0 ₁₆ ,dd (91 ₁₆ ,C0 ₁₆ ,dd)	3 (3)	7 (7)
(DIR,X)	STAB A,(dd),X	11 ₁₆ ,C1 ₁₆ ,dd (91 ₁₆ ,C1 ₁₆ ,dd)	3 (3)	8 (8)
(DIR),Y	STAB A,(dd),Y	C8 ₁₆ ,dd (81 ₁₆ ,C8 ₁₆ ,dd)	2 (3)	7 (8)
L(DIR)	STAB A,L(dd)	11 ₁₆ ,C2 ₁₆ ,dd (91 ₁₆ ,C2 ₁₆ ,dd)	3 (3)	9 (9)
L(DIR),Y	STAB A,L(dd),Y	C9 ₁₆ ,dd (81 ₁₆ ,C9 ₁₆ ,dd)	2 (3)	9 (10)
SR	STAB A,nn,S	11 ₁₆ ,C3 ₁₆ ,nn (91 ₁₆ ,C3 ₁₆ ,nn)	3 (3)	6 (6)
(SR),Y	STAB A,(nn),S,Y	11 ₁₆ ,C4 ₁₆ ,nn (91 ₁₆ ,C4 ₁₆ ,nn)	3 (3)	9 (9)
ABS	STAB A,mml	CE ₁₆ ,ll,mm (81 ₁₆ ,CE ₁₆ ,ll,mm)	3 (4)	4 (5)
ABS,X	STAB A,mml,X	CF ₁₆ ,ll,mm (81 ₁₆ ,CF ₁₆ ,ll,mm)	3 (4)	5 (6)
ABS,Y	STAB A,mml,Y	11 ₁₆ ,C6 ₁₆ ,ll,mm (91 ₁₆ ,C6 ₁₆ ,ll,mm)	4 (4)	6 (6)
ABL	STAB A,hhmml	CC ₁₆ ,ll,mm,hh (81 ₁₆ ,CC ₁₆ ,ll,mm,hh)	4 (5)	5 (6)
ABL,X	STAB A,hhmml,X	CD ₁₆ ,ll,mm,hh (81 ₁₆ ,CD ₁₆ ,ll,mm,hh)	4 (5)	6 (7)

注．この表はアキュムレータAを使用する場合について、書かれています。アキュムレータBを使用する場合は記述形式の“ A, ”の部分“ B, ”になり、機械語、バイト数、サイクル数は()内に示す値になります。

【記述例】

STAB A,MEM8 ; MEM8 AL

【機能】 ストア

【演算長】 32ビット

【動作】 $M32 \leftarrow E$



【説明】 Eの内容を32ビット長でメモリへ格納します。

Eの内容は変化しません。

この命令はmフラグの影響を受けません。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
DIR	STAD E,dd	EA ₁₆ ,dd	2	6
DIR,X	STAD E,dd,X	EB ₁₆ ,dd	2	7
(DIR)	STAD E,(dd)	11 ₁₆ ,E0 ₁₆ ,dd	3	9
(DIR,X)	STAD E,(dd,X)	11 ₁₆ ,E1 ₁₆ ,dd	3	10
(DIR),Y	STAD E,(dd),Y	E8 ₁₆ ,dd	2	9
L(DIR)	STAD E,L(dd)	11 ₁₆ ,E2 ₁₆ ,dd	3	11
L(DIR),Y	STAD E,L(dd),Y	E9 ₁₆ ,dd	2	11
SR	STAD E,nn,S	11 ₁₆ ,E3 ₁₆ ,nn	3	8
(SR),Y	STAD E,(nn,S),Y	11 ₁₆ ,E4 ₁₆ ,nn	3	11
ABS	STAD E,mml	EE ₁₆ ,ll,mm	3	6
ABS,X	STAD E,mml,X	EF ₁₆ ,ll,mm	3	7
ABS,Y	STAD E,mml,Y	11 ₁₆ ,E6 ₁₆ ,ll,mm	4	8
ABL	STAD E,hhmml	EC ₁₆ ,ll,mm,hh	4	7
ABL,X	STAD E,hhmml,X	ED ₁₆ ,ll,mm,hh	4	8

【記述例】

STAD E,MEM32 ; MEM32 E (MEM32H B, MEM32L A)

【機能】 特殊

【演算長】 -

【動作】 発振を停止します。

【説明】 発振制御フリップフロップをリセットし、発振回路の発振を停止します。
再起動するためには、割り込み要求が発生するか、又はハードウェアリセットすることでストップ状態が解除されます。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
IMP	STP	31 ₁₆ ,30 ₁₆	2	

【記述例】

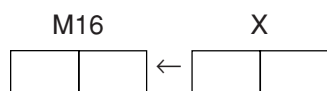
STP ;

【機能】 ストア

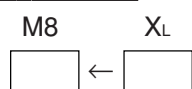
【演算長】 16ビット又は8ビット

【動作】 $M \leftarrow X$

・x=“0”のとき



・x=“1”のとき



【説明】 Xの内容をメモリへ格納します。Xの内容は変化しません。

この命令はmフラグの影響を受けません。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
DIR	STX dd	E2 ₁₆ ,dd	2	4
DIR,Y	STX dd,Y	41 ₁₆ ,E5 ₁₆ ,dd	3	6
ABS	STX mml	E7 ₁₆ ,ll,mm	3	4

【記述例】

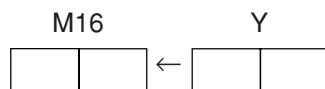
```
CLP      x
STX      MEM16          ; MEM16  X
SEP      x
STX      MEM8           ; MEM8   XL
```

【機能】 ストア

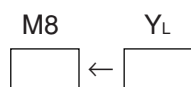
【演算長】 16ビット又は8ビット

【動作】 $M \leftarrow Y$

・x=“0”のとき



・x=“1”のとき



【説明】 Yの内容をメモリへ格納します。Yの内容は変化しません。
この命令はmフラグの影響を受けません。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
DIR	STY dd	F2 ₁₆ ,dd	2	4
DIR,X	STY dd,X	41 ₁₆ ,FB ₁₆ ,dd	3	6
ABS	STY mml	F7 ₁₆ ,ll,mm	3	4

【記述例】

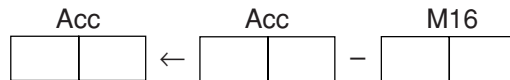
```
CLP      x
STY      MEM16          ; MEM16  Y
SEP      x
STY      MEM8           ; MEM8   YL
```

【機能】 減算

【演算長】 16ビット又は8ビット

【動作】 $Acc \leftarrow Acc - M$

・ $m = "0"$ のとき



・ $m = "1"$ のとき



このときAccHの内容は変化しません。

【説明】 Accの内容からメモリの内容を減算し、その結果をAccへ格納します。

10進演算を行うことはできません。この命令を使用する際には「D=0」にしてください。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	N	V					Z	C

N : 実行結果のMSBが“1”のとき、“1”になります。それ以外では“0”になります。

V : 符号付き演算と見なした結果が+32767、又は-32768（「m=1」のときは+127又は-128）を越えると“1”になります。それ以外では“0”になります。

Z : 実行結果が“0”のとき、“1”になります。それ以外では“0”になります。

C : ボローが発生すると、“0”になります。それ以外では“1”になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
IMM	SUB A,#imm	36 ₁₆ ,imm (81 ₁₆ ,36 ₁₆ ,imm)	2 (3)	1 (2)
DIR	SUB A,dd	3A ₁₆ ,dd (81 ₁₆ ,3A ₁₆ ,dd)	2 (3)	3 (4)
DIR,X	SUB A,dd,X	3B ₁₆ ,dd (81 ₁₆ ,3B ₁₆ ,dd)	2 (3)	4 (5)
(DIR)	SUB A,(dd)	11 ₁₆ ,30 ₁₆ ,dd (91 ₁₆ ,30 ₁₆ ,dd)	3 (3)	6 (6)
(DIR,X)	SUB A,(dd,X)	11 ₁₆ ,31 ₁₆ ,dd (91 ₁₆ ,31 ₁₆ ,dd)	3 (3)	7 (7)
(DIR),Y	SUB A,(dd),Y	11 ₁₆ ,38 ₁₆ ,dd (91 ₁₆ ,38 ₁₆ ,dd)	3 (3)	7 (7)
L(DIR)	SUB A,L(dd)	11 ₁₆ ,32 ₁₆ ,dd (91 ₁₆ ,32 ₁₆ ,dd)	3 (3)	8 (8)
L(DIR),Y	SUB A,L(dd),Y	11 ₁₆ ,39 ₁₆ ,dd (91 ₁₆ ,39 ₁₆ ,dd)	3 (3)	9 (9)
SR	SUB A,nn,S	11 ₁₆ ,33 ₁₆ ,nn (91 ₁₆ ,33 ₁₆ ,nn)	3 (3)	5 (5)
(SR),Y	SUB A,(nn,S),Y	11 ₁₆ ,34 ₁₆ ,nn (91 ₁₆ ,34 ₁₆ ,nn)	3 (3)	8 (8)
ABS	SUB A,mml	3E ₁₆ ,ll,mm (81 ₁₆ ,3E ₁₆ ,ll,mm)	3 (4)	3 (4)
ABS,X	SUB A,mml,X	3F ₁₆ ,ll,mm (81 ₁₆ ,3F ₁₆ ,ll,mm)	3 (4)	4 (5)
ABS,Y	SUB A,mml,Y	11 ₁₆ ,36 ₁₆ ,ll,mm (91 ₁₆ ,36 ₁₆ ,ll,mm)	4 (4)	5 (5)
ABL	SUB A,hhmml	11 ₁₆ ,3C ₁₆ ,ll,mm,hh (91 ₁₆ ,3C ₁₆ ,ll,mm,hh)	5 (5)	5 (5)
ABL,X	SUB A,hhmml,X	11 ₁₆ ,3D ₁₆ ,ll,mm,hh (91 ₁₆ ,3D ₁₆ ,ll,mm,hh)	5 (5)	6 (6)

注1. この表はアキュムレータAを使用する場合について、書かれています。アキュムレータBを使用する場合は記述形式の“A,”の部分で“B,”になり、機械語、バイト数、サイクル数は()内に示す値になります。

2. IMMアドレッシングモードで、mフラグが“0”の場合は、バイト数は1バイト増加します。

【記述例】

```
CLM
SUB.W   A,#IMM16           ; A   A - IMM16
SUB     B,MEM16           ; B   B - MEM16
SEM
SUB.B   A,#IMM8           ; AL  AL - IMM8
SUB     B,MEM8           ; BL  BL - MEM8
```

【機能】 減算

【演算長】 8ビット

【動作】 $AcCL \leftarrow AcCL - IMM8$



【説明】 AcCLの内容からイミディエイト値を8ビット長で減算し、その結果をAcCLへ格納します。

この命令はmフラグの影響を受けません。

AcCHの内容は変化しません。

10進演算を行うことはできません。この命令を使用する際には「D=0」にしてください。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	N	V					Z	C

N : 実行結果のMSBが“1”のとき、“1”になります。それ以外では“0”になります。

V : 符号付き演算と見なした結果が+127、又は-128を越えると“1”になります。それ以外では“0”になります。

Z : 実行結果が“0”のとき、“1”になります。それ以外では、“0”になります。

C : ボローが発生すると、“0”になります。それ以外では“1”になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
IMM	SUBB A,#imm	39 ₁₆ ,imm	2	1
IMM	SUBB B,#imm	81 ₁₆ ,39 ₁₆ ,imm	3	2

【記述例】

SUBB A,#IMM8 ; AL AL - IMM8
 SUBB B,#IMM8 ; BL BL - IMM8

【機能】 減算

【演算長】 32ビット

「4.4 32ビット演算命令実行時の注意」を参照してください。

【動作】 $E \leftarrow E - M32$



【説明】 Eの内容からメモリの内容を32ビット長で減算し、その結果をEへ格納します。

この命令はmフラグの影響を受けません。

10進演算を行うことはできません。この命令を使用する際には「D=0」にしてください。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	N	V					Z	C

N : 実行結果のMSBが“1”のとき、“1”になります。それ以外では“0”になります。

V : 符号付き演算と見なした結果が+2147483647、又は-2147483648を越えると“1”になります。それ以外では“0”になります。

Z : 実行結果が“0”のとき、“1”になります。それ以外では“0”になります。

C : ボローが発生すると、“0”になります。それ以外では“1”になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
IMM	SUBD E,#imm	3D ₁₆ ,immLL,immLH,immHL,immHH	5	3
DIR	SUBD E,dd	AA ₁₆ ,dd	2	6
DIR,X	SUBD E,dd,X	AB ₁₆ ,dd	2	7
(DIR)	SUBD E,(dd)	11 ₁₆ ,A0 ₁₆ ,dd	3	9
(DIR,X)	SUBD E,(dd,X)	11 ₁₆ ,A1 ₁₆ ,dd	3	10
(DIR),Y	SUBD E,(dd),Y	11 ₁₆ ,A8 ₁₆ ,dd	3	10
L(DIR)	SUBD E,L(dd)	11 ₁₆ ,A2 ₁₆ ,dd	3	11
L(DIR),Y	SUBD E,L(dd),Y	11 ₁₆ ,A9 ₁₆ ,dd	3	12
SR	SUBD E,nn,S	11 ₁₆ ,A3 ₁₆ ,nn	3	8
(SR),Y	SUBD E,(nn,S),Y	11 ₁₆ ,A4 ₁₆ ,nn	3	11
ABS	SUBD E,mml	AE ₁₆ ,ll,mm	3	6
ABS,X	SUBD E,mml,X	AF ₁₆ ,ll,mm	3	7
ABS,Y	SUBD E,mml,Y	11 ₁₆ ,A6 ₁₆ ,ll,mm	4	8
ABL	SUBD E,hhmmll	11 ₁₆ ,AC ₁₆ ,ll,mm,hh	5	8
ABL,X	SUBD E,hhmmll,X	11 ₁₆ ,AD ₁₆ ,ll,mm,hh	5	9

【記述例】

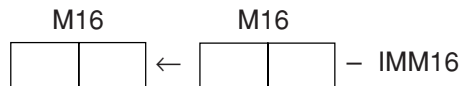
SUBD E,#IMM32 ; E E - IMM32 (B,A B,A - IMM32)
 SUBD E,MEM32 ; E E - MEM32 (B,A B,A - MEM32)

【機能】 減算

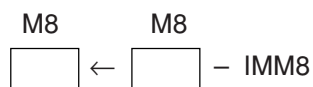
【演算長】 16ビット又は8ビット

【動作】 $M \leftarrow M - IMM$

・ m=“0”のとき



・ m=“1”のとき



【説明】 メモリの内容からイミディエイト値を減算し、その結果をメモリへ格納します。
10進演算を行うことはできません。この命令を使用する際には「D=0」にしてください。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	N	V					Z	C

- N : 実行結果のMSBが“1”のとき、“1”になります。それ以外では“0”になります。
V : 符号付き演算と見なした結果が+32767、又は-32768（「m=1」のときは+127又は-128）を越えると“1”になります。それ以外では“0”になります。
Z : 実行結果が“0”のとき、“1”になります。それ以外では“0”になります。
C : ボローが発生すると、“0”になります。それ以外では“1”になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
DIR	SUBM dd,#imm	51 ₁₆ ,13 ₁₆ ,dd,imm	4	7
ABS	SUBM mmlI,#imm	51 ₁₆ ,17 ₁₆ ,ll,mm,imm	5	7

注．mフラグが“0”の場合は、バイト数は1バイト増加します。

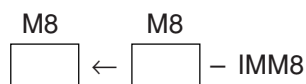
【記述例】

CLM
SUBM.W MEM16,#IMM16 ; MEM16 MEM16 - IMM16
SEM
SUBM.B MEM8,#IMM8 ; MEM8 MEM8 - IMM8

【機能】 減算

【演算長】 8ビット

【動作】 $M8 \leftarrow M8 - IMM8$



【説明】 メモリの内容からイミディエイト値を8ビット長で減算し、その結果をメモリへ格納します。
この命令はmフラグの影響を受けません。

10進演算を行うことはできません。この命令を使用する際には「D=0」にしてください。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	N	V					Z	C

N : 実行結果のMSBが“1”のとき、“1”になります。それ以外では“0”になります。

V : 符号付き演算と見なした結果が+127、又は-128を越えると“1”になります。
それ以外では“0”になります。

Z : 実行結果が“0”のとき、“1”になります。それ以外では“0”になります。

C : ボローが発生すると、“0”になります。それ以外では“1”になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
DIR	SUBMB dd,#imm	5 ₁₆ ,12 ₁₆ ,dd,imm	4	7
ABS	SUBMB mmll,#imm	5 ₁₆ ,16 ₁₆ ,ll,mm,imm	5	7

【記述例】

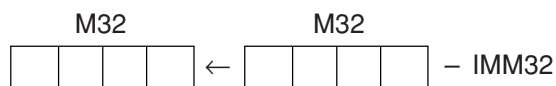
SUBMB MEM8,#IMM8 ; MEM8 MEM8 - IMM8

【機能】 減算

【演算長】 32ビット

「4.4 32ビット演算命令実行時の注意」を参照してください。

【動作】 $M32 \leftarrow M32 - IMM32$



【説明】 メモリの内容からイミディエイト値を32ビット長で減算し、その結果をメモリへ格納します。
この命令はmフラグの影響を受けません。
10進演算を行うことはできません。この命令を使用する際には「D=0」にしてください。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	N	V					Z	C

- N : 実行結果のMSBが“1”のとき、“1”になります。それ以外では“0”になります。
V : 符号付き演算と見なした結果が+2147483647、又は-2147483648を越えると“1”になります。それ以外では“0”になります。
Z : 実行結果が“0”のとき、“1”になります。それ以外では“0”になります。
C : ボローが発生すると、“0”になります。それ以外では“1”になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
DIR	SUBMD dd,#imm	51 ₁₆ ,93 ₁₆ ,dd,immLL,immLH,immHL,immHH	7	10
ABS	SUBMD mml,#imm	51 ₁₆ ,97 ₁₆ ,ll,mm,immLL,immLH,immHL,immHH	8	10

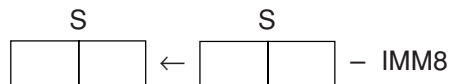
【記述例】

SUBMD MEM32,#IMM32 ; MEM32 MEM32 - IMM32

【機能】 減算

【演算長】 16ビット

【動作】 $S \leftarrow S - IMM8$



【説明】 Sの内容から8ビットのイミディエイト値を16ビット長で減算し、その結果をSに格納します。演算の際、イミディエイト値を16ビット長にゼロ拡張します。

この命令はmフラグの影響を受けません。

10進演算を行うことはできません。この命令を使用する際には「D=0」にしてください。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	N	V					Z	C

