

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、誤りが無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。

740ファミリ

ソフトウェアマニュアル

ルネサスマイクロコンピュータ

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

1. 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

1. 本資料は、お客様が用途に応じた適切なルネサス テクノロジ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ルネサス テクノロジ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。
2. 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、ルネサス テクノロジは責任を負いません。
3. 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、ルネサス テクノロジは、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 テクノロジ半導体製品のご購入に当たりましては、事前にルネサス テクノロジ、ルネサス販売または特約店へ最新の情報をご確認頂きますとともに、ルネサス テクノロジホームページ (<http://www.renesas.com>) などを通じて公開される情報に常にご注意ください。
4. 本資料に記載した情報は、正確を期すため、慎重に制作したものです。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、ルネサス テクノロジはその責任を負いません。
5. 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。ルネサス テクノロジは、適用可否に対する責任を負いません。
6. 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、ルネサス テクノロジ、ルネサス販売または特約店へご照会ください。
7. 本資料の転載、複製については、文書によるルネサス テクノロジの事前の承諾が必要です。
8. 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ルネサス テクノロジ、ルネサス販売または特約店までご照会ください。

改訂記録

740 ファミリ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
1.00	2001.01.06	-	PDF 初版発行 製本版（印刷番号：HU-009C）から以下を変更 “はじめに”を一部変更
		46	注2の文章を変更
		107	4.1 入出力ポートに関する注意事項を追加
		108	4.2 未使用端子の処理に関する注意事項を追加
		110	4.3.2 関連レジスタの設定変更を一部変更
		114	4.4.5 乗除算命令を追加 4.4.6 ポートを追加 4.4.7 命令の実行時間を追加
		186	MUL, DIV 命令に対応していない品種を追加
2.00	2006.08.01	-	ルネサス様式に変更 “はじめに”を“本書の使い方”に変更
		4	プロセッサステータスレジスタ説明文に注意文追加
		33	Vフラグの文言変更 10進演算モードの場合の注意文追加
		46	注2文変更
		62	注3文変更
		67	注意文追加
		74	注意文追加
		80	注意文追加
		86	注意文追加
		108	理由文変更 4.2 未使用端子の処理に関する注意事項に説明文追加 入力専用端子に説明文追加
		109	4.2.1 入出力ポートに説明文追加 4.2.2 入出力ポートの理由の一部削除
		110	図4.3.1中の文言変更 理由文変更
		111	図4.3.2中一部削除
		115	説明文変更 表一部追加、変更
		182	命令コード修正
		186	表中命令修正 対応していない品種表削除

本書の使い方

本書は740ファミリのソフトウェアマニュアルです。740ファミリのCPUコアをもつ全品種で共通に使用することができます。ただし、7600シリーズをご使用のユーザの皆様は、本マニュアルではなく「7600シリーズ和文ソフトウェアマニュアル(印刷番号：HU-056A)」を参照してください。

本書を使用する上で、電気回路、論理回路、およびマイクロコンピュータの基本的な知識が必要です。

740ファミリ関連ドキュメント

740ファミリでは次のドキュメントを用意しています。

ドキュメントの種類	記載内容
データシート	ハードウェアの概要と電気的特性 ハードウェアの仕様(ピン配置、メモリマップ、周辺機能の仕様、電気的特性、タイミング)
ソフトウェアマニュアル	命令(アセンブリ言語)の動作の詳細
アプリケーションノート	周辺機能の応用例 参考プログラム

目 次

1 . 概 要	1
2 . 中央演算処理装置	2
2.1 アキュムレータ(A)	3
2.2 インデックスレジスタX(X)、インデックスレジスタY(Y)	3
2.3 スタックポインタ(S)	3
2.4 プログラムカウンタ(PC)	3
2.5 プロセッサステータスレジスタ(PS)	4
3 . 命令の説明	7
3.1 アドレッシングモード(Addressing mode)	7
3.2 命令の体系	27
3.2.1 データ転送命令	27
3.2.2 演算命令	28
3.2.3 ビット処理命令	29
3.2.4 フラグ設定命令	29
3.2.5 ジャンプ、ブランチ、リターン命令	30
3.2.6 割り込み命令(ブレーク命令)	31
3.2.7 特殊命令	31
3.2.8 その他の命令	31
3.3 命令の説明	32
3.4 割り込み処理、サブルーチン処理に関する命令	104
3.4.1 割り込み処理に関する命令	104
3.4.2 割り込み制御に関する命令	104
3.4.3 サブルーチン処理に関する命令	105
4 . 使用上の注意事項	107
4.1 入出力ポートに関する注意事項	107
4.1.1 スタンバイ状態での使用	107
4.1.2 ビット処理命令による出力データの書き替え	108
4.2 未使用端子の処理に関する注意事項	108
4.2.1 未使用端子の適切な処理	108
4.2.2 処理上の留意事項	109
4.3 割り込みに関する注意事項	110
4.3.1 割り込み要求ビット及び許可ビットの設定	110
4.3.2 関連レジスタの設定変更	110
4.3.3 割り込み要求ビットの判定	111
4.4 プログラム作成に関する注意事項	112
4.4.1 プロセッサステータスレジスタ	112
4.4.2 BRK命令	113

4.4.3	10進演算	113
4.4.4	JMP命令	114
4.4.5	乗除算命令	114
4.4.6	ポート	114
4.4.7	命令の実行時間	114
付録 1 . アドレッシングモード別実行シーケンス		115
付録 2 .740ファミリ機械語命令一覧表		181
付録 3 .740ファミリ命令コード対応表		186

《アドレッシングモード》

イミディエイト	8	絶対 Y	15	スペシャルページ	22
アキュムレータ	9	インプライド	16	ゼロページ ビット	23
ゼロページ	10	相対	17	アキュムレータ ビット	24
ゼロページ X	11	間接 X	18	アキュムレータ ビット 相対 ..	25
ゼロページ Y	12	間接 Y	19	ゼロページ ビット 相対	26
絶対	13	間接	20		
絶対 X	14	ゼロページ 間接	21		

《命 令》

ADC	33	CLD	51	LDA	69	SEB	87
AND	34	CLI	52	LDM	70	SEC	88
ASL	35	CLT	53	LDX	71	SED	89
BBC	36	CLV	54	LDY	72	SEI	90
BBS	37	CMP	55	LSR	73	SET	91
BCC	38	COM	56	MUL	74	STA	92
BCS	39	CPX	57	NOP	75	STP	93
BEQ	40	CPY	58	ORA	76	STX	94
BIT	41	DEC	59	PHA	77	STY	95
BMI	42	DEX	60	PHP	78	TAX	96
BNE	43	DEY	61	PLA	79	TAY	97
BPL	44	DIV	62	PLP	80	TST	98
BRA	45	EOR	63	ROL	81	TSX	99
BRK	46	INC	64	ROR	82	TXA	100
BVC	47	INX	65	RRF	83	TXS	101
BVS	48	INY	66	RTI	84	TYA	102
CLB	49	JMP	67	RTS	85	WIT	103
CLC	50	JSR	68	SBC	86		

1 . 概 要

CMOS 8ビットマイクロコンピュータ740ファミリのソフトウェアの特長を要約すると次のようになります。

ROM領域が有効に使える効率的な命令群と豊富なアドレッシングモード
アキュムレータ、メモリ、I/Oを問わず処理できるビット操作命令と、ビットテスト/ブランチ命令
豊富な割り込みソースと処理機能
バイト単位処理、テーブル参照機能に優れたインデックス アドレッシング機能
ソフトウェア補正を必要としない10進演算機能
アキュムレータを経由せずに実行のできるメモリ間、I/O間、そしてメモリとI/O間の演算機能

2 . 中央演算処理装置

740ファミリのCPU(中央演算処理装置)は図2.1.1に示されるような6個のレジスタを持っています。

プログラムカウンタ(PC)が16ビットで構成されているほかはアキュムレータ(A)、インデックスレジスタX(X)、インデックスレジスタY(Y)、スタックポインタ(S)、プロセッサステータスレジスタ(PS)の5つのレジスタは、すべて8ビットで構成されています。

ハードウェアリセット直後は、Iフラグを除いたこれらのレジスタの内容は不定状態となっており、プログラムによって初期化する必要があります(リセット直後I = 1)。

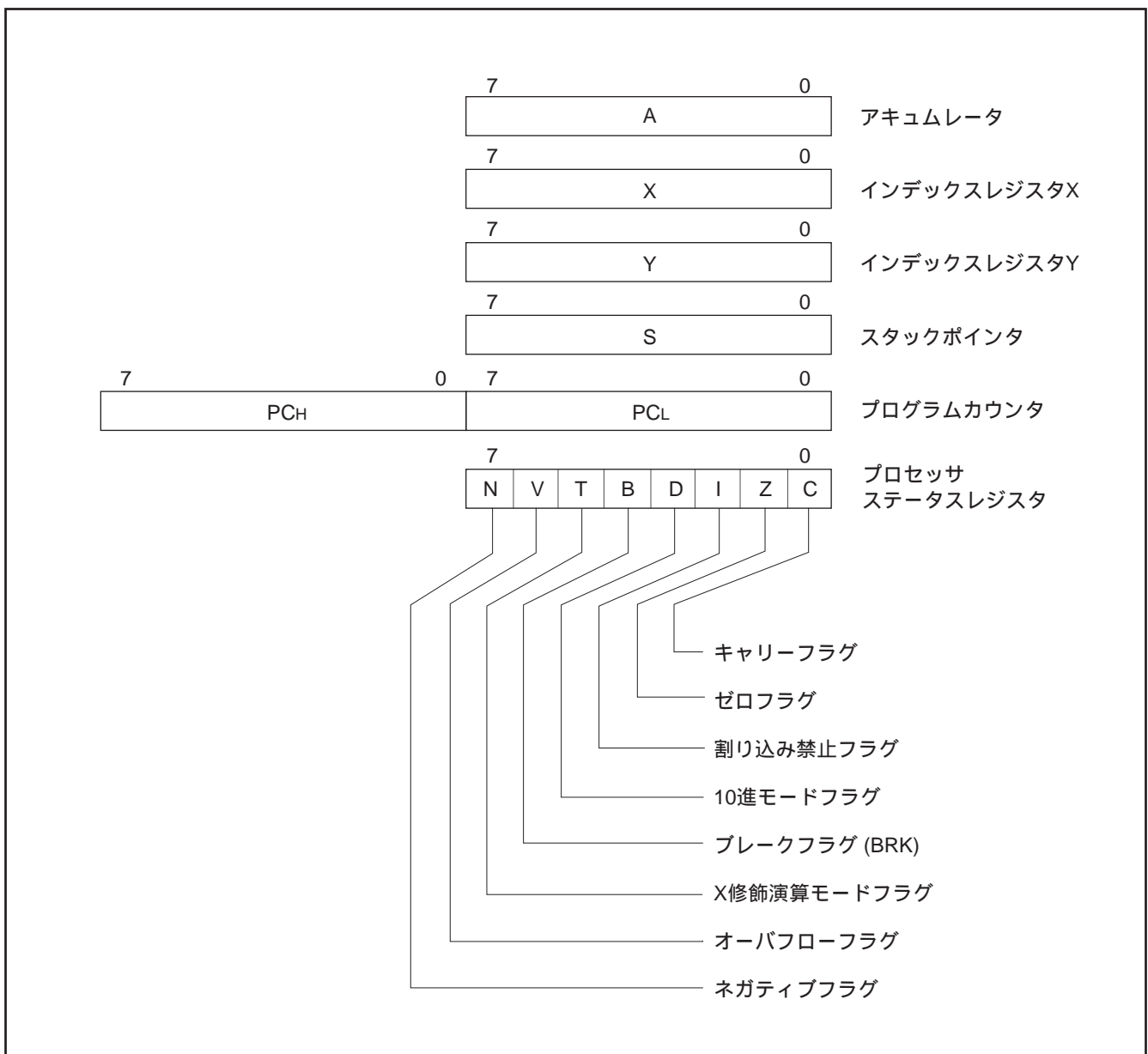


図2.1.1 740ファミリ レジスタ構成図

2.1 アキュムレータ(A)

アキュムレータはマイクロコンピュータの中心となるレジスタで、8ビットで構成されています。算術演算、データ転送、一時退避、条件判断などに用いられ、最も使用頻度の高い汎用のレジスタです。

2.2 インデックスレジスタX(X)、インデックスレジスタY(Y)

740ファミリのCPUは、各8ビットで構成されたインデックスレジスタXとインデックスレジスタYを持っています。

これらのインデックスレジスタを用いたアドレッシングモードを使用した場合、オペランドで指定した番地に、インデックスレジスタの内容を加算した番地がアクセスされます。このモードはサブルーチンテーブル参照やメモリテーブル参照に力を発揮します。

インデックスレジスタにはインクリメント、デクリメント、比較、データ転送機能があるため簡単なアキュムレータとして使用することもできます。

2.3 スタックポインタ(S)

スタックポインタ(S)は8ビットのレジスタです。

スタックポインタは割り込み発生時やサブルーチン呼び出しの際、それらの処理の終了時の戻り先番地(プログラムカウンタ値)などが退避されるメモリ番地を指し示すために使用されます。

割り込み処理、及びサブルーチン処理の際のスタックポインタ(S)の動きは3章4節「割り込み処理、サブルーチン処理に関する命令」を参照してください。

2.4 プログラムカウンタ(PC)

プログラムカウンタはPC_HとPC_Lからなる16ビットのカウンタであり、PC_H、PC_Lはそれぞれ8ビットです。

プログラムカウンタは次に実行する命令の格納されている番地を示しています。

740ファミリのCPUはストアードプログラム方式を採っているため、新しい動作を始めるためには、まず命令及びそれに付随するデータをメモリからCPUへ転送する必要があります。

通常、プログラムカウンタは次に転送すべきメモリ番地を示すようコントロールされており、1つの命令を実行した後、次々に必要な命令を呼び出してくることができます。

このように740ファミリのプログラムカウンタはほとんど自動的にコントロールされます。しかしスタックポインタを操作したり、直接プログラムカウンタの内容を変化させた場合は、プログラムの流れとプログラムカウンタの内容に食い違いが生じないように注意してください。

2.5 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は8ビットのレジスタで、内部CPUに関して演算直後の状態を示す5つのフラグと、動作を決定する3つのフラグで構成されています。リセット直後は、割り込み禁止フラグのみ“1”で、他のフラグは不定です。このため、プログラムの実行に影響を与えるフラグを初期化してください。特に、TフラグとDフラグについては、演算そのものに影響を与えるため、必ず初期化してください。

以下、各フラグについて説明します。なお、各フラグをセット又はクリアする命令の一覧を表2.5.1に示します。また、これらのフラグの変化の様子は、このユーザーズマニュアル内の3章3節「命令の説明」、あるいは付録2「740ファミリ機械語命令一覧表」を参照してください。

《キャリフラグ C》_____ ビット0

演算処理後の算術論理ユニットからのキャリ又はボローを保持します。シフト命令、ローテート命令でも変化します。

“SEC”命令でセットされ、“CLC”命令でクリアされます。

《ゼロフラグ Z》_____ ビット1

演算処理、データ転送の結果が“0”のときセットされ、“0”でないときクリアされます。

《割り込み禁止フラグ I》_____ ビット2

割り込み禁止のためのフラグで、フラグが“1”のときはいっさいの割り込みは禁止されます。割り込みを受け付けると同時に“1”になります。

“SEI”命令でセットされ、“CLI”命令でクリアされます。

《10進演算フラグ D》_____ ビット3

加減算を2進で行うか10進で行うかを定めるフラグです。フラグの内容が“0”のときは通常の2進演算を行い、フラグの内容が“1”のときは1語を2桁の10進数として演算を行います。このとき10進補正は自動的に行われます。

“SED”命令でセットされ、“CLD”命令でクリアされます。

10進演算が行えるのはADC命令とSBC命令のみです。これらの命令により10進演算を行った場合N、V、Zの3つのフラグは無効となりますので注意してください。

《ブレイクフラグ B》_____ ビット4

BRK命令で割り込んだかどうかを識別するためのフラグです。BRK命令で割り込んだ場合は自動的にフラグBの内容を“1”にし、それ以外の割り込みでは“0”にしてスタックに退避します。

《X修飾演算モードフラグ T》 _____ ビット5

演算を行う際にアキュムレータを経由して演算を行うか、メモリとメモリの間で直接演算を行うかを決定するフラグです。フラグの内容が“0”のときはアキュムレータとメモリ間で演算が行われます。また、“1”のときはメモリとメモリの間で直接演算ができます。

このフラグは“SET”命令でセットされ、“CLT”命令でクリアされます。

Tフラグ = 0の場合

A	A * M ₂	[* : 演算を表す。 A : アキュムレータの内容 M ₂ : 演算に付随したアドレッシングによって指定されたメモリの内容]

Tフラグ = 1の場合

M ₁	M ₁ * M ₂	[* : 演算を表す。 M ₁ : インデックスレジスタXにより直接指定されたメモリの内容 M ₂ : 演算に付随したアドレッシングによって指定されたメモリの内容]

《オーバーフローフラグ V》 _____ ビット6

サイン付きの演算の結果オーバーフローが生じたときに“1”にセットされます。加減算の結果が + 127(7F₁₆) または | - 128(80₁₆) | を超えるとオーバーフローが生じます。

オーバーフローフラグをクリアする命令は、“CLV”です。セットする命令はありません。

上記以外にBIT命令を実行した場合、BIT命令が実行されたメモリの“ビット6”がオーバーフローフラグに入ります。

加減算の結果が上記数値より小さい場合や、異符号間の加算ではオーバーフローは生じません。

《ネガティブフラグ N》 _____ ビット7

データ操作あるいは演算処理の結果、その結果のサインビット(ビット7)の状態と等しくなるようにセットされています。BIT命令を実行した場合、実行されたメモリの“ビット7”がネガティブフラグに入ります。このフラグを利用することによって演算結果のプラス・マイナスの判断を行うだけでなく、単純なビットのテストを行うことができます。

表2.5.1 プロセッサステータスレジスタの各フラグをセット又はクリアする命令

	Cフラグ	Zフラグ	Iフラグ	Dフラグ	Bフラグ	Tフラグ	Vフラグ	Nフラグ
“1”にセットする命令	SEC	-	SEI	SED	-	SET	-	-
“0”にクリアする命令	CLC	-	CLI	CLD	-	CLT	CLV	-

3 . 命令の説明

3.1 アドレッシングモード(Addressing mode)

740ファミリは、19種のアドレッシングモードを持っており、たいへん強力なメモリアクセス能力を有しています。

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く必要があります。また、プログラムの進行上、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。

次に、アドレッシングに必要なデータおよびレジスタについて説明します。図3.1.1に示すように、命令を構成するために必要なプログラムメモリ上のバイト数で740ファミリの命令を分類したとき、1バイト命令、2バイト命令、3バイト命令の3種類になります。いずれも1バイト目はオペコード(Operation code)と呼ばれ、命令の基本をなす部分です。2バイト目、あるいは3バイト目はオペランド(Operand)と呼ばれ、アドレッシングに影響する部分です。また、インデックスレジスタX及びYの内容もアドレッシングに影響します。

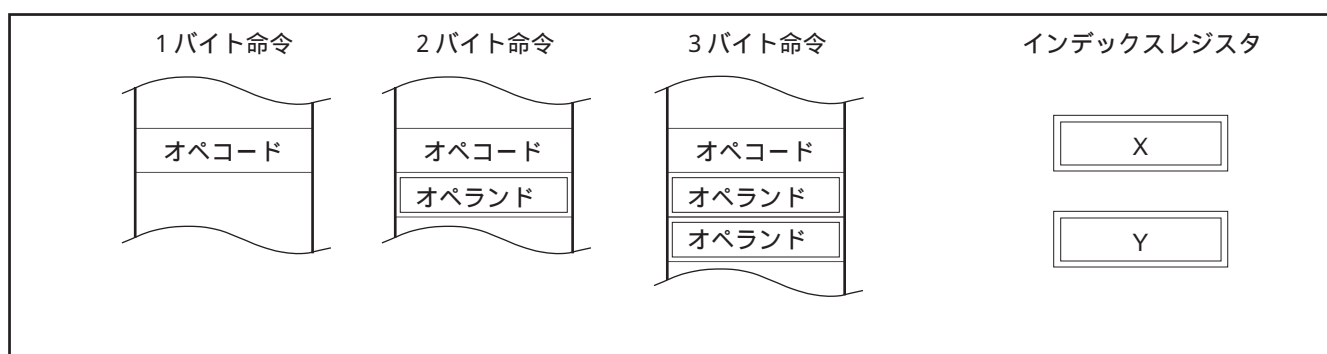


図3.1.1 命令のバイト構成

アドレッシングモードが何種類あろうと、「特定のメモリを指定する」という意味においてはまったく同じことを行っています。違うのは、メモリを指定したり、ジャンプ先を指定するために、オペランドを用いるのか、インデックスレジスタの内容を用いるのか、あるいはこれらの組み合わせによるのか、ということです。この3つの方法を基本として、ビット操作命令、ジャンプ命令、演算命令といった命令の組み合わせによってバリエーションを増やし、740ファミリの操作性能を高めています。1バイト命令の場合は、アキュムレータ又はレジスタを指定しますので、メモリを指定する部分である「オペランド」は持っていません。

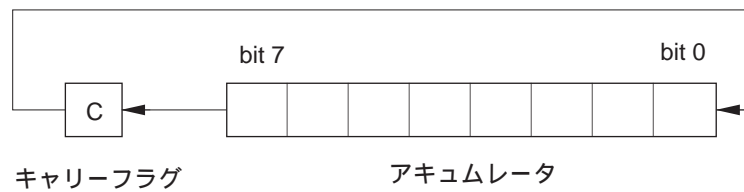
アキュムレータ(Accumulator)

名 称 : アキュムレータ アドレッシングモード

機 能 : 演算データとしてアキュムレータの内容を直接指定します。

該当命令 : ASL, DEC, INC, LSR, ROL, ROR

例 : ニーモニック 機械語
 ROL A 2A₁₆



ゼロページ Y(Zero page Y)

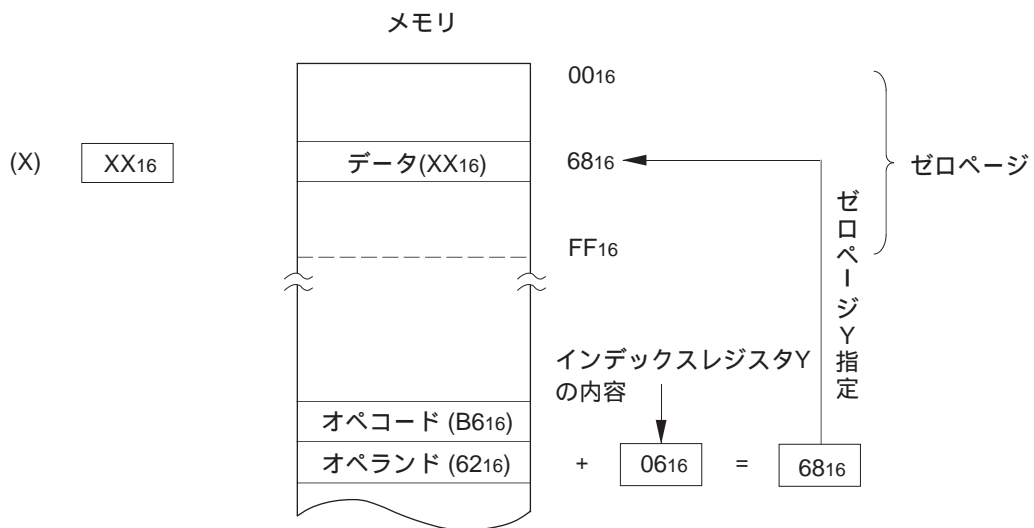
名 称 : ゼロページ Y アドレッシングモード

機 能 : 演算データとして、オペランド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示すゼロページ領域内のメモリ内容を指定します。

該当命令 : LDX, STX

例 : ニーモニック
LDX \$ 62, Y

機械語
B6₁₆ 62₁₆



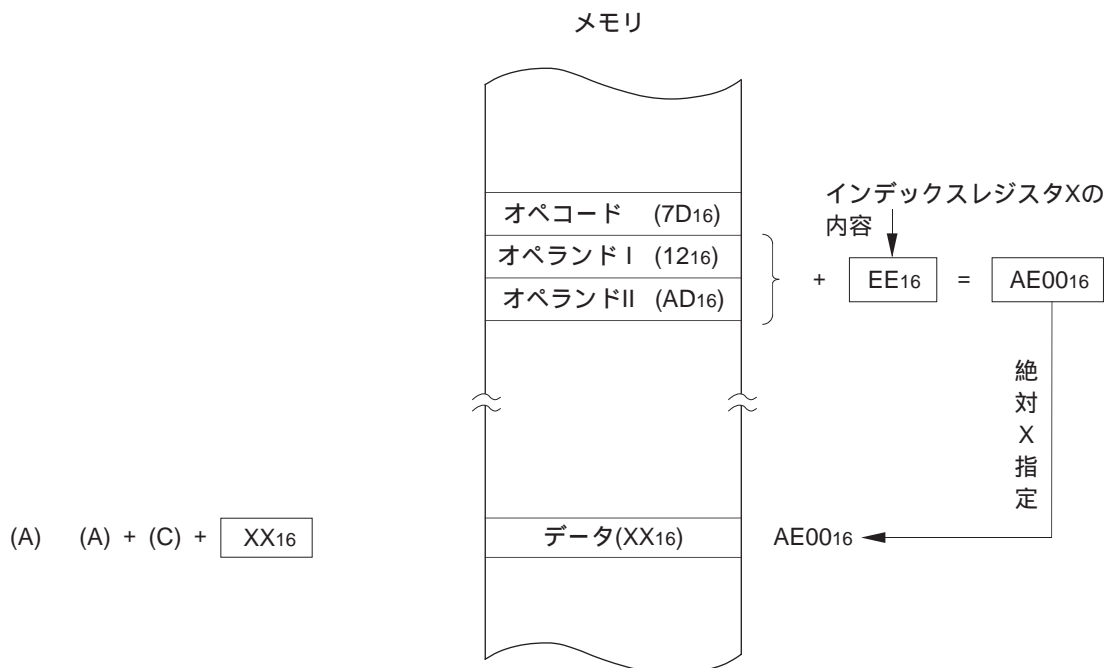
絶対 X(Absolute X)

名称 : アブソリュート X アドレッシングモード

機能 : 演算データとして、第1オペランドと第2オペランドで表わされる数値に、インデックスレジスタXの内容を加えた値で示されるメモリの内容を指定します。

該当命令 : ADC, AND, ASL, CMP, DEC, EOR, INC, LDA, LDY, LSR, ORA, ROL, ROR, SBC, STA

例	: ニーモニック	機械語
	: ADC \$AD12, X	7D ₁₆ 12 ₁₆ AD ₁₆



インプライド(Implied)

名 称 : インプライド アドレッシングモード

機 能 : 1命令で、レジスタなどの操作をします。

該当命令 : BRK, CLC, CLD, CLI, CLT, CLV, DEX, DEY, INX, INY, NOP, PHA, PHP, PLA, PLP, RTI, RTS, SEC, SED, SEI, SET, STP, TAX, TAY, TSX, TXA, TXS, TYA, WIT

例 : ニーモニック 機械語
CLC 1816



キャリーフラグのリセット

相対(Relative)

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、オペランドの内容を加えた値で示される番地にジャンプします。