N : 実行結果のMSBが“1”のとき、“1”になります。それ以外では“0”になります。

V : 符号付き演算と見なした結果が+32767、又は-32768を越えると“1”になります。それ以外では“0”になります。

Z : 実行結果が“0”のとき、“1”になります。それ以外では“0”になります。

C : ボローが発生すると、“0”になります。それ以外では“1”になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
IMM	SUBS #imm	31 ₁₆ ,0B ₁₆ ,imm	3	2

【記述例】

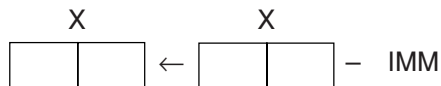
SUBS #IMM8 ; S S - IMM8

【機能】 減算

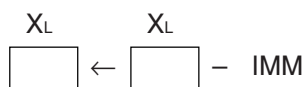
【演算長】 16ビット又は8ビット

【動作】 $X \leftarrow X - IMM$ ($IMM=0 \sim 31$)

・ $x=“0”$ のとき



・ $x=“1”$ のとき



このとき X_H の内容は変化しません。

【説明】 X の内容からイミディエイト値 (0~31) を減算し、その結果を X へ格納します。

この命令は m フラグの影響を受けません。

10進演算を行うことはできません。この命令を使用する際には「 $D=0$ 」にしてください。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	N	V					Z	C

- N : 実行結果のMSBが“1”のとき、“1”になります。それ以外では“0”になります。
- V : 符号付き演算と見なした結果が+32767、又は-32768 (「 $x=1$ 」のときは+127又は-128) を越えると“1”になります。それ以外では“0”になります。
- Z : 実行結果が“0”のとき、“1”になります。それ以外では“0”になります。
- C : ボローが発生すると、“0”になります。それ以外では“1”になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
IMM	SUBX #imm	$01_{16}, imm+40_{16}$	2	2

注 . immには0~31の値が設定できます。

【記述例】

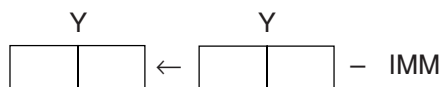
```
CLP      x
SUBX     #IMM      ; X   X - IMM(0~31)
SEP      x
SUBX     #IMM      ; X_L X_L - IMM(0~31)
```

【機能】 減算

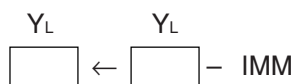
【演算長】 16ビット又は8ビット

【動作】 $Y \leftarrow Y - IMM$ ($IMM=0 \sim 31$)

・ $x=“0”$ のとき



・ $x=“1”$ のとき



このとき Y_H の内容は変化しません。

【説明】 Yの内容からイミディエイト値(0~31)を減算し、その結果をYへ格納します。

この命令はmフラグの影響を受けません。

10進演算を行うことはできません。この命令を使用する際には「D=0」にしてください。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	N	V					Z	C

N : 実行結果のMSBが“1”のとき、“1”になります。それ以外では“0”になります。

V : 符号付き演算と見なした結果が+32767、又は-32768(「x=1」のときは+127又は-128)を越えると“1”になります。それ以外では“0”になります。

Z : 実行結果が“0”のとき、“1”になります。それ以外では“0”になります。

C : ボローが発生すると、“0”になります。それ以外では“1”になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
IMM	SUBY #imm	01 ₁₆ , imm+60 ₁₆	2	2

注 . immには0~31の値が設定できます。

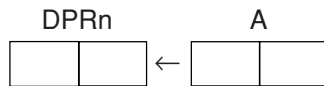
【記述例】

```
CLP      x
SUBY     #IMM      ; Y   Y - IMM(0~31)
SEP      x
SUBY     #IMM      ; YL  YL - IMM(0~31)
```

【機能】 レジスタ間転送

【演算長】 16ビット

【動作】 $DPRn \leftarrow A$ (n=0~3)



【説明】 指定されたDPRn (DPR0~DPR3)へAの内容を16ビット長で転送します。

DPR0~DPR3のいずれか1つを転送先に指定します。

Aの内容は変化しません。

この命令はmフラグの影響を受けません。

従来の7700ファミリのTAD命令の機能はこの命令に含まれます。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
IMP	TAD n	31 ₁₆ ,n2 ₁₆	2	3

注．nには0~3の値が設定できます。

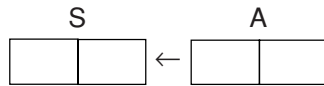
【記述例】

TAD 0 ; DPR0 A
TAD 1 ; DPR1 A

【機能】 レジスタ間転送

【演算長】 16ビット

【動作】 $S \leftarrow A$



【説明】 Aの内容を16ビット長でSへ転送します。Aの内容は変化しません。
この命令はmフラグの影響を受けません。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
IMP	TAS	31 ₁₆ , 82 ₁₆	2	2

【記述例】

TAS ; S ← A

【機能】 レジスタ間転送

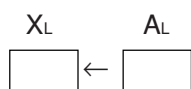
【演算長】 16ビット又は8ビット

【動作】 $X \leftarrow A$

・x=“0”のとき



・x=“1”のとき



このときX_Hの内容は変化しません。

【説明】 Aの内容をXへ転送します。Aの内容は変化しません。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	N						Z	

N : 実行結果のMSBが“1”のとき、“1”になります。それ以外では“0”になります。

Z : 実行結果が“0”のとき、“1”になります。それ以外では“0”になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
IMP	TAX	C4 ₁₆	1	1

【記述例】

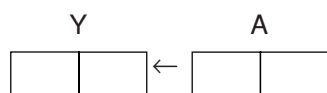
```
CLP      x
TAX
SEP      x
TAX      ; X  A
          ; XL AL
```

【機能】 レジスタ間転送

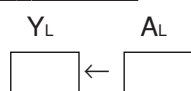
【演算長】 16ビット又は8ビット

【動作】 $Y \leftarrow A$

・ $x = "0"$ のとき



・ $x = "1"$ のとき



このとき Y_H の内容は変化しません。

【説明】 Aの内容をYへ転送します。Aの内容は変化しません。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	N						Z	

N : 実行結果のMSBが“1”のとき、“1”になります。それ以外では“0”になります。

Z : 実行結果が“0”のとき、“1”になります。それ以外では“0”になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
IMP	TAY	D4 ₁₆	1	1

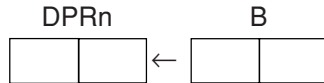
【記述例】

```
CLP      x
TAY                      ; Y  A
SEP      x
TAY                      ; YL AL
```

【機能】 レジスタ間転送

【演算長】 16ビット

【動作】 $DPRn \leftarrow B$ (n=0~3)



【説明】 指定されたDPRn (DPR0~DPR3) へBの内容を16ビット長で転送します。

DPR0~DPR3のいずれか1つを転送先に指定します。

Bの内容は変化しません。

この命令はmフラグの影響を受けません。

従来の7700ファミリのTBD命令の機能はこの命令に含まれます。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
IMP	TBD n	B1 ₁₆ ,n2 ₁₆	2	3

注．nには0~3の値が設定できます。

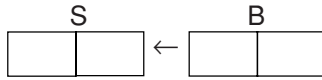
【記述例】

TBD 0 ; DPR0 B
TBD 1 ; DPR1 B

【機能】 レジスタ間転送

【演算長】 16ビット

【動作】 $S \leftarrow B$



【説明】 Bの内容を16ビット長でSへ転送します。Bの内容は変化しません。
この命令はmフラグの影響を受けません。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
IMP	TBS	B1 ₁₆ ,B2 ₁₆	2	2

【記述例】

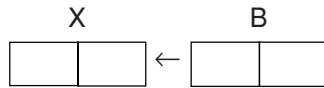
TBS ; S B

【機能】 レジスタ間転送

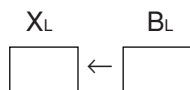
【演算長】 16ビット又は8ビット

【動作】 $X \leftarrow B$

・x=“0”のとき



・x=“1”のとき



このときX_Hの内容は変化しません。

【説明】 Bの内容をXへ転送します。Bの内容は変化しません。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	N						Z	

N : 実行結果のMSBが“1”のとき、“1”になります。それ以外では“0”になります。

Z : 実行結果が“0”のとき、“1”になります。それ以外では“0”になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
IMP	TBX	81 ₁₆ ,C4 ₁₆	2	2

【記述例】

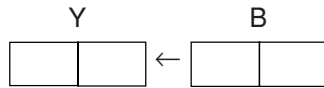
```
CLP      x
TBX                      ; X  B
SEP      x
TBX                      ; XL BL
```

【機能】 レジスタ間転送

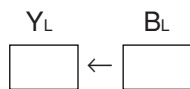
【演算長】 16ビット又は8ビット

【動作】 $Y \leftarrow B$

・ $x = "0"$ のとき



・ $x = "1"$ のとき



このとき Y_H の内容は変化しません。

【説明】 Bの内容をYへ転送します。Bの内容は変化しません。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	N						Z	

N : 実行結果のMSBが“1”のとき、“1”になります。それ以外では“0”になります。

Z : 実行結果が“0”のとき、“1”になります。それ以外では“0”になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
IMP	TBY	81 ₁₆ , D4 ₁₆	2	2

【記述例】

```
CLP      x
TBY
SEP      x
TBY
```

; Y B

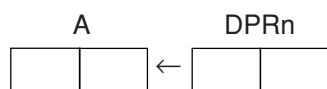
; YL BL

【機能】 レジスタ間転送

【演算長】 16ビット又は8ビット

【動作】 $A \leftarrow \text{DPRn}$ (n=0~3)

・ m=“0”のとき



・ m=“1”のとき



このときAHの内容は変化しません。

【説明】 指定されたDPRn (DPR0~DPR3) の内容をAへ転送します。

DPR0~DPR3のいずれか1つを転送元に指定します。

DPRnの内容は変化しません。

従来の7700ファミリのTDA命令の機能はこの命令に含まれます。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	N						Z	

N : 実行結果のMSBが“1”のとき、“1”になります。それ以外では“0”になります。

Z : 実行結果が“0”のとき、“1”になります。それ以外では“0”になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
IMP	TDA n	31 ₁₆ ,n2 ₁₆ +40 ₁₆	2	2

注．nには0~3の値が設定できます。

【記述例】

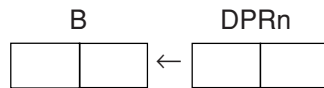
TDA 0 ; A DPR0
TDA 1 ; A DPR1

【機能】 レジスタ間転送

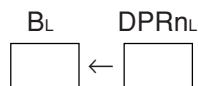
【演算長】 16ビット又は8ビット

【動作】 $B \leftarrow \text{DPRn}$ (n=0~3)

・ m=“0”のとき



・ m=“1”のとき



このときB_Hの内容は変化しません。

【説明】 指定されたDPRn (DPR0~DPR3) の内容をBへ転送します。

DPR0~DPR3のいずれか1つを転送元に指定します。

DPRnの内容は変化しません。

従来の7700ファミリのTDB命令の機能はこの命令に含まれます。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	N						Z	

N : 実行結果のMSBが“1”のとき、“1”になります。それ以外では“0”になります。

Z : 実行結果が“0”のとき、“1”になります。それ以外では“0”になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
IMP	TDB n	B1 ₁₆ ,n2 ₁₆ +40 ₁₆	2	2

注．nには0~3の値が設定できます。

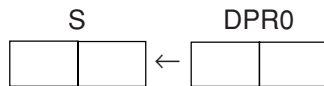
【記述例】

TDB 0 ; B DPR0
TDB 1 ; B DPR1

【機能】 レジスタ間転送

【演算長】 16ビット

【動作】 $S \leftarrow DPR0$



【説明】 DPR0の内容を16ビット長でSへ転送します。
DPR0の内容は変化しません。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C

ア・レッシング・モード	記述形式	機械語	バイト数	サイクル数
IMP	TDS	31 ₁₆ ,73 ₁₆	2	2

【記述例】

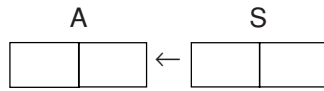
TDS ; S DPR0

【機能】 レジスタ間転送

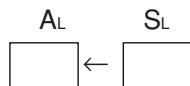
【演算長】 16ビット又は8ビット

【動作】 $A \leftarrow S$

・ $m = "0"$ のとき



・ $m = "1"$ のとき



A_H の内容は変化しません。

【説明】 Sの内容をAへ転送します。Sの内容は変化しません。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	N						Z	

N : 実行結果のMSBが“1”のとき、“1”になります。それ以外では“0”になります。

Z : 実行結果が“0”のとき、“1”になります。それ以外では“0”になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
IMP	TSA	31 ₁₆ ,92 ₁₆	2	2

【記述例】

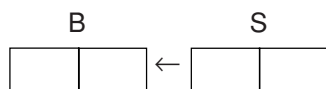
```
CLM
TSA                ; A  S
SEM
TSA                ; AL SL
```

【機能】 レジスタ間転送

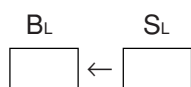
【演算長】 16ビット又は8ビット

【動作】 $B \leftarrow S$

・ $m = "0"$ のとき



・ $m = "1"$ のとき



B_H の内容は変化しません。

【説明】 S の内容を B へ転送します。 S の内容は変化しません。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	N						Z	

N : 実行結果のMSBが“1”のとき、“1”になります。それ以外では“0”になります。

Z : 実行結果が“0”のとき、“1”になります。それ以外では“0”になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
IMP	TSB	B1 ₁₆ ,92 ₁₆	2	2

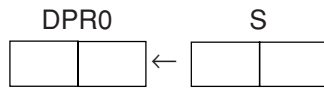
【記述例】

```
CLM
TSB                ; B  S
SEM
TSB                ; B_L S_L
```


【機能】 レジスタ間転送

【演算長】 16ビット

【動作】 DPR0 ← S



【説明】 Sの内容をDPR0へ16ビット長で転送します。
Sの内容は変化しません。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C

ア・レッシング・モード	記述形式	機械語	バイト数	サイクル数
IMP	TSD	31 ₁₆ , 70 ₁₆	2	4

【記述例】

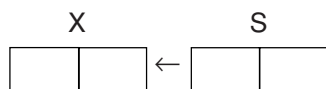
TSD ; DPR0 S

【機能】 レジスタ間転送

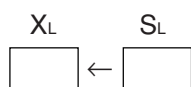
【演算長】 16ビット又は8ビット

【動作】 $X \leftarrow S$

・x=“0”のとき



・x=“1”のとき



X_Hの内容は変化しません。

【説明】 Sの内容をXへ転送します。Sの内容は変化しません。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	N						Z	

N : 実行結果のMSBが“1”のとき、“1”になります。それ以外では“0”になります。

Z : 実行結果が“0”のとき、“1”になります。それ以外では“0”になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
IMP	TSX	31 ₁₆ ,F2 ₁₆	2	2

【記述例】

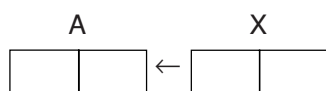
```
CLP      x
TSX                      ; X  S
SEP      x
TSX                      ; XL SL
```

【機能】 レジスタ間転送

【演算長】 16ビット又は8ビット

【動作】 $A \leftarrow X$

・ $m = "0"$ かつ $x = "0"$ のとき

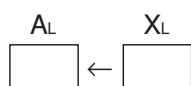


・ $m = "0"$ かつ $x = "1"$ のとき



A_H には“00₁₆”が入ります。

・ $m = "1"$ のとき



A_H の内容は変化しません。

【説明】 Xの内容をAへ転送します。Xの内容は変化しません。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	N						Z	

N : 実行結果のMSBが“1”のとき、“1”になります。それ以外では“0”になります。

Z : 実行結果が“0”のとき、“1”になります。それ以外では“0”になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
IMP	TXA	A4 ₁₆	1	1

【記述例】

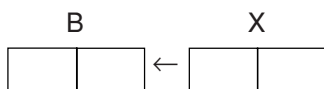
TXA ; A ← X

【機能】 レジスタ間転送

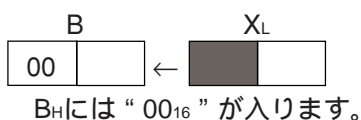
【演算長】 16ビット又は8ビット

【動作】 $B \leftarrow X$

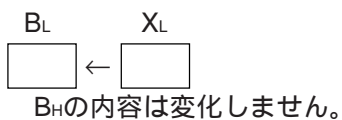
・ $m = "0"$ かつ $x = "0"$ のとき



・ $m = "0"$ かつ $x = "1"$ のとき



・ $m = "1"$ のとき



【説明】 Xの内容をBへ転送します。Xの内容は変化しません。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	N						Z	

N : 実行結果のMSBが "1" のとき、"1" になります。それ以外では "0" になります。

Z : 実行結果が "0" のとき、"1" になります。それ以外では "0" になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
IMP	TXB	8 ₁₆ ,A4 ₁₆	2	2

【記述例】

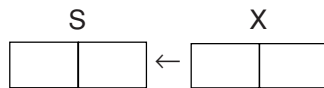
TXB ; B X

【機能】 レジスタ間転送

【演算長】 16ビット又は8ビット

【動作】 $S \leftarrow X$

・ $x = "0"$ のとき



・ $x = "1"$ のとき



S_H には“00₁₆”が入ります。

【説明】 Xの内容をSへ転送します。Xの内容は変化しません。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
IMP	TXS	31 ₁₆ , E2 ₁₆	2	2

【記述例】

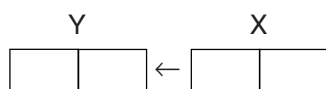
```
CLP      x
TXS
SEP      x
TXS      ; SL  XL, SH  0016
          ; S   X
```

【機能】 レジスタ間転送

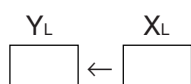
【演算長】 16ビット又は8ビット

【動作】 $Y \leftarrow X$

・x=“0”のとき



・x=“1”のとき



Y_Hの内容は変化しません。

【説明】 Xの内容をYへ転送します。Xの内容は変化しません。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	N						Z	

N : 実行結果のMSBが“1”のとき、“1”になります。それ以外では“0”になります。

Z : 実行結果が“0”のとき、“1”になります。それ以外では“0”になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
IMP	TXY	31 ₁₆ ,C2 ₁₆	2	2

【記述例】

```
CLP      x
TXY                      ; Y  X
SEP      x
TXY                      ; YL XL
```

【機能】 レジスタ間転送

【演算長】 16ビット又は8ビット

【動作】 $A \leftarrow Y$

・ $m = "0"$ かつ $x = "0"$ のとき

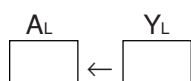


・ $m = "0"$ かつ $x = "1"$ のとき



A_H には“00₁₆”が入ります。

・ $m = "1"$ のとき



A_H の内容は変化しません。

【説明】 Yの内容をAへ転送します。Yの内容は変化しません。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	N						Z	

N : 実行結果のMSBが“1”のとき、“1”になります。それ以外では“0”になります。

Z : 実行結果が“0”のとき、“1”になります。それ以外では“0”になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
IMP	TYA	B4 ₁₆	1	1

【記述例】

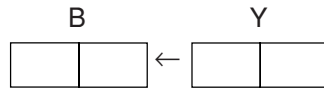
TYA ; A ← Y

【機能】 レジスタ間転送

【演算長】 16ビット又は8ビット

【動作】 $B \leftarrow Y$

・ $m = "0"$ かつ $x = "0"$ のとき



・ $m = "0"$ かつ $x = "1"$ のとき



B_H には " 00_{16} "が入ります。

・ $m = "1"$ のとき



B_H の内容は変化しません。

【説明】 Yの内容をBへ転送します。Yの内容は変化しません。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	N						Z	

N : 実行結果のMSBが "1" のとき、"1" になります。それ以外では "0" になります。

Z : 実行結果が "0" のとき、"1" になります。それ以外では "0" になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
IMP	TYB	$8_{16}, B_{4_{16}}$	2	2

【記述例】

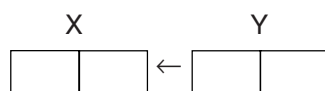
TYB ; B ← Y

【機能】 レジスタ間転送

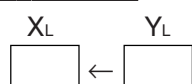
【演算長】 16ビット又は8ビット

【動作】 $X \leftarrow Y$

・ $x = "0"$ のとき



・ $x = "1"$ のとき



X_H の内容は変化しません。

【説明】 Yの内容をXへ転送します。Yの内容は変化しません。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	N						Z	

N : 実行結果のMSBが“1”のとき、“1”になります。それ以外では“0”になります。

Z : 実行結果が“0”のとき、“1”になります。それ以外では“0”になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
IMP	TYX	31 ₁₆ , D2 ₁₆	2	2

【記述例】

```
CLP      x
TYX
SEP      x
TYX      ; X  Y
          ; XL YL
```

【機能】 特殊

【演算長】 -

【動作】 CPUクロックを停止します。

【説明】 内部クロックを停止しますが、発振回路の発振は停止しません。
再起動するためには、割り込み要求が発生するか、又はハードウェアリセットすることでウェイト状態が解除されます。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
IMP	WIT	31 ₁₆ ,10 ₁₆	2	

【記述例】

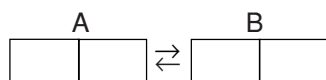
WIT ;

【機能】 レジスタ間転送

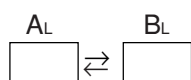
【演算長】 16ビット又は8ビット

【動作】 $A \rightleftharpoons B$

・ $m = "0"$ のとき



・ $m = "1"$ のとき



A_H 及び B_H の内容は変化しません。

【説明】 AとBの内容を交換します。

【プロセッサステータスレジスタ】

IPL	N	V	m	x	D	I	Z	C
	N						Z	

N : 実行結果のMSBが“1”のとき、“1”になります。それ以外では“0”になります。

Z : 実行結果が“0”のとき、“1”になります。それ以外では“0”になります。

アドレッシングモード	記述形式	機械語	バイト数	サイクル数
IMP	XAB	55 ₁₆	1	2

【記述例】

CLM
 XAB ; A B
 SEM
 XAB ; AL BL

4.3 ソフトウェア開発上の注意

4.3.1 命令実行サイクル

本マニュアルに記載の命令実行サイクルは理想的状態で動作した場合の値です。

実際の命令実行サイクルは、命令キューバッファ・メモリアクセスのデータバス幅・ウェイト数(バスサイクル)の状態により変動します。マニュアルの数値を用いて机上でプログラム実行速度を見積もる場合やソフトウェアでタイマを実現する場合等には、その実行時間はあくまで目安と考えてください。

4.3.2 mフラグ、xフラグの状態

mフラグの内容が1(データ長8ビット)であるのにもかかわらず、命令のオペランドに16ビットイミディエイト値を記述したり、mフラグの内容が0(データ長16ビット)であるのにもかかわらず、命令のオペランドに8ビットイミディエイト値を記述するとプログラム暴走の原因になります。

xフラグについても同様です。ご使用になるアセンブラのユーザズマニュアルを参考にし、フラグの状態と演算データ長に矛盾が生じないようにご注意ください。

4.3.3 データ領域配置の工夫

(1)ダイレクトページレジスタの下位8ビット(DPR_{nL})の内容を00₁₆以外の値に設定すると、00₁₆に設定した場合よりも処理時間が1サイクル長くなります。

したがって、可能であればダイレクトページレジスタの下位8ビット(DPR_{nL})の内容を00₁₆に設定すると、プログラムの実行速度向上に効果があります。

(2)16ビットデータは偶数番地境界に配置すると、プログラム実行速度向上に効果的です。

奇数番地境界に16ビットデータを配置しそのデータをアクセスすると、バスサイクルが2回発生しますのでプログラム実行速度が低下します。

4.3.4 10進演算を行う場合

(1)10進演算モードフラグDを1に設定すると10進演算を行うことができます。

ただし、10進演算を行うことができる命令は以下の4命令です。

- ・ADC
- ・ADCB
- ・SBC
- ・SBCB

(2)10進演算を行う際にはフラグの動作に注意が必要です。

10進演算を行った結果はキャリーフラグCには正しく反映されますが、Z, N, Vの各フラグには結果が反映されません。

4.4 32ビット演算命令実行時の注意

32ビット演算命令(注)を使用してメモリをアクセスするときは、データバス幅が異なる領域をまたいでアクセスしないでください。すなわち、データバス幅16ビットの領域(内部領域を含む)と、データバス幅8ビットの領域が連続して配置されているとき、データバス幅が異なる領域の境界前3バイトに対してはダブルワード(32ビット)アクセスをしないでください。

注 . ADCD, ADDD, ADDMD, ANDMD, CMPD, CMPMD, EORMD, LDAD, ORAMD, SBCD, STAD, SUBD, SUBMD

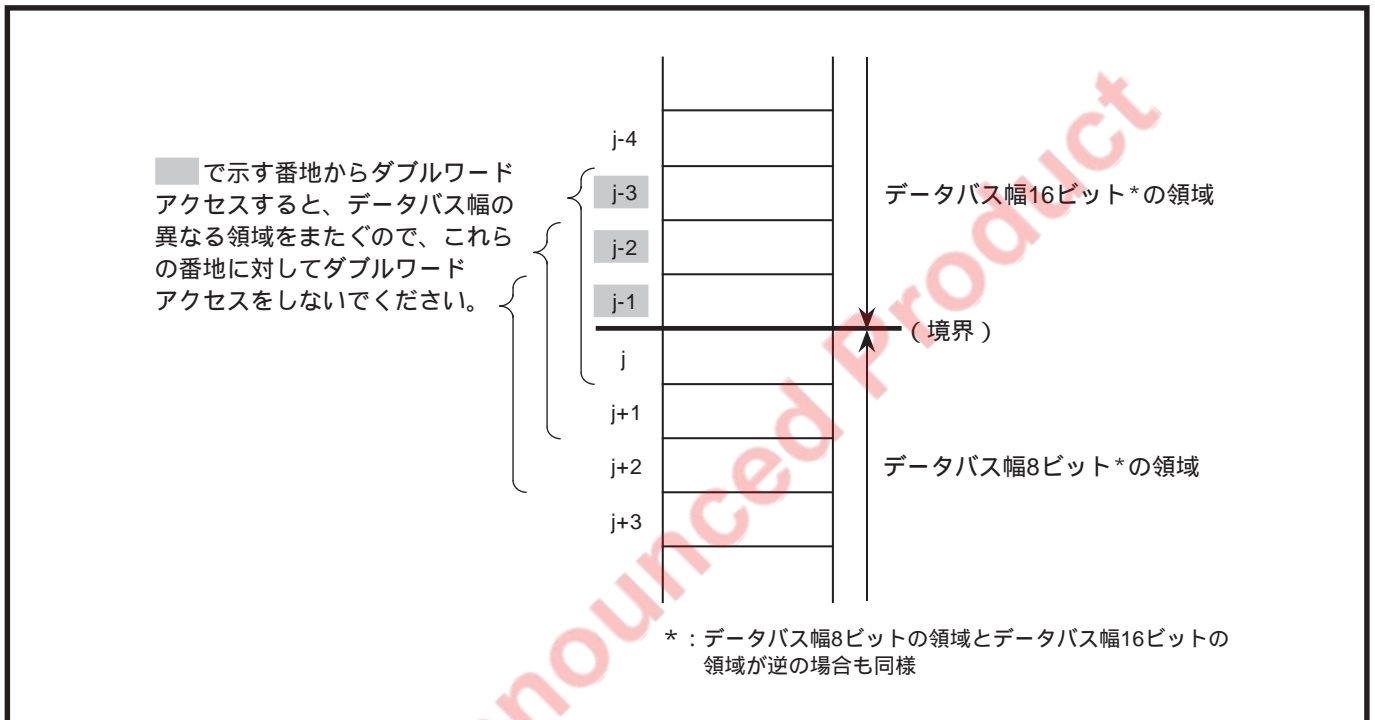


図4.4.1 データバス幅が異なる領域をまたぐアクセス例

付 録

付録1 . 機械語命令一覧表

付録2 . 16進命令コード対応表

付録1．機械語命令一覧表

注．演算長(ビット)が16/8の命令は、 $m=0$ 又は $x=0$ のとき16ビット長、 $m=1$ 又は $x=1$ のとき8ビット長で演算します。

演算長(ビット)が8又は32の命令は、 m フラグ及び x フラグの内容にかかわらず、8ビット長又は32ビット長で演算します。

EOL announced Product

記号	内容	記号	内容
IMP	インプライド・アドレッシングモード	Acc	アキュムレータ
IMM	イミディエイト・アドレッシングモード	AccH	アキュムレータの上位8ビット
A	アキュムレータ・アドレッシングモード	AccL	アキュムレータの下位8ビット
DIR	ダイレクト・アドレッシングモード	A	アキュムレータA
DIR, X	ダイレクト・インデクストX・アドレッシングモード	AH	アキュムレータAの上位8ビット
DIR, Y	ダイレクト・インデクストY・アドレッシングモード	AL	アキュムレータAの下位8ビット
(DIR)	ダイレクト・インダイレクト・アドレッシングモード	B	アキュムレータB
(DIR, X)	ダイレクト・インデクストX・インダイレクト・アドレッシングモード	BH	アキュムレータBの上位8ビット
(DIR), Y	ダイレクト・インダイレクト・インデクストY・アドレッシングモード	BL	アキュムレータBの下位8ビット
L(DIR)	ダイレクト・インダイレクトロング・アドレッシングモード	E	アキュムレータE
L(DIR), Y	ダイレクト・インダイレクトロング・インデクストY・アドレッシングモード	EH	アキュムレータEの上位16ビット (アキュムレータB)
ABS	アブソリュート・アドレッシングモード	EL	アキュムレータEの下位16ビット (アキュムレータA)
ABS, X	アブソリュート・インデクストX・アドレッシングモード	X	インデックスレジスタX
ABS, Y	アブソリュート・インデクストY・アドレッシングモード	XH	インデックスレジスタXの上位8ビット
ABL	アブソリュートロング・アドレッシングモード	XL	インデックスレジスタXの下位8ビット
ABL, X	アブソリュートロング・インデクストX・アドレッシングモード	Y	インデックスレジスタY
(ABS)	アブソリュート・インダイレクト・アドレッシングモード	YH	インデックスレジスタYの上位8ビット
L(ABS)	アブソリュート・インダイレクトロング・アドレッシングモード	YL	インデックスレジスタYの下位8ビット
(ABS, X)	アブソリュート・インデクストX・インダイレクト・アドレッシングモード	S	スタックポインタ
STK	スタック・アドレッシングモード	REL	相対アドレス
REL	レラティブ・アドレッシングモード	PC	プログラムカウンタ
DIR, b, R	ダイレクト・ビット・レラティブ・アドレッシングモード	PCH	プログラムカウンタの上位8ビット
ABS, b, R	アブソリュート・ビット・レラティブ・アドレッシングモード	PCl	プログラムカウンタの下位8ビット
SR	スタックポインタ・レラティブ・アドレッシングモード	PG	プログラムバンクレジスタ
(SR), Y	スタックポインタ・レラティブ・インダイレクト・インデクストY・アドレッシングモード	DT	データバンクレジスタ
BLK	ブロック転送・アドレッシングモード	DPR0	ダイレクトページレジスタ0
積和	積和アドレッシングモード	DPR0H	ダイレクトページレジスタ0の上位8ビット
op	命令コード (オペコード)	DPR0L	ダイレクトページレジスタ0の下位8ビット
n	サイクル数	DPRn	ダイレクトページレジスタn
#	バイト数	DPRnH	ダイレクトページレジスタnの上位8ビット
C	キャリーフラグ	DPRnL	ダイレクトページレジスタnの下位8ビット
Z	ゼロフラグ	PS	プロセッサステータスレジスタ
I	割り込み禁止フラグ	PSH	プロセッサステータスレジスタの上位8ビット
D	10進演算モードフラグ	PSL	プロセッサステータスレジスタの下位8ビット
x	インデックスレジスタ長選択フラグ	PSL(bit n)	プロセッサステータスレジスタ中のnビット目
m	データ長選択フラグ	M	メモリの内容
V	オーバフローフラグ	M(S)	スタックポインタの内容の示す番地のメモリの内容
N	ネガティブフラグ	M(bit n)	メモリのnビット目
IPL	プロセッサ割り込み優先レベル	Mn	nビットのメモリの番地、又は内容
+	加算	IMM	イミディエイト値 (8ビット又は16ビット)
-	減算	IMMn	nビットのイミディエイト値
×	乗算	IMMH	16ビットのイミディエイト値の上位8ビット
÷	除算	IMML	16ビットのイミディエイト値の下位8ビット
	論理積	ADH	24ビットのアドレスの上位A ₂₃ ~ A ₁₆ の値
	論理和	ADM	24ビットのアドレスの中位A ₁₅ ~ A ₈ の値
	排他的論理和	ADL	24ビットのアドレスの下位A ₇ ~ A ₀ の値
	絶対値	EAR	実効アドレス (16ビット)
	否定	EARH	実効アドレスの上位8ビット
	矢印の方向に移動することを示す	EARL	実効アドレスの下位8ビット
	矢印の方向に移動することを示す	imm	8ビットイミディエイト値
→	交換	immn	nビットイミディエイト値
←		dd	DPRに対する変位 (8ビット又は6ビット)
		i	転送バイト数、回転数、又は繰り返し演算回数
		i1, i2	退避、又は復帰するレジスタ数
		source	転送元を指定するオペランド (ソース)
		dest	転送先を指定するオペランド (デスティネーション)

7900シリーズ 機械語命令一覧表

命令記号	動作	演算長 (ビット)	アドレッシングモード																								
			IMP		IMM		A		DIR		DIR, X		DIR, Y		(DIR)		(DIR, X)		(DIR, Y)		L(DIR)		L(DIR), Y				
			op	n #	op	n #	op	n #	op	n #	op	n #	op	n #	op	n #	op	n #	op	n #	op	n #	op	n #	op	n #	
ABS (注1)	Acc Acc	16/8					E1	3	1																		
							81	4	2																		
							E1																				
ABSD	E E	32					31	5	2																		
							90																				
ADC (注1, 注2)	Acc Acc + M + C	16/8			31	3	3					21	5	3	21	6	3										
					87							8A			8B												
					B1	3	3					A1	7	3	A1	8	3										
					87							8A			8B												
ADCB (注1)	AccL AccL + IMM8 + C	8			31	3	3																				
					1A																						
					B1	3	3																				
					1A																						
ADCD	E E + M32 + C	32			31	4	6					21	7	3	21	8	3										
					1C							9A			9B												
ADD (注1, 注2)	Acc Acc + M	16/8			26	1	2					2A	3	2	2B	4	2										
					81	2	3					81	4	3	81	5	3										
					26							2A			2B												
ADDB (注1)	AccL AccL + IMM8	8			29	1	2																				
					81	2	3																				
					29																						
ADDD	E E + M32	32			2D	3	5					9A	6	2	9B	7	2										
ADDM (注3)	M M + IMM	16/8										51	7	4													
												03															
ADDMB	M8 M8 + IMM8	8										51	7	4													
												02															
ADDMD	M32 M32 + IMM32	32										51	10	7													
												83															
ADDS	S S + IMM8	16			31	2	3																				
					0A																						
ADDX	X X + IMM(IMM=0 ~ 31)	16/8			01	2	2																				
ADDY (注4)	Y Y + IMM(IMM=0 ~ 31)	16/8			01	2	2																				
					20																						
					+																						
					imm																						

アドレッシングモード																				プロセッサステータスレジスタ									
ABS	ABS, X	ABS, Y	ABL	ABL, X	(ABS)	L(ABS)	(ABS, X)	STK	REL	DIR, b, R	ABS, b, R	SR	(SR), Y	BLK	積和	10	9	8	7	6	5	4	3	2	1	0			
op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	IPL	N	V	m	x	D	I	Z	C					
																										. . . 0 V Z 0			
																										. . . 0 V Z 0			
21 5 4 8E	21 6 4 8F	21 6 4 86	21 6 4 8C	21 7 5 8D								21 6 3 83	21 9 3 84				. . . N V Z C												
A1 7 4 8E	A1 8 4 8F	A1 8 4 86	A1 8 4 8C	A1 9 5 8D								A1 8 3 83	A1 11 3 84				. . . N V Z C												
																									. . . N V Z C				
21 7 4 9E	21 8 4 9F	21 8 4 96	21 8 4 9C	21 9 5 9D								21 8 3 93	21 11 3 94				. . . N V Z C												
2E 3 3 8E	2F 4 3 8F	11 5 5 26	11 5 5 2C	11 6 5 2D								11 5 3 23	11 8 3 24				. . . N V Z C												
81 4 4 2E	81 5 4 2F	81 5 4 26	81 5 4 2C	81 6 5 2D								81 5 3 23	81 8 3 24				. . . N V Z C												
																									. . . N V Z C				
9E 6 3	9F 7 3	11 8 4 96	11 8 4 9C	11 9 5 9D								11 8 3 93	11 11 3 94				. . . N V Z C												
51 7 5 07																									. . . N V Z C				
51 7 5 06																									. . . N V Z C				
51 10 8 87																									. . . N V Z C				
																									. . . N V Z C				
																									. . . N V Z C				
																									. . . N V Z C				