該当命令 : BCC, BCS, BEQ, BMI, BNE, BPL, BRA, BVC, BVS

例 : ニーモニック

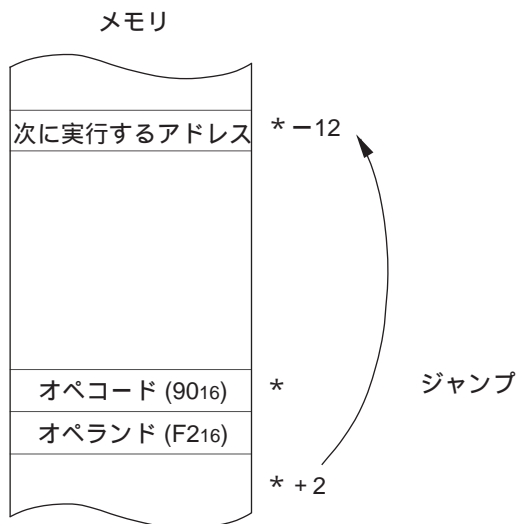
BCC * - 12

┌───┐
↑
10進数

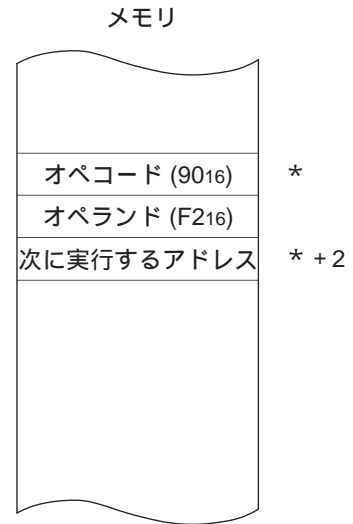
機械語

90₁₆ F2₁₆

キャリフラグ(C)がクリアされているとき *
- 12 番地にジャンプします。



キャリフラグ(C)がセットされているとき *
+ 2 番地に進みます。



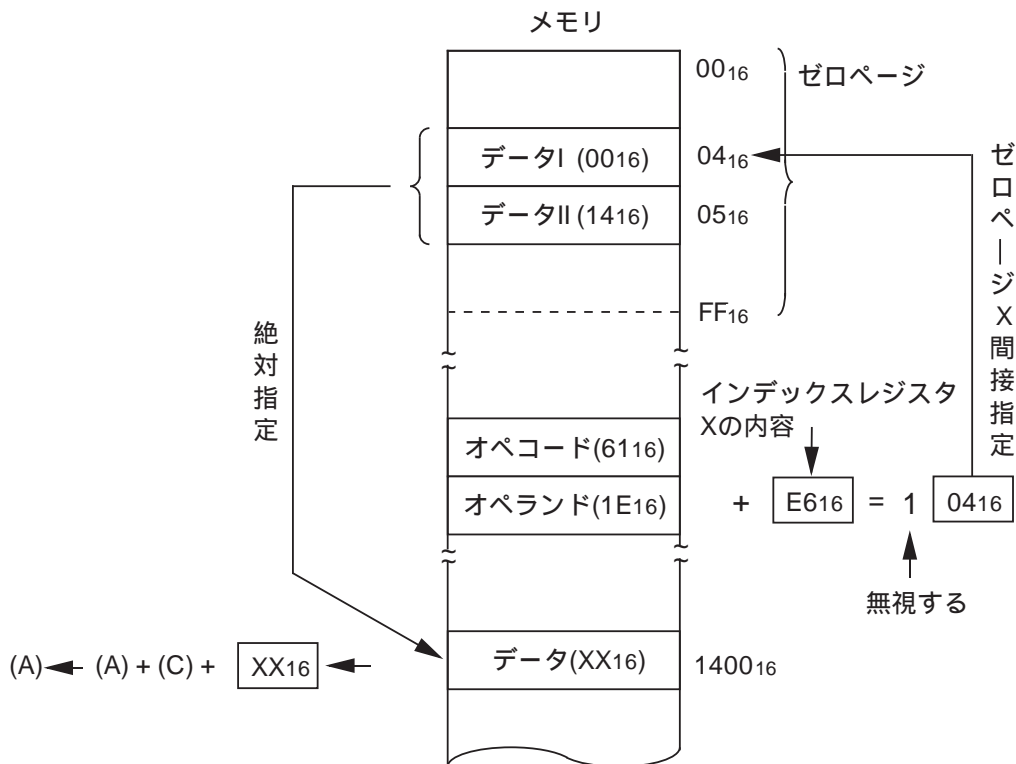
間接 X(Indirect X)

名 称 : インダイレクトX アドレッシングモード

機 能 : オペランドの内容に、インデックスレジスタ X の内容を加えた値でゼロページ領域内の連続する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番地のメモリの内容を、演算データとして指定します。

該当命令 : ADC, AND, CMP, EOR, LDA, ORA, SBC, STA

例	: ニーモニック	機械語
	: ADC (\$ 1E, X)	: 61 ₁₆ 1E ₁₆



この例では、データとして“00₁₆”が、データとして“14₁₆”が、あらかじめ格納されていたものとします。

間接(Indirect)

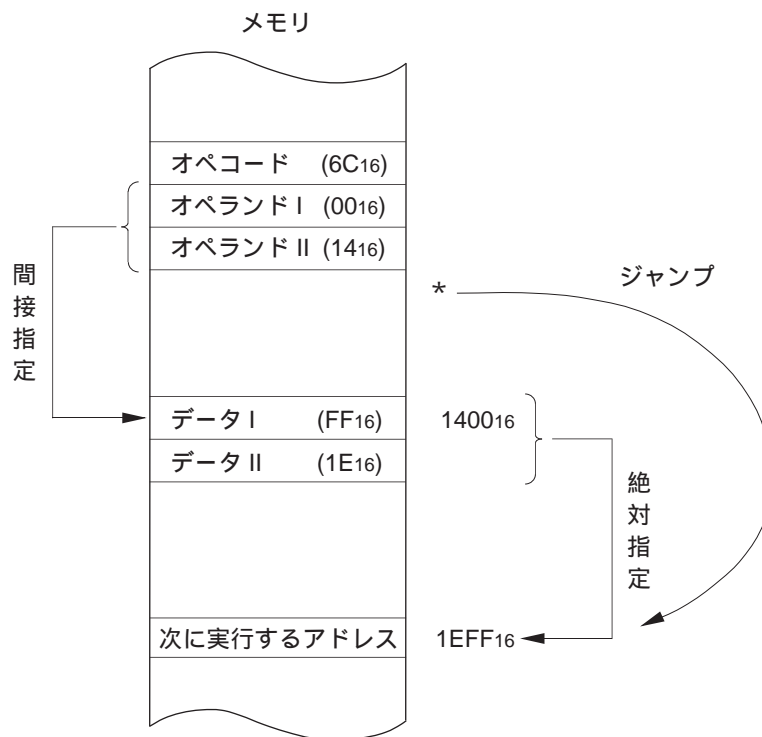
名称 : インダイレクト アブソリュート アドレッシングモード

機能 : 第1オペランドと第2オペランドの内容で、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番地にジャンプします。

該当命令 : JMP

例 : ニーモニック
JMP (\$ 1400)

機械語
6C₁₆ 00₁₆ 14₁₆



この例では、データとして“FF₁₆”が、データとして“1E₁₆”が、あらかじめ格納されていたものとします。

注 .間接指定アドレスとしてページの最終番地(XXFF₁₆番地)は指定できません。つまり、JMP(\$ XXFF)を実行することはできませんので、ご注意ください。

ゼロページ 間接(Zero page indirect)

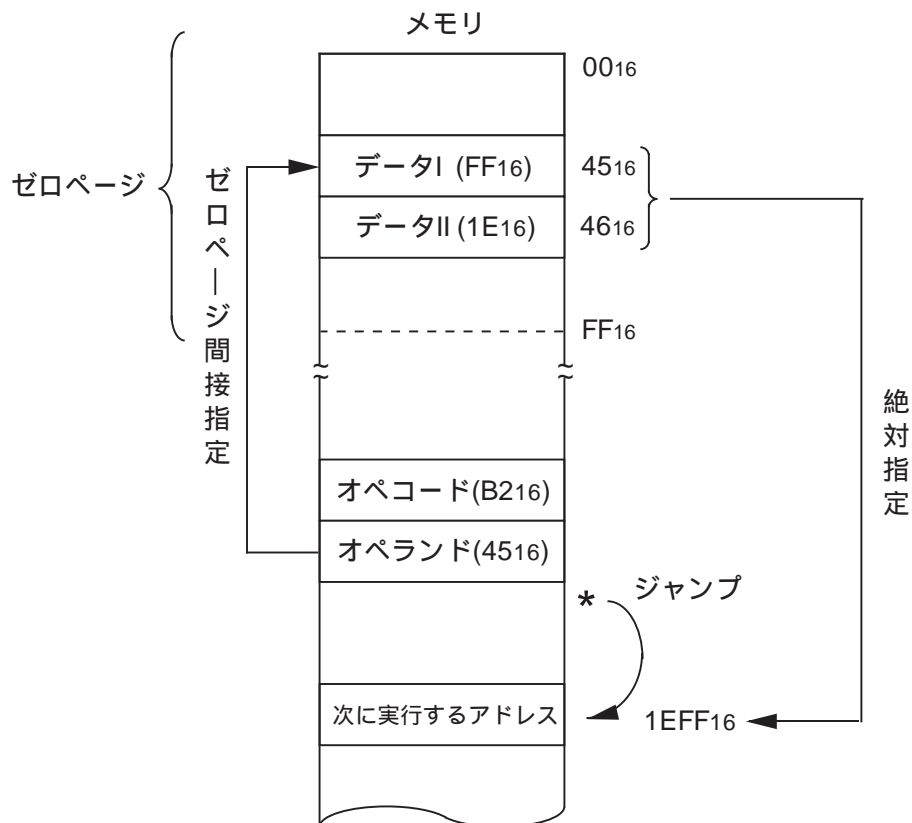
名称 : ゼロページ インダイレクト アブソリュート アドレッシングモード

機能 : オペランドの内容で、ゼロページ領域内の連続する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番地にジャンプします。

該当命令 : JMP, JSR

例 : ニーモニック
JMP (\$ 45)

機械語
B2₁₆ 45₁₆



この例では、データとして“FF₁₆”が、データとして“1E₁₆”が、あらかじめ格納されていたものとします。

ゼロページ ビット(Zero page bit)

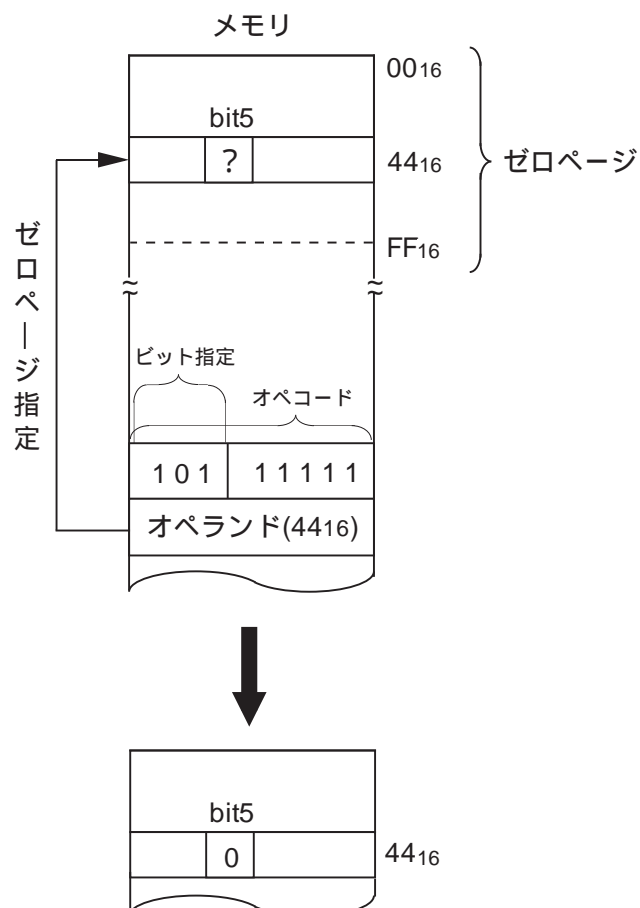
名称 : ゼロページ ビット アドレッシングモード

機能 : 第1オペランドの内容で示されたゼロページ領域のメモリの特定ビットを、オペコードの上位3ビットで指定します。

該当命令 : CLB, SEB

例 : ニーモニック
CLB 5, \$44

機械語
BF₁₆ 44₁₆



アキュムレータ ビット 相対(Accumulator bit relative)

名 称 : アキュムレータ ビット レラティブ アドレッシングモード

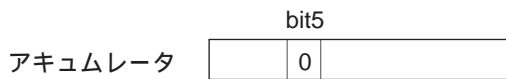
機 能 : オペコードの上位3ビットでアキュムレータのビット位置を指定し、そのビットの状態によって、プログラムカウンタの内容に、オペランドの内容を加えた値で示される番地にジャンプします。

該当命令 : BBC, BBS

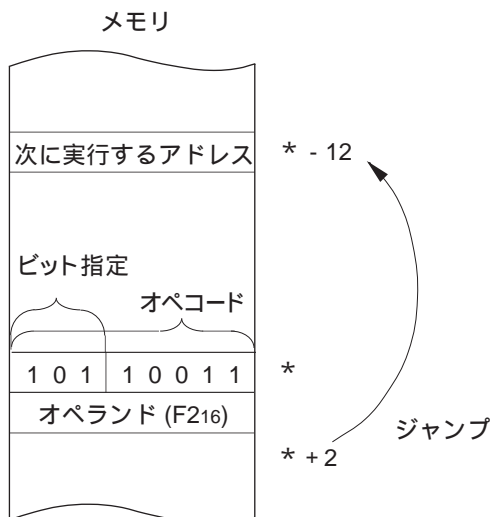
例 : ニーモニック
BBC 5, A, * - 12

機械語
B316 F216

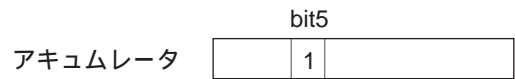
アキュムレータのビット5がクリアされているとき



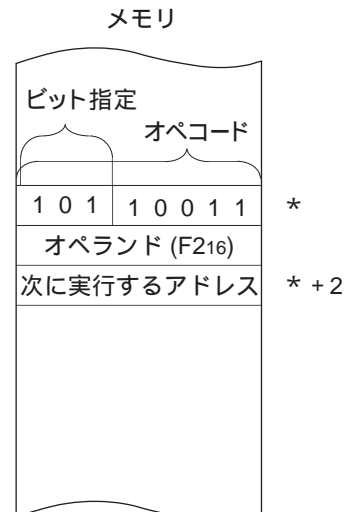
* - 12番地にジャンプします。



アキュムレータのビット5がセットされているとき



* + 2番地に進みます。



ゼロページ ビット 相対(Zero page bit relative)

名 称 : ゼロページ ビット レラティブ アドレッシングモード

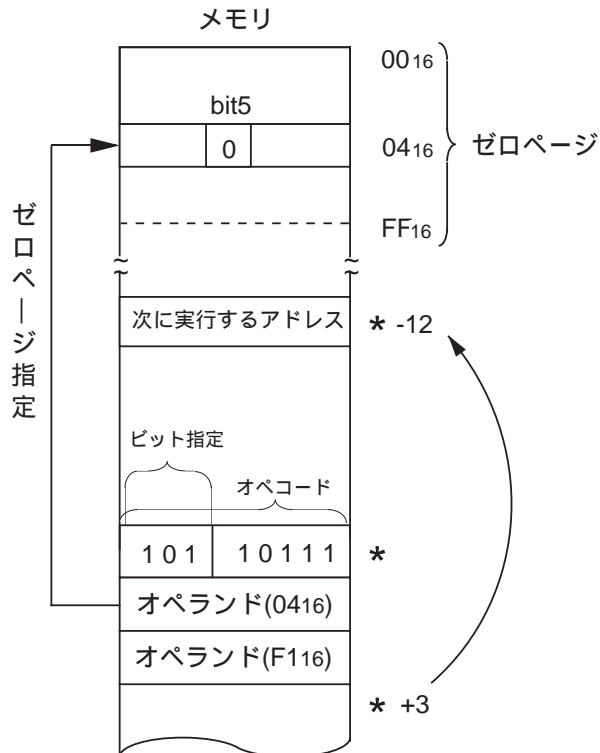
機 能 : 第1オペランドの内容で示されたゼロページ領域のメモリの特定ビットを、オペコードの上位3ビットで指定し、この特定ビットの状態によって、プログラムカウンタの内容に、第2オペランドの内容を加えた値で示される番地にジャンプします。

該当命令 : BBC, BBS

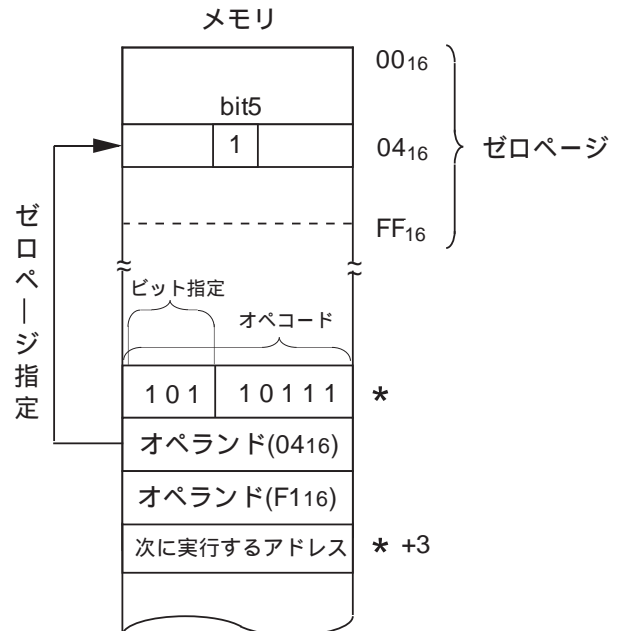
例 : ニーモニック
BBC 5, \$04, * - 12

機械語
B7₁₆ 04₁₆ F1₁₆

04₁₆ 番地のビット5がクリアされているとき * - 12 番地にジャンプします。



04₁₆ 番地のビット5がセットされているとき * + 3 番地に進みます。



3.2 命令の体系

740ファミリは、これから述べるような71種類の命令をもっています。これらの命令の詳細な説明は、3.3節に述べてあります。

マイコンの品種により使用できない命令がありますので注意してください。

3.2.1 データ転送命令

これらの命令は、データをレジスタ・レジスタ間、レジスタ・メモリ間、又は、メモリ・メモリ間で移動させます。

データ転送命令には、以下のようなものがあります。

	命 令	内 容
ロ ー ド	LDA	メモリをアキュムレータ又はインデックスXの示すメモリへロード
	LDM	イミディエイト値をメモリへロード
	LDX	メモリをインデックスXへロード
	LDY	メモリをインデックスYへロード
ス ト ア	STA	アキュムレータをメモリへ格納
	STX	インデックスXをメモリへ格納
	STY	インデックスYをメモリへ格納
転 送	TAX	アキュムレータをインデックスXへ転送
	TXA	インデックスXをアキュムレータへ転送
	TAY	アキュムレータをインデックスYへ転送
	TYA	インデックスYをアキュムレータへ転送
	TSX	スタックポインタをインデックスXへ転送
	TXS	インデックスXをスタックポインタへ転送
ス 操 タ ッ ク 作	PHA	アキュムレータをスタックへ退避
	PHP	プロセッサステータスをスタックへ退避
	PLA	アキュムレータをスタックから復帰
	PLP	プロセッサステータスをスタックから復帰

3.2.2 演算命令

これらの命令は、レジスタやメモリにあるデータの、加減、論理、比較、回転、シフトの演算を行います。

演算命令には、以下のようなものがあります。

	命 令	内 容
加 減 算	ADC	メモリとCフラグを共にアキュムレータ、又はインデックスXの示すメモリへ加算
	SBC	アキュムレータ、又はインデックスXの示すメモリからメモリとCフラグの補数を共に減算
	INC	メモリ又はアキュムレータに1を加算
	DEC	メモリ又はアキュムレータから1を減算
	INX	インデックスXに1を加算
	DEX	インデックスXから1を減算
	INY	インデックスYに1を加算
	DEY	インデックスYから1を減算
乗 除 算	MUL (注)	アキュムレータの内容とゼロページXアドレッシングモードによって指定されたメモリの内容とを掛け合わせ、積の上位8ビットをスタックに、下位8ビットをアキュムレータに格納
	DIV (注)	ゼロページXアドレッシングモードによって指定されたメモリとその次の番地のメモリの内容で表されるワードデータをアキュムレータの内容で割り、商をアキュムレータ、剰余の1の補数をスタックに格納
論 理 演 算	AND	メモリとアキュムレータ又はインデックスXの示すメモリとの論理AND
	ORA	メモリとアキュムレータ又はインデックスXの示すメモリとの論理OR
	EOR	メモリとアキュムレータ又はインデックスXの示すメモリとの排他的論理OR
	COM	メモリの1の補数をメモリに格納
	BIT	メモリとアキュムレータとの論理AND(結果はどこにも格納されません)
	TST	メモリの内容が“0”かどうかのテスト
比 較	CMP	メモリとアキュムレータ又はインデックスXの示すメモリとの比較
	CPX	メモリとインデックスXとの比較
	CPY	メモリとインデックスYとの比較

注. 一部の品種では、乗除算命令が使用できません。

	命 令	内 容
シフト・回転	ASL	メモリ又はアキュムレータを1ビット左シフト
	LSR	メモリ又はアキュムレータを1ビット右シフト
	ROL	メモリ又はアキュムレータとCフラグをつなげて、左へ1ビット回転
	ROR	メモリ又はアキュムレータとCフラグをつなげて、右へ1ビット回転
	RRF	メモリの内容を右へ4ビット回転

3.2.3 ビット処理命令

これらの命令は、アキュムレータ又はメモリの指定されたビットの内容を“1”又は、“0”にします。ビット処理命令には、以下の2つがあります。

	命 令	内 容
ビット処理	CLB	アキュムレータ又はメモリの指定ビットを“0”にクリア
	SEB	アキュムレータ又はメモリの指定ビットを“1”にセット

3.2.4 フラグ設定命令

これらの命令は、Cフラグ、Dフラグ、Iフラグ、Tフラグ、Vフラグを“0”又は“1”に設定します。フラグ設定命令には、以下のようなものがあります。

	命 令	内 容	
フ ラ グ 設 定	CLC	Cフラグを“0”にクリア	Cフラグ : キャリフラグ
	SEC	Cフラグを“1”にセット	
	CLD	Dフラグを“0”にクリア	Dフラグ : 10進モードフラグ
	SED	Dフラグを“1”にセット	
	CLI	Iフラグを“0”にクリア	Iフラグ : 割り込み禁止フラグ
	SEI	Iフラグを“1”にセット	
	CLT	Tフラグを“0”にクリア	Tフラグ : X修飾演算モードフラグ
	SET	Tフラグを“1”にセット	
	CLV	Vフラグを“0”にクリア	Vフラグ : オーバフローフラグ

3.2.5 ジャンプ、ブランチ、リターン命令

これらの命令により、プログラムのシーケンスが変わります。
ジャンプ、ブランチ、リターン命令には、以下のようなものがあります。

	命 令	内 容	
ジャンプ	JMP	新番地へジャンプ	
	BRA	新番地へジャンプ	
	JSR	復帰アドレスを確保して、新番地へジャンプ	
ブランチ	BBC	アキュムレータ又はメモリの指定ビットが“0”のとき、分岐	
	BBS	アキュムレータ又はメモリの指定ビットが“1”のとき、分岐	
	BCC	Cフラグが“0”のとき、分岐	Cフラグ : キャリフラグ
	BCS	Cフラグが“1”のとき、分岐	
	BNE	Zフラグが“0”のとき、分岐	Zフラグ : ゼロフラグ
	BEQ	Zフラグが“1”のとき、分岐	
	BPL	Nフラグが“0”のとき、分岐	Nフラグ : ネガティブフラグ
	BMI	Nフラグが“1”のとき、分岐	
	BVC	Vフラグが“0”のとき、分岐	Vフラグ : オーバフローフラグ
	BVS	Vフラグが“1”のとき、分岐	
リターン	RTI	割り込みからの復帰	
	RTS	サブルーチンからの復帰	

3.2.6 割り込み命令(ブ레이크命令)

この命令は、ソフトウェア割り込みを行います。

	命 令	内 容
割り込み	BRK	ソフトウェア割り込みを実行

3.2.7 特殊命令

これらの命令は、発振及び内部クロックの制御を行います。

	命 令	内 容
特殊	WIT STP	内部クロックを停止 発振器の発振を停止

3.2.8 その他の命令

	命 令	内 容
その他	NOP	プログラムカウンタを進めるのみ

3.3 命令の説明

740ファミリの各命令についての詳しい説明をします。この章は、命令のニーモニックを標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。

各命令については、その命令の動作と説明、ステータスフラグの変化が記述してあります。また、その命令で使われる、各アドレッシングモードについて、アセンブラ記述形式、機械語、バイト数、サイクル数の一覧表も記載しています。

説明に用いた記号及びその内容を下記一覧表に示します。

記号	内 容	記号	内 容
A	アキュムレータ	hh	アドレス上位バイト 0 ~ 255 のデータ
Ai	アキュムレータのビット i	ll	アドレス下位バイト 0 ~ 255 のデータ
PC	プログラムカウンタ	zz	ゼロページアドレス 0 ~ 255 のデータ
PCL	プログラムカウンタの下位バイト	nn	0 ~ 255 のデータ
PCH	プログラムカウンタの上位バイト	i	0 ~ 7 のデータ
PS	プロセッサステータスレジスタ	*	プログラムカウンタの内容 タブ又はスペース
S	スタックポインタ	#	イミディエイトモード
X、Y	インデックスレジスタ	¥	スペシャルページモード
M	メモリ	\$	16進表記
Mi	メモリのビット i	+	加算
C	キャリフラグ	-	減算
Z	ゼロフラグ	x	乗算
I	割り込み禁止フラグ	/	除算
D	10進モードフラグ		論理積
B	ブレークフラグ		論理和
T	X 修飾演算モードフラグ		排他的論理和
V	オーバフローフラグ	()	レジスタ、メモリなどの内容
N	ネガティブフラグ		データ転送
Rel	相対アドレス		
BADRS	ブレークアドレス		

動作 : (T) = 0 のとき (A) (A) + (M) + (C)
 (T) = 1 のとき (M(X)) (M(X)) + (M) + (C)

説明 : Tフラグが0のとき、AとMとCフラグの内容を加算して、結果をA及びCフラグに入れます。
 Tフラグが1のとき、M(X)とMとCフラグの内容を加算して、結果をM(X)及びCフラグに入れます。このとき、Aの内容は変化しませんが、ステータスフラグは変化します。ただし、M(X)はXが示す番地のメモリの内容です。

ステータスフラグ : N : 演算の結果、ビット7が1のとき、1になります。それ以外の場合は0になります。
 V : 演算の結果が + 127 または | - 128 | を越えると、1になります。それ以外の場合は0になります。
 T : 変化しません。
 B : 変化しません。
 D : 変化しません。
 I : 変化しません。
 Z : 演算の結果が0のとき、1になります。それ以外の場合は0になります。
 C : 2進加算で、結果が255を越えたとき、又は10進加算で結果が99を越えたとき、1になります。それ以外の場合は、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
イミディエイト	ADC # \$ nn	69 ₁₆ , nn ₁₆	2	2
ゼロページ	ADC \$ zz	65 ₁₆ , zz ₁₆	2	3
ゼロページ X	ADC \$ zz, X	75 ₁₆ , zz ₁₆	2	4
絶対	ADC \$ hhll	6D ₁₆ , ll ₁₆ , hh ₁₆	3	4
絶対 X	ADC \$ hhll, X	7D ₁₆ , ll ₁₆ , hh ₁₆	3	5
絶対 Y	ADC \$ hhll, Y	79 ₁₆ , ll ₁₆ , hh ₁₆	3	5
(間接 X)	ADC (\$ zz, X)	61 ₁₆ , zz ₁₆	2	6
(間接 Y)	ADC (\$ zz), Y	71 ₁₆ , zz ₁₆	2	6

注 1. Tフラグが1のときサイクル数を + 3します。

2. 10進演算モード(Dフラグ = 1)でADC命令を実行させた場合、SEC, CLC, CLD命令は、ADC命令から一命令以上後に行ってください。

また、10進演算モードでは、N, V, Zフラグが無効となります。

動作 : (T) = 0 のとき (A) (A) (M)
(T) = 1 のとき (M(X)) (M(X)) (M)

説明 : Tフラグが0のとき、AとMの内容のビットごとの論理積をとり、結果をAに入れます。
Tフラグが1のとき、M(X)とMの内容のビットごとの論理積をとり、結果をM(X)に入れます。このとき、Aの内容は変化しませんが、ステータスフラグは変化します。ただし、M(X)はXが示す番地のメモリの内容です。

ステータスフラグ : N : 演算の結果、ビット7が1のとき、1になります。それ以外の場合は0になります。
V : 変化しません。
T : 変化しません。
B : 変化しません。
D : 変化しません。
I : 変化しません。
Z : 演算の結果が0のとき、1になります。それ以外の場合は0になります。
C : 変化しません。

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
イミディエイト	AND # \$ nn	2916, nn16	2	2
ゼロページ	AND \$ zz	2516, zz16	2	3
ゼロページ X	AND \$ zz, X	3516, zz16	2	4
絶対	AND \$ hhll	2D16, ll16, hh16	3	4
絶対 X	AND \$ hhll, X	3D16, ll16, hh16	3	5
絶対 Y	AND \$ hhll, Y	3916, ll16, hh16	3	5
(間接 X)	AND (\$ zz, X)	2116, zz16	2	6
(間接 Y)	AND (\$ zz), Y	3116, zz16	2	6

注 .Tフラグが1のとき、サイクル数を+3します。

動作 :