命令記号	動作	演算長 (ビット)	アドレッシングモード																															
			IMP		IMM		A		DIR		DIR, X		DIR, Y		(DIR)		(DIR, X)		(DIR), Y		L(DIR)		L(DIR), Y											
			op	n #	op	n #	op	n #	op	n #	op	n #	op	n #	op	n #	op	n #	op	n #	op	n #	op	n #	op	n #								
AND (注1, 注2)	Acc Acc M	16/8			66	1	2			6A	3	2	6B	4	2			11	6	3	11	7	3	11	7	3	11	8	3	11	9	3		
					81	2	3			81	4	3	81	5	3			91	6	3	91	7	3	91	7	3	91	8	3	91	9	3		
					66					6A			6B					60			61			68			62			69				
ANDB (注1)	AccL AccL IMM8	8			23	1	2																											
					81	2	3																											
					23																													
ANDM (注3)	M M IMM	16/8								51	7	4																						
										63																								
ANDMB	M8 M8 IMM8	8								51	7	4																						
										62																								
ANDMD	M32 M32 IMM32	32								51	10	7																						
										E3																								
ASL (注1)	1ビット算術左シフト m=0のとき Acc or M16 C $\boxed{b_{15} \dots b_0}$ 0 m=1のとき AccL or M8 C $\boxed{b_7 \dots b_0}$ 0	16/8								03	1	1	21	7	3	21	8	3																
										81	2	2																						
										03																								
ASL #n (注4)	n(n=0~15)ビット算術左シフト m=0のとき A C $\boxed{b_{15} \dots b_0}$ 0 m=1のとき A _L C $\boxed{b_7 \dots b_0}$ 0	16/8								C1	6	2																						
										40																								
										+																								
										imm																								
										imm																								
ASLD #n (注4)	n(n=0~31)ビット算術左シフト E C $\boxed{b_{31} \dots b_0}$ 0	32								D1	8	2																						
										40																								
										+																								
										imm																								
										imm																								
ASR (注1)	1ビット算術右シフト m=0のとき Acc or M16 $\boxed{b_{15} \dots b_0}$ C m=1のとき AccL or M8 $\boxed{b_7 \dots b_0}$ C	16/8								64	1	1	21	7	3	21	8	3																
										4A																								
										4B																								
										81	2	2																						
										64																								
ASR #n (注4)	n(n=0~15)ビット算術右シフト m=0のとき A $\boxed{b_{15} \dots b_0}$ C m=1のとき A _L $\boxed{b_7 \dots b_0}$ C	16/8								C1	6	2																						
										80																								
										+																								
										imm																								
										imm																								

命令記号	動作	演算長 (ビット)	アドレッシングモード																		
			IMP		IMM		A		DIR	DIR, X	DIR, Y	(DIR)	(DIR, X)	(DIR), Y	L(DIR)	L(DIR), Y					
			op	n #	op	n #	op	n #	op	n #	op	n #	op	n #	op	n #	op	n #	op	n #	op
ASRD #n (注4)	n(n=0~31)ビット算術右シフト E ┌──────────┐ [C] b31...b0	32					D1	8	2												
BBC (注3)	if M(bit n)=0 then PC PC + cnt + REL(-128 ~ +127) (cnt: 命令のバイト数)	16/8																			
BBCB	if M8(bit n)=0 then PC PC + cnt + REL(-128 ~ +127) (cnt: 命令のバイト数)	8																			
BBS (注3)	if M(bit n)=1 then PC PC + cnt + REL(-128 ~ +127) (cnt: 命令のバイト数)	16/8																			
BBSB	if M8(bit n)=1 then PC PC + cnt + REL(-128 ~ +127) (cnt: 命令のバイト数)	8																			
BCC	if C=0 then PC PC + 2 + REL(-128 ~ +127)	-																			
BCS	if C=1 then PC PC + 2 + REL(-128 ~ +127)	-																			
BEQ	if Z=1 then PC PC + 2 + REL(-128 ~ +127)	-																			
BGE	if N V=0 then PC PC + 2 + REL(-128 ~ +127)	-																			
BGT	if Z=0 and N V=0 then PC PC + 2 + REL(-128 ~ +127)	-																			
BGTU	if C=1 and Z=0 then PC PC + 2 + REL(-128 ~ +127)	-																			
BLE	if Z=1 or N V=1 then PC PC + 2 + REL(-128 ~ +127)	-																			
BLEU	if C=0 or Z=1 then PC PC + 2 + REL(-128 ~ +127)	-																			
BLT	if N V=1 then PC PC + 2 + REL(-128 ~ +127)	-																			

アドレッシングモード																		プロセッサステータスレジスタ									
ABS	ABS, X	ABS, Y	ABL	ABL, X	(ABS)	L(ABS)	(ABS, X)	STK	REL	DIR, b, R	ABS, b, R	SR	(SR), Y	BLK	積和	10	9	8	7	6	5	4	3	2	1	0	
op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	IPL	N	V	m	x	D	I	Z	C			
																											..N.....ZC
										41 9 5 5A	41 9 6 5E															
										52 8 4	57 8 5															
										41 9 5 4A	41 9 6 4E															
										42 8 4	47 8 5															
										90 6 2																
										B0 6 2																
										F0 6 2																
										C0 6 2																
										80 6 2																
										40 6 2																
										A0 6 2																
										60 6 2																
										E0 6 2																

命令記号	動作	演算長 (ビット)	アドレッシングモード																					
			IMP		IMM		A		DIR		DIR, X		DIR, Y		(DIR)		(DIR, X)		(DIR, Y)		L(DIR)		L(DIR), Y	
			op	n #	op	n #	op	n #	op	n #	op	n #	op	n #	op	n #	op	n #	op	n #	op	n #	op	n #
BMI	if N=1 then PC PC + 2 + REL(-128 ~ +127)	-																						
BNE	if Z=0 then PC PC + 2 + REL(-128 ~ +127)	-																						
BPL	if N=0 then PC PC + 2 + REL(-128 ~ +127)	-																						
BRA/BRAL (注5)	PC PC + cnt + REL (BRA:-128 ~ +127, BRAL:-32768 ~ +32767) (cnt : BRA/BRAL命令のバイト数) PG PG + 1 (キャリーが生じたとき) PG PG - 1 (ボローが生じたとき)	-																						
BRK (注6)	PC PC + 2 M(S) PG S S - 1 M(S) PCH S S - 1 M(S) PCL S S - 1 M(S) PSH S S - 1 M(S) PSL S S - 1 I 1 PCL ADL PCH ADM PG 00 ₁₆ or FF ₁₆	-	00	15	2																			
BSC (注7)	if A(bit n) or M(bit n)=0(n=0 ~ 15) then PC PC + cnt + REL(-128 ~ +127) (cnt : 命令のバイト数)	16/8							01	7	3	71	11	4										
BSR	(S) PC + 2 PC PC + 2 + REL(-1024 ~ +1023)	-							A0			A0												
BSS (注7)	if A(bit n) or M(bit n)=1(n=0 ~ 15) then PC PC + cnt + REL(-128 ~ +127) (cnt : 命令のバイト数)	16/8							80			80												
BVC	if V=0 then PC PC + 2 + REL(-128 ~ +127)	-																						
BVS	if V=1 then PC PC + 2 + REL(-128 ~ +127)	-																						

アドレッシングモード																				プロセッサステータスレジスタ									
ABS	ABS, X	ABS, Y	ABL	ABL, X	(ABS)	L(ABS)	(ABS, X)	STK	REL	DIR, b, R	ABS, b, R	SR	(SR), Y	BLK	積和	10	9	8	7	6	5	4	3	2	1	0			
op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	IPL	N	V	m	x	D	I	Z	C					
									30	6	2							
									D0	6	2							
									10	6	2							
									20	5	2							
									A7	5	3							
																		
71	10	5																
E0																		
+																		
n																		
									F8	7	2							
									}									
									FF									
71	10	5																
C0																		
+																		
n																		
									50	6	2							
																		
									70	6	2							
																		

アドレッシングモード																			プロセスステータスレジスタ									
ABS	ABS, X	ABS, Y	ABL	ABL, X	(ABS)	L(ABS)	(ABS, X)	STK	REL	DIR, b, R	ABS, b, R	SR	(SR), Y	BLK	積和	10	9	8	7	6	5	4	3	2	1	0		
op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	IPL	N	V	m	x	D	I	Z	C				
																...	N	V	Z	C	
																...	N	V	Z	C
																...	N	V	Z	C
																...	N	V	Z	C
																0	
																0	..	
																0
																指定したフラグ が“0”になり ます。	
																...	0	1	..	
																...	0	1	..	
D7	5	3														
C7	5	3														
																...	0	1	..	
																...	0	1	..	

命令記号	動作	演算長 (ビット)	アドレッシングモード																							
			IMP		IMM		A		DIR		DIR, X		DIR, Y		(DIR)		(DIR, X)		(DIR), Y		L(DIR)		L(DIR), Y			
			op	n #	op	n #	op	n #	op	n #	op	n #	op	n #	op	n #	op	n #	op	n #	op	n #	op	n #	op	n #
CLV	V 0	-	65	1 1																						
CMP (注1, 注2)	Acc - M	16/8			46	1 2			4A	3 2	4B	4 2			11	6 3	11	7 3	11	7 3	11	8 3	11	9 3		
					81	2 3			81	4 3	81	5 3			91	6 3	91	7 3	91	7 3	91	8 3	91	9 3		
CMPB (注1)	AccL - IMM8	8			38	1 2																				
					81	2 3																				
CMPD	E - M32	32			3C	3 5			BA	6 2	BB	7 2			11	9 3	11	10 3	11	10 3	11	11 3	11	12 3		
CMPM (注3)	M - IMM	16/8							51	5 4																
CMPMB	M8 - IMM8	8							51	5 4																
CMPMD	M32 - IMM32	32							51	7 7																
CPX (注8)	X - M	16/8			E6	1 2			22	3 2																
CPY (注8)	Y - M	16/8			F6	1 2			32	3 2																
DEBNE (注4)	M M - IMM(IMM=0~31) if M 0 then PC PC + cnt + REL(-128 ~ +127) (cnt : 命令のバイト数)	16/8							C1	12 4																
DEC (注1)	Acc Acc - 1 又は M M - 1	16/8			B3	1 1			92	6 2	41	8 3														
					81	2 2																				
DEX	X X - 1	16/8	E3	1 1																						
DEY	Y Y - 1	16/8	F3	1 1																						
DIV (注2, 注9, 注10)	A (商) (B, A) ÷ M B (剰余)	16/8			31	15 3			21	16 3	21	17 3			21	18 3	21	19 3	21	19 3	21	20 3	21	21 3		
					E7				EA		EB				E0		E1		E8		E2		E9			

アドレッシングモード																					プロセスステータスレジスタ									
ABS	ABS, X	ABS, Y	ABL	ABL, X	(ABS)	L(ABS)	(ABS, X)	STK	REL	DIR, b, R	ABS, b, R	SR	(SR), Y	BLK	積和	10	9	8	7	6	5	4	3	2	1	0				
op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	IPL	N	V	m	x	D	I	Z	C						
																	0													
4E	3	3	4F	4	3	11 46	5	4	11 4C	5	5	11 4D	6	5																
81 4E	4	4	81 4F	5	4	91 46	5	4	91 4C	5	5	91 4D	6	5																
												11 43	5	3	11 44	8	3													
												91 43	5	3	91 44	8	3													
BE	6	3	BF	7	3	11 B6	8	4	11 BC	8	5	11 BD	9	5																
												11 B3	8	3	11 B4	11	3													
51 27	5	5																												
51 26	5	5																												
51 A7	7	8																												
41 2E	4	4																												
41 3E	4	4																												
D1 E0 + imm	11	5																												
97	6	3	41 9F	8	4																									
21 EE	16	4	21 EF	17	4	21 E6	17	4	21 EC	17	5	21 ED	18	5																
												21 E3	17	3	21 E4	20	3													

命令記号	動作	演算長 (ビット)	アドレッシングモード																					
			IMP		IMM		A		DIR		DIR, X		DIR, Y		(DIR)		(DIR, X)		(DIR), Y		L(DIR)		L(DIR), Y	
			op	n #	op	n #	op	n #	op	n #	op	n #	op	n #	op	n #	op	n #	op	n #	op	n #	op	n #
DIVS (注2, 注9, 注10)	A (商) (B, A) ÷ M B (剰余) (符号付き)	16/8																						
DXBNE (注4)	X X - IMM(IMM=0~31) if X 0 then PC PC + cnt + REL(-128 ~ +127) (cnt: 命令のバイト数)	16/8																						
DYBNE (注4)	Y Y - IMM(IMM=0~31) if Y 0 then PC PC + cnt + REL(-128 ~ +127) (cnt: 命令のバイト数)	16/8																						
EOR (注1, 注2)	Acc Acc M	16/8																						
EORB (注1)	AccL AccL IMM8	8																						
EORM (注3)	M M IMM	16/8																						
EORMB	M8 M8 IMM8	8																						
EORMD	M32 M32 IMM32	32																						
EXTS (注1)	Acc AccL (符号拡張) (AccLのビット7)=0のとき b15 b7 b0 00000000 0 AccH AccL (AccLのビット7)=1のとき b15 b7 b0 11111111 1 AccH AccL	16																						
EXTSD	E EL(=A) (符号拡張) (Aのビット15)=0のとき b15 b0 b15 b0 0000 ₁₆ 0 EH(B) EL(A) (Aのビット15)=1のとき b15 b0 b15 b0 FFFF ₁₆ 1 EH(B) EL(A)	32																						
EXTZ (注1)	Acc AccL (ゼロ拡張) b15 b8 b7 b0 00000000 AccH AccL	16																						
EXTZD	E EL(=A) (ゼロ拡張) b15 b8 b7 b0 0000 ₁₆ EH(B) EL(A)	32																						

アドレッシングモード																						プロセスステータスレジスタ									
ABS	ABS, X	ABS, Y	ABL	ABL, X	(ABS)	L(ABS)	(ABS, X)	STK	REL	DIR, b, R	ABS, b, R	SR	(SR), Y	BLK	積和	10	9	8	7	6	5	4	3	2	1	0					
op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	IPL	N	V	m	x	D	I	Z	C							
21 FE	23 4	21 FF	24 4	21 F6	24 4	21 FC	24 5	21 FD	25 5							21 F3	24 3	21 F4	27 3								• • • N V • • • I Z C				
																										• • • • • • • • • •					
																											• • • • • • • • • •				
7E	3	3	7F	4	3	11	5	4	11	5	5	11	6	5		11	5	3	11	8	3						• • • N • • • • • Z •				
						76			7C			7D				73			74												
81	4	4	81	5	4	91	5	4	91	5	5	91	6	5		91	5	3	91	8	3										
7E			7F			76			7C			7D				73			74								• • • N • • • • • Z •				
																											• • • N • • • • • Z •				
																												• • • N • • • • • Z •			
																												• • • N • • • • • Z •			
																												• • • N • • • • • Z •			
																												• • • N • • • • • Z •			
																												• • • N • • • • • Z •			
																												• • • 0 • • • • • Z •			
																												• • • 0 • • • • • Z •			

命令記号	動作	演算長 (ビット)	アドレッシングモード																													
			IMP		IMM		A		DIR		DIR, X		DIR, Y		(DIR)		(DIR, X)		(DIR), Y		L(DIR)		L(DIR), Y									
			op	n #	op	n #	op	n #	op	n #	op	n #	op	n #	op	n #	op	n #	op	n #	op	n #	op	n #	op	n #						
INC (注1)	Acc Acc+1 又は M M+1	16/8																														
INX	X X+1	16/8	C3	1	1																											
INY	Y Y+1	16/8	D3	1	1																											
JMP/JMPL	ABSのとき PCL ADL PCH ADM ABLのとき PCL ADL PCH ADM PG ADH (ABS)のとき PCL (ADM, ADL) PCH (ADM, ADL + 1) L(ABS)のとき PCL (ADM, ADL) PCH (ADM, ADL + 1) PG (ADM, ADL + 2) (ABS, X)のとき PCL (ADM, ADL + X) PCH (ADM, ADL + X + 1)	-																														
JSR/JSRL	ABSのとき PC PC+3 M(S) PCH S S-1 M(S) PCL S S-1 PCL ADL PCH ADM ABLのとき PC PC+4 M(S) PG S S-1 M(S) PCH S S-1 M(S) PCL S S-1 PCL ADL PCH ADM PG ADH (ABS, X)のとき PC PC+3 M(S) PCH S S-1 M(S) PCL S S-1 PCL (ADM, ADL + X) PCH (ADM, ADL + X + 1)	-																														
LDA (注1, 注2)	Acc M	16/8			16	1	2			1A	3	2	1B	4	2			11	6	3	11	7	3	18	6	2	11	8	3	19	8	2
					81	2	3			81	4	3	81	5	3			91	6	3	91	7	3	81	7	3	91	8	3	81	9	3
					16					1A		1B						10			11		18		12		19					

命令記号	動作	演算長 (ビット)	アドレッシングモード																													
			IMP		IMM		A		DIR		DIR, X		DIR, Y		(DIR)		(DIR, X)		(DIR), Y		L(DIR)		L(DIR), Y									
			op	n #	op	n #	op	n #	op	n #	op	n #	op	n #	op	n #	op	n #	op	n #	op	n #	op	n #	op	n #						
LDAB (注1)	Acc M8 (ゼロ拡張)	16			28	1	2			0A	3	2	0B	4	2			11	6	3	11	7	3	08	6	2	11	8	3	09	8	2
					81	2	3			81	4	3	81	5	3			91	6	3	91	7	3	81	7	3	91	8	3	81	9	3
					2C	3	5			8A	6	2	8B	7	2			11	9	3	11	10	3	88	9	2	11	11	3	89	11	2
					80																											
LDAD	E M32	32																														
LDD n (注11, 注12)	DPRn IMM16 (n=0~3。複数指定可)	16			B8	13	4																									
					B8	11	2																									
					20	+	+																									
					2i	2i																										
LDT	DT IMM8	8			31	4	3																									
					4A																											
LDX (注8)	X M	16/8			C6	1	2			02	3	2			41	5	3			05												
LDXB	X IMM8 (ゼロ拡張)	16			27	1	2																									
LDY (注8)	Y M	16/8			D6	1	2			12	3	2	41	5	3			1B														
LDYB	Y IMM8 (ゼロ拡張)	16			37	1	2																									
LSR (注1)	1ビット論理右シフト m=0のとき Acc or M16 0 [b15...b0] C m=1のとき AccL or M8 0 [b7...b0] C	16/8			43	1	1	21	7	3	21	8	3																			
					81	2	2																									
					43																											
LSR #n (注4)	n(n=0~15)ビット論理右シフト m=0のとき A 0 [b15...b0] C m=1のとき AL 0 [b7...b0] C	16/8			C1	6	2																									
						+																										
						imm																										
LSRD #n (注4)	n(n=0~31)ビット論理右シフト E 0 [b31...b0] C	32			D1	8	2																									
						+																										
						imm																										

アドレッシングモード																	プロセッサステータレジスタ										
ABS	ABS, X	ABS, Y	ABL	ABL, X	(ABS)	L(ABS)	(ABS, X)	STK	REL	DIR, b, R	ABS, b, R	SR	(SR), Y	BLK	積和	10	9	8	7	6	5	4	3	2	1	0	
op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	IPL	N	V	m	x	D	I	Z	C			
0E	3	3	0F	4	3	11	5	4	0C	4	4	0D	5	4													
												11	5	3	11	8	3										
												03	04														
81	4	4	81	5	4	91	5	4	81	5	5	81	6	5													
0E			0F			06			0C			0D															
												91	5	3	91	8	3										
												03	04														
8E	6	3	8F	7	3	11	8	4	8C	7	4	8D	8	4													
												11	8	3	11	3											
												83	84														
07	3	3				41	5	4																			
						06																					
17	3	3	41	5	4																						
			1F																								
21	7	4	21	8	4																						
2E			2F																								

命令記号	動作	演算長 (ビット)		デスティネーション																								
				IMP		IMM		A		DIR		DIR, X		DIR, Y		(DIR)		(DIR, X)		(DIR), Y		L(DIR)		L(DIR), Y				
				op	n #	op	n #	op	n #	op	n #	op	n #	op	n #	op	n #	op	n #	op	n #	op	n #	op	n #	op	n #	
MOVM (注2)	m=0のとき M16(dest) M16(source) m=1のとき M8(dest) M8(source)	16/8	ソ ス	IMM							86	5	31	7	4													
				DIR							58	6	3															
				DIR, X																								
				ABS							5C	6	4															
				ABS, X							5D	7	4															
MOVMB	M8(dest) M8(source)	8	ソ ス	IMM							A9	5	31	7	4													
				DIR							48	6	3															
				DIR, X																								
				ABS							4C	6	4															
				ABS, X							4D	7	4															
MOVR (注7, 注13)	m=0のとき M16(dest1) M16(source1) ⋮ M16(dest n) M16(source n) m=1のとき M8(dest1) M8(source1) ⋮ M8(dest n) M8(source n) (n=0 ~ 15)	16/8	ソ ス	IMM							61	3	2															
				DIR							61	3	2															
				DIR, X							50	+	+															
				ABS							61	3	2															
				ABS, X							71	3	2															
MOVRB (注7)	M8(dest1) M8(source1) ⋮ M8(dest n) M8(source n) (n=0 ~ 15)	8	ソ ス	IMM							61	3	2															
				DIR							61	3	2															
				DIR, X							40	+	+															
				ABS							61	3	2															
				ABS, X							71	3	2															

デスティネーション																	プロセッサステータスレジスタ										
ABS	ABS, X	ABS, Y	ABL	ABL, X	(ABS)	L(ABS)	(ABS, X)	STK	REL	DIR, b, R	ABS, b, R	SR	(SR), Y	BLK	積和	10	9	8	7	6	5	4	3	2	1	0	
op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	IPL	N	V	m	x	D	I	Z	C			
96	4	4	31	6	5																						
			57																								
78	5	4																									
79	6	4																									
7C	5	5																									
B9	4	4	31	6	5																						
			3B																								
68	5	4																									
69	6	4																									
6C	5	5																									
61	3	2																									
30	+	+																									
	+	4n	3n																								
	n																										
61	3	2																									
70	+	+																									
	+	5n	3n																								
	n																										
71	3	2																									
70	+	+																									
	+	6n	3n																								
	n																										
61	3	2																									
B0	+	+																									
	+	5n	4n																								
	n																										
61	3	2																									
20	+	+																									
	+	4n	3n																								
	n																										
61	3	2																									
60	+	+																									
	+	5n	3n																								
	n																										
71	3	2																									
60	+	+																									
	+	6n	3n																								
	n																										
61	3	2																									
A0	+	+																									
	+	5n	4n																								
	n																										

命令記号	動作	演算長 (ビット)	アドレッシングモード																															
			IMP		IMM		A		DIR		DIR, X		DIR, Y		(DIR)		(DIR, X)		(DIR), Y		L(DIR)		L(DIR), Y											
			op	n #	op	n #	op	n #	op	n #	op	n #	op	n #	op	n #	op	n #	op	n #	op	n #	op	n #	op	n #								
MPY (注2, 注14)	(B, A) A × M	16/8			31	8	3			21	9	3	21	10	3			21	11	3	21	12	3	21	12	3	21	13	3	21	14	3		
MPYS (注2, 注14)	(B, A) A × M (符号付き)	16/8			31	8	3			21	9	3	21	10	3			21	11	3	21	12	3	21	12	3	21	13	3	21	14	3		
MVN (注15)	M(Y + k) M(X + k) k=0~i-1 (iはアキュムレータで 指定する転送バイト数)	16/8																																
MVP (注16)	M(Y - k) M(X - k) k=0~i-1 (iはアキュムレータで 指定する転送バイト数)	16/8																																
NEG (注1)	Acc -Acc	16/8						24	1	1																								
NEGD	E -E	32						31	4	2																								
NOP	PC PC + 1 PCにキャリーが生じた場合： PG PG + 1	-	74	1	1																													
ORA (注1, 注2)	Acc Acc M	16/8			56	1	2			5A	3	2	5B	4	2			11	6	3	11	7	3	11	7	3	11	8	3	11	9	3		
					81	2	3			81	4	3	81	5	3			91	6	3	91	7	3	91	7	3	91	8	3	91	9	3		
					56					5A			5B					50			51			58			52			59				
ORAB (注1)	AccL AccL IMM8	8			63	1	2																											
					81	2	3																											
					63																													
ORAM (注3)	M M IMM	16/8								51	7	4																						
										33																								
ORAMB	M8 M8 IMM8	8								51	7	4																						
										32																								
ORAMD	M32 M32 IMM32	32								51	10	7																						
										B3																								
PEA	M(S) IMM _H S S - 1 M(S) IMM _L S S - 1	16																																
PEI	M(S) M((DPR _n) + dd + 1) S S - 1 M(S) M((DPR _n) + dd) S S - 1 (n=0~3)	16																																

付録

付録1. 機械語命令一覧表

アドレッシングモード																	プロセッサステータスレジスタ										
ABS	ABS, X	ABS, Y	ABL	ABL, X	(ABS)	L(ABS)	(ABS, X)	STK	REL	DIR, b, R	ABS, b, R	SR	(SR), Y	BLK	積和	10	9	8	7	6	5	4	3	2	1	0	
op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	IPL	N	V	m	x	D	I	Z	C			
21 9 4 CE	21 10 4 CF	21 10 4 C6	21 10 4 CC	21 11 5 CD								21 10 3 C3	21 13 3 C4			...	N									Z 0	
21 9 4 DE	21 10 4 DF	21 10 4 D6	21 10 4 DC	21 11 5 DD								21 10 3 D3	21 13 3 D4			...	N									Z 0	
														31 5 4 2B + 5i		...											
														31 9 4 2A + 5i		...											
																...	N	V								Z C	
																...	N	V								Z C	
																...											
5E 3 3	5F 4 3	11 5 4 56	11 5 4 5C	11 6 5 5D								11 5 3 53	11 8 3 54			...	N									Z .	
81 4 4 5E	81 5 4 5F	91 5 4 56	91 5 4 5C	91 6 5 5D								91 5 3 53	91 8 3 54			...	N									Z .	
																...	N									Z .	
51 7 5 37																...	N									Z .	
51 7 5 36																...	N									Z .	
51 10 8 B7																...	N									Z .	
																...											
																...											

命令記号	動 作	演算長 (ビット)	ア ド レ ッ シ ン グ モ ー ド																							
			IMP		IMM		A		DIR		DIR, X		DIR, Y		(DIR)		(DIR, X)		(DIR), Y		L(DIR)		L(DIR), Y			
			op	n #	op	n #	op	n #	op	n #	op	n #	op	n #	op	n #	op	n #	op	n #	op	n #	op	n #	op	n #
PER	EAR PC + IMM16 M(S) EAR _H S S - 1 M(S) EAR _L S S - 1	16																								
PHA	m=0のとき M(S) A _H S S - 1 M(S) A _L S S - 1 m=1のとき M(S) A _L S S - 1	16/8																								
PHB	m=0のとき M(S) B _H S S - 1 M(S) B _L S S - 1 m=1のとき M(S) B _L S S - 1	16/8																								
PHD	M(S) DPR0 _H S S - 1 M(S) DPR0 _L S S - 1	16																								
PHD n (注11)	M(S) DPRn _H S S - 1 M(S) DPRn _L S S - 1 (n=0~3) 複数のDPRを指定した場合、 上記の繰り返し	16																								
PHG	M(S) PG S S - 1	8																								
PHLD n (注11)	M(S) DPRn _H S S - 1 M(S) DPRn _L S S - 1 DPRn IMM16 (n=0~3) 複数のDPRを指定した場合、 上記の繰り返し	16																								
PHP	M(S) P _S _H S S - 1 M(S) P _S _L S S - 1	16																								
PHT	M(S) DT S S - 1	8																								

アドレッシングモード																			プロセッサステータレジスタ									
ABS	ABS, X	ABS, Y	ABL	ABL, X	(ABS)	L(ABS)	(ABS, X)	STK	REL	DIR, b, R	ABS, b, R	SR	(SR), Y	BLK	積和	10	9	8	7	6	5	4	3	2	1	0		
op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	IPL	N	V	m	x	D	I	Z	C				
								31 4D	6	4							
								85	4	1							
								81 85	5	2							
								83	4	1							
								B8 01 }	12	2							
								B8 01 }	11	2							
								31 60	4	2							
								B8 01 }	14	4							
								B8 01 }	11	2							
								A5	4	1							
								31 40	4	2							

アドレッシングモード																	プロセッサステータスレジスタ									
ABS	ABS, X	ABS, Y	ABL	ABL, X	(ABS)	L(ABS)	(ABS, X)	STK	REL	DIR, b, R	ABS, b, R	SR	(SR), Y	BLK	積和	10	9	8	7	6	5	4	3	2	1	0
op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	IPL	N	V	m	x	D	I	Z	C		
								C5	4	1					
								E5	4	1					
								95	4	1						.	.	.	N	Z	.
								81	5	2						.	.	.	N	Z	.
								95							
								93	5	1					
								77	11	2					
								77	8	2					
								70	+	3i					
								85	5	1																
								31	6	2						.	.	.	N	Z	.
								50							

EOL announced Product

命令記号	動作	演算長 (ビット)	アドレッシングモード																							
			IMP		IMM		A		DIR		DIR, X		DIR, Y		(DIR)		(DIR, X)		(DIR), Y		L(DIR)		L(DIR), Y			
			op	n #	op	n #	op	n #	op	n #	op	n #	op	n #	op	n #	op	n #	op	n #	op	n #	op	n #	op	n #
PLX	<p>x=0のとき S S+1 XL M(S) S S+1 XH M(S)</p> <p>x=1のとき S S+1 XL M(S)</p>	16/8																								
PLY	<p>x=0のとき S S+1 YL M(S) S S+1 YH M(S)</p> <p>x=1のとき S S+1 YL M(S)</p>	16/8																								
PSH (注17)	<p>M(S - S - i + 1) A, B, X...</p> <p>S S - i</p> <p>i: スタックに退避するレジスタの バイト数(合計)</p>	16/8																								
PUL (注18)	<p>A, B, X... M(S + 1 - S + i)</p> <p>S S + i</p> <p>i: スタックから復帰するレジスタ のバイト数(合計)</p>	16/8																								
RLA (注3)	<p>nビット左回転</p> <p>m=0のとき (n=0 ~ 65535)</p> <div style="text-align: center;"> <p>$\left[\begin{matrix} A \\ b_{15} \dots b_0 \end{matrix} \right]$</p> </div> <p>m=1のとき (n=0 ~ 255)</p> <div style="text-align: center;"> <p>$\left[\begin{matrix} A_L \\ b_7 \dots b_0 \end{matrix} \right]$</p> </div>	16/8			31	5	3																			
RMPA (注19)	<p>m=0のとき</p> <p>Repeat</p> <p>(B, A) (B, A) + M(DT: X)</p> <p>× M(DT: Y) (符号付き)</p> <p>X X + 2</p> <p>Y Y + 2</p> <p>i i - 1</p> <p>Until i = 0</p> <p>m=1のとき</p> <p>Repeat</p> <p>(BL, AL) (BL, AL) + M(DT: X)</p> <p>× M(DT: Y) (符号付き)</p> <p>X X + 1</p> <p>Y Y + 1</p> <p>i i - 1</p> <p>Until i = 0</p> <p>i: 繰り返し回数。0 ~ 255</p>	16/8																								

アドレッシングモード																				プロセッサステータスレジスタ									
ABS	ABS, X	ABS, Y	ABL	ABL, X	(ABS)	L(ABS)	(ABS, X)	STK	REL	DIR, b, R	ABS, b, R	SR	(SR), Y	BLK	積和	10	9	8	7	6	5	4	3	2	1	0			
op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	IPL	N	V	m	x	D	I	Z	C					
								D5	4	1						..	.	N	Z	.			
								F5	4	1						..	.	N	Z	.			
								A8	11	2								
								+										
								2i	+	i2								
								67	13	2								
								+										
								3i										
																		
																31	5	3	..	.	N	V	.	.	.	Z	C		
																5A	+			
																14	imm			

命令記号	動作	演算長 (ビット)	アドレッシングモード																														
			IMP		IMM		A		DIR		DIR, X		DIR, Y		(DIR)		(DIR, X)		(DIR), Y		L(DIR)		L(DIR), Y										
			op	n #	op	n #	op	n #	op	n #	op	n #	op	n #	op	n #	op	n #	op	n #	op	n #	op	n #	op	n #							
ROL (注1)	1ビット左回転 m=0のとき m=1のとき 	16/8							13	1	1	21	7	3	21	8	3																
ROL #n (注4)	n(n=0~15)ビット左回転 m=0のとき m=1のとき 	16/8							C1	6	2																						
ROLD #n (注4)	n(n=0~31)ビット左回転 	32							D1	8	2																						
ROR (注1)	1ビット右回転 m=0のとき m=1のとき 	16/8							53	1	1	21	7	3	21	8	3																
ROR #n (注4)	n(n=0~15)ビット右回転 m=0のとき m=1のとき 	16/8							C1	6	2																						
RORD #n (注4)	n(n=0~31)ビット右回転 	32							D1	8	2																						

アドレッシングモード																プロセスステータレジスタ										
ABS	ABS, X	ABS, Y	ABL	ABL, X	(ABS)	L(ABS)	(ABS, X)	STK	REL	DIR, b, R	ABS, b, R	SR	(SR), Y	BLK	積和	10	9	8	7	6	5	4	3	2	1	0
op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	IPL	N	V	m	x	D	I	Z	C		
21 1E	7 4	21 1F	8 4																							
21 3E	7 4	21 3F	8 4																							

命令記号	動作	演算長 (ビット)	アドレッシングモード																																						
			IMP		IMM		A		DIR		DIR, X		DIR, Y		(DIR)		(DIR, X)		(DIR, Y)		L(DIR)		L(DIR), Y																		
			op	n #	op	n #	op	n #	op	n #	op	n #	op	n #	op	n #	op	n #	op	n #	op	n #	op	n #	op	n #															
RTI	S S+1 PSL M(S) S S+1 PSH M(S) S S+1 PCL M(S) S S+1 PCH M(S) S S+1 PG M(S)	-	F1	12	1																																				
RTL	S S+1 PCL M(S) S S+1 PCH M(S) S S+1 PG M(S)	-	94	10	1																																				
RTLD n (注11, 注12)	S S+1 DPRnL M(S) S S+1 DPRnH M(S) S S+1 PCL M(S) S S+1 PCH M(S) S S+1 PG M(S) (n=0 ~ 3。複数指定可)	16																																							
RTS	S S+1 PCL M(S) S S+1 PCH M(S)	-	84	7	1																																				
RTSD n (注11, 注12)	S S+1 DPRnL M(S) S S+1 DPRnH M(S) S S+1 PCL M(S) S S+1 PCH M(S) (n=0 ~ 3。複数指定可)	16																																							
SBC (注1, 注2)	Acc Acc - M - \bar{C}	16/8				31	3	3										21	7	3	21	8	3	21	8	3	21	9	3	21	10	3	21	11	3	21	12	3			
						A7												A0			A1			A8			A2			A9											
						B1	3	3										A1	7	3	A1	8	3				A1	10	3	A1	10	3	A1	11	3	A1	12	3			
						A7												AA			AB																				
SBCB (注1)	AccL AccL - IMM8 - \bar{C}	8				31	3	3																																	
						1B																																			
						B1	3	3																																	
						1B																																			
SBCD	E E - M32 - \bar{C}	32				31	4	6										21	7	3	21	8	3				21	9	3	21	10	3	21	10	3	21	11	3	21	12	3
						1D												BA			BB						B0			B1			B8			B2			B9		
SEC	C 1	-	04	1	1																																				
SEI	I 1	-	05	4	1																																				

アドレッシングモード																プロセッサステータスレジスタ										
ABS	ABS, X	ABS, Y	ABL	ABL, X	(ABS)	L(ABS)	(ABS, X)	STK	REL	DIR, b, R	ABS, b, R	SR	(SR), Y	BLK	積和	10	9	8	7	6	5	4	3	2	1	0
op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	IPL	N	V	m	x	D	I	Z	C		
																スタックから復帰した値になります。										
																									
								77 15 2 ?C																	
								77 12 2 ?C + 3i																	
																									
								77 14 2 ?8																	
								77 11 2 ?8 + 3i																	
21 5 4 21 6 4 21 6 4 21 6 5 21 7 5	AE	AF	A6	AC	AD								21 6 3 21 9 3	A3	A4	. . . N V Z C										
A1 7 4 A1 8 4 A1 8 4 A1 8 5 A1 9 5	AE	AF	A6	AC	AD								A1 8 3 A1 11 3	A3	A4	. . . N V Z C										
																									
21 7 4 21 8 4 21 8 4 21 8 5 21 9 5	BE	BF	B6	BC	BD								21 8 3 21 11 3	B3	B4	. . . N V Z C										
															 1										
															 1 . .										