C	←	b7									b0	←	0
---	---	----	--	--	--	--	--	--	--	--	----	---	---

説明 : A 又は M のすべてのビットを、1 ビット左へシフトします。このとき、A 又は M のビット 0 は 0 になります。また、C フラグには、A 又は M のビット 7 の内容が入ります。

ステータス フラグ : N : 実行の結果、ビット 7 が 1 のとき、1 になります。それ以外の場合は 0 になります。

V : 変化しません。

T : 変化しません。

B : 変化しません。

D : 変化しません。

I : 変化しません。

Z : 実行の結果が 0 のとき、1 になります。それ以外の場合は 0 になります。

C : 実行前の A 又は M のビット 7 が 1 のとき、1 になります。それ以外の場合は 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
アキュムレータ	ASL	A			0A16			1	2
ゼロページ	ASL	\$ zz			0616, zz16			2	5
ゼロページ X	ASL	\$ zz, X			1616, zz16			2	6
絶対	ASL	\$ hhll			0E16, ll16, hh16			3	6
絶対 X	ASL	\$ hhll, X			1E16, ll16, hh16			3	7

動作 : (Mi)又は(Ai) = 0 のとき (PC) (PC) + n + Rel
 (Mi)又は(Ai) = 1 のとき (PC) (PC) + n
 ただし、アドレッシングモードが
 ゼロページするとき、n = 3
 アキュムレータのとき、n = 2 となります。

説明 : M 又は A の指定されたビット i をテストします。
 そのビットが 0 であれば、指定されたアドレスに分岐します。
 分岐先のアドレスは、相対で示します。
 そのビットが 1 であれば、そのまま、次へ進みます。

ステータス : 変化しません。
 フラグ

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
アキュムレータビット 相対	BBC i, A, \$hhll	$(20i + 13)_{16}, rr_{16}$	2	4
ゼロページビット 相対	BBC i, \$zz, \$hhll	$(20i + 17)_{16}, zz_{16}, rr_{16}$	3	5

- 注 1 . $rr_{16} = \$hhll - (* + n)$ 。 rr_{16} は - 128 ~ + 127 までの範囲の値をとります。
 2 . ブランチ実行のとき、サイクル数を + 2 します。
 3 . 割り込み要求ビットの内容を変更した後BBC命令を実行する場合は、変更後1命令以上後に行ってください。

動作 : (Mi)又は(Ai) = 1 のとき (PC) (PC) + n + Rel
 (Mi)又は(Ai) = 0 のとき (PC) (PC) + n
 ただし、アドレッシングモードが
 ゼロページするとき、n = 3
 アキュムレータするとき、n = 2 となります。

説明 : M 又は A の指定されたビット i をテストします。
 そのビットが 1 であれば、指定されたアドレスに分岐します。
 分岐先のアドレスは、相対で示します。
 そのビットが 0 であれば、そのまま、次へ進みます。

ステータス : 変化しません。
 フラグ

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
アキュムレータビット 相対	BBS i, A, \$ hhl	(20i + 3) ₁₆ , rr ₁₆	2	4
ゼロページビット 相対	BBS i, \$ zz, \$ hhl	(20i + 7) ₁₆ , zz ₁₆ , rr ₁₆	3	5

- 注 1. rr₁₆ = \$ hhl - (* + n)。rr₁₆ は - 128 ~ + 127 までの範囲の値をとります。
 2. ブランチ実行のとき、サイクル数を + 2 します。
 3. 割り込み要求ビットの内容を変更した後 BBS 命令を実行する場合は、変更後 1 命令以上後に行ってください。

動作 : (C) = 0 のとき (PC) (PC) + 2 + Rel
(C) = 1 のとき (PC) (PC) + 2

説明 : Cフラグが0のとき、指定されたアドレスに分岐します。分岐先のアドレスは、相対で示します。
Cフラグが1のとき、そのまま、次へ進みます。

ステータス : 変化しません。
フラグ

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
相対	BCC \$hhll	9016, rr16	2	2

注 1 . rr16 = \$hhll - (* + 2)。rr16は - 128 ~ + 127までの範囲の値をとります。

2 . ブランチ実行のとき、サイクル数を + 2 します。

動作 : (C) = 1 のとき (PC) (PC) + 2 + Rel
(C) = 0 のとき (PC) (PC) + 2

説明 : Cフラグが1のとき、指定されたアドレスに分岐します。分岐先のアドレスは、相対で示します。
Cフラグが0のとき、そのまま次へ進みます。

ステータス : 変化しません。
フラグ

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
相対	BCS \$ hhll	B016 , rr16	2	2

- 注 1 . rr16 = \$ hhll - (* + 2)。rr16は - 128 ~ + 127までの範囲の値をとります。
2 . ブランチ実行のとき、サイクル数を + 2 します。

動作 : (Z) = 1 のとき (PC) (PC) + 2 + Rel
(Z) = 0 のとき (PC) (PC) + 2

説明 : Zフラグが1のとき、指定されたアドレスに分岐します。分岐先のアドレスは、相対で示します。
Zフラグが0のとき、そのまま次へ進みます。

ステータス : 変化しません。
フラグ

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
相対	BEQ \$hhll	F016, rr16	2	2

- 注 1 . rr16 = \$hhll - (* + 2)。rr16は - 128 ~ + 127までの範囲の値をとります。
2 . ブランチ実行のとき、サイクル数を + 2 します。

動作 : (A) (M)

説明 : A と M の内容のビットごとの論理積をとりますが、結果はどこにもストアされません。N フラグ、V フラグ、Z フラグの内容は変化しますが、A と M の内容は変化しません。

ステータス フラグ : N : M のビット 7 が 1 のとき、1 になります。それ以外のときは 0 になります。

V : M のビット 6 が 1 のとき、1 になります。それ以外のときは 0 になります。

T : 変化しません。

B : 変化しません。

D : 変化しません。

I : 変化しません。

Z : 論理積の結果が 0 のとき、1 になります。それ以外のときは 0 になります。

C : 変化しません。

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
ゼロページ	BIT \$ zz	24 ₁₆ , zz ₁₆	2	3
絶対	BIT \$ hhll	2C ₁₆ , ll ₁₆ , hh ₁₆	3	4

動作 : (N) = 1 のとき (PC) (PC) + 2 + Rel
 (N) = 0 のとき (PC) (PC) + 2

説明 : Nフラグが1のとき、指定されたアドレスに分岐します。分岐先のアドレスは、相対で示します。
 Nフラグが0のとき、そのまま次へ進みます。

ステータス : 変化しません。
 フラグ

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
相対	BMI \$hhll	3016, rr16	2	2

注 1 . rr16 = \$hhll - (* + 2)。rr16は - 128 ~ + 127までの範囲の値をとります。

2 . ブランチ実行のとき、サイクル数を + 2 します。

動作 : (Z) = 0 のとき (PC) (PC) + 2 + Rel
 (Z) = 1 のとき (PC) (PC) + 2

説明 : Zフラグが0のとき、指定されたアドレスに分岐します。分岐先のアドレスは、相対で示します。
 Zフラグが1のとき、そのまま次へ進みます。

ステータス : 変化しません。
 フラグ

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
相対	BNE \$hhll	D016 , rr16	2	2

- 注 1 . rr16 = \$hhll - (* + 2)。rr16 は - 128 ~ + 127 までの範囲の値をとります。
 2 . ブランチ実行のとき、サイクル数を + 2 します。

動作 : (N) = 0 のとき (PC) (PC) + 2 + Rel
(N) = 1 のとき (PC) (PC) + 2

説明 : Nフラグが0のとき、指定されたアドレスに分岐します。分岐先のアドレスは、相対で示します。
Nフラグが1のとき、そのまま次へ進みます。

ステータス : 変化しません。
フラグ

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
相対	BPL \$ hhll	1016 , rr16	2	2

- 注 1 . rr16 = \$ hhll - (* + 2)。rr16 は - 128 ~ + 127 までの範囲の値をとります。
2 . ブランチ実行のとき、サイクル数 + 2 します。

動作 : $(PC) + 2 + Rel$

説明 : 指定されたアドレスにジャンプします。ジャンプ先のアドレスは、相対で示します。

ステータス : 変化しません。
フラグ

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
相対	BRA \$hhll	8016, rr16	2	4

注 . $rr16 = \$hhll - (* + 2)$ 。rr16 は - 128 ~ + 127 までの範囲の値をとります。

動作 : (B) 1
(PC) (PC) + 2
(M(S)) (PCH)
(S) (S) - 1
(M(S)) (PCL)
(S) (S) - 1
(M(S)) (PS)
(S) (S) - 1
(I) 1
(PC) BADRS

説明 : BRK命令を実行すると、CPUは現在のPCの内容をスタックに退避し、割り込みベクトルで、指定されたアドレス(BADRS)をPCに格納します。

ステータス : N : 変化しません。
フラグ V : 変化しません。
T : 変化しません。
B : 1になります。
D : 変化しません。
I : 1になります。
Z : 変化しません。
C : 変化しません。

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
インプライド	BRK	0016	1	7

- 注1 . BRK命令の実行によって退避されるPCの値は、BRK命令のアドレス + 2です。したがって、BRKルーチンから復帰してきたとき、BRK命令の次のバイトは実行されません。
- 2 . 一部の品種では、BRK命令と他の割り込み要因が同一割り込みベクトルになっています。その場合、どちらも割り込みベクトルで指定されたアドレスに分岐しますが、スタックに退避されたプロセッサステータスレジスタの内容のBフラグをテストすることにより、BRK命令を実行したことが分かります。

動作 : (V) = 0 のとき (PC) (PC) + 2 + Rel
 (V) = 1 のとき (PC) (PC) + 2

説明 : Vフラグが0のとき、指定されたアドレスに分岐します。分岐先のアドレスは、相対で示します。
 Vフラグが1のとき、そのまま次へ進みます。

ステータス : 変化しません。
 フラグ

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
相対	BVC \$hhll	5016, rr16	2	2

- 注 1 . rr16 = \$hhll - (* + 2)。rr16は - 128 ~ + 127までの範囲の値をとります。
 2 . ブランチ実行のとき、サイクル数を + 2 します。

動作 : (V) = 1 のとき (PC) (PC) + 2 + Rel
 (V) = 0 のとき (PC) (PC) + 2

説明 : Vフラグが1のとき、指定されたアドレスに分岐します。分岐先のアドレスは、相対で示します。
 Vフラグが0のとき、そのまま次へ進みます。

ステータス : 変化しません。
 フラグ

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
相対	BVS \$hhll	7016, rr16	2	2

- 注 1 . rr16 = \$hhll - (* + 2)。rr16は - 128 ~ + 127までの範囲の値をとります。
 2 . ブランチ実行のとき、サイクル数を + 2 します。

動作 : (Ai) 0 又は、(Mi) 0

説明 : A 又は M の、指定されたビット i の内容を 0 にします。

ステータス : 変化しません。
フラグ

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
アキュムレータ ビット	CLB i, A	$(20i + 1B)_{16}$	1	2
ゼロページ ビット	CLB i, \$zz	$(20i + 1F)_{16}, zz_{16}$	2	5

動作 : (C) 0

説明 : Cフラグの内容を0にします。

ステータス : N : 変化しません。

フラグ V : 変化しません。

T : 変化しません。

B : 変化しません。

D : 変化しません。

I : 変化しません。

Z : 変化しません。

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
インブライド	CLC	1816	1	2

動作 : (D) 0

説明 : Dフラグの内容を0にします。

ステータス : N : 変化しません。

フラグ V : 変化しません。

T : 変化しません。

B : 変化しません。

D : 0になります。

I : 変化しません。

Z : 変化しません。

C : 変化しません。

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
インブライド	CLD	D816	1	2

動作 : (I) 0

説明 : Iフラグの内容を0にします。

ステータス : N : 変化しません。

フラグ V : 変化しません。

T : 変化しません。

B : 変化しません。

D : 変化しません。

I : 0になります。

Z : 変化しません。

C : 変化しません。

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
インプライド	CLI	5816	1	2

動作 : (T) 0

説明 : Tフラグの内容を0にします。

ステータス : N : 変化しません。

フラグ V : 変化しません。

T : 0になります。

B : 変化しません。

D : 変化しません。

I : 変化しません。

Z : 変化しません。

C : 変化しません。

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
インプライド	CLT	1216	1	2

動作 : (V) 0

説明 : Vフラグの内容を0にします。

ステータス : N : 変化しません。
フラグ V : 0になります。
T : 変化しません。
B : 変化しません。
D : 変化しません。
I : 変化しません。
Z : 変化しません。
C : 変化しません。

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
インプライド	CLV	B816	1	2

動作 : (T) = 0 のとき (A) - (M)
(T) = 1 のとき (M(X)) - (M)

説明 : Tフラグが0のとき、Aの内容からMの内容を減算しますが、結果はどこにもストアされません。このとき、AとMの内容は変化しません。Tフラグが1のとき、M(X)の内容からMの内容を減算しますが、結果はどこにもストアされません。このとき、M(X)とM及びAの内容は変化しません。ただし、M(X)はXが示す番地のメモリの内容です。

ステータス フラグ : N : 減算の結果、ビット7が1のとき、1になります。それ以外の場合は0になります。
V : 変化しません。
T : 変化しません。
B : 変化しません。
D : 変化しません。
I : 変化しません。
Z : 減算の結果が0のとき、1になります。それ以外の場合は0になります。
C : 減算の結果が0より大きいか等しいとき、1になります。それ以外の場合は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
イミディエイト	CMP # \$ nn	C916 , nn16	2	2
ゼロページ	CMP \$ zz	C516 , zz16	2	3
ゼロページ X	CMP \$ zz , X	D516 , zz16	2	4
絶対	CMP \$ hhll	CD16 , ll16 , hh16	3	4
絶対 X	CMP \$ hhll , X	DD16 , ll16 , hh16	3	5
絶対 Y	CMP \$ hhll , Y	D916 , ll16 , hh16	3	5
(間接 X)	CMP (\$ zz , X)	C116 , zz16	2	6
(間接 Y)	CMP (\$ zz) , Y	D116 , zz16	2	6

注 . Tフラグが1のとき、サイクル数を + 1 します。

動作 : (M) (\bar{M})

説明 : Mの内容の、1の補数をMに入れます。

ステータス フラグ : N : 実行の結果、Mのビット7が1のとき、1になります。それ以外
のときは0になります。

V : 変化しません。

T : 変化しません。

B : 変化しません。

D : 変化しません。

I : 変化しません。

Z : 実行の結果が0のとき、1になります。それ以外の場合は0になります。

C : 変化しません。

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
ゼロページ	COM \$ZZ	4416, ZZ16	2	5

動作 : (X) - (M)

説明 : Xの内容からMの内容を減算しますが、結果はどこにもストアされません。このとき、XとMの内容は変化しません。

ステータスフラグ : N : 減算の結果、ビット7が1のとき、1になります。それ以外の場合は0になります。

V : 変化しません。

T : 変化しません。

B : 変化しません。

D : 変化しません。

I : 変化しません。

Z : 減算の結果が0のとき、1になります。それ以外の場合は0になります。

C : 減算の結果が0より大きいか等しいとき、1になります。それ以外の場合は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
イミディエイト	CPX # \$ nn	E0 ₁₆ , nn ₁₆	2	2
ゼロページ	CPX \$ zz	E4 ₁₆ , zz ₁₆	2	3
絶対	CPX \$ hhll	EC ₁₆ , ll ₁₆ , hh ₁₆	3	4

動作 : (Y) - (M)

説明 : Yの内容からMの内容を減算しますが、結果はどこにもストアされません。このとき、YとMの内容は変化しません。

ステータスフラグ : N : 減算の結果、ビット7が1のとき、1になります。それ以外の場合は0になります。

V : 変化しません。

T : 変化しません。

B : 変化しません。

D : 変化しません。

I : 変化しません。

Z : 減算の結果が0のとき、1になります。それ以外の場合は0になります。

C : 減算の結果が0より大きいか等しいとき、1になります。それ以外の場合は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
イミディエイト	CPY # \$ nn	C016 , nn16	2	2
ゼロページ	CPY \$ zz	C416 , zz16	2	3
絶対	CPY \$ hhll	CC16 , ll16 , hh16	3	4

動作 : (A) (A) - 1 又は、(M) (M) - 1

説明 : A 又は M の内容を 1 減少させます。

ステータス フラグ : N : 実行の結果、ビット 7 が 1 のとき、1 になります。それ以外
のときは 0 になります。

V : 変化しません。

T : 変化しません。

B : 変化しません。

D : 変化しません。

I : 変化しません。

Z : 実行の結果が 0 のとき、1 になります。それ以外
のときは 0 になります。

C : 変化しません。

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
アキュムレータ	DEC A	1A ₁₆	1	2
ゼロページ	DEC \$zz	C6 ₁₆ , ZZ ₁₆	2	5
ゼロページ X	DEC \$zz, X	D6 ₁₆ , ZZ ₁₆	2	6
絶対	DEC \$hhll	CE ₁₆ , ll ₁₆ , hh ₁₆	3	6
絶対 X	DEC \$hhll, X	DE ₁₆ , ll ₁₆ , hh ₁₆	3	7

動作 : (X) (X) - 1

説明 : Xの内容を1減少させます。

ステータス フラグ : N : 実行の結果、ビット7が1のとき、1になります。それ以外
のときは0になります。

V : 変化しません。

T : 変化しません。

B : 変化しません。

D : 変化しません。

I : 変化しません。

Z : 実行の結果が0のとき、1になります。それ以外
のときは0になります。

C : 変化しません。

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
インプライド	DEX	CA16	1	2

動作 : (Y) (Y) - 1

説明 : Yの内容を1減少させます。

ステータス フラグ : N : 実行の結果、ビット7が1のとき、1になります。それ以外
のときは0になります。

V : 変化しません。

T : 変化しません。

B : 変化しません。

D : 変化しません。

I : 変化しません。

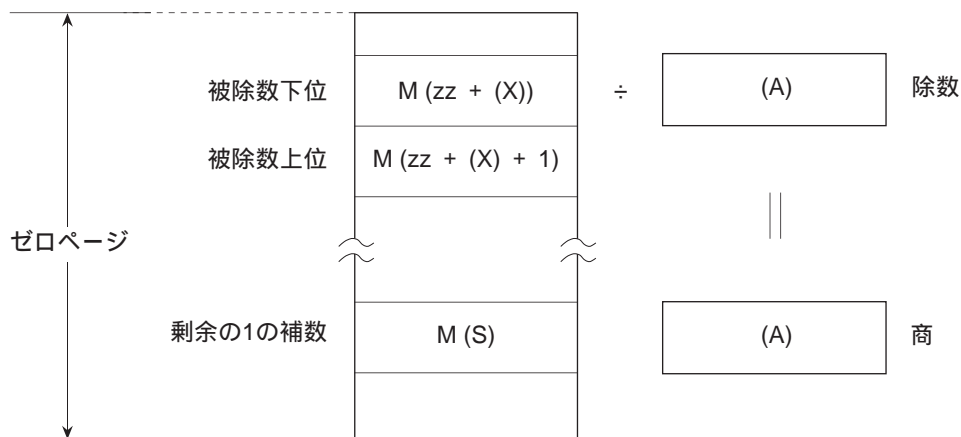
Z : 実行の結果が0のとき、1になります。それ以外
のときは0になります。

C : 変化しません。

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
インプライド	DEY	8816	1	2

動作 : (A) (M(zz + (X) + 1), M(zz + (X))) / (A)
M(S) 剰余の1の補数
(S) (S) - 1

説明 : 上位 8 ビットを M(zz + (X) + 1)、下位 8 ビットを M(zz + (X)) の内容とする 16 ビットのデータをアキュムレータの内容で除算します。除算の結果、商はアキュムレータに、剰余の1の補数はスタックに入ります。



ステータス : 変化しません。
フラグ

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
ゼロページ X	DIV \$zz, X	E216, ZZ16	2	16

- 注 1. 商のオーバーフロー及び0ディビジョンは検出しませんので、この命令を実行する前にプログラム上で処理をしてください。
また、この命令を実行した場合、アキュムレータ及びスタックポインタの内容が変化しますのでご注意ください。
- 一部の品種では、この命令は使用できません。
 - Tフラグ、Dフラグの影響を受けません。

動作 : (T) = 0 のとき (A) (A) (M)
(T) = 1 のとき (M(X)) (M(X)) (M)

説明 : Tフラグが0のとき、AとMの内容のビットごとの排他的論理和をとり、結果をAに入れます。
Tフラグが1のとき、M(X)とMの内容のビットごとの排他的論理和をとり、結果をM(X)に入れます。このとき、Aの内容は変化しませんが、ステータスフラグの内容は変化します。ただし、M(X)はXが示す番地のメモリの内容です。

ステータスフラグ : N : 演算の結果、ビット7が1のとき、1になります。それ以外の場合は0になります。
V : 変化しません。
T : 変化しません。
B : 変化しません。
D : 変化しません。
I : 変化しません。
Z : 演算の結果が0のとき、1になります。それ以外の場合は、0になります。
C : 変化しません。

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
イミディエイト	EOR # \$ nn	4916, nn16	2	2
ゼロページ	EOR \$ zz	4516, zz16	2	3
ゼロページ X	EOR \$ zz, X	5516, zz16	2	4
絶対	EOR \$ hhll	4D16, ll16, hh16	3	4
絶対 X	EOR \$ hhll, X	5D16, ll16, hh16	3	5
絶対 Y	EOR \$ hhll, Y	5916, ll16, hh16	3	5
(間接 X)	EOR (\$ zz, X)	4116, zz16	2	6
(間接 Y)	EOR (\$ zz), Y	5116, zz16	2	6

注 . Tフラグが1のとき、サイクル数を + 3します。

動作 : (A) (A) + 1 又は、(M) (M) + 1

説明 : A 又は M の内容を、1 増加させます。

ステータス フラグ : N : 実行の結果、ビット7が1のとき、1になります。それ以外
のときは0になります。

V : 変化しません。

T : 変化しません。

B : 変化しません。

D : 変化しません。

I : 変化しません。

Z : 実行の結果が0のとき、1になります。それ以外
のときは0になります。

C : 変化しません。

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
アキュムレータ	INC A	3A16	1	2
ゼロページ	INC \$zz	E616, ZZ16	2	5
ゼロページ X	INC \$zz, X	F616, ZZ16	2	6
絶対	INC \$hhll	EE16, ll16, hh16	3	6
絶対 X	INC \$hhll, X	FE16, ll16, hh16	3	7

動作 : (X) (X) + 1

説明 : Xの内容を、1増加させます。

ステータス フラグ : N : 実行の結果、ビット7が1のとき、1になります。それ以外
のときは0になります。

V : 変化しません。

T : 変化しません。

B : 変化しません。

D : 変化しません。

I : 変化しません。

Z : 実行の結果が0のとき、1になります。それ以外
のときは0になります。

C : 変化しません。

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
インプライド	INX	E816	1	2

動作 : (Y) (Y) + 1

説明 : Yの内容を、1増加させます。

ステータス フラグ : N : 実行の結果、ビット7が1のとき、1になります。それ以外
のときは0になります。

V : 変化しません。

T : 変化しません。

B : 変化しません。

D : 変化しません。

I : 変化しません。

Z : 実行の結果が0のとき、1になります。それ以外
のときは0になります。

C : 変化しません。

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
インプライド	INY	C816	1	2

動作 : アドレッシングモードが
絶対のとき、
(PC) hhll

間接のとき
(PCL) (hhll)
(PCH) (hhll + 1)

ゼロページ間接のとき
(PCL) (zz)
(PCH) (zz + 1)

説明 : 絶対、間接、及び、ゼロページ間接の各アドレッシングモードで示されるアドレスにジャンプします。

ステータス : 変化しません。
フラグ

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
絶対	JMP \$hhll	4C ₁₆ , ll ₁₆ , hh ₁₆	3	3
間接	JMP (\$hhll)	6C ₁₆ , ll ₁₆ , hh ₁₆	3	5
ゼロページ 間接	JMP (\$zz)	B2 ₁₆ , zz ₁₆	2	4

注 . 間接指定アドレスとしてページの最終番地(XXFF₁₆ 番地)は指定できません。つまり、JMP(\$XXFF)を実行することはできませんので、ご注意ください。

動作 : (M(S)) (PCH)
 (S) (S) - 1
 (M(S)) (PCL)
 (S) (S) - 1

上記を行った後、アドレッシングモードが絶対のときは、

(PC) hhll

スペシャルページのときは、

(PCL) ll

(PCH) FF16

ゼロページ間接のときは、

(PCL) (zz)

(PCH) (zz + 1)

を行う。

説明 : PCの内容をスタックに退避した後、絶対、スペシャルページ、及びゼロページ間接の各アドレッシングモードで示されるアドレスにジャンプします。

ステータス : 変化しません。
 フラグ

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
絶対	JSR \$hhll	2016, ll16, hh16	3	6
スペシャルページ	JSR ¥\$hhll(注)	2216, ll16	2	5
ゼロページ 間接	JSR (\$zz)	0216, zz16	2	7

注 . スペシャルページを示すキーワードは、¥です。(¥は、JIS 8 コードで 0DC16 に該当します。ASCII コードでは、このコードは \ になるので注意が必要です)。また、スペシャルページのとき、hh16はFF16としてください。

動作 : (T) = 0 のとき (A) (M)
(T) = 1 のとき (M(X)) (M)

説明 : Tフラグが0のとき、Mの内容をAに入れます。
Tフラグが1のとき、Mの内容をM(X)に入れます。このとき、Aの内容は変化しませんが、ステータスフラグは変化します。ただし、M(X)はXが示す番地のメモリの内容です。

ステータスフラグ : N : 実行の結果、ビット7が1のとき、1になります。それ以外の場合は0になります。

V : 変化しません。

T : 変化しません。

B : 変化しません。

D : 変化しません。

I : 変化しません。

Z : 実行の結果が0のとき、1になります。それ以外の場合は0になります。

C : 変化しません。

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
イミディエイト	LDA # \$ nn	A9 ₁₆ , nn ₁₆	2	2
ゼロページ	LDA \$ zz	A5 ₁₆ , zz ₁₆	2	3
ゼロページ X	LDA \$ zz, X	B5 ₁₆ , zz ₁₆	2	4
絶対	LDA \$ hhll	AD ₁₆ , ll ₁₆ , hh ₁₆	3	4
絶対 X	LDA \$ hhll, X	BD ₁₆ , ll ₁₆ , hh ₁₆	3	5
絶対 Y	LDA \$ hhll, Y	B9 ₁₆ , ll ₁₆ , hh ₁₆	3	5
(間接 X)	LDA (\$ zz, X)	A1 ₁₆ , zz ₁₆	2	6
(間接 Y)	LDA (\$ zz), Y	B1 ₁₆ , zz ₁₆	2	6

注 . Tフラグが1のとき、サイクル数を + 2します。

動作 : (M) nn

説明 : イミディエイト値を M にロードします。

ステータス : 変化しません。
フラグ

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
ゼロページ	LDM # \$ nn , \$ zz	3C16 , nn16 , zz16	3	4

動作 : (X) (M)

説明 : Mの内容を、Xにロードします。

ステータス : N : 実行の結果、ビット7が1のとき、1になります。それ以外
のときは0になります。

V : 変化しません。

T : 変化しません。

B : 変化しません。

D : 変化しません。

I : 変化しません。

Z : 実行の結果が0のとき、1になります。それ以外
のときは0になります。

C : 変化しません。

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
イミディエイト	LDX # \$ nn	A216 , nn16	2	2
ゼロページ	LDX \$ zz	A616 , zz16	2	3
ゼロページ Y	LDX \$ zz , Y	B616 , zz16	2	4
絶対	LDX \$ hhll	AE16 , ll16 , hh16	3	4
絶対 Y	LDX \$ hhll , Y	BE16 , ll16 , hh16	3	5

動作 : (Y) (M)

説明 : Mの内容を、Yにロードします。

ステータス : N : 実行の結果、ビット7が1のとき、1になります。それ以外のときは0になります。

V : 変化しません。

T : 変化しません。

B : 変化しません。

D : 変化しません。

I : 変化しません。

Z : 実行の結果が0のとき、1になります。それ以外のときは0になります。

C : 変化しません。

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
イミディエイト	LDY # \$ nn	A0 ₁₆ , nn ₁₆	2	2
ゼロページ	LDY \$ zz	A4 ₁₆ , zz ₁₆	2	3
ゼロページ X	LDY \$ zz, X	B4 ₁₆ , zz ₁₆	2	4
絶対	LDY \$ hhll	AC ₁₆ , ll ₁₆ , hh ₁₆	3	4
絶対 X	LDY \$ hhll, X	BC ₁₆ , ll ₁₆ , hh ₁₆	3	5

動作 : 0 →

b7							b0
----	--	--	--	--	--	--	----

 →

C

説明 : A 又は M のすべてのビットを、1 ビット右へシフトします。このとき、A 又は M のビット 7 は 0 になります。また、C フラグには、A 又は M のビット 0 の内容が入ります。