命令記号	動作	演算長 (ビット)	アドレッシングモード																																		
			IMP		IMM		A		DIR		DIR, X		DIR, Y		(DIR)		(DIR, X)		(DIR), Y		L(DIR)		L(DIR), Y														
			op	n #	op	n #	op	n #	op	n #	op	n #	op	n #	op	n #	op	n #	op	n #	op	n #	op	n #	op	n #											
SEM	m 1	-	25	3	1																																
SEP	PS _i (bit n) 1 (n=0~7.複数ビット指定可)	-				99	3	2																													
STA (注1)	M Acc	16/8							DA	4	2	DB	5	2					11	7	3	11	8	3	D8	7	2	11	9	3	D9	9	2				
									81	5	3	81	6	3					91	7	3	91	8	3	81	8	3	91	9	3	81	10	3				
									DA		DB							D0		D1		D8		D2		D9											
STAB (注1)	M8 Acc.	8							CA	4	2	CB	5	2					11	7	3	11	8	3	C8	7	2	11	9	3	C9	9	2				
									81	5	3	81	6	3					91	7	3	91	8	3	81	8	3	91	9	3	81	10	3				
									CA		CB							C0		C1		C8		C2		C9											
STAD	M32 E	32							EA	6	2	EB	7	2					11	9	3	11	10	3	E8	9	2	11	11	3	E9	11	2				
																		E0		E1																	
STP	発振停止	-	31	-	2																																
			30																																		
STX	M X	16/8							E2	4	2									41	6	3															
																				E5																	
STY	M Y	16/8							F2	4	2	41	6	3																							
																				FB																	
SUB (注1,注2)	Acc Acc - M	16/8				36	1	2				3A	3	2	3B	4	2				11	6	3	11	7	3	11	7	3	11	8	3	11	9	3		
						81	2	3				81	4	3	81	5	3				91	6	3	91	7	3	91	7	3	91	8	3	91	9	3		
						36						3A			3B						30			31			38			32			39				
SUBB (注1)	AccL AccL - IMM8	8				39	1	2																													
						81	2	3																													
						39																															
SUBD	E E - M32	32				3D	3	5				AA	6	2	AB	7	2				11	9	3	11	10	3	11	10	3	11	11	3	11	12	3		
																					A0		A1		A8		A2		A9								
SUBM (注3)	M M - IMM	16/8							51	7	4																										
									13																												
SUBMB	M8 M8 - IMM8	8							51	7	4																										
									12																												
SUBMD	M32 M32 - IMM32	32							51	10	7																										
									93																												

命令記号	動作	演算長 (ビット)	アドレッシングモード																					
			IMP		IMM		A		DIR		DIR, X		DIR, Y		(DIR)		(DIR, X)		(DIR), Y		L(DIR)		L(DIR), Y	
			op	n #	op	n #	op	n #	op	n #	op	n #	op	n #	op	n #	op	n #	op	n #	op	n #	op	n #
SUBS	S S - IMM8	16			31	2	3																	
SUBX (注4)	X X - IMM(IMM=0~31)	16/8			01	2	2																	
SUBY (注4)	Y Y - IMM(IMM=0~31)	16/8			01	2	2																	
TAD n (注20)	DPRn A (n=0~3)	16	31	3	2																			
TAS	S A	16	31	2	2																			
TAX	X A	16/8	C4	1	1																			
TAY	Y A	16/8	D4	1	1																			
TBD n (注20)	DPRn B (n=0~3)	16	B1	3	2																			
TBS	S B	16	B1	2	2																			
TBX	X B	16/8	81	2	2																			
TBY	Y B	16/8	81	2	2																			
TDA n (注20)	A DPRn (n=0~3)	16/8	31	2	2																			
TDB n (注20)	B DPRn (n=0~3)	16/8	B1	2	2																			
TDS	S DPR0	16	31	2	2																			

アドレッシングモード																プロセッサステータスレジスタ											
ABS	ABS, X	ABS, Y	ABL	ABL, X	(ABS)	L(ABS)	(ABS, X)	STK	REL	DIR, b, R	ABS, b, R	SR	(SR), Y	BLK	積和	10	9	8	7	6	5	4	3	2	1	0	
op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	IPL	N	V	m	x	D	I	Z	C			
																.	.	.	N	V	Z	C
																.	.	.	N	V	Z	C
																.	.	.	N	V	Z	C
															
															
																.	.	.	N	Z	.
																.	.	.	N	Z	.
															
															
																.	.	.	N	Z	.
																.	.	.	N	Z	.
																.	.	.	N	Z	.
															
															

命令記号	動作	演算長 (ビット)	アドレッシングモード																						
			IMP		IMM		A		DIR		DIR, X		DIR, Y		(DIR)		(DIR, X)		(DIR), Y		L(DIR)		L(DIR), Y		
			op	n #	op	n #	op	n #	op	n #	op	n #	op	n #	op	n #	op	n #	op	n #	op	n #	op	n #	op
TSA	A S	16/8	31	2	2																				
TSB	B S	16/8	B1	2	2																				
TSD	DPR0 S	16	31	4	2																				
TSX	X S	16/8	31	2	2																				
TXA	A X	16/8	A4	1	1																				
TXB	B X	16/8	81	2	2																				
TXS	S X	16/8	31	2	2																				
TXY	Y X	16/8	31	2	2																				
TYA	A Y	16/8	B4	1	1																				
TYB	B Y	16/8	81	2	2																				
TYX	X Y	16/8	31	2	2																				
WIT	CPUクロック停止	-	31	-	2																				
XAB	A ₂ →B	16/8	55	2	1																				

アドレッシングモード																プロセッサステータスレジスタ										
ABS	ABS, X	ABS, Y	ABL	ABL, X	(ABS)	L(ABS)	(ABS, X)	STK	REL	DIR, b, R	ABS, b, R	SR	(SR), Y	BLK	積和	10	9	8	7	6	5	4	3	2	1	0
op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	op n #	IPL	N	V	m	x	D	I	Z	C		
																..N.....Z.										
																..N.....Z.										
																									
																..N.....Z.										
																..N.....Z.										
																..N.....Z.										
																									
																..N.....Z.										
																..N.....Z.										
																									
																..N.....Z.										
																..N.....Z.										
																									
																..N.....Z.										

【機械語命令一覧表参照時の注意】

表中の命令サイクル数は最短の場合を示しています。サイクル数は以下の条件によって変化します。

- ・ダイレクトページレジスタの下位バイト(DPR_{nL})の値
DPR_n($n=0\sim 3$)に関するアドレッシングモードのサイクル数は、DPR_{nL}=0の場合の値です。DPR_{nL} ≠ 0の場合は、サイクル数を+1してください。
- ・命令キューバッファに取り込まれている命令のバイト数
- ・メモリの読み出し、書き込み時の番地(偶数が奇数か)
- ・外部データバス幅8ビットで外部領域をアクセスした場合
- ・1バスサイクルの長さ

- 注 1. アキュムレータAを使用する場合は、上段のオペコードを使用してください。
アキュムレータBを使用する場合は、下段のオペコードを使用してください。
2. IMMアドレッシングモードを使用して、 $m=0$ で16ビットデータを扱う場合は、バイト数(#)を+1してください。
3. $m=0$ で16ビットデータを扱う場合は、バイト数(#)を+1してください。
4. immはオペランドで指定した即値です。
5. -128 ~ +127の範囲のブランチでは、上段のオペコードを使用してください。
-32768 ~ +32767の範囲のブランチでは、下段のオペコードを使用してください。
6. BRK命令はデバッグツール用の予約命令です。エミュレータ使用時には使用できません。
7. 機械語欄のnには0 ~ 15の値が設定されます。
8. IMMアドレッシングモードを使用して、 $x=0$ で16ビットデータを扱う場合は、バイト数(#)を+1してください。
9. サイクル数は16ビット÷8ビット演算の場合の値です。
32ビット÷16ビット演算の場合は、サイクル数を+8してください。
10. 0除算割り込みが発生した場合のサイクル数は、演算のデータ長にかかわらず、16サイクルになります。
11. 特定の1つのダイレクトページレジスタに値を設定する場合は上段の機械語を使用してください。
複数のダイレクトページレジスタに値を設定する場合は下段の機械語を使用してください。
iは指定するDPR_nの数(1 ~ 4)です。
12. ?は指定したDPR_nに対応するビットが"1"になります。
13. ソースがIMMアドレッシングモードでかつ $m=0$ の場合、バイト数(#)は $n(0\sim 15)$ バイト増加します。
14. サイクル数は8ビット×8ビット演算の場合の値です。
16ビット×16ビット演算の場合は、サイクル数を+4してください。

- 15．サイクル数は、転送バイト数(i)が偶数の場合の値です。
転送バイト数(i)が奇数の場合のサイクル数は、 $5 \times i + 10$ となります。
- 16．サイクル数は、転送バイト数(i)が偶数の場合の値です。
転送バイト数(i)が奇数の場合のサイクル数は、 $5 \times i + 14$ となります。
ただし、1バイト転送時は10サイクルです。
- 17．退避するレジスタに対応したサイクル数を追加してください。
 i_1 はA, B, X, Y, DPR0, PSのうち退避するレジスタの個数を示します。
 i_2 はDT, PGのうち退避するレジスタの個数を示します。
- 18． i_1 は復帰するレジスタの個数を示します。
- 19．サイクル数は $m = 1$ の場合の値です。
 $m = 0$ の場合のサイクル数は、 $18 \times imm + 5$ (imm : 繰り返し回数)になります。
- 20．機械語欄の n には0～3の値が設定されます。

EOL announced Product

付録2 . 16進命令コード対応表

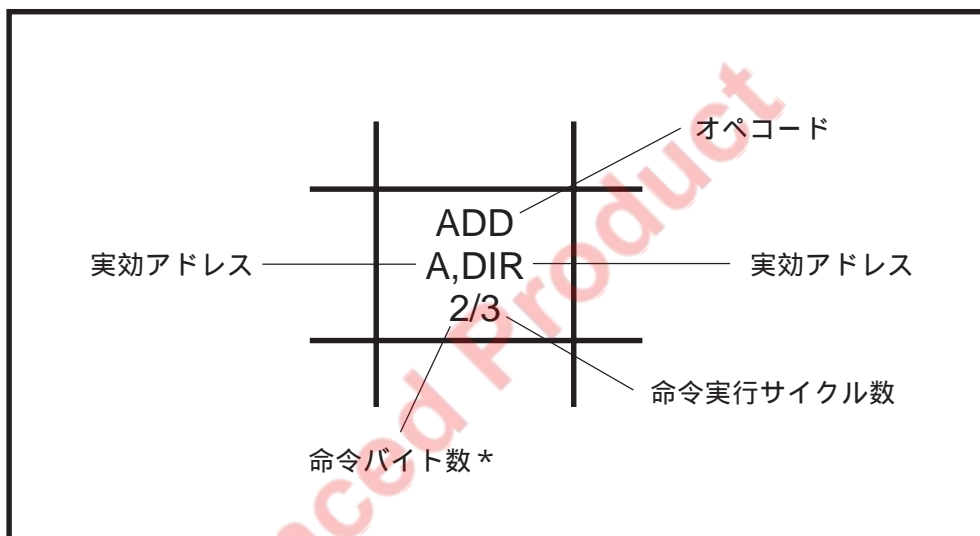
<表の見方>

まず、命令コード対応表 0-Aを参照します。

オペコードが2バイト長の命令は2バイト目のコードが別表に示されています。

別表に示されている命令の1バイト目は命令コード対応表0-Aにおいて、“PAGE XX”の形式で示されています。

表には下記の形式で記載しています。



*()で示す数値はm=0、又はx=0で16ビットデータを扱う場合です。
特に断らない場合は、m, xフラグの影響は受けません。

命令コード対応表 0-A

D7 - D4 16進 表記	D3 - D0 16進 表記	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	0	BRK(注1) IMP 2/15	PAGE10	LXD DIR 2/3	ASL A 1/1	SEC IMP 1/1	SEI IMP 1/4	LXD ABS 3/3	LDAB A,(DIR),Y 2/6	LDAB A,L(DIR),Y 2/6	LDAB A,DIR 2/3	LDAB A,DIR,X 2/4	LDAB A,ABL 4/4	LDAB A,ABL,X 4/5	LDAB A,ABS 3/3	LDAB A,ABS,X 3/4	
0001	1	BPL REL 2/6	PAGE1-A	LDY DIR 2/3	ROL A 1/1	CLC IMP 1/1	CLI IMP 1/3	LDA A,IMM 2(3)/1	LDY ABS 3/3	LDA A,(DIR),Y 2/6	LDA A,L(DIR),Y 2/6	LDA A,DIR 2/3	LDA A,DIR,X 2/4	LDA A,ABL 4/4	LDA A,ABL,X 4/5	LDA A,ABS 3/3	LDA A,ABS,X 3/4
0010	2	BRA REL 2/5	PAGE2-A	CPX DIR 2/3	ANDB A,IMM 2/1	NEG A 1/1	SEM IMP 1/3	ADD A,IMM 2(3)/1	LDXB IMM 2/1	LDAB A,IMM 2/1	ADDB A,IMM 2/1	ADD A,DIR 2/3	ADD A,DIR,X 2/4	LDAD E,IMM 5/3	ADDD E,IMM 5/3	ADD A,ABS 3/3	ADD A,ABS,X 3/4
0011	3	BMI REL 2/6	PAGE3-A	CPY DIR 2/3	EORB A,IMM 2/1	EXTZ A 1/1	EXTS A 1/1	SUB A,IMM 2(3)/1	LDYB IMM 2/1	CMPB A,IMM 2/1	SUBB A,IMM 2/1	SUB A,DIR 2/3	SUB A,DIR,X 2/4	CMPD E,IMM 5/3	SUBD E,IMM 5/3	SUB A,ABS 3/3	SUB A,ABS,X 3/4
0100	4	BGTU REL 2/6	PAGE4	BBSB DIR,b,REL 4/8	LSR A 1/1	CLRB A 1/1	CLM IMP 1/3	CMP A,IMM 2(3)/1	BBSB ABS,b,REL 5/8	MOVMB DIR/DIR 3/6		CMP A,DIR 2/3	CMP A,DIR,X 2/4	MOVMB DIR/ABS 4/6	MOVMB DIR/ABS,X 4/7	CMP A,ABS 3/3	CMP A,ABS,X 3/4
0101	5	BVC REL 2/6	PAGE5	BBCB DIR,b,REL 4/8	ROR A 1/1	CLR A 1/1	XAB IMP 1/2	ORA A,IMM 2(3)/1	BBCB ABS,b,REL 5/8	MOVMB DIR/DIR 3/6		ORA A,DIR 2/3	ORA A,DIR,X 2/4	MOVMB DIR/ABS 4/6	MOVMB DIR/ABS,X 4/7	ORA A,ABS 3/3	ORA A,ABS,X 3/4
0110	6	BLEU REL 2/6	PAGE6	CBEQB DIR/IMM,REL 4/8	ORAB A,IMM 2/1	ASR A 1/1	CLV IMP 1/1	AND A,IMM 2(3)/1	PUL STK 2(注2)	MOVMB ABS/DIR 4/5	MOVMB ABS/DIR,X 4/6	AND A,DIR 2/3	AND A,DIR,X 2/4	MOVMB ABS/ABS 5/5	MOVMB ABS/ABS,X 5/6	AND A,ABS 3/3	AND A,ABS,X 3/4
0111	7	BVS REL 2/6	PAGE7	CBNEB DIR/IMM,REL 4/8		NOP IMP 1/1		EOR A,IMM 2(3)/1	PLD STK 2(注3)	MOVMB ABS/DIR 4/5	MOVMB ABS/DIR,X 4/6	EOR A,DIR 2/3	EOR A,DIR,X 2/4	MOVMB ABS/ABS 5/5	MOVMB ABS/ABS,X 5/6	EOR A,ABS 3/3	EOR A,ABS,X 3/4
1000	8	BGT REL 2/6	PAGE0-B	INC DIR 2/6	PHD STK 1/4	RTS IMP 1/7	PHA STK 1/4	MOVMB DIR/IMM 3(4)/5	INC ABS 3/6	LDAD E,(DIR),Y 2/11	LDAD E,L(DIR),Y 2/11	LDAD E,DIR 2/6	LDAD E,DIR,X 2/7	LDAD E,ABL 4/7	LDAD E,ABL,X 4/8	LDAD E,ABS 3/6	LDAD E,ABS,X 3/7
1001	9	BCC REL 2/6	PAGE1-B	DEC DIR 2/6	PLD STK 1/5	RTL IMP 1/10	PLA STK 1/4	MOVMB ABS/IMM 4(5)/4	DEC ABS 3/6	CLP IMM 2/4	SEP IMM 2/3	ADDD E,DIR 2/6	ADDD E,DIR,X 2/7	JMP ABS 3/4	JSR ABS 3/6	ADDD E,ABS 3/6	ADDD E,ABS,X 3/7
1010	A	BLE REL 2/6	PAGE2-B	CBEQB A/IMM,REL 3/6	INC A 1/1	TXA IMP 1/1	PHM STK 1/4	CBEQ A/IMM,REL 3(4)/6	BRAL REL 3/5	PSH STK 2(注4)	MOVMB DIR/IMM 3/5	SUBD E,DIR 2/6	SUBD E,DIR,X 2/7	JMPL ABL 4/5	JSR ABL 4/7	SUBD E,ABS 3/6	SUBD E,ABS,X 3/7
1011	B	BCL REL 2/6	PAGE3-B	CBNEB A/IMM,REL 3/6	DEC A 1/1	TYA IMP 1/1	PLP STK 1/5	CBNE A/IMM,REL 3(4)/6		LDAD PHD STK/IMM (注5)(注6)	MOVMB ABS/IMM 4/4	CMPD E,DIR 2/6	CMPD E,DIR,X 2/7	JMP (ABS,X) 3/7	JSR (ABS,X) 3/8	CMPD E,ABS 3/6	CMPD E,ABS,X 3/7
1100	C	BGE REL 2/6	PAGE8	CLRMB DIR 2/5	INX IMP 1/1	TAX IMP 1/1	PHS STK 1/4	LDX IMM 2(3)/1	CLRMB ABS 3/5	STAB A,(DIR),Y 2/7	STAB A,L(DIR),Y 2/9	STAB A,DIR 2/4	STAB A,DIR,X 2/5	STAB A,ABL 4/5	STAB A,ABL,X 4/6	STAB A,ABS 3/4	STAB A,ABS,X 3/5
1101	D	BNE REL 2/6	PAGE9	CLRMB DIR 2/5	INX IMP 1/1	TAY IMP 1/1	PLX STK 1/4	LDY IMM 2(3)/1	CLRMB ABS 3/5	STA A,(DIR),Y 2/7	STA A,L(DIR),Y 2/9	STA A,DIR 2/4	STA A,DIR,X 2/5	STA A,ABL 4/5	STA A,ABL,X 4/6	STA A,ABS 3/4	STA A,ABS,X 3/5
1110	E	BLT REL 2/6	ABS A 1/3	STX DIR 2/4	DEX IMP 1/1	CLR IMP 1/1	PHY STK 1/4	CPX IMM 2(3)/1	STX ABS 3/4	STAD E,(DIR),Y 2/9	STAD E,L(DIR),Y 2/11	STAD E,DIR 2/6	STAD E,DIR,X 2/7	STAD E,ABL 4/7	STAD E,ABL,X 4/8	STAD E,ABS 3/6	STAD E,ABS,X 3/7
1111	F	BEQ REL 2/6	RTI DIR 1/12	STY DIR 2/4	DEY IMP 1/1	CLRY IMP 1/1	PLY STK 1/4	CPY IMM 2(3)/1	STY ABS 3/4				BSR REL 2/7				

命令コード対応表 1-A (PAGE 1-A)

D7 - D4 16進 表記	D3 - D0 16進 表記	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	0	LDAB A,(DIR) 3/6	LDAB A,(DIR),X 3/7	LDAB A,L(DIR) 3/8	LDAB A,SR 3/5	LDAB A,SR,Y 3/6		LDAB A,ABS,Y 4/5									
0001	1	LDA A,(DIR) 3/6	LDA A,(DIR),X 3/7	LDA A,L(DIR) 3/8	LDA A,SR 3/5	LDA A,SR,Y 3/6		LDA A,ABS,Y 4/5									
0010	2	ADD A,(DIR) 3/6	ADD A,(DIR),X 3/7	ADD A,L(DIR) 3/8	ADD A,SR 3/5	ADD A,SR,Y 3/6		ADD A,ABS,Y 4/5	ADD A,(DIR),Y 3/7	ADD A,L(DIR),Y 3/9			ADD A,ABL 5/5	ADD A,ABL,X 5/6			
0011	3	SUB A,(DIR) 3/6	SUB A,(DIR),X 3/7	SUB A,L(DIR) 3/8	SUB A,SR 3/5	SUB A,SR,Y 3/6		SUB A,ABS,Y 4/5	SUB A,(DIR),Y 3/7	SUB A,L(DIR),Y 3/9			SUB A,ABL 5/5	SUB A,ABL,X 5/6			
0100	4	CMP A,(DIR) 3/6	CMP A,(DIR),X 3/7	CMP A,L(DIR) 3/8	CMP A,SR 3/5	CMP A,SR,Y 3/6		CMP A,ABS,Y 4/5	CMP A,(DIR),Y 3/7	CMP A,L(DIR),Y 3/9			CMP A,ABL 5/5	CMP A,ABL,X 5/6			
0101	5	ORA A,(DIR) 3/6	ORA A,(DIR),X 3/7	ORA A,L(DIR) 3/8	ORA A,SR 3/5	ORA A,SR,Y 3/6		ORA A,ABS,Y 4/5	ORA A,(DIR),Y 3/7	ORA A,L(DIR),Y 3/9			ORA A,ABL 5/5	ORA A,ABL,X 5/6			
0110	6	AND A,(DIR) 3/6	AND A,(DIR),X 3/7	AND A,L(DIR) 3/8	AND A,SR 3/5	AND A,SR,Y 3/6		AND A,ABS,Y 4/5	AND A,(DIR),Y 3/7	AND A,L(DIR),Y 3/9			AND A,ABL 5/5	AND A,ABL,X 5/6			
0111	7	EOR A,(DIR) 3/6	EOR A,(DIR),X 3/7	EOR A,L(DIR) 3/8	EOR A,SR 3/5	EOR A,SR,Y 3/6		EOR A,ABS,Y 4/5	EOR A,(DIR),Y 3/7	EOR A,L(DIR),Y 3/9			EOR A,ABL 5/5	EOR A,ABL,X 5/6			
1000	8	LDAD E,(DIR) 3/9	LDAD E,(DIR),X 3/10	LDAD E,L(DIR) 3/11	LDAD E,SR 3/8	LDAD E,SR,Y 3/11		LDAD E,ABS,Y 4/8									
1001	9	ADDD E,(DIR) 3/9	ADDD E,(DIR),X 3/10	ADDD E,L(DIR) 3/11	ADDD E,SR 3/8	ADDD E,SR,Y 3/11		ADDD E,ABS,Y 4/8	ADDD E,(DIR),Y 3/10	ADDD E,L(DIR),Y 3/12			ADDD E,ABL 5/8	ADDD E,ABL,X 5/9			
1010	A	SUBD E,(DIR) 3/9	SUBD E,(DIR),X 3/10	SUBD E,L(DIR) 3/11	SUBD E,SR 3/8	SUBD E,SR,Y 3/11		SUBD E,ABS,Y 4/8	SUBD E,(DIR),Y 3/10	SUBD E,L(DIR),Y 3/12			SUBD E,ABL 5/8	SUBD E,ABL,X 5/9			
1011	B	CMPD E,(DIR) 3/9	CMPD E,(DIR),X 3/10	CMPD E,L(DIR) 3/11	CMPD E,SR 3/8	CMPD E,SR,Y 3/11		CMPD E,ABS,Y 4/8	CMPD E,(DIR),Y 3/10	CMPD E,L(DIR),Y 3/12			CMPD E,ABL 5/8	CMPD E,ABL,X 5/9			
1100	C	STAB A,(DIR) 3/7	STAB A,(DIR),X 3/8	STAB A,L(DIR) 3/9	STAB A,SR 3/6	STAB A,SR,Y 3/9		STAB A,ABS,Y 4/6									
1101	D	STA A,(DIR) 3/7	STA A,(DIR),X 3/8	STA A,L(DIR) 3/9	STA A,SR 3/6	STA A,SR,Y 3/9		STA A,ABS,Y 4/6									
1110	E	STAD E,(DIR) 3/9	STAD E,(DIR),X 3/10	STAD E,L(DIR) 3/11	STAD E,SR 3/8	STAD E,SR,Y 3/11		STAD E,ABS,Y 4/8									
1111	F																

命令コード対応表 2-A (PAGE 2-A)

D3 - D0 16進 表記	D7 - D4	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	0											ASL DIR 3/7	ASL DIR,X 3/8			ASL ABS 4/7	ASL ABS,X 4/8
0001	1											ROL DIR 3/7	ROL DIR,X 3/8			ROL ABS 4/7	ROL ABS,X 4/8
0010	2											LSR DIR 3/7	LSR DIR,X 3/8			LSR ABS 4/7	LSR ABS,X 4/8
0011	3											ROR DIR 3/7	ROR DIR,X 3/8			ROR ABS 4/7	ROR ABS,X 4/8
0100	4											ASR DIR 3/7	ASR DIR,X 3/8			ASR ABS 4/7	ASR ABS,X 4/8
0101	5																
0110	6																
0111	7																
1000	8	ADC A,(DIR) 3/7	ADC A,(DIR,X) 3/8	ADC A,L(DIR) 3/9	ADC A,SR 3/6	ADC A,(SR),Y 3/9		ADC A,ABS,Y 4/6		ADC A,(DIR),Y 3/8	ADC A,L(DIR),Y 3/10	ADC A,DIR 3/5	ADC A,DIR,X 3/6	ADC A,ABL 5/6	ADC A,ABL,X 5/7	ADC A,ABS 4/5	ADC A,ABS,X 4/6
1001	9	ADCD E,(DIR) 3/9	ADCD E,(DIR,X) 3/10	ADCD E,L(DIR) 3/11	ADCD E,SR 3/8	ADCD E,(SR),Y 3/11		ADCD E,ABS,Y 4/8		ADCD E,(DIR),Y 3/10	ADCD E,L(DIR),Y 3/12	ADCD E,DIR 3/7	ADCD E,DIR,X 3/8	ADCD E,ABL 5/8	ADCD E,ABL,X 5/9	ADCD E,ABS 4/7	ADCD E,ABS,X 4/8
1010	A	SBC A,(DIR) 3/7	SBC A,(DIR,X) 3/8	SBC A,L(DIR) 3/9	SBC A,SR 3/6	SBC A,(SR),Y 3/9		SBC A,ABS,Y 4/6		SBC A,(DIR),Y 3/8	SBC A,L(DIR),Y 3/10	SBC A,DIR 3/5	SBC A,DIR,X 3/6	SBC A,ABL 5/6	SBC A,ABL,X 5/7	SBC A,ABS 4/5	SBC A,ABS,X 4/6
1011	B	SBCD E,(DIR) 3/9	SBCD E,(DIR,X) 3/10	SBCD E,L(DIR) 3/11	SBCD E,SR 3/8	SBCD E,(SR),Y 3/11		SBCD E,ABS,Y 4/8		SBCD E,(DIR),Y 3/10	SBCD E,L(DIR),Y 3/12	SBCD E,DIR 3/7	SBCD E,DIR,X 3/8	SBCD E,ABL 5/8	SBCD E,ABL,X 5/9	SBCD E,ABS 4/7	SBCD E,ABS,X 4/8
1100	C	MPY (DIR) 3/11(注7)	MPY (DIR,X) 3/12(注7)	MPY L(DIR) 3/13(注7)	MPY SR 3/10(注7)	MPY (SR),Y 3/13(注7)		MPY ABS,Y 4/10(注7)		MPY (DIR),Y 3/12(注7)	MPY L(DIR),Y 3/14(注7)	MPY DIR 3/9(注7)	MPY DIR,X 3/10(注7)	MPY ABL 5/10(注7)	MPY ABL,X 5/11(注7)	MPY ABS 4/9(注7)	MPY ABS,X 4/10(注7)
1101	D	MPYS (DIR) 3/11(注7)	MPYS (DIR,X) 3/12(注7)	MPYS L(DIR) 3/13(注7)	MPYS SR 3/10(注7)	MPYS (SR),Y 3/13(注7)		MPYS ABS,Y 4/10(注7)		MPYS (DIR),Y 3/12(注7)	MPYS L(DIR),Y 3/14(注7)	MPYS DIR 3/9(注7)	MPYS DIR,X 3/10(注7)	MPYS ABL 5/10(注7)	MPYS ABL,X 5/11(注7)	MPYS ABS 4/9(注7)	MPYS ABS,X 4/10(注7)
1110	E	DIV (DIR) 3/18(注8,9)	DIV (DIR,X) 3/19(注8,9)	DIV L(DIR) 3/20(注8,9)	DIV SR 3/17(注8,9)	DIV (SR),Y 3/20(注8,9)		DIV ABS,Y 4/17(注8,9)		DIV (DIR),Y 3/19(注8,9)	DIV L(DIR),Y 3/21(注8,9)	DIV DIR 3/16(注8,9)	DIV DIR,X 3/17(注8,9)	DIV ABL 5/17(注8,9)	DIV ABL,X 5/18(注8,9)	DIV ABS 4/16(注8,9)	DIV ABS,X 4/17(注8,9)
1111	F	DIVS (DIR) 3/25(注8,9)	DIVS (DIR,X) 3/26(注8,9)	DIVS L(DIR) 3/27(注8,9)	DIVS SR 3/24(注8,9)	DIVS (SR),Y 3/27(注8,9)		DIVS ABS,Y 4/24(注8,9)		DIVS (DIR),Y 3/26(注8,9)	DIVS L(DIR),Y 3/28(注8,9)	DIVS DIR 3/23(注8,9)	DIVS DIR,X 3/24(注8,9)	DIVS ABL 5/24(注8,9)	DIVS ABL,X 5/25(注8,9)	DIVS ABS 4/23(注8,9)	DIVS ABS,X 4/24(注8,9)

命令コード対応表 3-A (PAGE 3-A)

D3 - D0 16進 表記	D7 - D4	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	0			TAD,0 IMP 2/3					RLA A 3(4)/n+5(注10)			ADDS IMM 3/2	SUBS IMM 3/2				
0001	1	WIT IMP 2/-		TAD,1 IMP 2/3								ADCB A,IMM 3/3	SBCB A,IMM 3/3	ADCD E,IMM 6/4	SBCD E,IMM 6/4		
0010	2			TAD,2 IMP 2/3								MVP BLK 4/5+9(注11)	MVN BLK 4/5+5(注12)				
0011	3	STP IMP 2/-		TAD,3 IMP 2/3								MOVMB DIR,X/IMM 4/7	MOVMB ABS,X/IMM 5/6				
0100	4	PHT STK 2/4		TDA,0 IMP 2/2					MOV DIR,X/IMM 4(5)/7			LDT IMM 3/4	PEI STK 3/7	PEA STK 4/5	PER STK 4/6		
0101	5	PLT STK 2/6		TDA,1 IMP 2/2					MOV ABS,X/IMM 5(6)/6			RMPA 値和 3/14imm+5 (注13)		JMP (ABS) 4/7	JMPL L(ABS) 4/9		
0110	6	PHG STK 2/4		TDA,2 IMP 2/2													
0111	7	TSD IMP 2/4		TDA,3 IMP 2/2	TDS IMP 2/2												
1000	8	NEGD E 2/4		TAS IMP 2/2					ADC A,IMM 3(4)/3								
1001	9	ABSD E 2/5		TSA IMP 2/2													
1010	A	EXTZD E 2/3							SBC A,IMM 3(4)/3								
1011	B	EXTSD E 2/5															
1100	C			TXY IMP 2/2					MPY IMM 3(4)/8(注7)								
1101	D			TYX IMP 2/2					MPYS IMM 3(4)/8(注7)								
1110	E			TXS IMP 2/2					DIV IMM 3(4)/15(注8,9)								
1111	F			TSX IMP 2/2					DIVS IMM 3(4)/22(注8,9)								

命令コード対応表 4 (PAGE 4)

D3 - D0 16進 表記	D7 - D4 表記	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	0						LDX DIR, Y 3/5	LDX ABS, Y 4/5									
0001	1											LDY DIR, X 3/5					LDY ABS, X 4/5
0010	2															CPX ABS 4/4	
0011	3															CPY ABS 4/4	
0100	4										BBS DIR, b, REL 5(6)/9					BBS ABS, b, REL 6(7)/9	
0101	5										BBC DIR, b, REL 5(6)/9					BBC ABS, b, REL 6(7)/9	
0110	6										CBEQ DIR/IMM, REL 5(6)/9						
0111	7										CBNE DIR/IMM, REL 5(6)/9						
1000	8											INC DIR, X 3/8					INC ABS, X 4/8
1001	9											DEC DIR, X 3/8					DEC ABS, X 4/8
1010	A																
1011	B																
1100	C																
1101	D																
1110	E						STX DIR, Y 3/6										
1111	F											STY DIR, X 3/6					

命令コード対応表 5 (PAGE 5)

D3 - D0 16進 表記	D7 - D4 表記	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	0			ADDMB DIR/IMM 4/7	ADDM DIR/IMM 4(5)/7			ADDMB ABS/IMM 5/7	ADDM ABS/IMM 5(6)/7								
0001	1			SUBMB DIR/IMM 4/7	SUBM DIR/IMM 4(5)/7			SUBMB ABS/IMM 5/7	SUBM ABS/IMM 5(6)/7								
0010	2			CMPMB DIR/IMM 4/5	CMPM DIR/IMM 4(5)/5			CMPMB ABS/IMM 5/5	CMPM ABS/IMM 5(6)/5								
0011	3			ORAMB DIR/IMM 4/7	ORAM DIR/IMM 4(5)/7			ORAMB ABS/IMM 5/7	ORAM ABS/IMM 5(6)/7								
0100	4																
0101	5																
0110	6			ANDMB DIR/IMM 4/7	ANDM DIR/IMM 4(5)/7			ANDMB ABS/IMM 5/7	ANDM ABS/IMM 5(6)/7								
0111	7			EORMB DIR/IMM 4/7	EORM DIR/IMM 4(5)/7			EORMB ABS/IMM 5/7	EORM ABS/IMM 5(6)/7								
1000	8				ADDMD DIR/IMM 7/10				ADDMD ABS/IMM 8/10								
1001	9				SUBMD DIR/IMM 7/10				SUBMD ABS/IMM 8/10								
1010	A				CMPMD DIR/IMM 7/7				CMPMD ABS/IMM 8/7								
1011	B				ORAMD DIR/IMM 7/10				ORAMD ABS/IMM 8/10								
1100	C																
1101	D																
1110	E				ANDMD DIR/IMM 7/10				ANDMD ABS/IMM 8/10								
1111	F				EORMD DIR/IMM 7/10				EORMD ABS/IMM 8/10								

命令コード対応表 6 (PAGE 6)

D7 - D4 16進 表記	D3 - D0 16進 表記	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	0	MOVRR DIR/IMM $2n+2/5n+3$ (注14)															→
0001	1	MOVR DIR/IMM $2n(3n+2/5n+3)$ (注14)															→
0010	2	MOVRR ABS/IMM $3n+2/4n+3$ (注14)															→
0011	3	MOVR ABS/IMM $3n(4n+2/4n+3)$ (注14)															→
0100	4	MOVRR DIR/DIR $2n+2/6n+3$ (注14)															→
0101	5	MOVR DIR/DIR $2n+2/6n+3$ (注14)															→
0110	6	MOVRR ABS/DIR $3n+2/5n+3$ (注14)															→
0111	7	MOVR ABS/DIR $3n+2/5n+3$ (注14)															→
1000	8	MOVRR DIR/ABS $3n+2/6n+3$ (注14)															→
1001	9	MOVR DIR/ABS $3n+2/6n+3$ (注14)															→
1010	A	MOVRR ABS/ABS $4n+2/5n+3$ (注14)															→
1011	B	MOVR ABS/ABS $4n+2/5n+3$ (注14)															→
1100	C																
1101	D																
1110	E																
1111	F																

命令コード対応表 7 (PAGE 7)

D7 - D4 16進 表記	D3 - D0 16進 表記	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	0	MOVRR DIR/ABS.X $3n+2/6n+3$ (注14)															→
0001	1	MOVR DIR/ABS.X $3n+2/6n+3$ (注14)															→
0010	2																
0011	3																
0100	4																
0101	5																
0110	6	MOVRR ABS/DIR.X $3n+2/6n+3$ (注14)															→
0111	7	MOVR ABS/DIR.X $3n+2/6n+3$ (注14)															→
1000	8									BSS DIR,b,REL 4/11							
1001	9																
1010	A									BSS DIR,b,REL 4/11							
1011	B																
1100	C									BSS ABS,b,REL 5/10							
1101	D																
1110	E									BSS ABS,b,REL 5/10							
1111	F																

命令コード対応表 8 (PAGE 8)

D7 - D4 16進 表記	D3 - D0 16進 表記	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	0								LSR, #n A 2/imm+6 (注15)								
0001	1																
0010	2								ROR, #n A 2/imm+6 (注15)								
0011	3																
0100	4								ASL, #n A 2/imm+6 (注15)								
0101	5																
0110	6								ROL, #n A 2/imm+6 (注15)								
0111	7																
1000	8								ASR, #n A 2/imm+6 (注15)								
1001	9																
1010	A								DEBNE DIR/IMM_REL 4/12								
1011	B																
1100	C																
1101	D																
1110	E																
1111	F																

命令コード対応表 9 (PAGE 9)

D7 - D4 16進 表記	D3 - D0 16進 表記	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	0									LSRD, #n E 2/imm+8 (注16)							
0001	1																
0010	2									RORD, #n E 2/imm+8 (注16)							
0011	3																
0100	4									ASLD, #n E 2/imm+8 (注16)							
0101	5																
0110	6									ROLD, #n E 2/imm+8 (注16)							
0111	7																
1000	8									ASRD, #n E 2/imm+8 (注16)							
1001	9																
1010	A																
1011	B																
1100	C																
1101	D																
1110	E									DEBNE ABS/IMM_REL 5/11							
1111	F																

命令コード対応表 10 (PAGE 10)

D3-D0 16進 表記	D7-D4 16進 表記	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	0																
0001	1								ADDX IMM 2/2								
0010	2								ADDX IMM 2/2								
0011	3																
0100	4								SUBX IMM 2/2								
0101	5																
0110	6								SUBY IMM 2/2								
0111	7																
1000	8								BSS A,b,REL 3/7								
1001	9																
1010	A								BSC A,b,REL 3/7								
1011	B																
1100	C								DXBNE IMM,REL 3/7								
1101	D																
1110	E								DYBNE IMM,REL 3/7								
1111	F																

命令コード対応表 0-B (PAGE 0-B)