ステータス : N : 0 になります。

フラグ V : 変化しません。

T : 変化しません。

B : 変化しません。

D : 変化しません。

I : 変化しません。

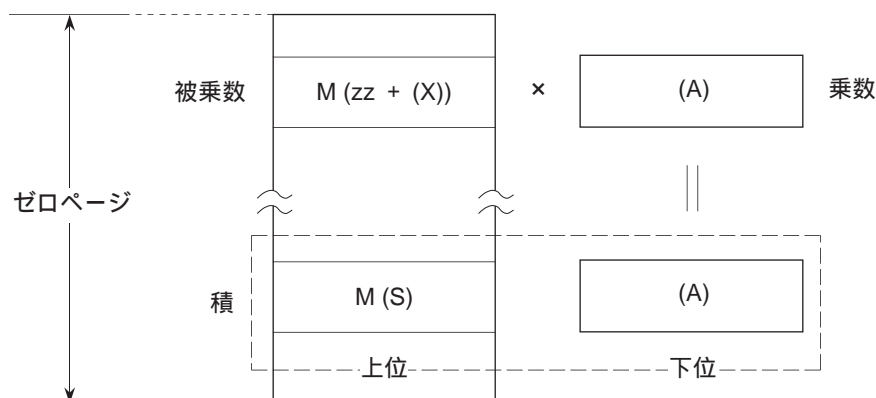
Z : 実行の結果が 0 のとき、1 になります。それ以外の場合は 0 になります。

C : 実行前の A 又は M のビット 0 が 1 のとき、1 になります。それ以外の場合は 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
アキュムレータ	LSR A	4A16	1	2
ゼロページ	LSR \$zz	4616, zz16	2	5
ゼロページ X	LSR \$zz, X	5616, zz16	2	6
絶対	LSR \$hhll	4E16, ll16, hh16	3	6
絶対 X	LSR \$hhll, X	5E16, ll16, hh16	3	7

動作 : $M(S) \cdot (A) \quad (A) \times M(zz + (X))$
 (S) (S) - 1

説明 : アキュムレータとゼロページ X アドレッシングモードによって指定されたメモリの内容を乗算します。乗算の結果、積の上位 8 ビットはスタックに、下位 8 ビットはアキュムレータに入ります。



ステータス : 変化しません。
 フラグ

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
ゼロページ X	MUL \$zz, X	6216 zz16	2	15

- 注 1. この命令を実行した場合、アキュムレータ及びスタックポインタの内容が変化しますのでご注意ください。
2. 一部の品種では、この命令は使用できません。
3. Tフラグ、Dフラグの影響を受けません。

動作 : (PC) (PC) + 1

説明 : PC を、1 増加させるだけで、その他は、何もしません。

ステータス : 変化しません。
フラグ

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
インプライド	NOP	EA16	1	2

動作 : (T) = 0 のとき (A) (A) (M)
(T) = 1 のとき (M(X)) (M(X)) (M)

説明 : Tフラグが0のとき、AとMの内容のビットごとの論理和をとり、結果をAに入れます。
Tフラグが1のとき、M(X)とMの内容のビットごとの論理和をとり、結果をM(X)に入れます。このとき、Aの内容は変化しませんが、ステータスフラグは変化します。ただし、M(X)はXが示す番地のメモリの内容です。

ステータス フラグ : N : 演算の結果、ビット7が1のとき、1になります。それ以外の場合は0になります。
V : 変化しません。
T : 変化しません。
B : 変化しません。
D : 変化しません。
I : 変化しません。
Z : 演算の結果が0のとき、1になります。それ以外の場合は、0になります。
C : 変化しません。

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
イミディエイト	ORA # \$ nn	09 ₁₆ , nn ₁₆	2	2
ゼロページ	ORA \$ zz	05 ₁₆ , zz ₁₆	2	3
ゼロページ X	ORA \$ zz, X	15 ₁₆ , zz ₁₆	2	4
絶対	ORA \$ hhll	0D ₁₆ , ll ₁₆ , hh ₁₆	3	4
絶対 X	ORA \$ hhll, X	1D ₁₆ , ll ₁₆ , hh ₁₆	3	5
絶対 Y	ORA \$ hhll, Y	19 ₁₆ , ll ₁₆ , hh ₁₆	3	5
(間接 X)	ORA (\$ zz, X)	01 ₁₆ , zz ₁₆	2	6
(間接 Y)	ORA (\$ zz), Y	11 ₁₆ , zz ₁₆	2	6

注 . Tフラグが1のとき、サイクル数を + 3します。

動作 : (M(S)) (A)
(S) (S) - 1

説明 : Sの示す番地のメモリへ、Aの内容を退避し、Sの内容を、1減少させます。

ステータス : 変化しません。
フラグ

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
インブライド	PHA	4816	1	3

動作 : (M(S)) (PS)
(S) (S) - 1

説明 : Sの示す番地のメモリへ、PSの内容を退避し、Sの内容を、1減少させます。

ステータス : 変化しません。
フラグ

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
インブライド	PHP	0816	1	3

動作 : (S) (S) + 1
(A) (M(S))

説明 : Sの内容を、1増加させ、Sの示す番地のメモリの内容を、Aに入れます。

ステータスフラグ : N : 実行の結果、ビット7が1のとき、1になります。それ以外
のときは0になります。

V : 変化しません。

T : 変化しません。

B : 変化しません。

D : 変化しません。

I : 変化しません。

Z : 実行の結果が0のとき、1になります。それ以外
のときは0になります。

C : 変化しません。

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
インブライド	PLA	6816	1	4

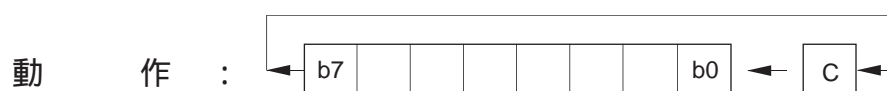
動作 : (S) (S) + 1
(PS) (M(S))

説明 : Sの内容を、1増加させ、Sの示す番地のメモリの内容を、PSに入れます。

ステータス : スタックに退避していた値に変化します。
フラグ

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
インブライド	PLP	2816	1	4

注．PLP命令実行後には必ずNOPを入れてください。



説明 : A 又は M を C フラグとつなげて、その内容を左へ1ビット回転します。A 又は M のビット 0 には、C フラグの内容が入り、C フラグには、A 又は M のビット 7 の内容が入ります。

ステータス フラグ : N : 実行前のビット 6 が 1 のとき、1 になります。それ以外の場合は、0 になります。

V : 変化しません。

T : 変化しません。

B : 変化しません。

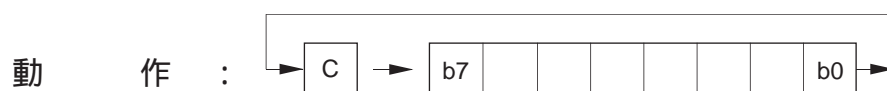
D : 変化しません。

I : 変化しません。

Z : 実行の結果が 0 のとき、1 になります。それ以外の場合は、0 になります。

C : 実行前のビット 7 が 1 のとき、1 になります。それ以外の場合は、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
アキュムレータ	ROL A	2A16	1	2
ゼロページ	ROL \$zz	2616, zz16	2	5
ゼロページ X	ROL \$zz, X	3616, zz16	2	6
絶対	ROL \$hhll	2E16, ll16, hh16	3	6
絶対 X	ROL \$hhll, X	3E16, ll16, hh16	3	7



説明 : A 又は M を C フラグとつなげて、その内容を右へ1ビット回転します。A 又は M のビット7には、C フラグの内容が入り、C フラグには、A 又は M のビット0の内容が入ります。

ステータス フラグ : N : 実行前のCフラグの内容が1のとき、1になります。それ以外のときは、0になります。

V : 変化しません。

T : 変化しません。

B : 変化しません。

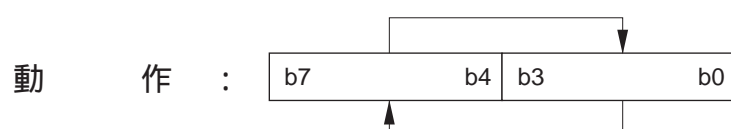
D : 変化しません。

I : 変化しません。

Z : 実行の結果が0のとき、1になります。それ以外のときは、0になります。

C : 実行前のビット0が1のとき、1になります。それ以外のときは、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
アキュムレータ	ROR	A	6A16	1	2
ゼロページ	ROR	\$ zz	6616, zz16	2	5
ゼロページ X	ROR	\$ zz, X	7616, zz16	2	6
絶対	ROR	\$ hhll	6E16, ll16, hh16	3	6
絶対 X	ROR	\$ hhll, X	7E16, ll16, hh16	3	7



説明 : Mの内容を右へ4ビット回転します。

ステータス : 変化しません。

フラグ

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
ゼロページ	RRF \$zz	8216, ZZ16	2	8

動作 : (S) (S) + 1
 (PS) (M(S))
 (S) (S) + 1
 (PCL) (M(S))
 (S) (S) + 1
 (PCH) (M(S))

説明 : 割り込みが受け付けられたときにスタックに退避したステータスフラグとPCの内容を復帰し、割り込みが受け付けられる前と同じ状態に戻します。
 ただし、M(S)はSが示す番地のメモリの内容です。

ステータスフラグ : スタックに退避していた値に変化します。

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
インプライド	RTI	40 ₁₆	1	6

動作 : (S) (S) + 1
 (PCL) (M(S))
 (S) (S) + 1
 (PCH) (M(S))
 (PC) (PC) + 1

説明 : サブルーチンへジャンプしたときに、スタックに退避した内容をPCへ格納します。そして、PCを1増加します。このとき、PCはJSRの次の命令を指しています。ただし、M(S)はSが示す番地のメモリの内容です。

ステータス : 変化しません。
 フラグ

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
インブライド	RTS	6016	1	6

動作 : (T) = 0 のとき (A) (A) - (M) - (\bar{C})
 (T) = 1 のとき (M(X)) (M(X)) - (M) - (\bar{C})

説明 : Tフラグが0のとき、Aの内容から、Mの内容及び、Cフラグの内容の補数を減算して、結果をA及びCフラグに入れます。
 Tフラグが1のとき、M(X)の内容から、Mの内容及び、Cフラグの内容の補数を減算して、結果をM(X)及びCフラグに入れます。このとき、Aの内容は変化しませんが、ステータスフラグは変化します。ただし、M(X)はXが示す番地のメモリの内容です。

ステータスフラグ : N : 演算の結果、ビット7が1のとき、1になります。それ以外の場合は0になります。
 V : 演算の結果が + 127 または | - 128 | を越えると、1になります。それ以外の場合は0になります。
 T : 変化しません。
 B : 変化しません。
 D : 変化しません。
 I : 変化しません。
 Z : 演算の結果が0のとき、1になります。それ以外の場合は0になります。
 C : 結果が0より大きいか等しいとき、1になります。それ以外の場合は0になり、ボローを示します。

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
イミディエイト	SBC # \$ nn	E9 ₁₆ , nn ₁₆	2	2
ゼロページ	SBC \$ zz	E5 ₁₆ , zz ₁₆	2	3
ゼロページ X	SBC \$ zz, X	F5 ₁₆ , zz ₁₆	2	4
絶対	SBC \$ hhll	ED ₁₆ , ll ₁₆ , hh ₁₆	3	4
絶対 X	SBC \$ hhll, X	FD ₁₆ , ll ₁₆ , hh ₁₆	3	5
絶対 Y	SBC \$ hhll, Y	F9 ₁₆ , ll ₁₆ , hh ₁₆	3	5
(間接 X)	SBC (\$ zz, X)	E1 ₁₆ , zz ₁₆	2	6
(間接 Y)	SBC (\$ zz), Y	F1 ₁₆ , zz ₁₆	2	6

注 1. Tフラグが1のときサイクル数を + 3します。

2. 10進演算モード(Dフラグ = 1)でSBC命令を実行させた場合、SEC, CLC, CLD命令は、SBC命令から一命令以上後に行ってください。

また、10進演算モードでは、N, V, Zフラグが無効となります。

動作 : (Ai) 1 又は、(Mi) 1

説明 : A 又は M の、指定されたビット i の内容を 1 にします。

ステータス : 変化しません。
フラグ

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
アキュムレータ・ビット	SEB i, A	$(20i + B)_{16}$	1	2
ゼロページ・ビット	SEB i, \$zz	$(20i + F)_{16}, zz_{16}$	2	5

動作 : (C) 1

説明 : Cフラグの内容を1にします。

ステータス : N : 変化しません。
フラグ V : 変化しません。
T : 変化しません。
B : 変化しません。
D : 変化しません。
I : 変化しません。
Z : 変化しません。
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
インプライド	SEC	3816	1	2

動作 : (D) 1

説明 : Dフラグの内容を1にします。

ステータス : N : 変化しません。
フラグ V : 変化しません。
T : 変化しません。
B : 変化しません。
D : 1になります。
I : 変化しません。
Z : 変化しません。
C : 変化しません。

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
インプライド	SED	F816	1	2

動作 : (I) 1

説明 : Iフラグの内容を1にします。

ステータス : N : 変化しません。
 フラグ V : 変化しません。
 T : 変化しません。
 B : 変化しません。
 D : 変化しません。
 I : 1になります。
 Z : 変化しません。
 C : 変化しません。

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
インプライド	SEI	7816	1	2

動作 : (T) 1

説明 : Tフラグの内容を1にします。

ステータス : N : 変化しません。
フラグ V : 変化しません。
T : 1になります。
B : 変化しません。
D : 変化しません。
I : 変化しません。
Z : 変化しません。
C : 変化しません。

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
インプライド	SET	3216	1	2

動作 : (M) (A)

説明 : Aの内容をMに格納します。このとき、Aは変化しません。

ステータス : 変化しません。

フラグ

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
ゼロページ	STA \$zz	8516, zz16	2	4
ゼロページ X	STA \$zz, X	9516, zz16	2	5
絶対	STA \$hhll	8D16, ll16, hh16	3	5
絶対 X	STA \$hhll, X	9D16, ll16, hh16	3	6
絶対 Y	STA \$hhll, Y	9916, ll16, hh16	3	6
(間接 X)	STA (\$zz, X)	8116, zz16	2	7
(間接 Y)	STA (\$zz, Y)	9116, zz16	2	7

動作 : 発振器の発振を停止します。

説明 : STP命令を実行することにより発振制御フリップフロップがリセットされるため、発振が停止します。再起動するためには、割り込みをかけるかリセットする必要があります。

ステータス : 変化しません。
フラグ

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
インブライド	STP	4216	1	2

注 . STP命令が無効な品種があります。STP命令が無効な場合にSTP命令を実行するとNOP命令の実行と同一(サイクル数は2)です。

動作 : (M) (X)

説明 : Xの内容をMに格納します。このとき、Xは変化しません。

ステータス : 変化しません。
フラグ

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
ゼロページ	STX \$zz	8616, zz16	2	4
ゼロページ Y	STX \$zz, Y	9616, zz16	2	5
絶対	STX \$hhll	8E16, ll16, hh16	3	5

動作 : (M) (Y)

説明 : Yの内容をMに格納します。このとき、Yは変化しません。

ステータス : 変化しません。
フラグ

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
ゼロページ	STY \$zz	84 ₁₆ , zz ₁₆	2	4
ゼロページ X	STY \$zz, X	94 ₁₆ , zz ₁₆	2	5
絶対	STY \$hhll	8C ₁₆ , ll ₁₆ , hh ₁₆	3	5

動作 : (X) (A)

説明 : Aの内容をXへ転送します。このとき、Aは変化しません。

ステータス フラグ : N : 転送した結果、ビット7が1のとき、1になります。それ以外のときは0になります。

V : 変化しません。

T : 変化しません。

B : 変化しません。

D : 変化しません。

I : 変化しません。

Z : 転送の結果が0のとき、1になります。それ以外のときは0になります。

C : 変化しません。

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
インプライド	TAX	AA16	1	2

動作 : (Y) (A)

説明 : Aの内容をYへ転送します。このとき、Aは変化しません。

ステータスフラグ : N : 転送の結果、ビット7が1のとき、1になります。それ以外
のときは0になります。

V : 変化しません。

T : 変化しません。

B : 変化しません。

D : 変化しません。

I : 変化しません。

Z : 転送の結果が0のとき、1になります。それ以外
のときは0になります。

C : 変化しません。

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
インプライド	TAY	A816	1	2

動作 : $M = 0?$

説明 : Mの内容が0かどうかテストして、NフラグとZフラグを変化させます。

ステータス : N : Mの内容のビット7が1のとき、1になります。それ以外のときは0になります。

V : 変化しません。

T : 変化しません。

B : 変化しません。

D : 変化しません。

I : 変化しません。

Z : Mの内容が0のとき、1になります。それ以外のときは0になります。

C : 変化しません。

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
ゼロページ	TST \$ZZ	6416, ZZ16	2	3

動作 : (X) (S)

説明 : Sの内容をXへ転送します。

ステータス フラグ : N : 実行の結果、ビット7が1のとき、1になります。それ以外
のときは0になります。

V : 変化しません。

T : 変化しません。

B : 変化しません。

D : 変化しません。

I : 変化しません。

Z : 実行の結果が0のとき、1になります。それ以外
のときは0になります。

C : 変化しません。

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
インプライド	TSX	BA16	1	2

動作 : (A) (X)

説明 : Xの内容をAへ転送します。

ステータス フラグ : N : 実行の結果、ビット7が1のとき、1になります。それ以外
のときは0になります。

V : 変化しません。

T : 変化しません。

B : 変化しません。

D : 変化しません。

I : 変化しません。

Z : 実行の結果が0のとき、1になります。それ以外
のときは0になります。

C : 変化しません。

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
インプライド	TXA	8A16	1	2

動作 : (S) (X)

説明 : Xの内容をSへ転送します。

ステータス : 変化しません。
フラグ

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
インプライド	TXS	9A16	1	2

動作 : (A) (Y)

説明 : Yの内容をAへ転送します。

ステータス フラグ : N : 実行の結果、ビット7が1のとき、1になります。それ以外
のときは0になります。

V : 変化しません。

T : 変化しません。

B : 変化しません。

D : 変化しません。

I : 変化しません。

Z : 実行の結果が0のとき、1になります。それ以外
のときは0になります。

C : 変化しません。

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
インプライド	TYA	9816	1	2

動作 : 内部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、割り込みをかけるかリセットする必要があります。

ステータス : 変化しません。
フラグ

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
インブライド	WIT	C216	1	2

3.4 割り込み処理、サブルーチン処理に関する命令

3.4.1 割り込み処理に関する命令

割り込みを受け付けるとプロセッサステータスレジスタの内容はスタックポインタで示されるメモリに退避されますので PHP 命令を実行する必要はありません。

アキュムレータの内容を退避する必要がある場合は割り込みルーチン内で PHA 命令を実行します(アキュムレータを操作する命令より前に実行します)。このような PHA などスタック操作命令を割り込みルーチン内で実行した場合、必ずその命令に対する PLA などの命令を同じ割り込みルーチン内で実行してください。

割り込みルーチンから戻る場合は RTI 命令を実行してください。

3.4.2 割り込み制御に関する命令

割り込みを制御する要因として割り込み禁止フラグ(I)、各割り込み要因に対応する割り込み許可ビット、要求ビットがあります(BRK 命令によるソフトウェア割り込みは除く)。

(1) 割り込みの禁止

割り込みを禁止するには、SEI 命令によって割り込み禁止フラグIをセット("1")する方法と、割り込み許可ビットを LDM 命令、CLB 命令など(この他にもいろいろな命令が使用できます)によってクリア("0")する方法があります。

(2) 割り込みの許可

割り込みを許可するには、割り込み許可ビットをLDM命令、SEB命令などによってセット("1")し、さらにCLI命令によって割り込み禁止フラグIをクリア("0")します。

(3) 割り込み要求の解除

割り込みが発生すると、その割り込み要因に対応する割り込み要求ビットが自動的にセット("1")され、割り込み受け付けと同時に自動的にクリア("0")されますので割り込みルーチン内での、プログラムによる割り込み要求ビットのクリア操作は不要です。

割り込み禁止状態においても割り込みが発生すると、割り込み要求ビットがセット("1")されます。そのまま割り込み許可状態(割り込み禁止フラグI=0かつ割り込み許可ビット=1の状態)にすると、同時に割り込みが受け付けられますが、その割り込みを受け付けたくない場合は割り込み許可状態にする前に LDM 命令、CLB 命令などによって割り込み要求ビットをクリア("0")します。この場合、以下の点に注意してください。

割り込み禁止フラグIが"0"である場合、割り込み要求ビットのクリア("0")と割り込み許可ビットのクリア("0")をLDM命令などにより同時に行いますと、実際には要求ビットのクリアよりも早く割り込み許可状態になるため割り込みを受け付けてしまいます。

このような場合はCLB命令などにより、まず要求ビットをクリア("0")してから割り込み禁止状態の解除を行ってください。

(4) 割り込みルーチン内での割り込み制御

割り込みを受け付けたあと割り込みルーチンに入ったとき、割り込み禁止フラグは自動的にセット("1")され、多重割り込みが禁止されます。多重割り込みを許可する場合は割り込みルーチン内で CLI 命令を用いて I フラグをクリア("0")してください。

3.4.3 サブルーチン処理に関する命令

サブルーチンへのジャンプは通常 JSR 命令を使用します。この命令を実行すると現在のプログラムカウンタの値が自動的に PCH、PCL の順でスタックに退避され、スタックポインタも、その分だけ移動します。ただし、割り込み処理時と異なりプロセッサステータスレジスタの自動退避が、サブルーチン呼び出しの際には行われませんので、もしプロセッサステータスレジスタを退避する必要がある場合は PHP 命令を実行してください。プロセッサステータスレジスタの内容は JSR 命令を実行しても変化しませんので PHP 命令によるプロセッサステータスレジスタの退避は、JSR 命令の直前でも直後(サブルーチンの先頭)でも、どちらでも構いません。ただし、サブルーチンの中で、このようなスタック操作命令を実行する場合、必ずそのサブルーチンから戻る前に(サブルーチンの中で)逆のスタック操作を行ってください。

サブルーチンから復帰するには RTS 命令を実行します。この命令を実行すると JSR 命令実行により退避されていた戻り先番地が自動的にプログラムカウンタに戻されます。この場合も割り込み処理時とは違ってプロセッサステータスレジスタの内容は戻されません。もしサブルーチン内で PHP 命令、PHA 命令によるプロセッサステータスレジスタ、アキュムレータの退避を行った場合、必ずサブルーチンから戻る前に(サブルーチンの中で)PLP 命令、PLA 命令など逆のスタック操作を行うようにしてください。

割り込みおよびサブルーチン処理時のスタックへの退避及び復帰動作を図 3.4.1 に示します。アキュムレータとプロセッサステータスレジスタの退避及び復帰命令を表 3.4.1 に示します。

割り込み処理、サブルーチン処理に関する命令

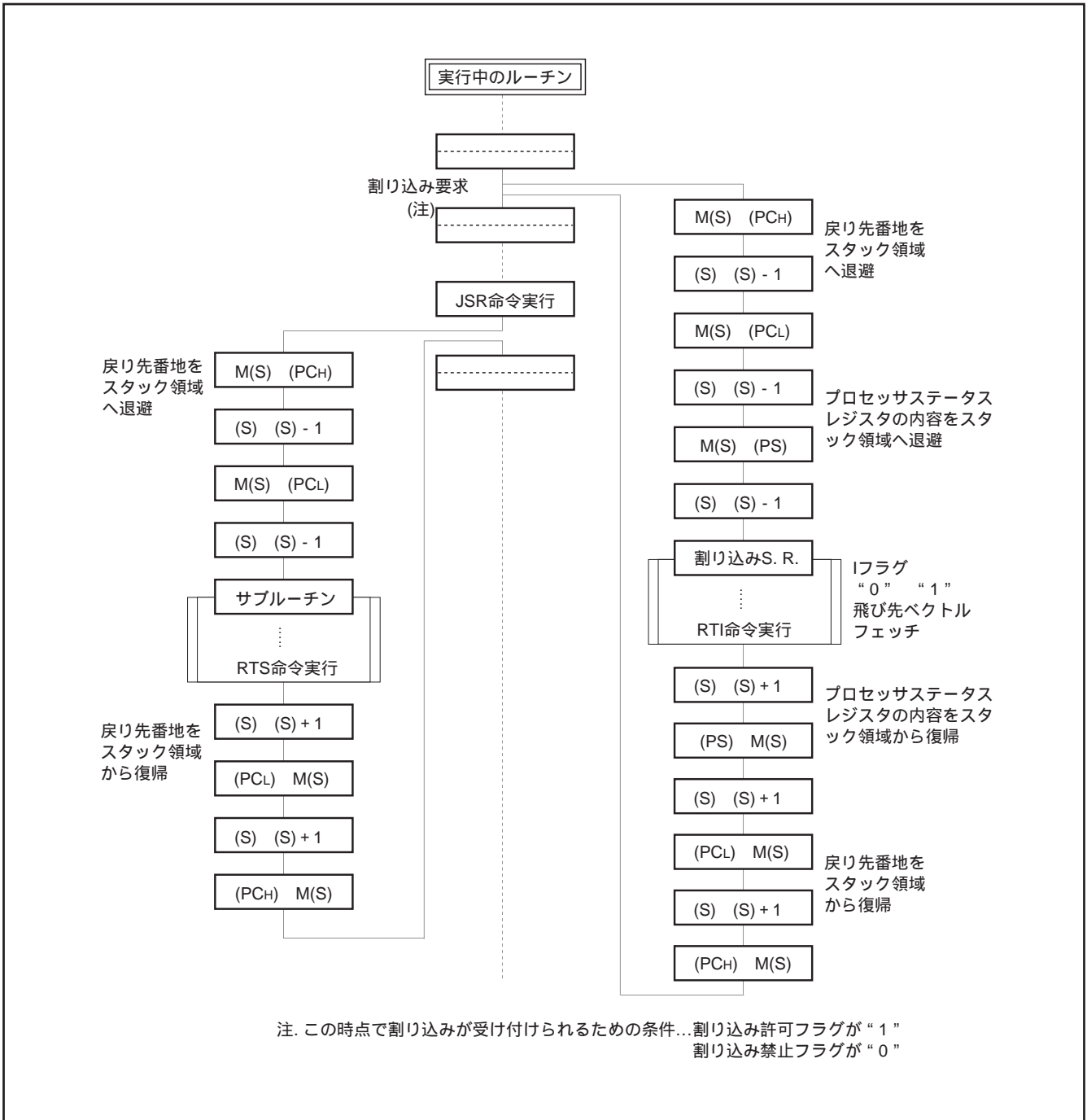


図3.4.1 スタックへの退避及び復帰動作

表3.4.1 アキュムレータとプロセッサステータスレジスタの退避命令及び復帰命令

	スタックに退避する命令	スタックより復帰する命令
アキュムレータ	PHA	PLA
プロセッサステータスレジスタ	PHP	PLP

4. 使用上の注意事項

740ファミリ共通の注意事項のみ掲載しています。ご使用の品種の注意事項を併せて参照ください。

4.1 入出力ポートに関する注意事項

4.1.1 スタンバイ状態での使用

スタンバイ状態^{*1}で使用する場合は、入出力ポートの入力レベルを不定の状態にしないでください。特にNチャンネルオープンドレインの入出力ポートでは注意が必要です。また、Nチャンネルオープンドレインの入出力ポートでは、出力に設定している場合でも、同様の注意が必要です。

この場合、抵抗を介してポートをプルアップ(V_{CC} に接続)又はプルダウン(V_{SS} に接続)してください。

抵抗値を決定する際は、以下の2点に留意してください。

- ・ 外付け回路
- ・ 通常動作時の出力レベルの変動

理由

方向レジスタで入力ポートに設定している場合はトランジスタがOFF状態になるため、ポートはハイインピーダンス状態になります。このとき、入力レベルを不定の状態にすると、マイコン内部の入力バッファに入力される電位が不安定となるため、電源電流が流れることがあります。

また、Nチャンネルオープンドレインの入出力ポートではポートラッチの内容が 1 の場合、方向レジスタで出力ポートに設定していても、入力ポートと同様の現象がおこります。

^{*1}スタンバイ状態：STP命令実行によるストップモード
WIT命令実行によるウェイトモード

4.1.2 ビット処理命令による出力データの書き替え

入出力ポートのポートラッチをビット処理命令*1を用いて書き替える場合、指定していないビットの値が変化することがあります。

理由

入出力ポートは、ビット単位で入力モード又は出力モードを設定できます。ポートレジスタに読み出し、書き込みを行うと次のように動作します。

- ・入力モードのポート

読み出し：端子のレベルを読む。

書き込み：ポートラッチへ書く。

- ・出力モードのポート

読み出し：ポートラッチを読む、又は、周辺機能の出力を読む(ポートにより仕様が異なる)。

書き込み：ポートラッチへ書く(ポートラッチの内容を端子から出力する)。

一方、ビット処理命令はリード・モディファイ・ライト命令*2ですので、ポートレジスタにビット処理命令を実行した場合、命令で指定していないビットにも同時に読み出し及び書き込みが行われます。

指定していないビットが入力モードの場合は、端子のレベルを読み、その値をポートラッチへ書きます。このとき、元のポートラッチの内容と、端子のレベルが違う場合は、ポートラッチの内容が変化します。

指定していないビットが出力モードの場合は、基本的にはポートラッチを読みますが、周辺機能の出力を読むポートもあり、その値をポートラッチへ書きます。このとき、元のポートラッチの内容と、周辺機能の出力が違う場合は、ポートラッチの内容が変化します。

*1ビット処理命令：SEB命令、CLB命令

*2リード・モディファイ・ライト命令：メモリをバイト単位で読み(リード)加工して(モディファイ)元のメモリにバイト単位で書く(ライト)命令

4.2 未使用端子の処理に関する注意事項

未使用端子処理はマイコンの端子からできるだけ短い配線(20mm以内)で処理してください。システムへの影響については、ユーザサイドで十分なシステム評価を行ってください。

4.2.1 未使用端子の適切な処理

①出力専用端子

開放してください。

入力専用端子

各端子ごとに1k~10k(参考値)の抵抗を介して、V_{CC}又はV_{SS}に接続してください。内蔵プルアップ/プルダウン抵抗が選択可能なポートでは、内蔵プルアップ/プルダウン抵抗も使用できます。

また、電圧レベルが動作モードに影響を与える端子(CNV_{SS}端子、INT端子など)は、モードを検討の上、V_{CC}又はV_{SS}を選択してください。

入出力ポート

入力モードに設定し、1k~10k（参考値）の抵抗を介して V_{CC} 又は V_{SS} に接続してください。内蔵プルアップ/プルダウン抵抗が選択可能なポートでは、内蔵プルアップ/プルダウン抵抗も使用できます。

出力モードに設定する場合は、“L”又は“H”出力状態で開放してください。

- ・出力モードに設定して開放する場合、リセット後プログラムによってポートを出力モードに切り替えるまでは、初期状態の入力モードのままです。そのため端子の電圧レベルが不定となり、ポートが入力モードになっている間、電源電流が増加する場合があります。システムへの影響については、ユーザサイドで十分なシステム評価を行ってください。
- ・ノイズやノイズによって引き起こされる暴走などにより方向レジスタが変化する場合を考慮し、定期的に方向レジスタをプログラムで再設定することによって更にプログラムの信頼度が高まります。

A/Dコンバータを使用しない場合のA/D変換用電源端子 AV_{SS}

A/Dコンバータを使用しない場合、A/D変換用電源端子 AV_{SS} 、 AV_{CC} は以下のように処理してください。

- ・ AV_{SS} : V_{SS} に接続
- ・ AV_{CC} : V_{CC} に接続

4.2.2 処理上の留意事項

入力ポート及び入出力ポート

入力モードで開放しないでください。

理由：

- ・初段回路によっては電源電流が増加する場合があります。
- ・上記適切な処理(1)の に比べ、ノイズの影響を受け易くなります。

入出力ポート

入力モードに設定した場合、 V_{CC} 又は V_{SS} に直結しないでください。

理由：

暴走、ノイズなどによって、方向レジスタが出力モードに変化した場合、短絡する可能性があります。

入出力ポート

入力モードに設定した場合、複数ポートをまとめて抵抗を介し、 V_{CC} 又は V_{SS} に接続しないでください。

理由：

暴走、ノイズなどによって、方向レジスタが出力モードに変化した場合、ポート間で短絡する可能性があります。

4.3 割り込みに関する注意事項

4.3.1 割り込み要求ビット及び許可ビットの設定

割り込み準備のための割り込み要求ビットの設定と割り込み許可ビットの設定は以下の順番に従って別々の命令で実行してください。

割り込み要求ビットを“0”(割り込み要求なし)にする。

割り込み許可ビットを“1”(割り込み許可)にする。

理由

上記の設定を1つの命令で行った場合、割り込み要求ビットが“0”になる前に割り込み許可ビットが“1”(割り込み許可)になるため、不要な割り込み処理ルーチンが実行されます。

4.3.2 関連レジスタの設定変更

外部割り込みのアクティブエッジを設定する場合、及び同一割り込みベクトルに複数の割り込み要因が割り当てられている場合にそのベクトルの割り込み要因を切り替える場合、これらの設定に同期した割り込み発生が不要なら、以下の手順で設定してください。

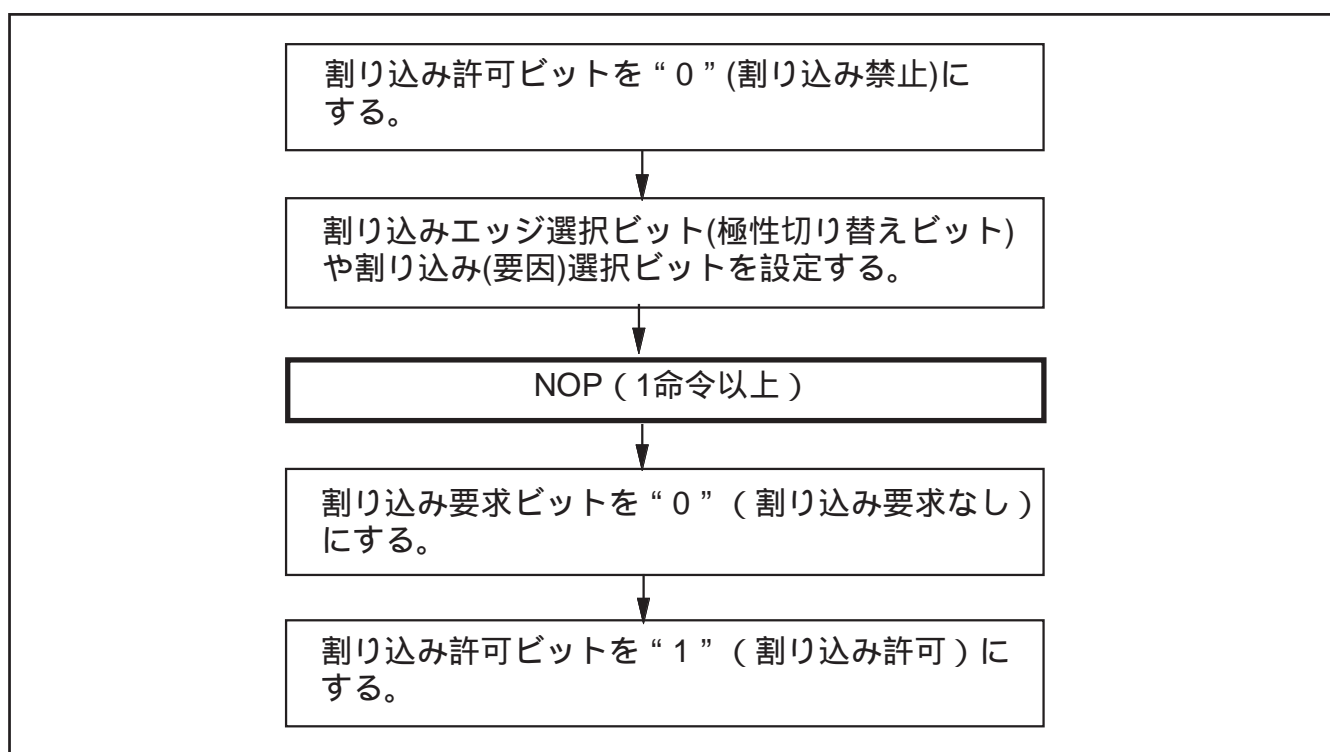


図4.3.1 関連レジスタの設定変更手順

理由

次の場合、割り込み要求ビットが“1”になる場合があります。

- ・ 外部割り込みのアクティブエッジを切り替える際
- ・ 同一割り込みベクトルに複数の割り込み要因が割り当てられたベクトルの割り込み要因を切り替える際

4.3.3 割り込み要求ビットの判定

割り込み要求レジスタの割り込み要求ビットを“0”にした直後、BBC命令又はBBS命令をこの割り込み要求ビットに対して実行する場合は、BBC命令又はBBS命令を実行する前に、1命令実行してください。

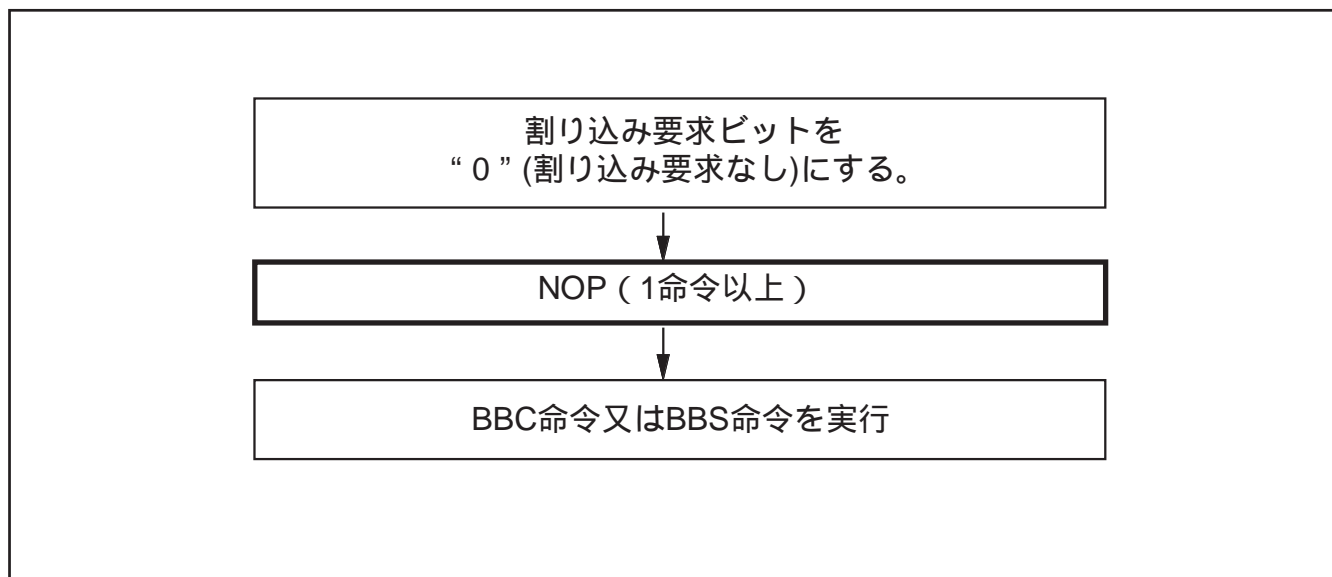


図4.3.2 割り込み要求ビットの判定手順

理由

割り込み要求レジスタの割り込み要求ビットを“0”にした直後にBBC命令又はBBS命令を実行すると、“0”になる前の割り込み要求ビットの値を判定します。

4.4 プログラム作成に関する注意事項

4.4.1 プロセッサステータスレジスタ

(1) プロセッサステータスレジスタの初期化

プログラムの実行に影響を与えるプロセッサステータスレジスタ(PS)のフラグを初期化しておく必要があります。

特にTフラグとDフラグは、演算そのものに影響を与えるため、初期化が必須となります。

理由

プロセッサステータスレジスタ(PS)は、Iフラグが“1”であるのを除いて、リセット直後は不定です。

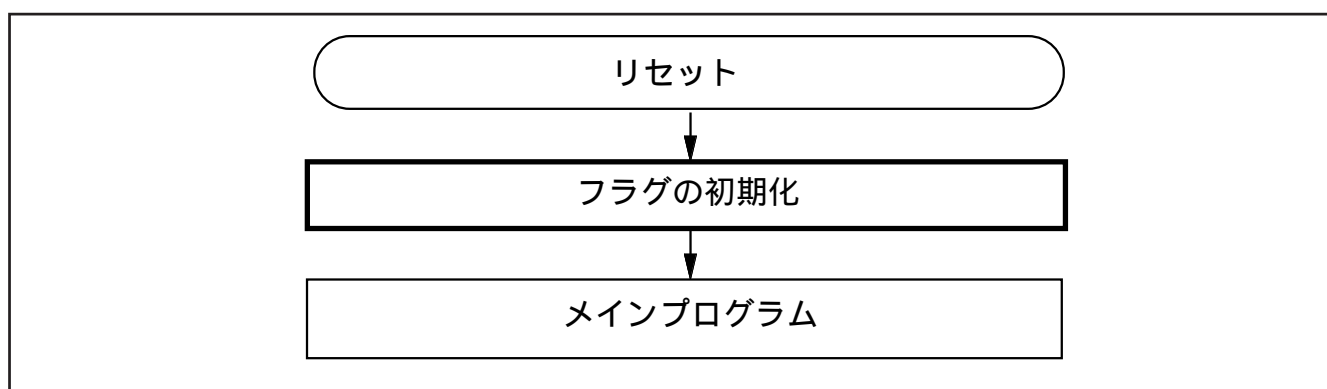


図4.4.1 プロセッサステータスレジスタのフラグの初期化

(2) プロセッサステータスレジスタの参照方法

プロセッサステータスレジスタ(PS)の内容を参照したい場合には、一度PHP命令を実行した後で、(S)+1の内容を読み出します。さらに必要な場合にはPLP命令の実行により退避したPSを元に戻します。

PLP命令実行後には、必ずNOP命令を入れてください。

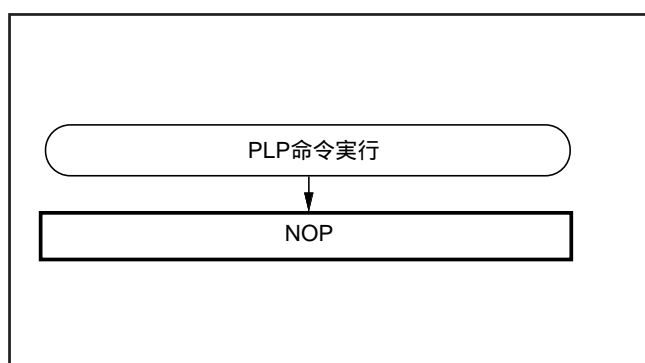


図4.4.2 PLP命令実行時の手順

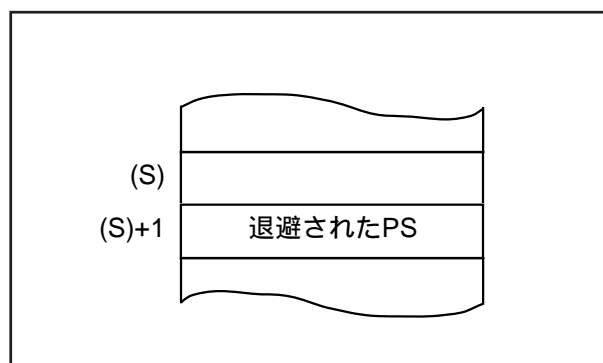


図4.4.3 PHP命令実行後のスタックメモリの内容

4.4.2 BRK命令

(1) 割り込み要因の識別方法

BRK命令と他の割り込み要因が同一割り込みベクトルの場合、割り込み発生要因がBRK命令の割り込みであるか、優先順位が最下位の割り込みであるかを識別する方法として、退避されたBフラグの内容によって判断できますが、この判断は割り込みルーチンの中で行ってください。

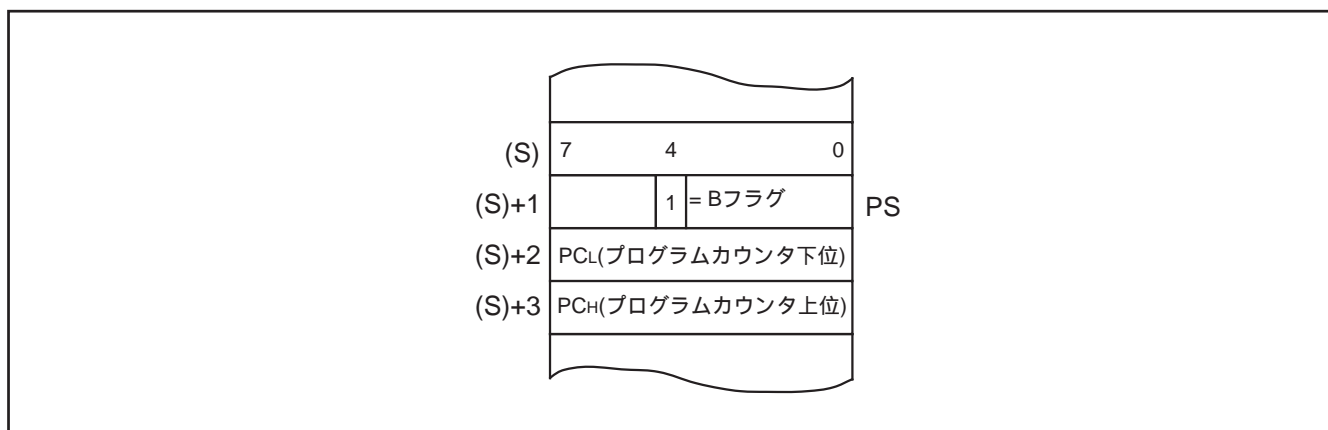


図4.4.4 割り込み処理ルーチン内のスタックメモリの内容

(2) 割り込み優先順位

下記2つの状態である時にBRK命令を実行すると、その要因の中で最も優先順位の高い要因の割り込みベクトルの番地から割り込みの実行を開始します。

- ・ 割り込み要求ビット、割り込み許可ビットが共に“1”
- ・ Iフラグを“1”にして割り込みを禁止

4.4.3 10進演算

(1) 10進演算時の命令

10進演算を行う場合、SED命令により10進モードフラグDを“1”にセットして、ADC命令又はSBC命令を実行します。その場合、SEC命令、CLC命令、又はCLD命令は、ADC命令又はSBC命令よりも一命令後に行ってください。

(2) 10進演算時のステータスフラグ

10進モード(Dフラグ=1)時にADC, SBC命令を実行したとき、ステータスフラグのうちN, V, Zの3つのフラグは無効となります。

また、C(キャリ)フラグは演算の結果、桁上がりが発生すると“1”にセット、桁借りが発生すると“0”にクリアされますので、演算結果の桁上がり、桁借りを判定させるフラグとして利用できます。また、演算前にはCフラグの初期化を行ってください。

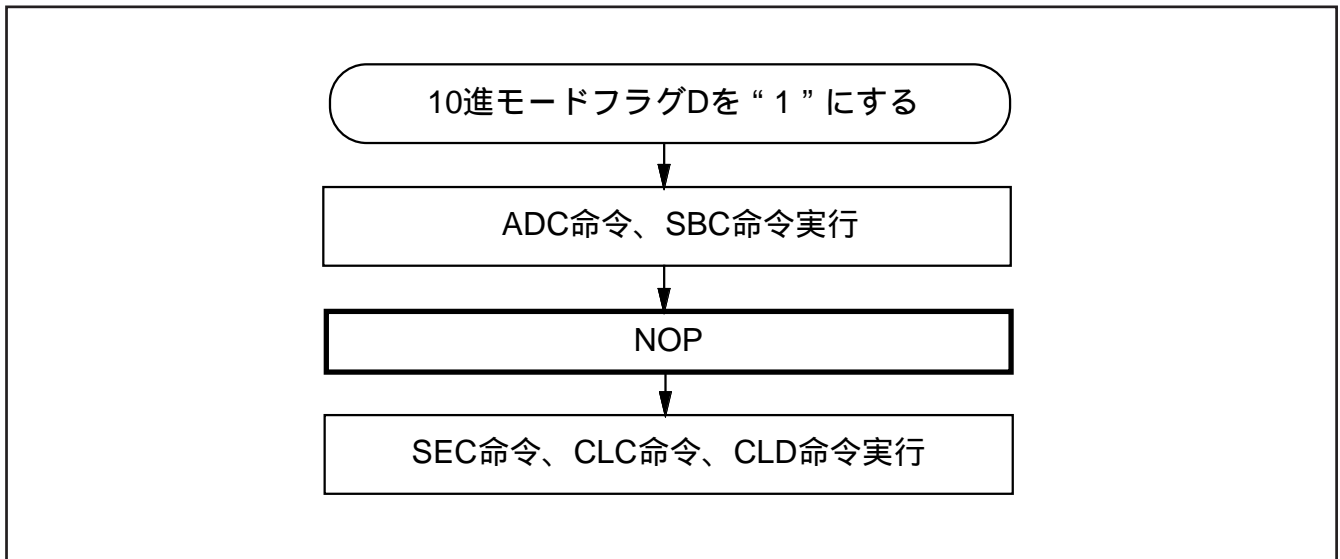


図4.4.5 10進演算時のステータスフラグ

4.4.4 JMP命令

JMP命令(間接アドレッシングモード)を使用する場合、下位8ビットが“FF₁₆”となるアドレスをオペランドに指定しないでください。

4.4.5 乗除算命令

- ・MUL、DIV命令は、T、Dフラグの影響を受けません。
- ・乗除算命令の実行ではプロセッサステータスレジスタの内容は変化しません。

4.4.6 ポート

ポート方向レジスタの値は読み出すことができません。すなわち、LDA命令をはじめ、Tフラグが“1”の場合のメモリ演算命令、方向レジスタの値を修飾値とするアドレッシングモード、BBC、BBSなどのビットテスト命令は使用できません。また、CLB、SEBなどのビット操作命令、RORなどの演算を始めとする方向レジスタのリード・モディファイ・ライト命令も使用できません。方向レジスタの設定はLDM命令、STA命令などを使用してください。

4.4.7 命令の実行時間

命令の実行時間は機械語命令一覧表に記載されているサイクル数に内部クロックの周期をかけることによって得られます。

付録 1 . アドレッシングモード別実行シーケンス

740ファミリは、クロック を基準クロックとして動作します。

各命令のフェッチ時に、クロック の一周期間だけSYNC信号を出力します。SYNC信号出力と同時に、各命令の納められている先頭アドレス(PC)を出力します。PCの出力に、クロック の半周期分の時間だけ遅れて、オペコードのフェッチ期間が存在します。

CPUは、このオペコードをデコードすることにより、その命令が何バイトで構成されているかを知ると共に、以降の動作を決定します。

以下、各命令の動作遷移状態をアドレッシングモード別に示します。

必要バイト数、必要サイクル数は、それぞれその命令に必要なバイト数、クロック のサイクル数を示します。

図中の 、SYNC、R/W(\overline{RD} , \overline{WR})、ADDR(ADDRH, ADDRL)、DATAの各信号は、内部バスの状態を示しています。これらの信号は、シングルチップモードでは直接見ることはできませんが、マイクロプロセッサモードを使用できる製品では、確認することができます。

なお、これらの信号の組み合わせは品種によって異なります。各品種において有効な信号を以下に示します。

各品種において有効な信号

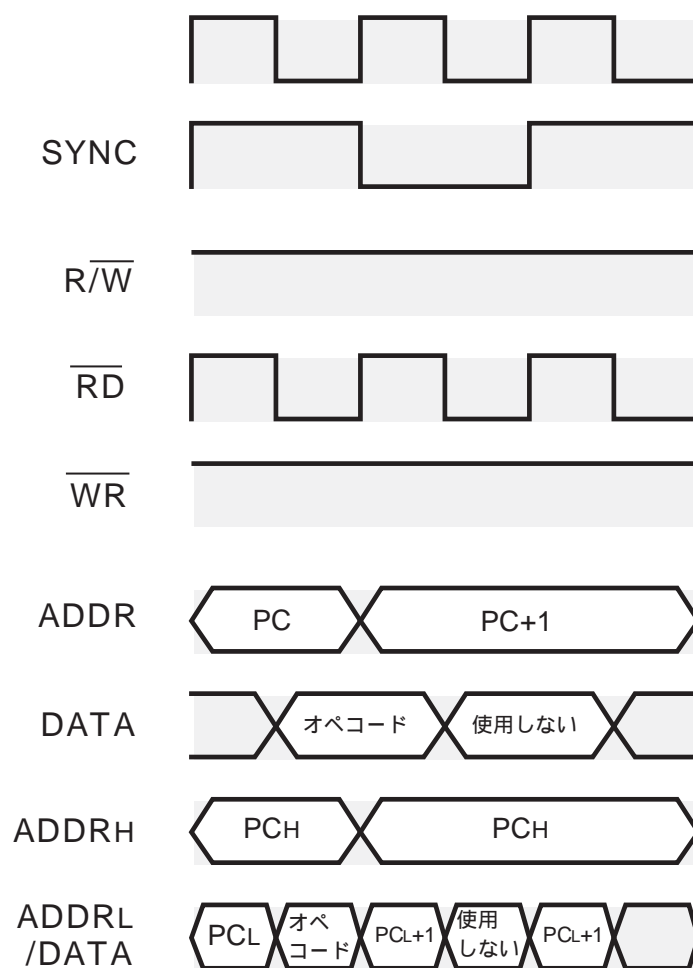
品 種		SYNC	R/W	\overline{RD}	\overline{WR}	ADDR	DATA	ADDRH	ADDRL/DATA
M507XX M509XX M374XX (M37451 を除く)									
M38XXX M375XX M372XX M371XX									
M37451				(注)	(注)				
M50734									

注 . 80ピン版のみ

インプライド(Implied)

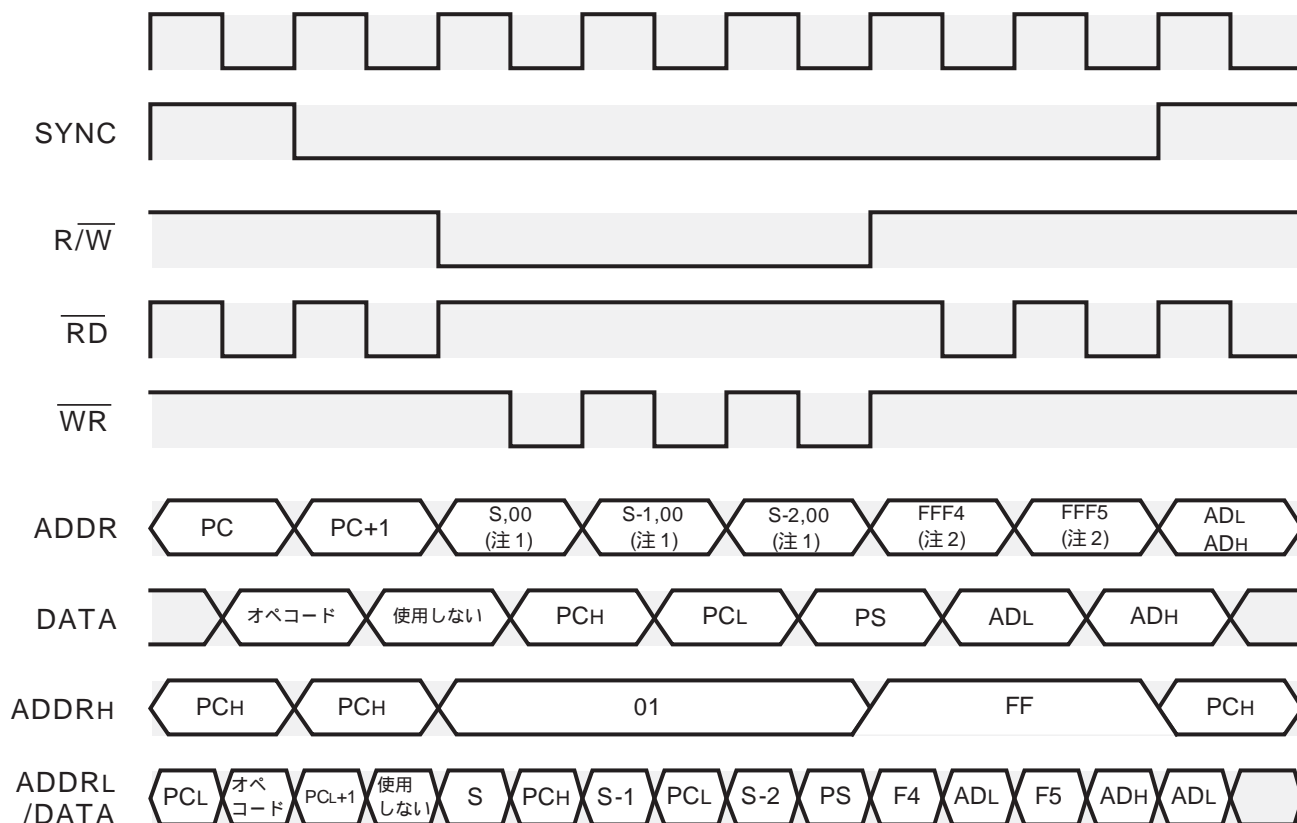
該当命令	:	CLC	SEC
		CLD	SED
		CLI	SEI
		CLT	SET
		CLV	TAX
		DEX	TAY
		DEY	TSX
		INX	TXA
		INY	TXS
		NOP	TYA