D3-D0 16進 表記	D7-D4 16進 表記	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	0				ASL B 2/2					LDAB B,(DIR),Y 3/7	LDAB B,L(DIR),Y 3/9	LDAB B,DIR 3/4	LDAB B,DIR,X 3/5	LDAB B,ABL 5/5	LDAB B,ABL,X 5/6	LDAB B,ABS 4/4	LDAB B,ABS,X 4/5
0001	1			ROL B 2/2			LDA B,IMM 3(4)/2			LDA B,(DIR),Y 3/7	LDA B,L(DIR),Y 3/9	LDA B,DIR 3/4	LDA B,DIR,X 3/5	LDA B,ABL 5/5	LDA B,ABL,X 5/6	LDA B,ABS 4/4	LDA B,ABS,X 4/5
0010	2			ANDB B,IMM 3/2	NEG B 2/2		ADD B,IMM 3(4)/2			LDAB B,IMM 3/2	ADDB B,IMM 3/2	ADD B,DIR 3/4	ADD B,DIR,X 3/5			ADD B,ABS 4/4	ADD B,ABS,X 4/5
0011	3			EORB B,IMM 3/2	EXTZ B 2/2	EXTS B 2/2	SUB B,IMM 3(4)/2			CMPB B,IMM 3/2	SUBB B,IMM 3/2	SUB B,DIR 3/4	SUB B,DIR,X 3/5			SUB B,ABS 4/4	SUB B,ABS,X 4/5
0100	4			LSR B 2/2	CLRB B 2/2		CMP B,IMM 3(4)/2					CMP B,DIR 3/4	CMP B,DIR,X 3/5			CMP B,ABS 4/4	CMP B,ABS,X 4/5
0101	5			ROR B 2/2	CLR B 2/2		ORA B,IMM 3(4)/2					ORA B,DIR 3/4	ORA B,DIR,X 3/5			ORA B,ABS 4/4	ORA B,ABS,X 4/5
0110	6			ORAB B,IMM 3/2	ASR B 2/2		AND B,IMM 3(4)/2					AND B,DIR 3/4	AND B,DIR,X 3/5			AND B,ABS 4/4	AND B,ABS,X 4/5
0111	7						EOR B,IMM 3(4)/2					EOR B,DIR 3/4	EOR B,DIR,X 3/5			EOR B,ABS 4/4	EOR B,ABS,X 4/5
1000	8						PHB STK 2/5										
1001	9						PLB STK 2/5										
1010	A			CBEOB B/IMM,REL 4/7	INC B 2/2	TXB IMP 2/2		CBEO B/IMM,REL 4(5)/7									
1011	B			CBNEB B/IMM,REL 4/7	DEC B 2/2	TYB IMP 2/2		CBNE B/IMM,REL 4(5)/7									
1100	C					TBX IMP 2/2				STAB B,(DIR),Y 3/8	STAB B,L(DIR),Y 3/10	STAB B,DIR 3/5	STAB B,DIR,X 3/6	STAB B,ABL 5/6	STAB B,ABL,X 5/7	STAB B,ABS 4/5	STAB B,ABS,X 4/6
1101	D					TBY IMP 2/2				STA B,(DIR),Y 3/8	STA B,L(DIR),Y 3/10	STA B,DIR 3/5	STA B,DIR,X 3/6	STA B,ABL 5/6	STA B,ABL,X 5/7	STA B,ABS 4/5	STA B,ABS,X 4/6
1110	E		ABS B 2/4														
1111	F																

命令コード対応表 1-B (PAGE 1-B)

D7 - D4 16進 表記	D3 - D0 16進 表記	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	0	LDAB B,(DIR) 3/6	LDAB B,(DIR,X) 3/7	LDAB B,L(DIR) 3/8	LDAB B,SR 3/5	LDAB B,(SR),Y 3/8		LDAB B,ABS,Y 4/5									
0001	1	LDA B,(DIR) 3/6	LDA B,(DIR,X) 3/7	LDA B,L(DIR) 3/8	LDA B,SR 3/5	LDA B,(SR),Y 3/8		LDA B,ABS,Y 4/5									
0010	2	ADD B,(DIR) 3/6	ADD B,(DIR,X) 3/7	ADD B,L(DIR) 3/8	ADD B,SR 3/5	ADD B,(SR),Y 3/8		ADD B,ABS,Y 4/5		ADD B,(DIR),Y 3/7	ADD B,L(DIR),Y 3/9			ADD B,ABL 5/5	ADD B,ABL,X 5/6		
0011	3	SUB B,(DIR) 3/6	SUB B,(DIR,X) 3/7	SUB B,L(DIR) 3/8	SUB B,SR 3/5	SUB B,(SR),Y 3/8		SUB B,ABS,Y 4/5		SUB B,(DIR),Y 3/7	SUB B,L(DIR),Y 3/9			SUB B,ABL 5/5	SUB B,ABL,X 5/6		
0100	4	CMP B,(DIR) 3/6	CMP B,(DIR,X) 3/7	CMP B,L(DIR) 3/8	CMP B,SR 3/5	CMP B,(SR),Y 3/8		CMP B,ABS,Y 4/5		CMP B,(DIR),Y 3/7	CMP B,L(DIR),Y 3/9			CMP B,ABL 5/5	CMP B,ABL,X 5/6		
0101	5	ORA B,(DIR) 3/6	ORA B,(DIR,X) 3/7	ORA B,L(DIR) 3/8	ORA B,SR 3/5	ORA B,(SR),Y 3/8		ORA B,ABS,Y 4/5		ORA B,(DIR),Y 3/7	ORA B,L(DIR),Y 3/9			ORA B,ABL 5/5	ORA B,ABL,X 5/6		
0110	6	AND B,(DIR) 3/6	AND B,(DIR,X) 3/7	AND B,L(DIR) 3/8	AND B,SR 3/5	AND B,(SR),Y 3/8		AND B,ABS,Y 4/5		AND B,(DIR),Y 3/7	AND B,L(DIR),Y 3/9			AND B,ABL 5/5	AND B,ABL,X 5/6		
0111	7	EOR B,(DIR) 3/6	EOR B,(DIR,X) 3/7	EOR B,L(DIR) 3/8	EOR B,SR 3/5	EOR B,(SR),Y 3/8		EOR B,ABS,Y 4/5		EOR B,(DIR),Y 3/7	EOR B,L(DIR),Y 3/9			EOR B,ABL 5/5	EOR B,ABL,X 5/6		
1000	8																
1001	9																
1010	A																
1011	B																
1100	C	STAB B,(DIR) 3/7	STAB B,(DIR,X) 3/8	STAB B,L(DIR) 3/9	STAB B,SR 3/6	STAB B,(SR),Y 3/8		STAB B,ABS,Y 4/6									
1101	D	STA B,(DIR) 3/7	STA B,(DIR,X) 3/8	STA B,L(DIR) 3/9	STA B,SR 3/6	STA B,(SR),Y 3/8		STA B,ABS,Y 4/6									
1110	E																
1111	F																

命令コード対応表 2-B (PAGE 2-B)

D7 - D4 16進 表記	D3 - D0 16進 表記	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	0																
0001	1																
0010	2																
0011	3																
0100	4																
0101	5																
0110	6																
0111	7																
1000	8	ADC B,(DIR) 3/9	ADC B,(DIR,X) 3/10	ADC B,L(DIR) 3/11	ADC B,SR 3/8	ADC B,(SR),Y 3/11		ADC B,ABS,Y 4/8		ADC B,(DIR),Y 3/10	ADC B,L(DIR),Y 3/12	ADC B,DIR 3/7	ADC B,DIR,X 3/8	ADC B,ABL 5/8	ADC B,ABL,X 5/9	ADC B,ABS 4/7	ADC B,ABS,X 4/8
1001	9																
1010	A	SBC B,(DIR) 3/9	SBC B,(DIR,X) 3/10	SBC B,L(DIR) 3/11	SBC B,SR 3/8	SBC B,(SR),Y 3/11		SBC B,ABS,Y 4/8		SBC B,(DIR),Y 3/10	SBC B,L(DIR),Y 3/12	SBC B,DIR 3/7	SBC B,DIR,X 3/8	SBC B,ABL 5/8	SBC B,ABL,X 5/9	SBC B,ABS 4/7	SBC B,ABS,X 4/8
1011	B																
1100	C																
1101	D																
1110	E																
1111	F																

命令コード対応表 3-B (PAGE 3-B)

D7-D4 16進 表記	D3-D0	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	0			TBD,0 IMP 2/3													
0001	1			TBD,1 IMP 2/3								ADCB B,IMM 3/3	SBCB B,IMM 3/3				
0010	2			TBD,2 IMP 2/3													
0011	3			TBD,3 IMP 2/3													
0100	4			TDB,0 IMP 2/2													
0101	5			TDB,1 IMP 2/2													
0110	6			TDB,2 IMP 2/2													
0111	7			TDB,3 IMP 2/2													
1000	8			TBS IMP 2/2					ADC B,IMM 3(4)/3								
1001	9			TSB IMP 2/2													
1010	A								SBC B,IMM 3(4)/3								
1011	B																
1100	C																
1101	D																
1110	E																
1111	F																

【16進命令コード対応表参照時の注意】

表中の命令サイクル数は最短の場合を示しています。

DPRn (n=0~3)に関するアドレッシングモードのサイクル数は、DPRnL=0の場合の値です。DPRnL=0の場合は、サイクル数を+1します。

サイクル数は、上記の場合以外に、命令キューバッファに何バイトの命令が取り込まれているか、又はアクセスするメモリが偶数番地であるか奇数番地であるかによって変化します。また、BYTE="H"で外部領域をアクセスした場合も変化します。

注 1 . BRK命令はデバッグツール用の予約命令です。エミュレータ使用時には使用できません。

2 . $3i + 13$ i : 復帰するレジスタの個数

3 . PLD n : 11, PLD(n₁,...,n_i) : $3i + 8$ (n₁,...,n_i)の括弧内 : 0~3(DPRnの番号)
 RTLD n : 15, RTLD(n₁,...,n_i) : $3i + 12$ i : 指定するDPRnの数(1~4)
 RTSD n : 14, RTSD(n₁,...,n_i) : $3i + 11$

4 . $2i_1 + i_2 + 11$ 退避するレジスタに対応したサイクル数を追加してください。
i₁ : A, B, X, Y, DPR0, PSのうち退避するレジスタの個数
i₂ : DT, PGのうち退避するレジスタの個数

- 5 . LDD n : 4, LDD(n_1, \dots, n_i) : $2i + 2$ (n_1, \dots, n_i の括弧内 : 0 ~ 3(DPRnの番号)
 PHD n : 2, PHD(n_1, \dots, n_i) : 2 i : 指定する DPRn の数(1 ~ 4)
 PHLD n : 4, PHLD(n_1, \dots, n_i) : $2i + 2$
- 6 . LDD n : 13, LDD(n_1, \dots, n_i) : $2i + 11$ (n_1, \dots, n_i の括弧内 : 0 ~ 3(DPRnの番号)
 PHD n : 12, PHD(n_1, \dots, n_i) : $i + 11$ i : 指定する DPRn の数(1 ~ 4)
 PHLD n : 14, PHLD(n_1, \dots, n_i) : $3i + 11$
- 7 . サイクル数は8ビット×8ビット演算の場合の値です。
 16ビット×16ビット演算の場合は、サイクル数を+4してください。
- 8 . サイクル数は16ビット÷8ビット演算の場合の値です。
 32ビット÷16ビット演算の場合は、サイクル数を+8してください。
- 9 . 0除算割り込みが発生した場合のサイクル数は、演算のデータ長にかかわらず、16サイクルになります。
- 10 . n : immで指定する回転数
 m = 0のとき、n = 0 ~ 65535
 m = 1のとき、n = 0 ~ 255
- 11 . サイクル数は、転送バイト数(i)が偶数の場合の値です。
 転送バイト数(i)が奇数の場合のサイクル数は、 $5i + 14$ となります。
 ただし、1バイト転送時は10サイクルです。
- 12 . サイクル数は、転送バイト数(i)が偶数の場合の値です。
 転送バイト数(i)が奇数の場合のサイクル数は、 $5i + 10$ となります。
- 13 . サイクル数はm = 1の場合の値です。
 m = 0の場合のサイクル数は、 $18imm + 5$ (imm : 繰り返し回数, 0 ~ 255)になります。
- 14 . n = 0 ~ 15
- 15 . imm = 0 ~ 15
- 16 . imm = 0 ~ 31

Memo

EOL announced Product

EOL announced Product

三菱16ビットシングルチップマイクロコンピュータ
ソフトウェアマニュアル
7900シリーズ

2001年5月22日 Rev.2.0発行
発行所 三菱電機株式会社半導体営業統括部
〒100-8310 東京都千代田区丸の内2-2-3
TEL 03-3218-9450

禁無断転載

本説明書の一部又は全部を、当社に断りなく、いかなる形でも転載又は複製
することを堅くお断りします。

© 2001 MITSUBISHI ELECTRIC CORPORATION

EOL announced Product

7900 シリーズ
ソフトウェアマニュアル



ルネサスエレクトロニクス株式会社
神奈川県川崎市中原区下沼部1753 〒211-8668

改訂履歴

7900シリーズソフトウェアマニュアル

Rev.	発行日	改訂内容	
		ページ	ポイント
1.0	9804		印刷物(印刷番号: HU-084A)発行
2.0	010522		PDFファイル初版発行(印刷物なし) Rev.1.0(印刷物)からの改訂箇所は下記の通りです。表現のみの変更は除きます。 具体的な改訂箇所・改訂内容は、「前版との差分資料」を参照してください。 なお、テクニカルニュースで発行した関連情報は下記のように示しています。 ・REV.D...No.M7700-140-0010掲載の印刷物の正誤表<REV.D>の内容 ・...No.M7700-124-9911掲載の注意事項「4.4」として本文に追加のある命令
		目次の前	「ソフトウェアマニュアルご使用の前に.....」追加
		2-3	2.1.1(1)、2.1.2:一部削除(各命令(第4章)で記載したため)
		2-4	2.1.4 4行目:「メモリ空間」を「アクセス空間」に変更(ユーザーズマニュアルと統一) ・23行目:リセット時の「S」の内容変更(機種によって異なるため)
		2-5	・2.1.5 3行目:リセットのベクトル番地変更(機種によって異なるため) ・2.1.6 1行目:「メモリ空間」を「アクセス空間」に変更(ユーザーズマニュアルと統一) ・7行目:リセット時の「PG」の値変更(機種によって異なるため)
		2-7	3行目:リセット時の「DPRn」の値変更(REV.D)
		2-10	・2.2 タイトル:「メモリ空間」を「アクセス空間」に変更(ユーザーズマニュアルと統一) ・1, 4, 5行目:「メモリ空間」を「アクセス空間」に変更(ユーザーズマニュアルと統一) ・図2.2.1 タイトル内:「メモリ空間」を「アクセス空間」に変更(ユーザーズマニュアルと統一) ・注1, 2追加(REV.D)
		2-16	・【機能】(2箇所)の後:補足(REV.D) ・図タイトルを「<ダイレクトページレジスタ切り替えビットが0の場合>」に変更(REV.D)
		2-17	図タイトルを「<ダイレクトページレジスタ切り替えビットが1の場合>」に変更(REV.D)
		2-18	図の上にタイトル「<ダイレクトページレジスタ切り替えビットが0の場合>」を追加
		2-21	ただし、REV.Dで掲載したタイトル以外の追加分は未反映(REV.D)
		2-22	
		2-24	
		2-27	
		2-30	
		2-32	
		2-55	
		3-2	3.1.1 9行目:「境界を越えてバンク1をアクセス可能」に修正
		4-3	ブロック転送 MVN、MVP 動作:修正(REV.D)
		4-5	比較 CMPD 動作:修正(REV.D)
		4-6	・無条件分岐 JMP、JMPL 動作:一部削除(REV.D) ・サブルーチン呼び出し ・BSR 動作:修正(REV.D) ・JSR、JSRL 動作:一部削除(REV.D) 及び修正
		4-8	4.2:「図4.2.1 命令の記述例」追加(注2、注3は次ページへ移動)
		4-11	・M(oprd):修正(REV.D) ・dd:修正
		4-17	ADCD 注記追加()
		4-20	ADDD 注記追加()
		4-23	ADDMD 注記追加()
		4-31	ANDMD 注記追加()

Rev.	発行日	改訂内容	
		ページ	ポイント
2.0	010522	4-38	BBC アドレッシングモード：変更(2-11ページと統一のため)(REV.D)
		4-39	BBCB アドレッシングモード：変更(2-11ページと統一のため)(REV.D)
		4-40	BBS アドレッシングモード：変更(2-11ページと統一のため)(REV.D)
		4-41	BBSB アドレッシングモード：変更(2-11ページと統一のため)(REV.D)
		4-54	BRA/BRAL 【記述例】：修正(REV.D)
		4-57	BSR 【動作】：修正(REV.D)
		4-62	CBEQB 【記述例】：修正(REV.D)
		4-64	CBNEB 【記述例】：修正(REV.D)
		4-78	CMPD 注記追加() 【動作】【説明】：修正(REV.D)
		4-81	CMPMD 注記追加()
		4-90	DIVS 【動作】：補足(REV.D)
		4-98	EORMD 注記追加()
		4-100	EXTSD 【動作】：修正(REV.D)
		4-106	JMP/JMPL 【動作】【記述例】：修正(REV.D)
		4-107	JSR/JSRL 【動作】【記述例】：修正(REV.D)
		4-110	LDAD 注記追加()
		4-113	LDX 【記述例】：修正(REV.D)
		4-115	LDY 【記述例】：修正(REV.D)
		4-125	MPYS 【動作】：補足(REV.D)
		4-126	MVN 【動作】【説明】(図)：修正(REV.D) 【説明】：一部追加
		4-127	MVP 【動作】【説明】(図)：修正(REV.D) 【説明】：一部追加
		4-135	ORAMD 注記追加()
		4-137	PEI 【説明】：一部追加(REV.D) 【記述例】：修正(REV.D)
		4-159	PUL サイクル数及び注記：「i」を「i _h 」に変更(REV.D)
		4-162	RMPA 【説明】：修正(REV.D) 及び一部追加
		4-163	ROL 【プロセッサステータスレジスタ】C, 記述形式：修正(REV.D)
		4-166	ROR 【プロセッサステータスレジスタ】C, 記述形式：修正(REV.D)
		4-177	SBCD 注記追加()
		4-184	STAD 注記追加()
		4-190	SUBD 注記追加()
		4-193	SUBMD 注記追加() 【記述例】：修正(REV.D)
		4-207	TDS 【説明】：修正(REV.D)
		4-210	TSD 【説明】：修正(REV.D)
		4-220	XAB 【記述例】：修正(REV.D)
		4-222	「4.4 32ビット演算命令実行時の注意」：追加
		5-10	BSR 動作：修正(REV.D)
		5-14	CMPD 動作：修正(REV.D)
		5-16	EXTSD、EXTZD 動作：修正(REV.D)
		5-18	JSR/JSRL 動作：一部追加(「LDAB」は5-20/5-21ページへ移動)
		5-24	MVN、MVP 動作：修正(REV.D)
		5-31	PUL アドレッシングモード(STK)：修正(REV.D)
		5-43	注18 「i」を「i _h 」に変更(REV.D)