必要バイト数 : 1
 必要サイクル数 : 2
 タイミング :



インプライド(Implied)

該当命令 : BRK
 必要バイト数 : 1
 必要サイクル数 : 7
 タイミング :



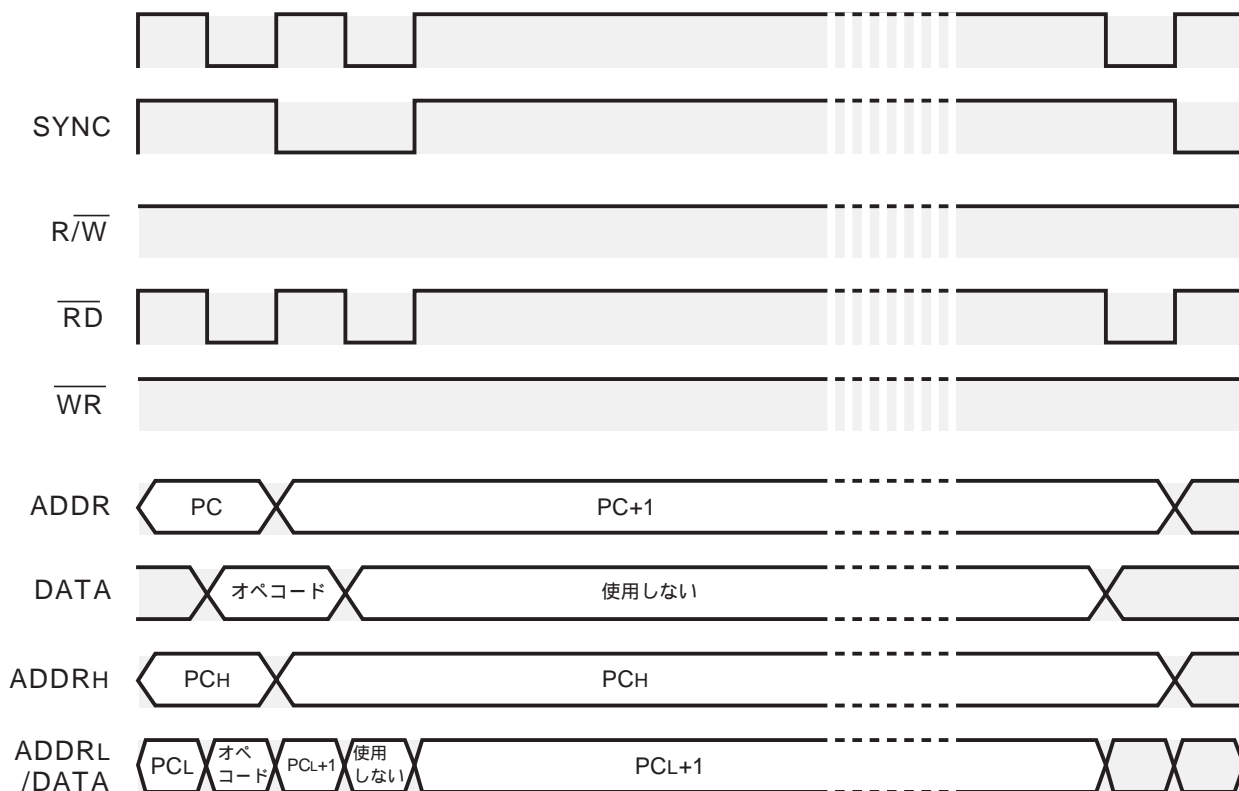
- 注 1. 品種によって“01”又はSPSフラグの内容となります。
 2. 品種によってアドレスが異なる場合があります。

インプライド(Implied)

該当命令 : STP
WIT

必要バイト数 : 1

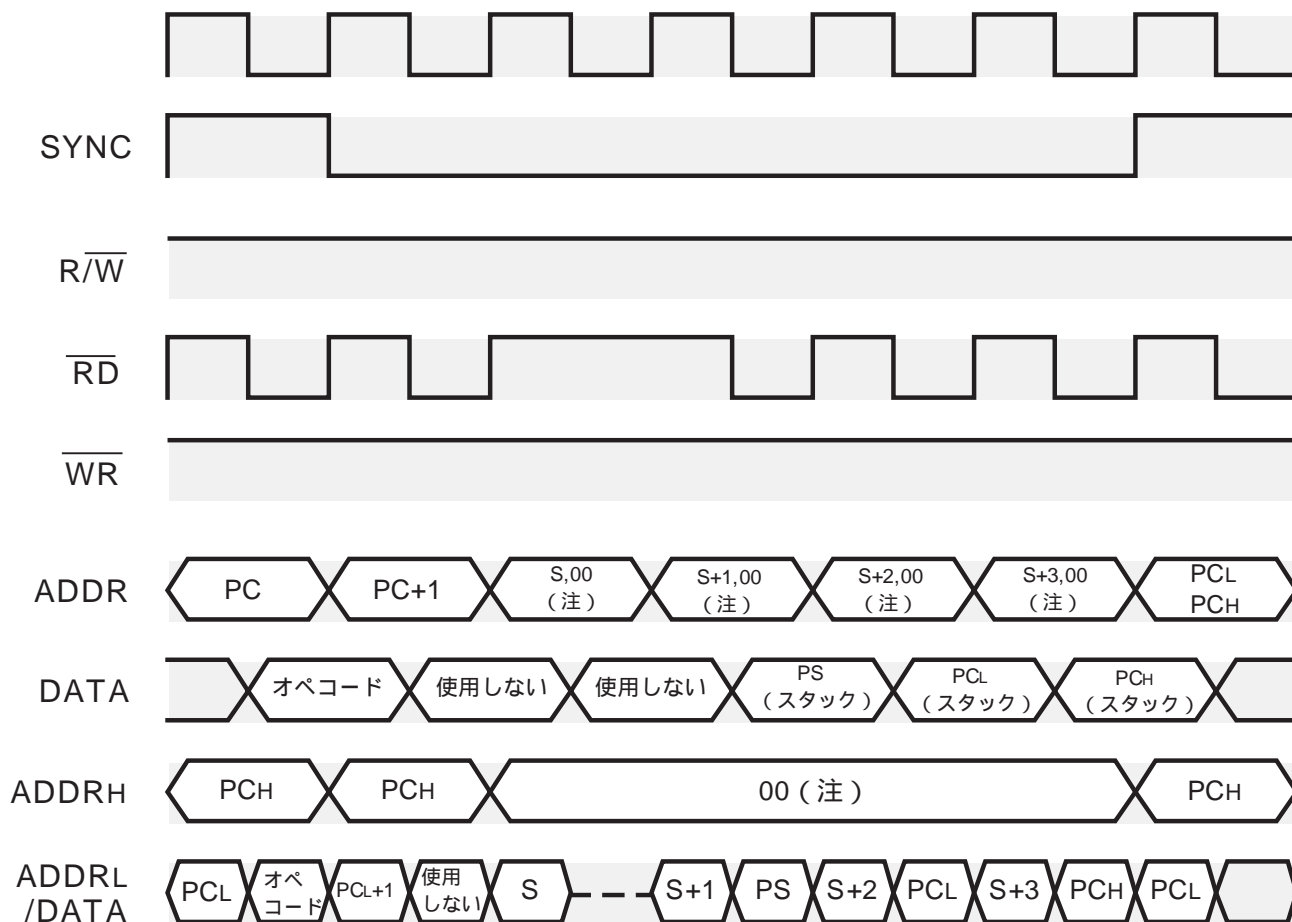
タイミング :



スタンバイ状態からの復帰は外部割り込みにより行われる。
ウェイト状態からの復帰は内/外部割り込みにより行われる。

インプライド(Implied)

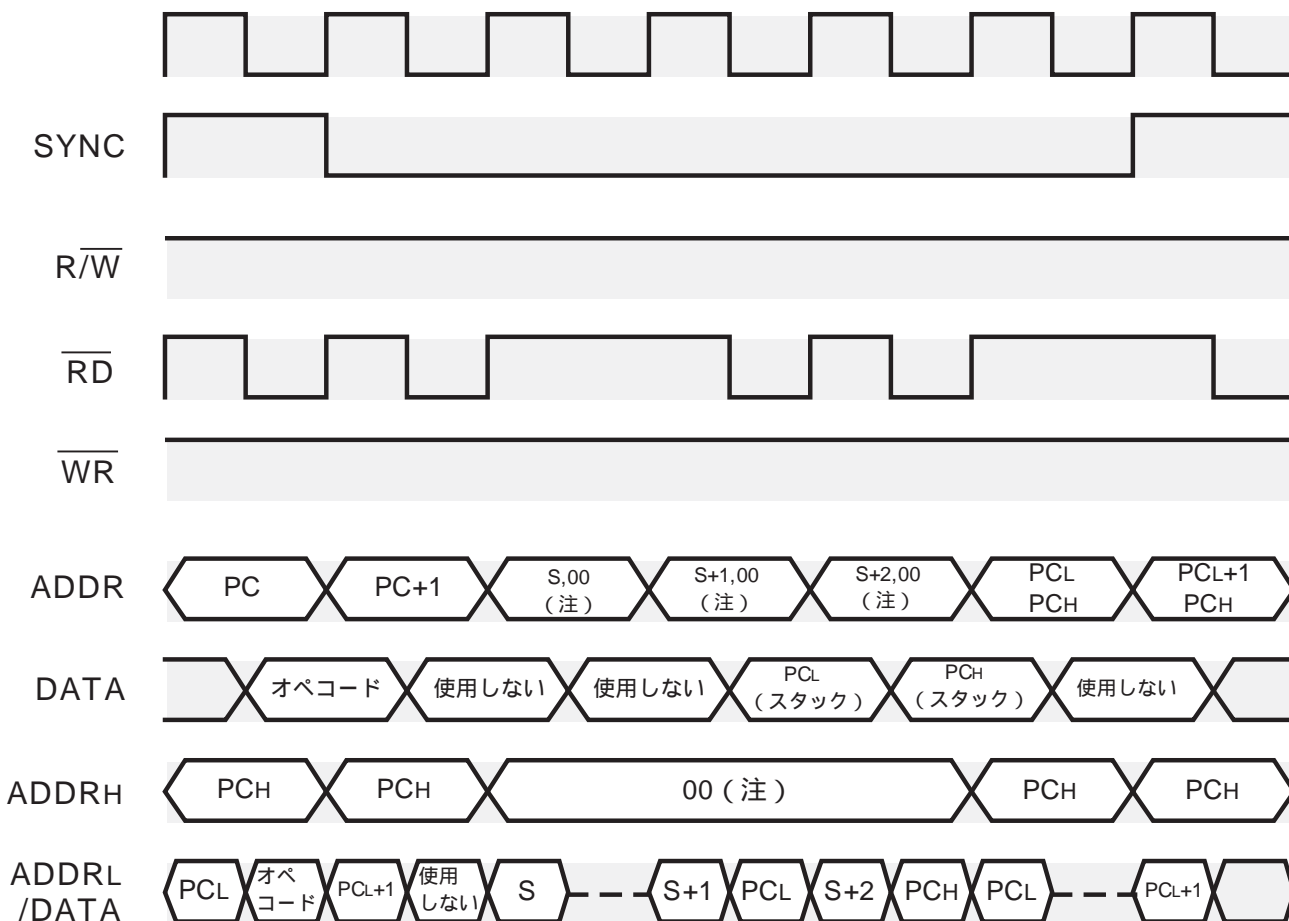
該当命令 : RTI
 必要バイト数 : 1
 必要サイクル数 : 6
 タイミング :



注. 品種によって、“01”又はSPSフラグの内容となります。

インプライド(Implied)

該当命令 : RTS
 必要バイト数 : 1
 必要サイクル数 : 6
 タイミング :



注. 品種によって、“01”又はSPSフラグの内容となります。

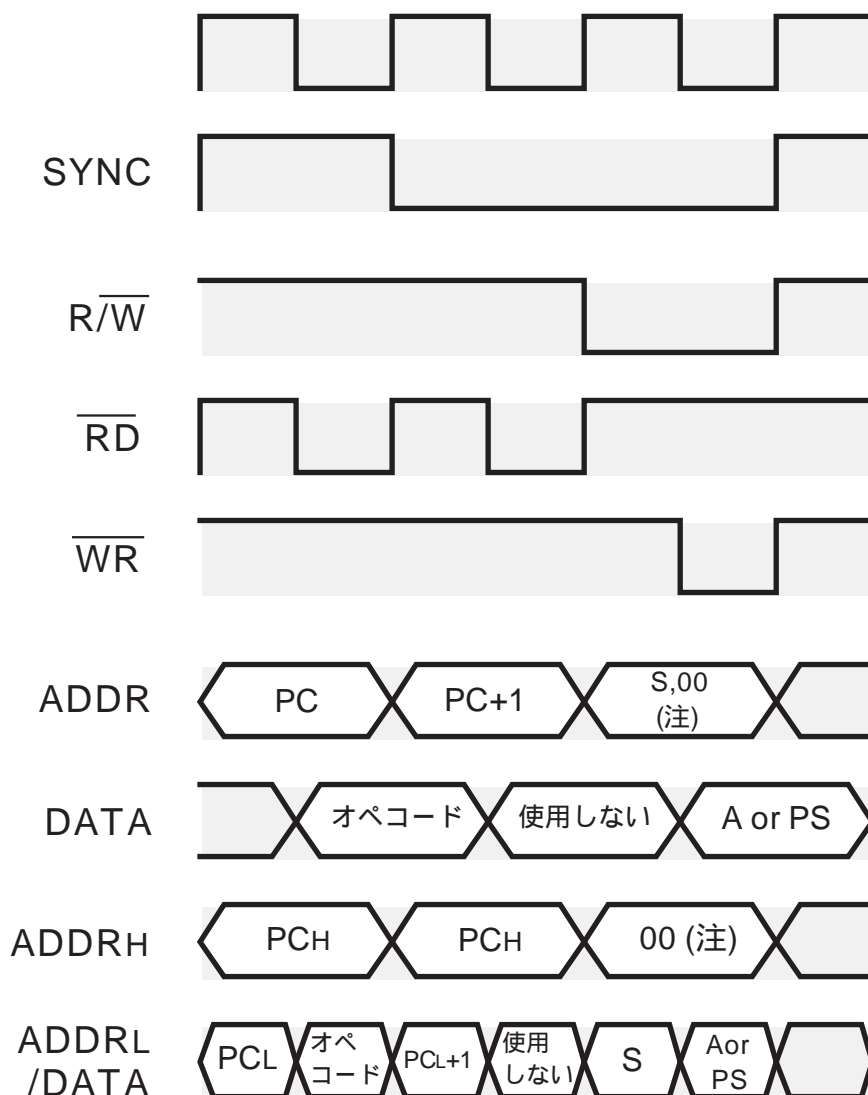
インプライド(Implied)

該当命令 : PHA
PHP

必要バイト数 : 1

必要サイクル数 : 3

タイミング :



注. 品種によって、“01”又はSPSフラグの内容となります。

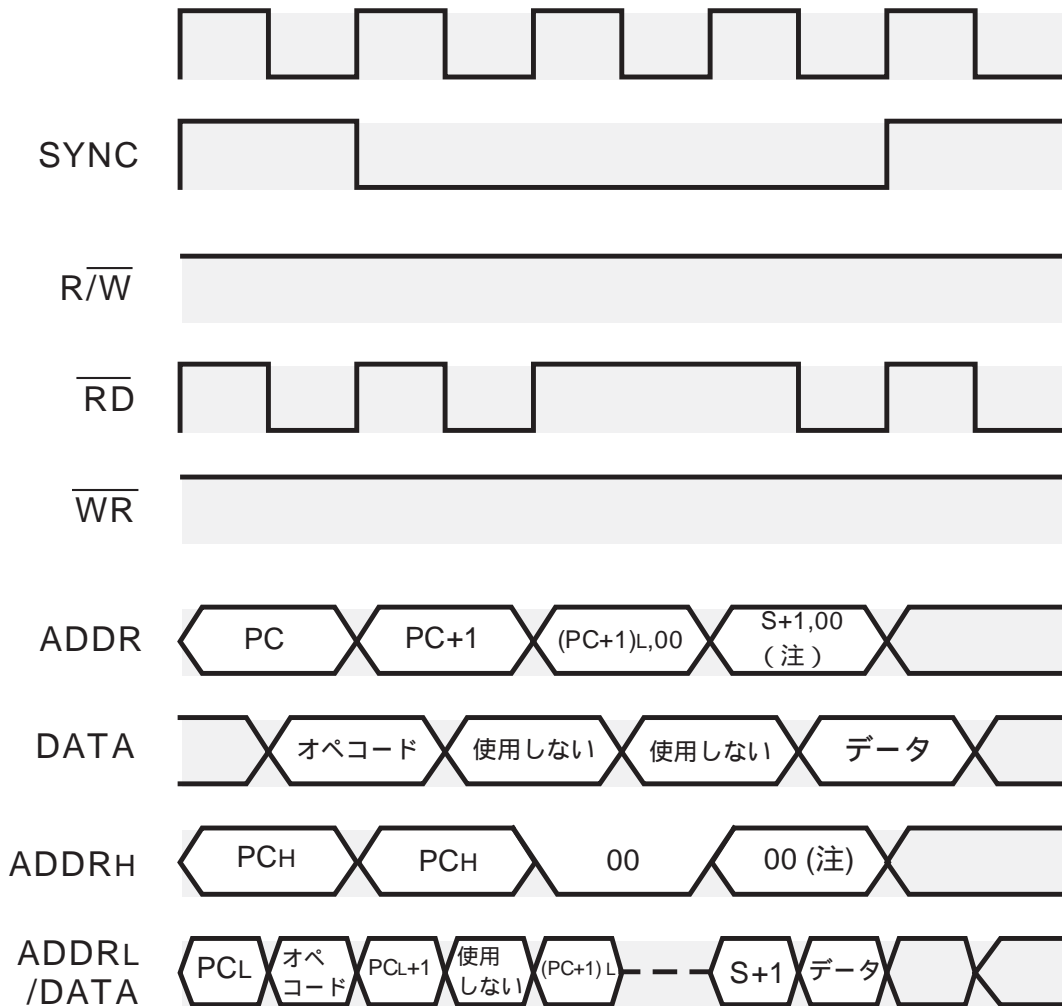
インプライド(Implied)

該当命令 : PLA
PLP

必要バイト数 : 1

必要サイクル数 : 4

タイミング :



注. 品種によって、“01”又はSPSフラグの内容となります。

[T=0]

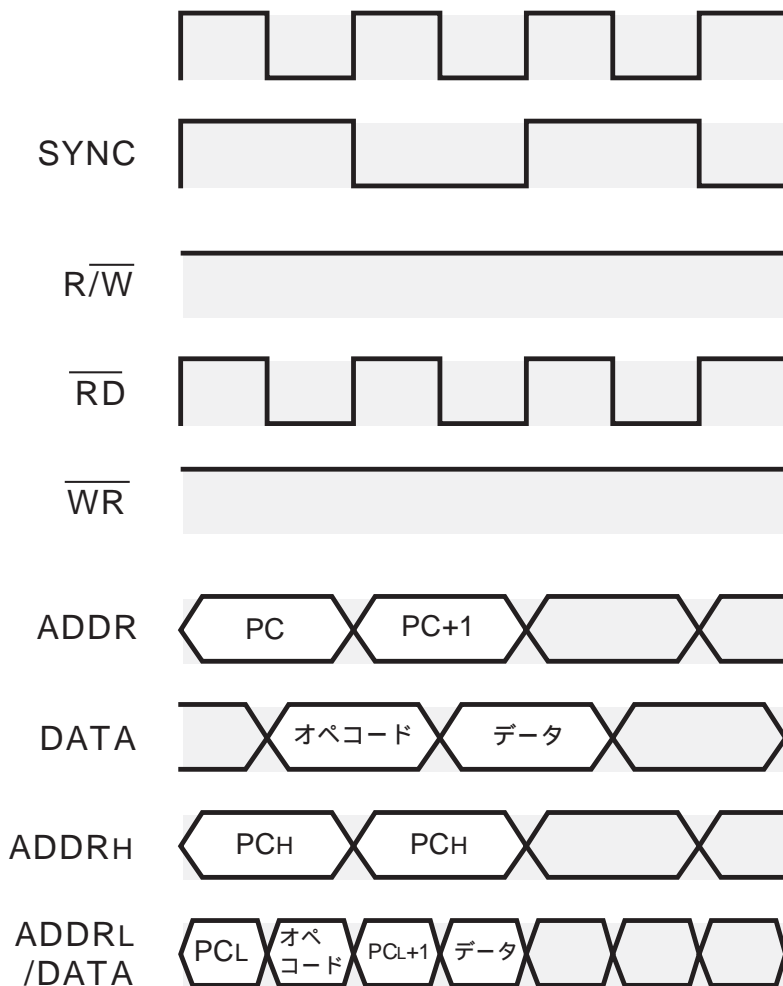
イミディエイト(Immediate)

該当命令 : ADC # \$ nn (T = 0)
 AND # \$ nn (T = 0)
 CMP # \$ nn (T = 0)
 CPX # \$ nn
 CPY # \$ nn
 EOR # \$ nn (T = 0)
 LDA # \$ nn (T = 0)
 LDX # \$ nn
 LDY # \$ nn
 ORA # \$ nn (T = 0)
 SBC # \$ nn (T = 0)

必要バイト数 : 2

必要サイクル数 : 2

タイミング :



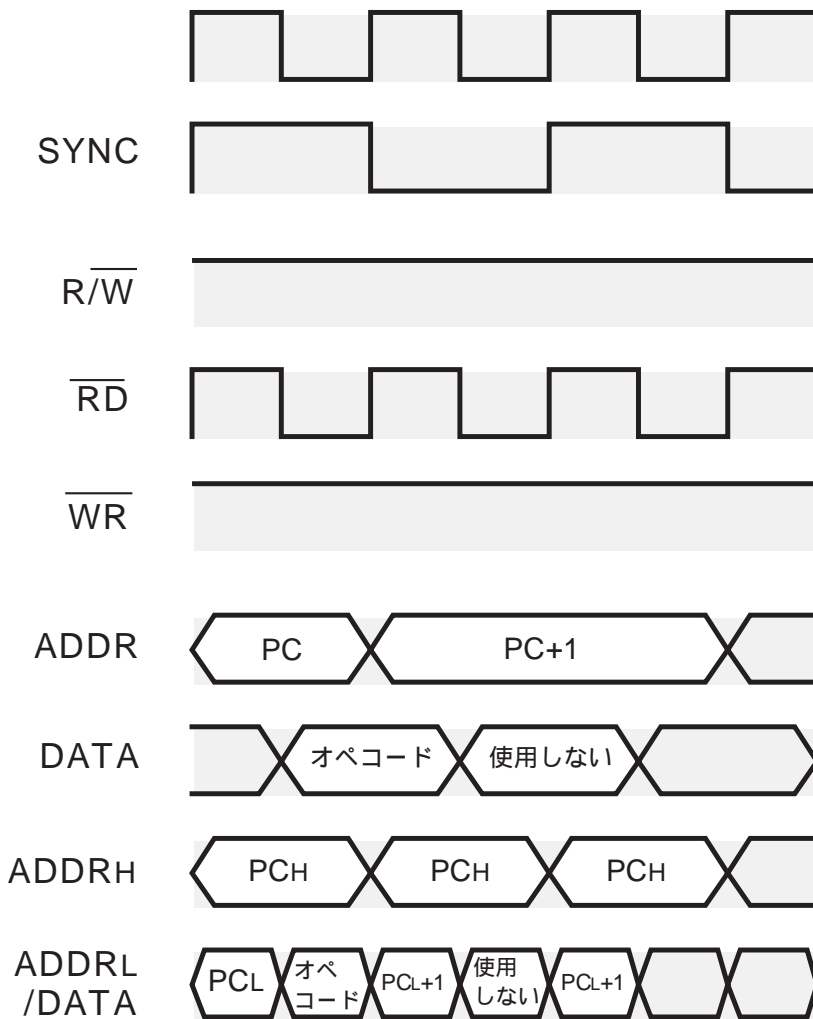
アキュムレータ(Accumulator)

該当命令 : ASL A
 : DEC A
 : INC A
 : LSR A
 : ROL A
 : ROR A

必要バイト数 : 1

必要サイクル数 : 2

タイミング :



アキュムレータ ビット 相対(Accumulator bit relative)

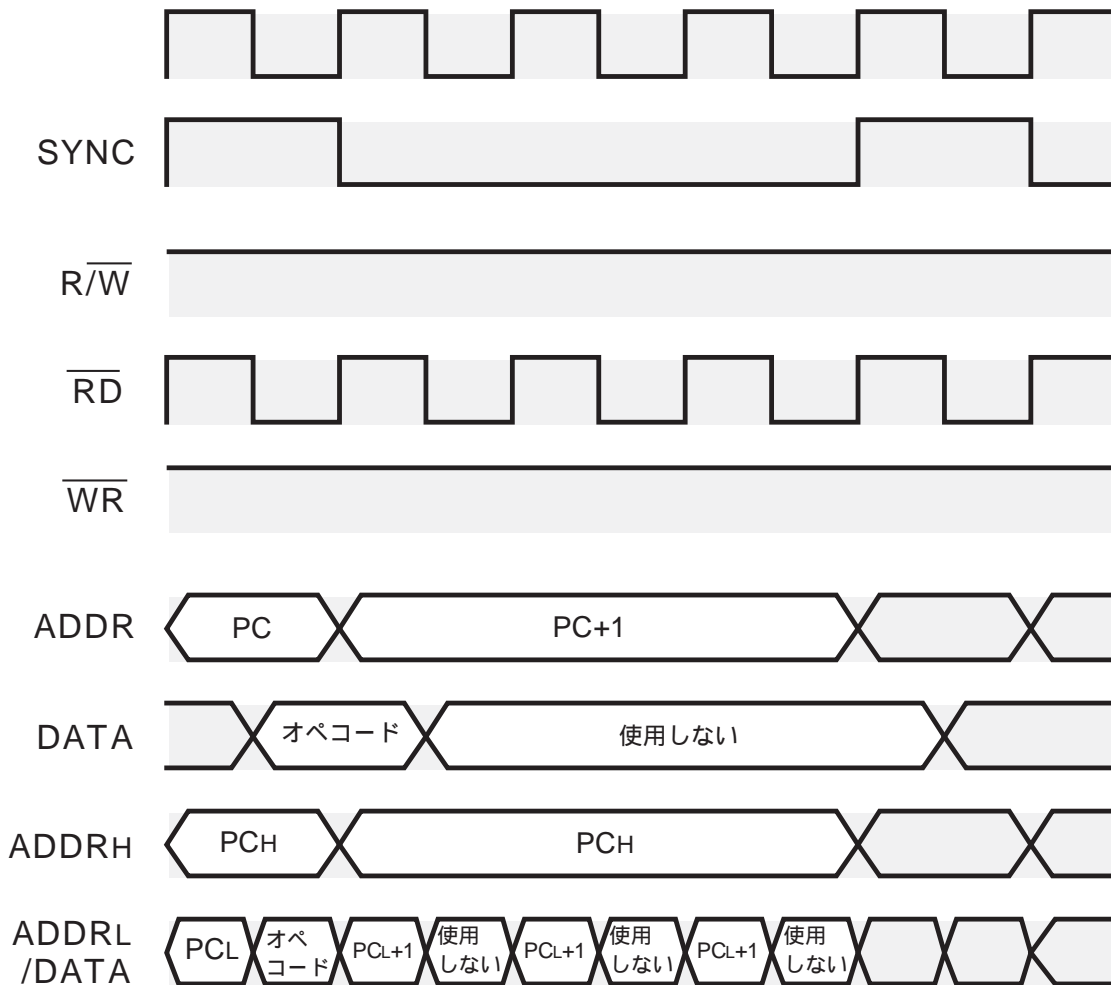
該当命令 : BBC i, A, \$ hhll
 BBS i, A, \$ hhll

必要バイト数 : 2

(1) ブランチしない場合

必要サイクル数 : 4

タイミング :



アキュムレータ ビット 相対(Accumulator bit relative)

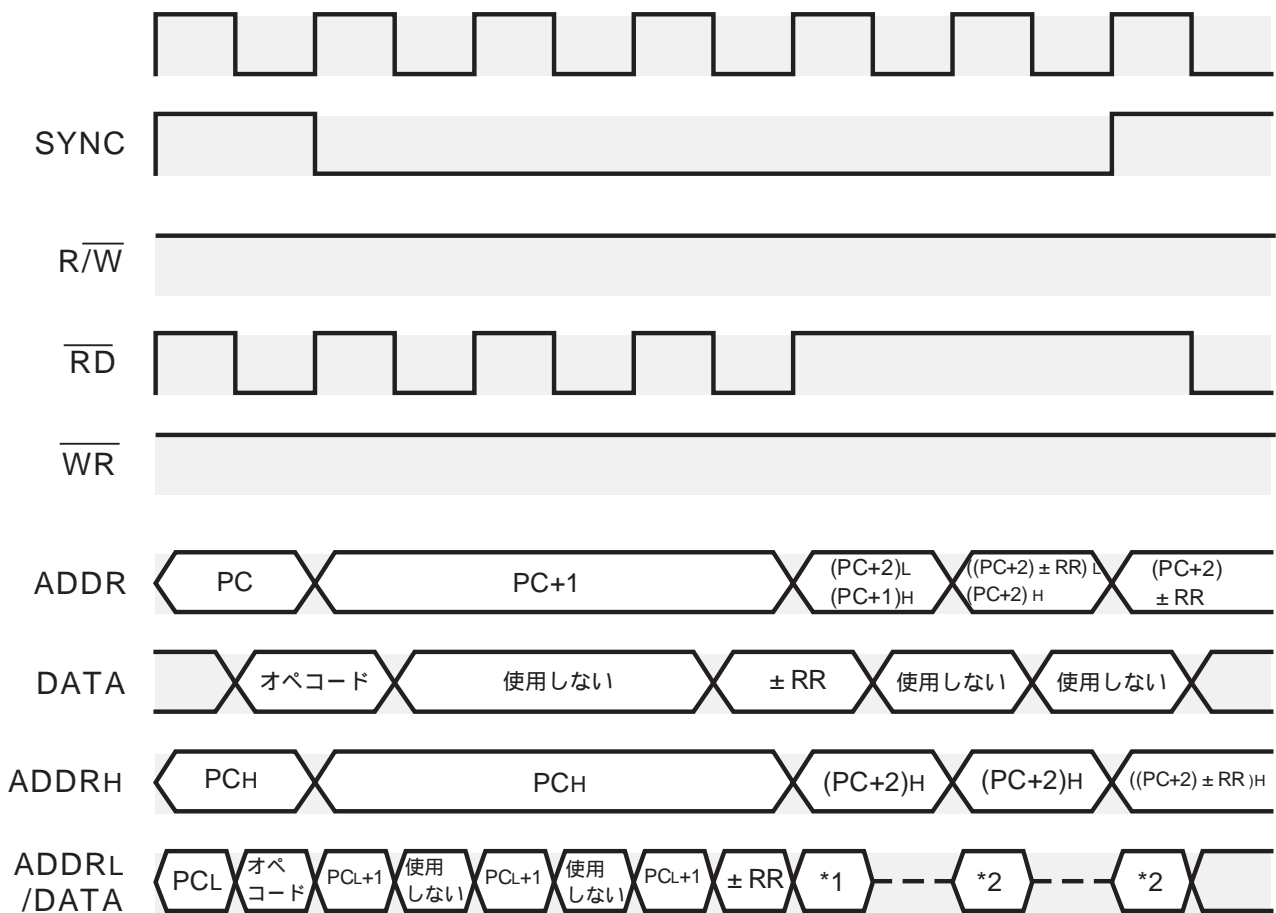
該当命令 : BBC i, A, \$hhll
 BBS i, A, \$hhll

必要バイト数 : 2

(2) ブランチする場合

必要サイクル数 : 6

タイミング :



RR:オフセットの値

*1 : (PC+1)_L

*2 : ((PC+2) ± RR)_L

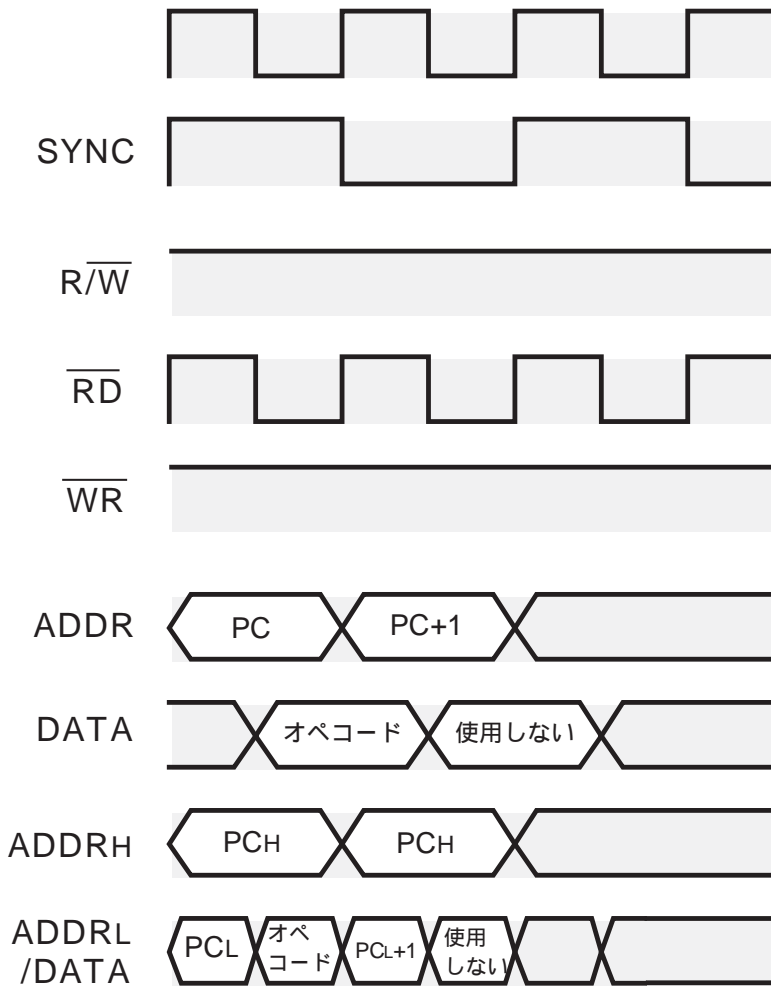
アキュムレータ ビット(Accumulator bit)

該当命令 : CLB i, A
 SEB i, A

必要バイト数 : 1

必要サイクル数 : 2

タイミング :



ビット 相対(Bit relative)

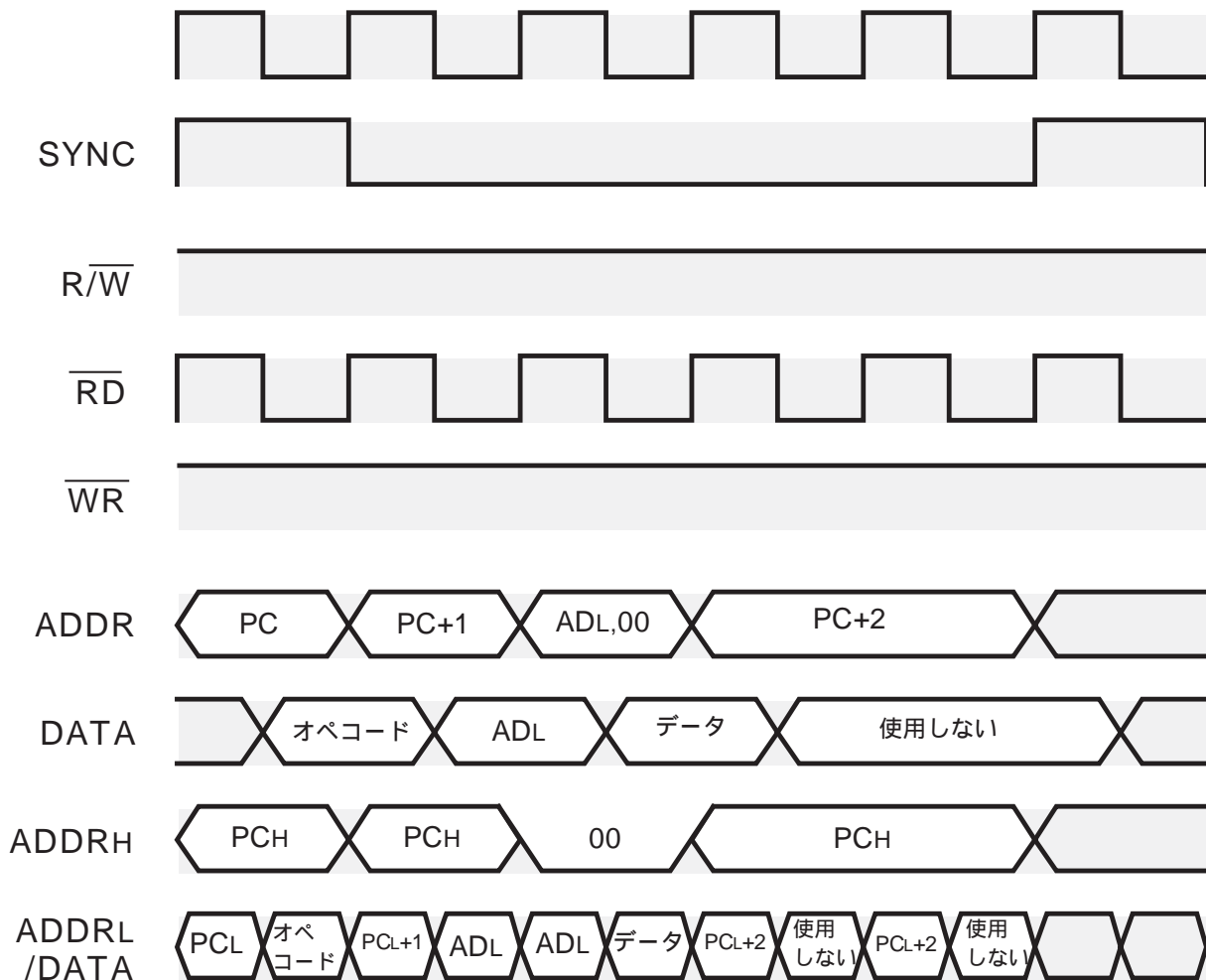
該当命令 : BBC i, \$ zz, \$ hhll
 BBS i, \$ zz, \$ hhll

必要バイト数 : 3

(1) ブランチしない場合

必要サイクル数 : 5

タイミング :



ビット 相対(Bit relative)

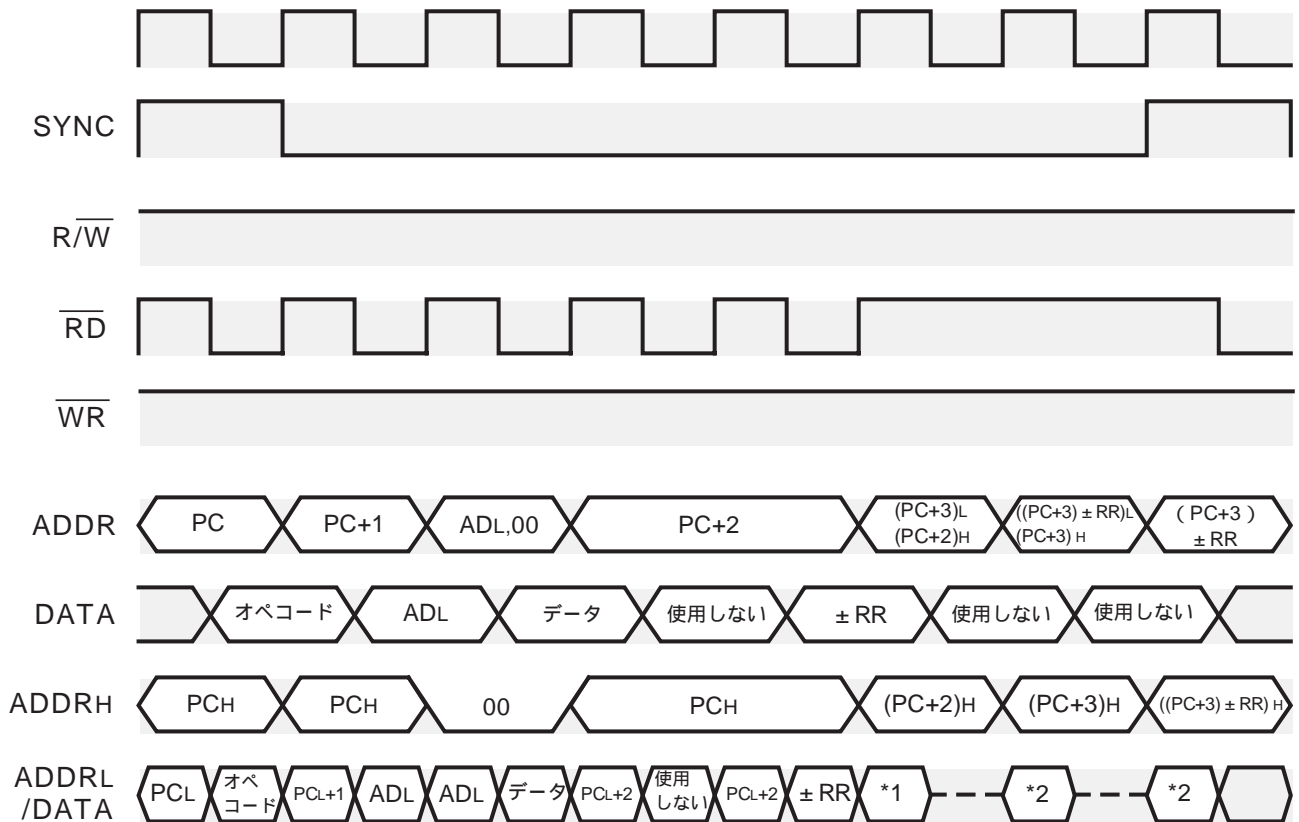
該当命令 : BBC i, \$ zz, \$ hhll
 BBS i, \$ zz, \$ hhll

必要バイト数 : 3

(2) ブランチする場合

必要サイクル数 : 7

タイミング :



RR:オフセットの値

*1 : (PC+3)_L

*2 : ((PC+3) ± RR)_L

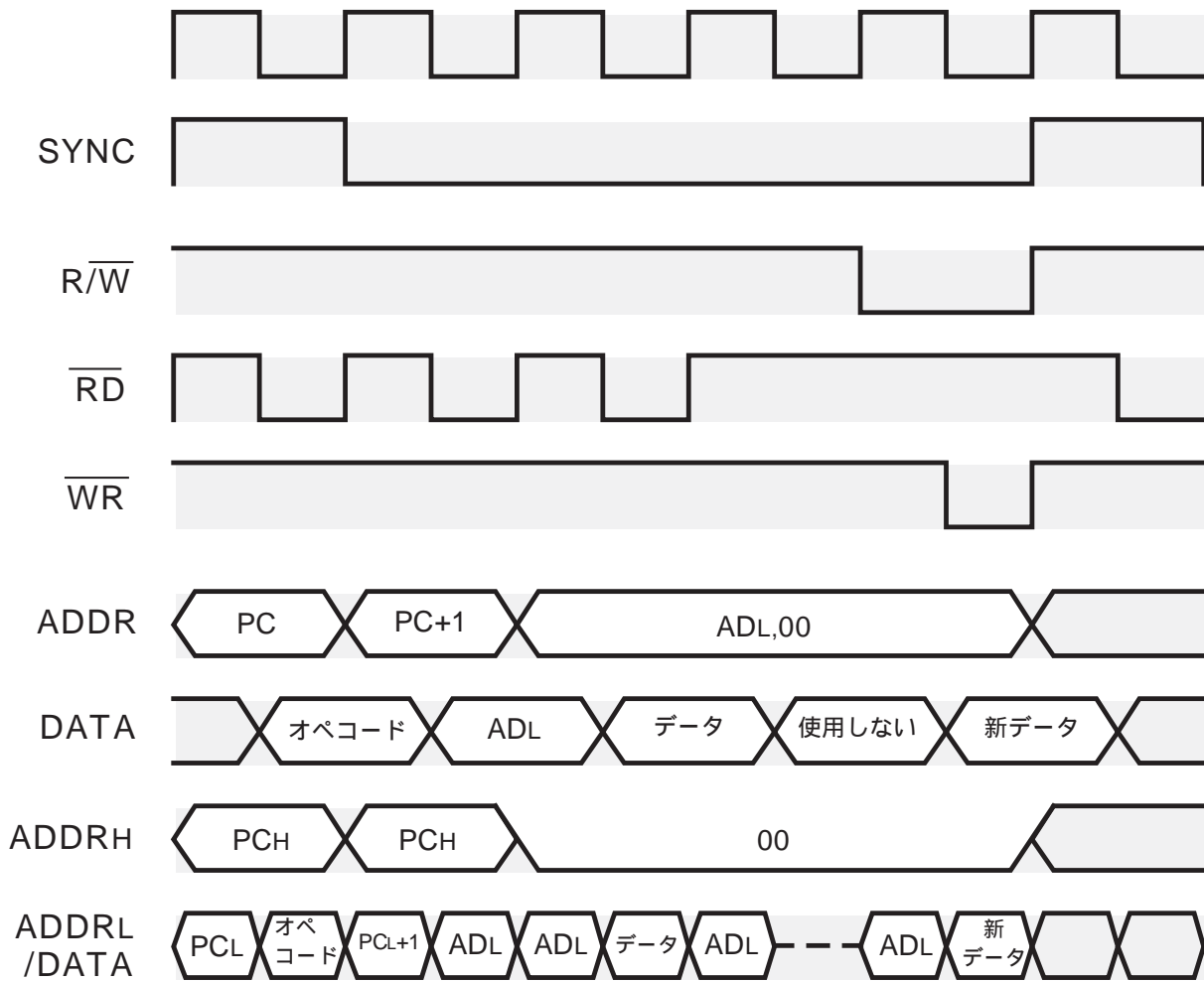
ゼロページ ビット(Zero page bit)

該当命令 : CLB i, \$zz
 SEB i, \$zz

必要バイト数 : 2

必要サイクル数 : 5

タイミング :

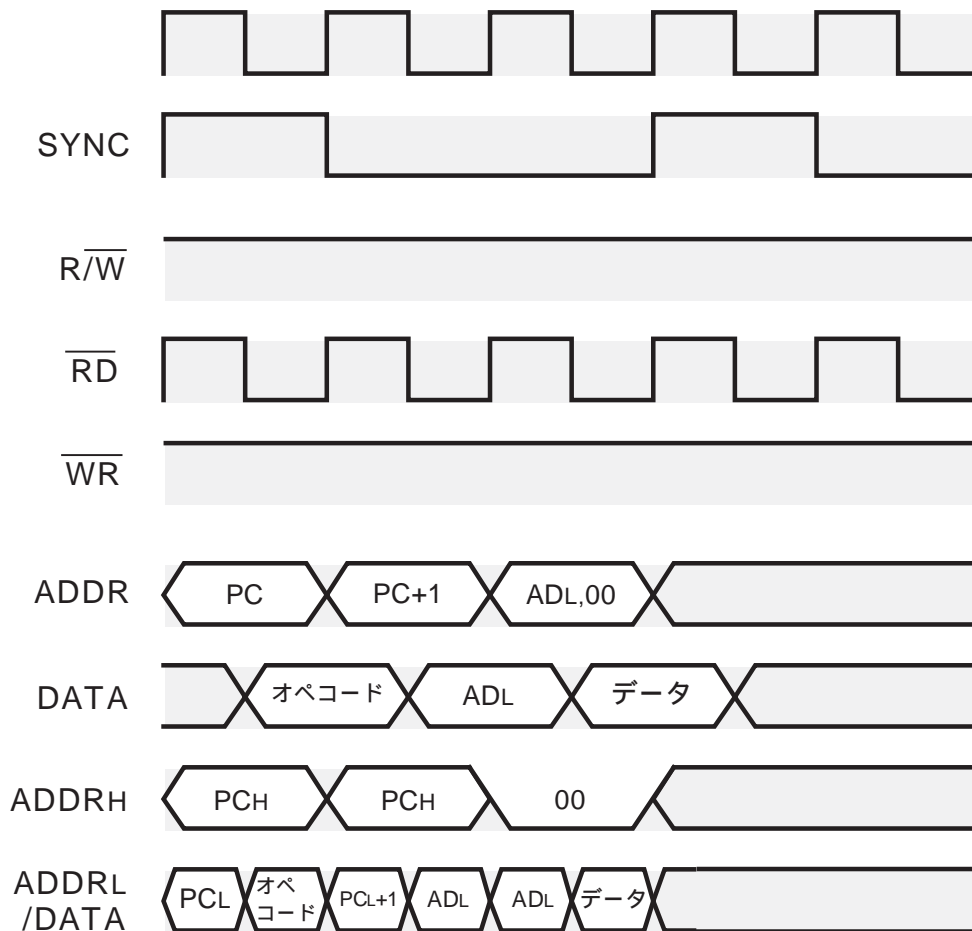


該当命令 : ADC \$zz (T = 0)
 AND \$zz (T = 0)
 BIT \$zz
 CMP \$zz (T = 0)
 CPX \$zz
 CPY \$zz
 EOR \$zz (T = 0)
 LDA \$zz (T = 0)
 LDX \$zz
 LDY \$zz
 ORA \$zz (T = 0)
 SBC \$zz (T = 0)
 TST \$zz

必要バイト数 : 2

必要サイクル数 : 3

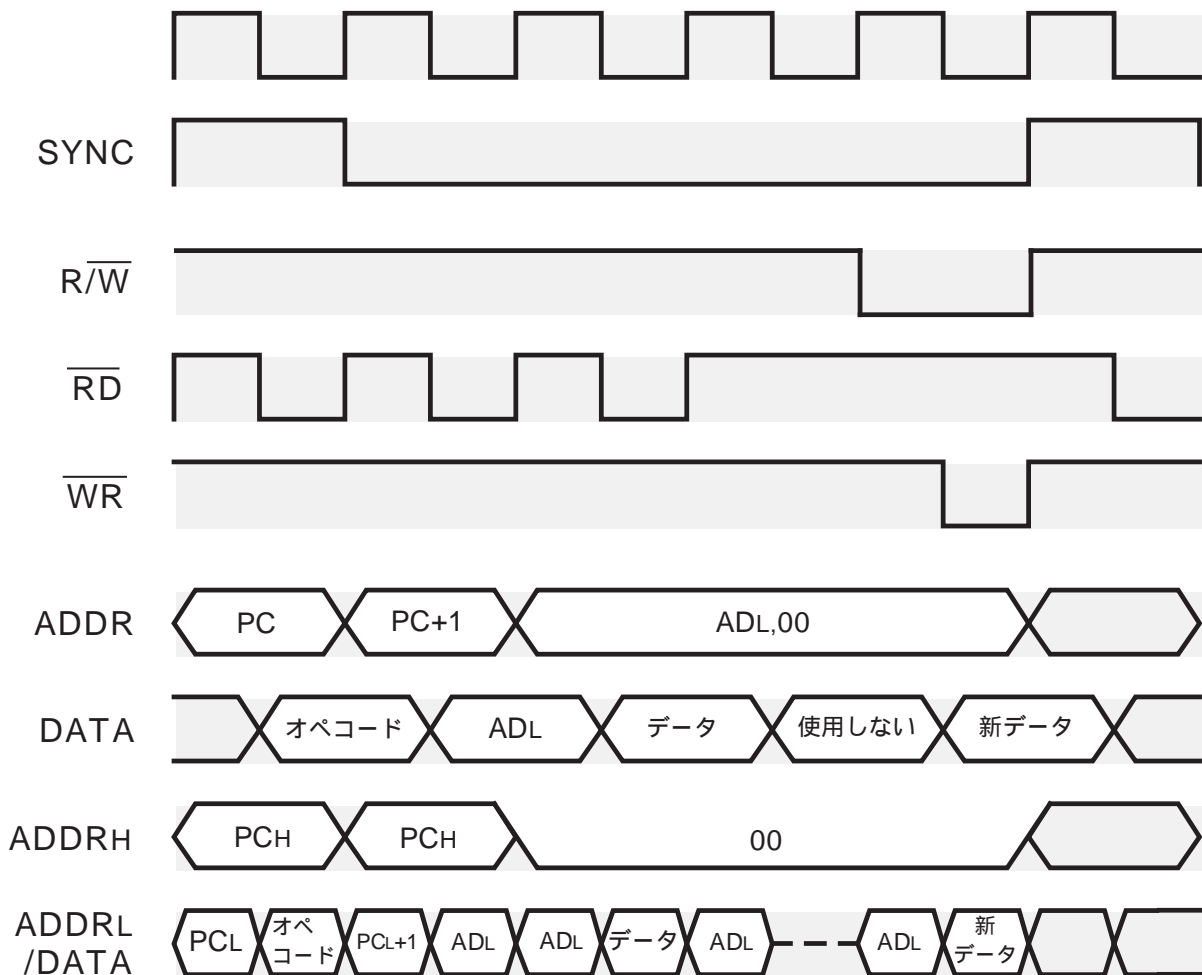
タイミング :



ゼロページ (Zero page)

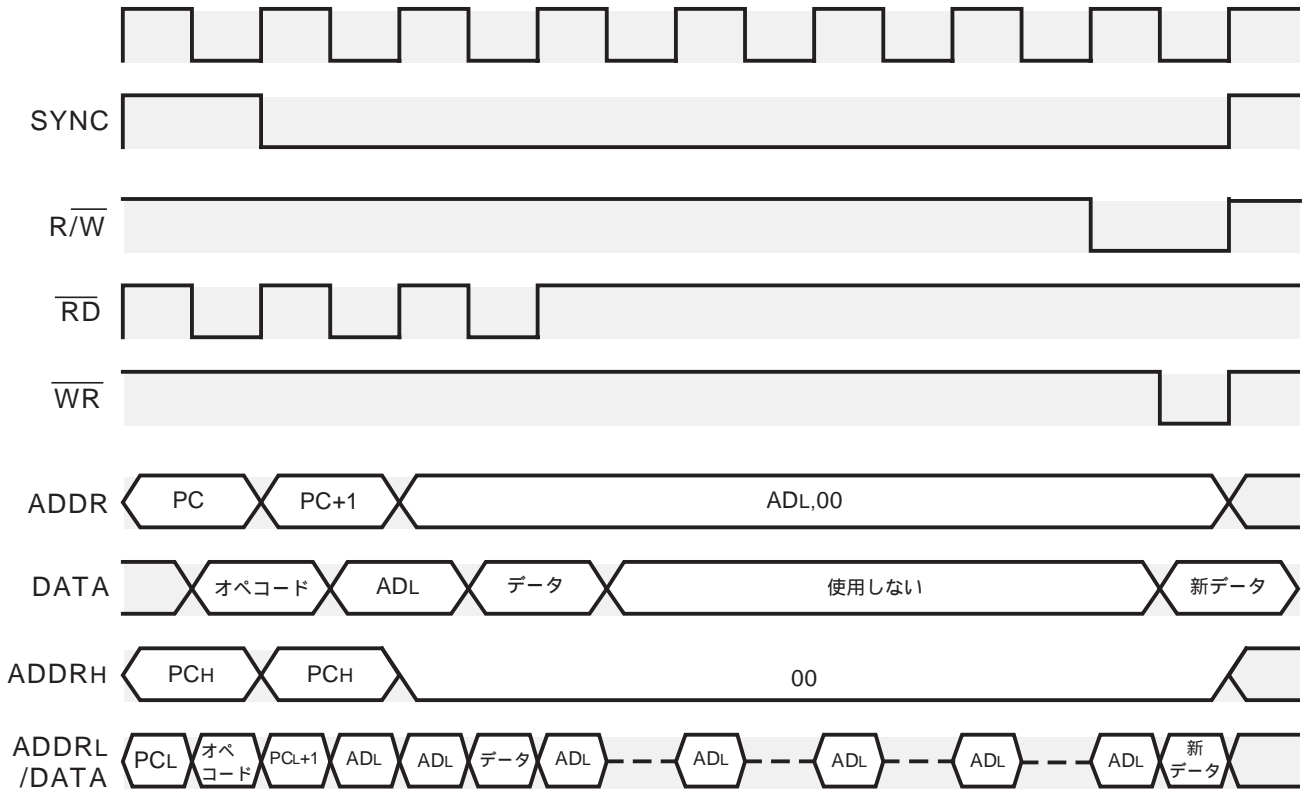
該当命令 : ASL \$ZZ
 : COM \$ZZ
 : DEC \$ZZ
 : INC \$ZZ
 : LSR \$ZZ
 : ROL \$ZZ
 : ROR \$ZZ

必要バイト数 : 2
 必要サイクル数 : 5
 タイミング :



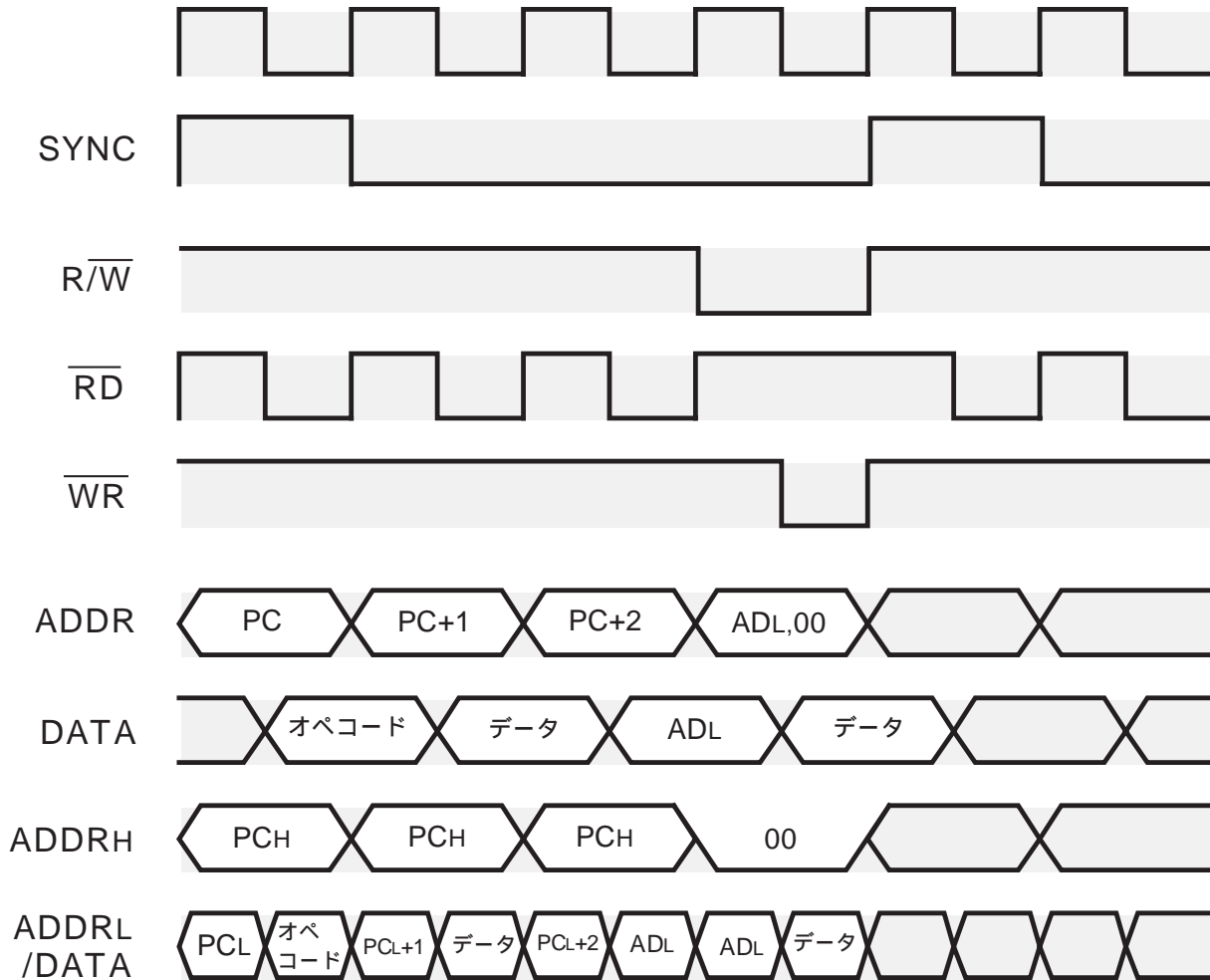
ゼロページ (Zero page)

該当命令 : RRF \$zz
 必要バイト数 : 2
 必要サイクル数 : 8
 タイミング :



ゼロページ (Zero page)

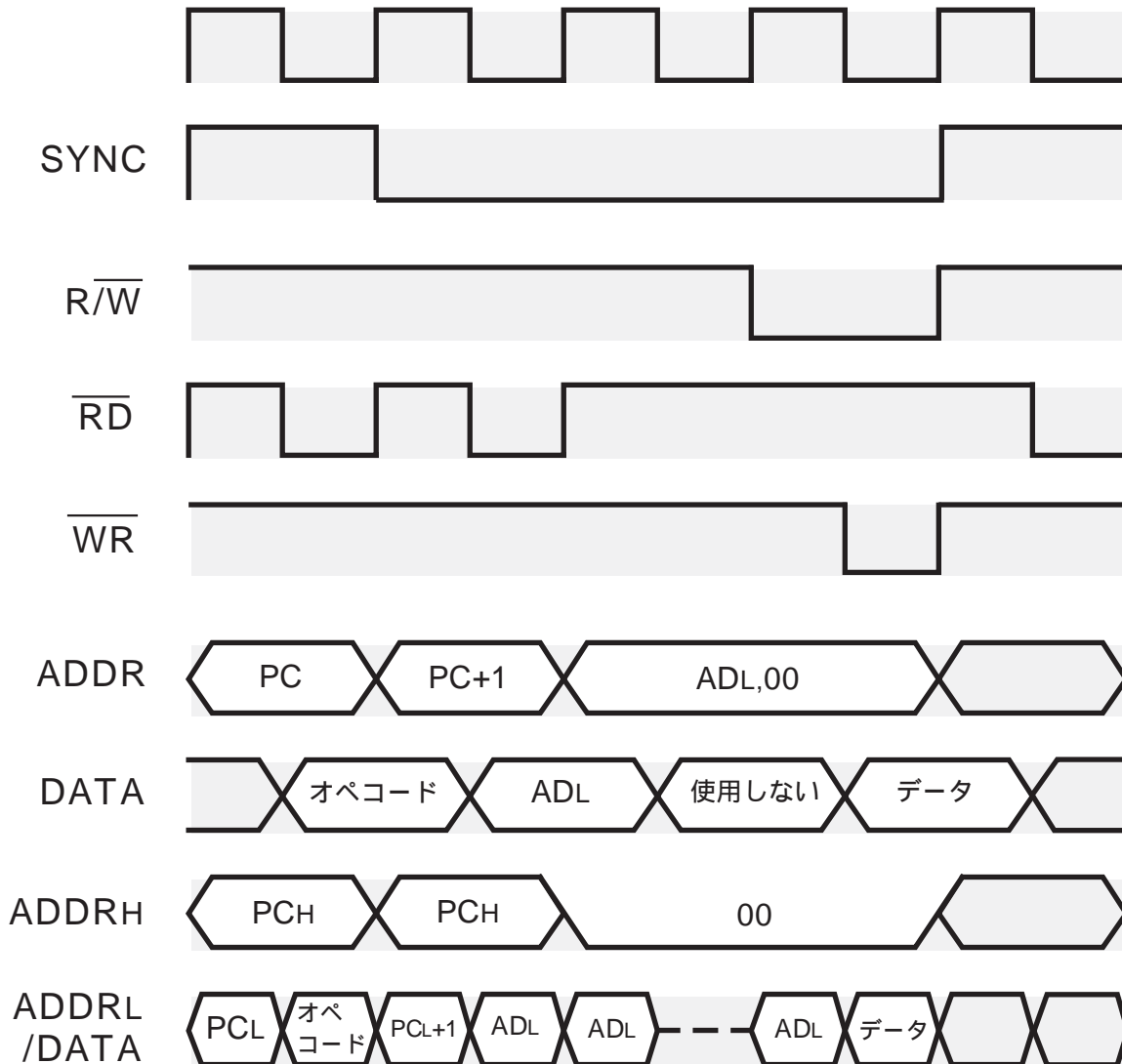
該当命令 : LDM # \$ nn, \$ zz
 必要バイト数 : 3
 必要サイクル数 : 4
 タイミング :



ゼロページ (Zero page)

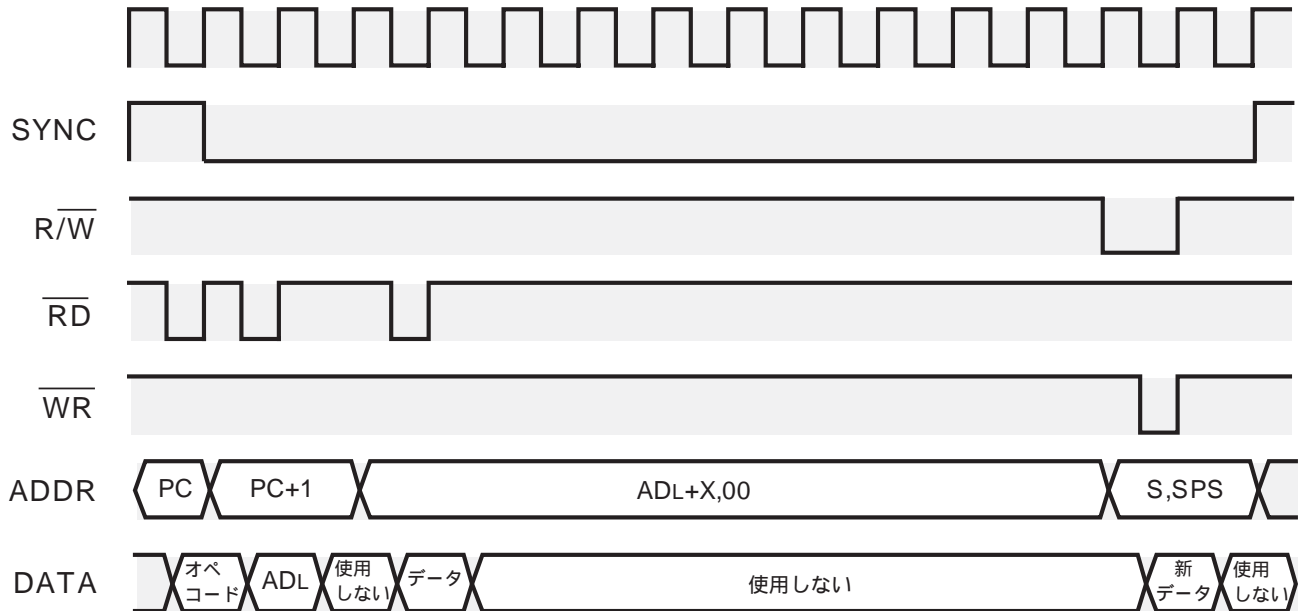
該当命令 : STA \$ZZ
 : STX \$ZZ
 : STY \$ZZ

必要バイト数 : 2
 必要サイクル数 : 4
 タイミング :



ゼロページ X(Zero page X)

該当命令 : MUL \$zz, X
 必要バイト数 : 2
 必要サイクル数 : 15
 タイミング :

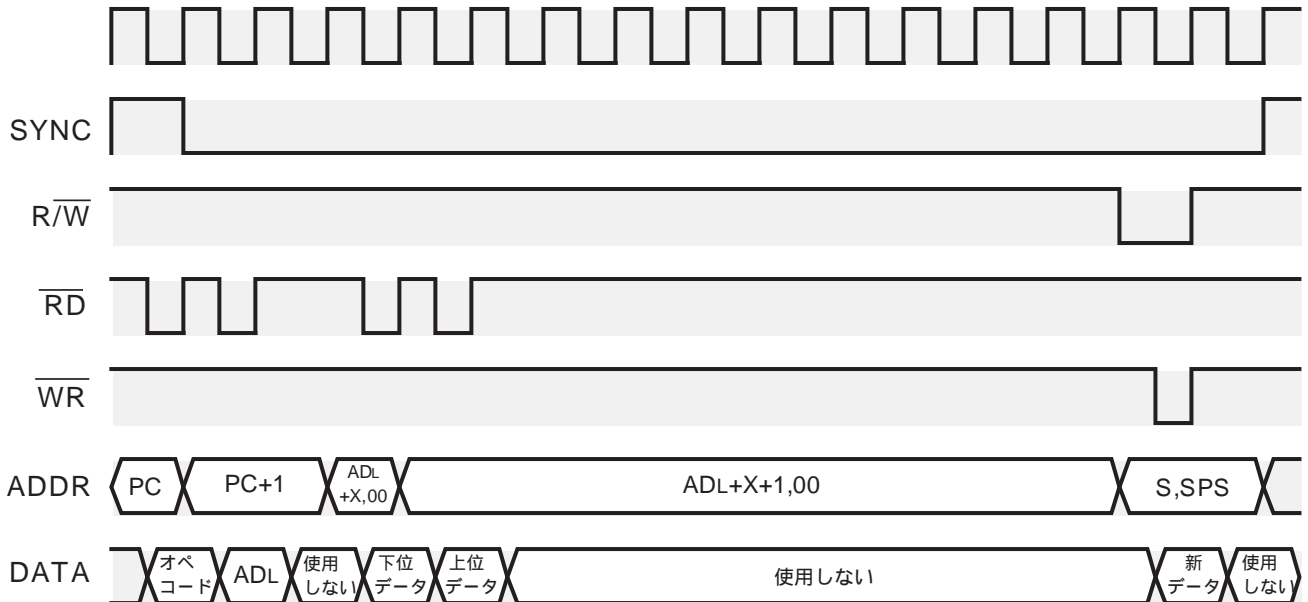


SPS:CPU モードレジスタのスタックページ選択ビットで選択されたページ

注. 一部の品種では、この命令は使用できません。

ゼロページ X(Zero page X)

該当命令 : DIV \$zz, X
 必要バイト数 : 2
 必要サイクル数 : 16
 タイミング :



SPS:CPU モードレジスタのスタックページ選択ビットで選択されたページ

注. 一部の品種では、この命令は使用できません。

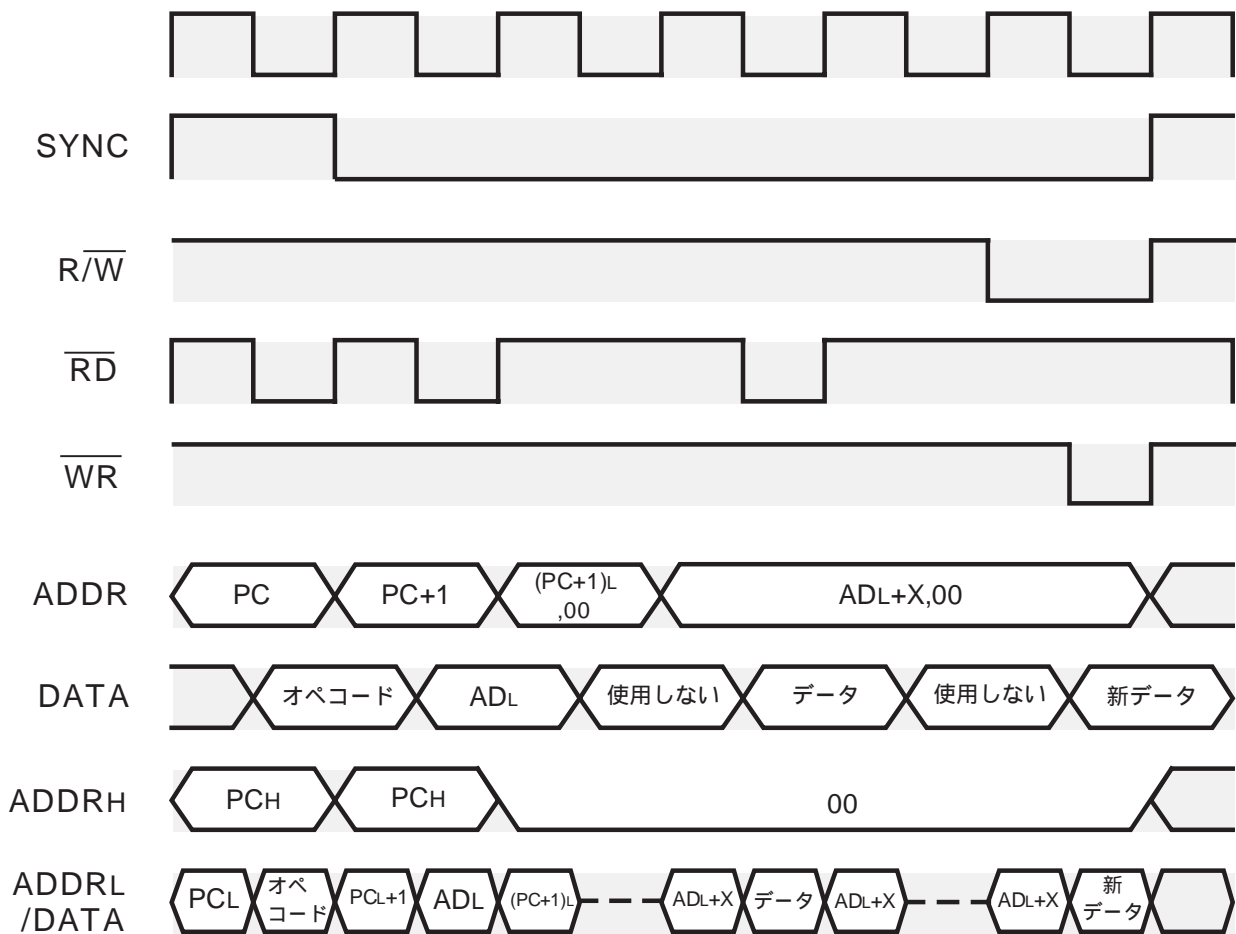
ゼロページ X(Zero page X)

該当命令 : ASL \$ZZ, X
 : DEC \$ZZ, X
 : INC \$ZZ, X
 : LSR \$ZZ, X
 : ROL \$ZZ, X
 : ROR \$ZZ, X

必要バイト数 : 2

必要サイクル数 : 6

タイミング :



ゼロページ X(Zero page X)

[T=0]

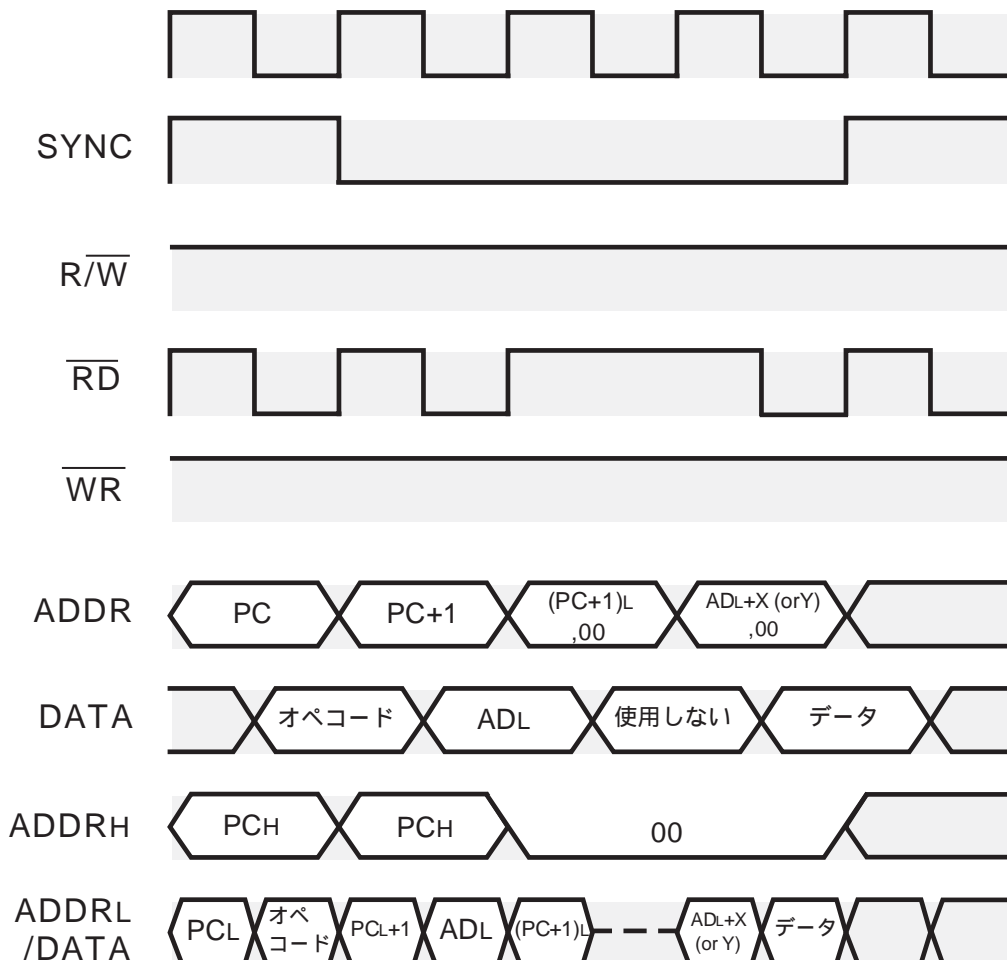
ゼロページ Y(Zero page Y)

該当命令 : ADC \$zz, X (T = 0)
AND \$zz, X (T = 0)
CMP \$zz, X (T = 0)
EOR \$zz, X (T = 0)
LDA \$zz, X (T = 0)
LDX \$zz, Y
LDY \$zz, X
ORA \$zz, X (T = 0)
SBC \$zz, X (T = 0)

必要バイト数 : 2

必要サイクル数 : 4

タイミング :



ゼロページ X(Zero page X)

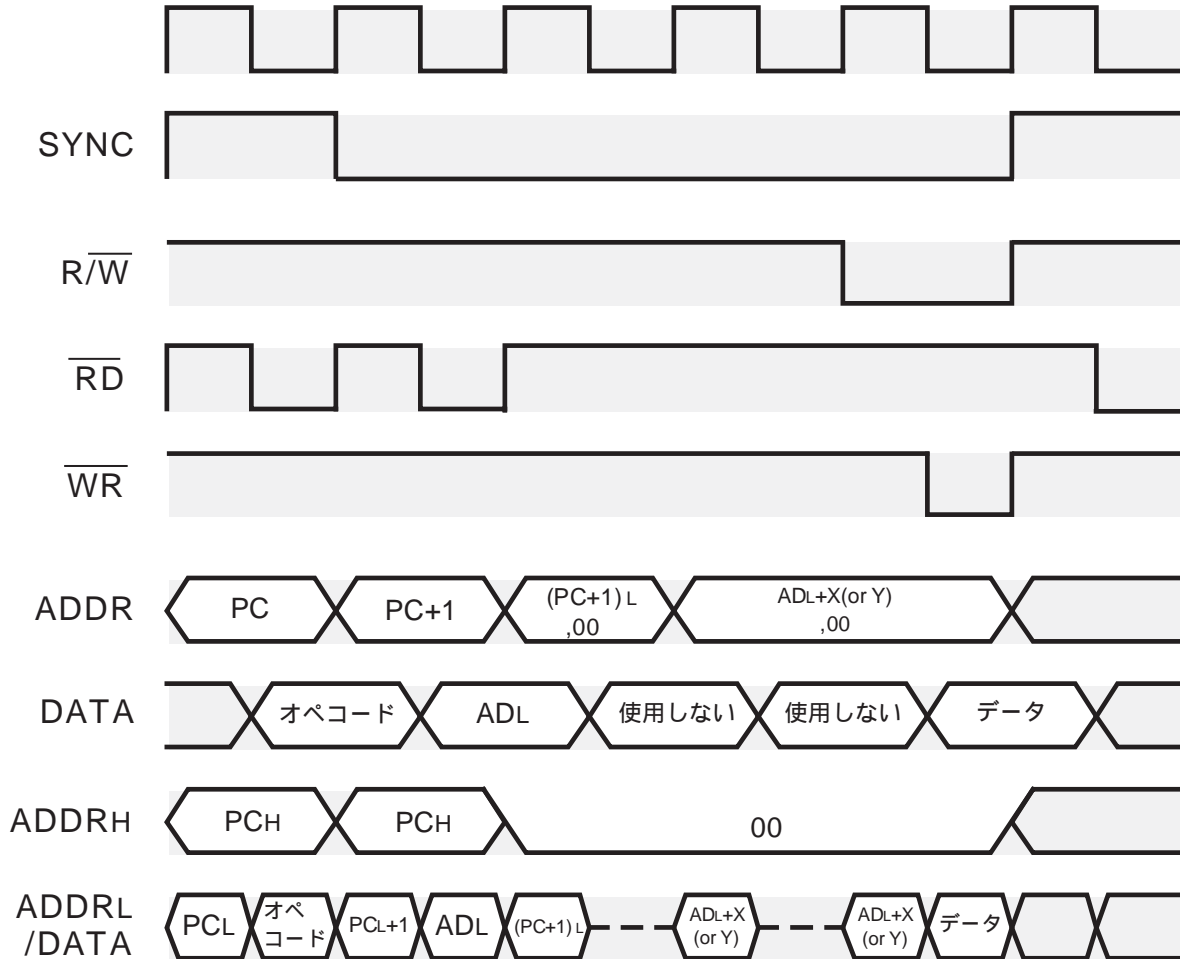
ゼロページ Y(Zero page Y)

該当命令 : STA \$zz, X
STX \$zz, Y
STY \$zz, X

必要バイト数 : 2

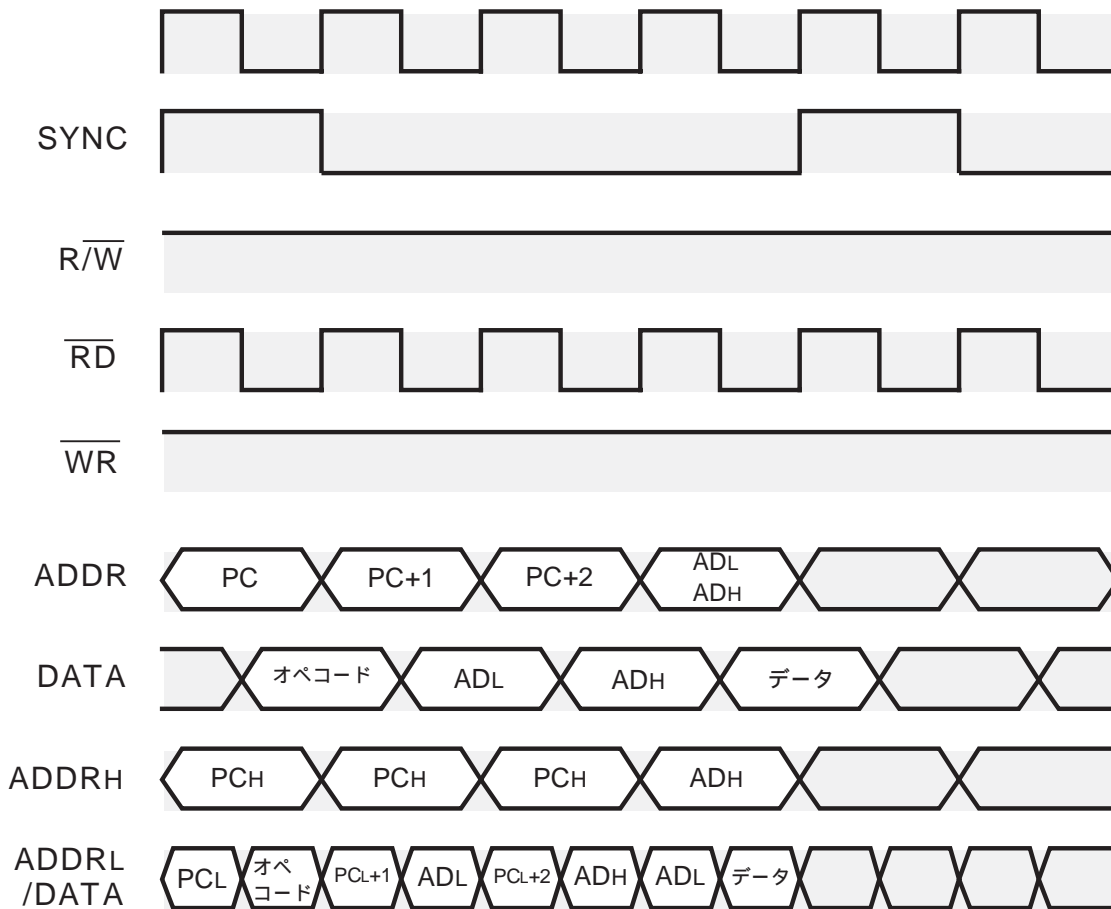
必要サイクル数 : 5

タイミング :



該当命令 : ADC \$hhll (T = 0)
 AND \$hhll (T = 0)
 BIT \$hhll
 CMP \$hhll (T = 0)
 CPX \$hhll
 CPY \$hhll
 EOR \$hhll (T = 0)
 LDA \$hhll (T = 0)
 LDX \$hhll
 LDY \$hhll
 ORA \$hhll (T = 0)
 SBC \$hhll (T = 0)

必要バイト数 : 3
 必要サイクル数 : 4
 タイミング :



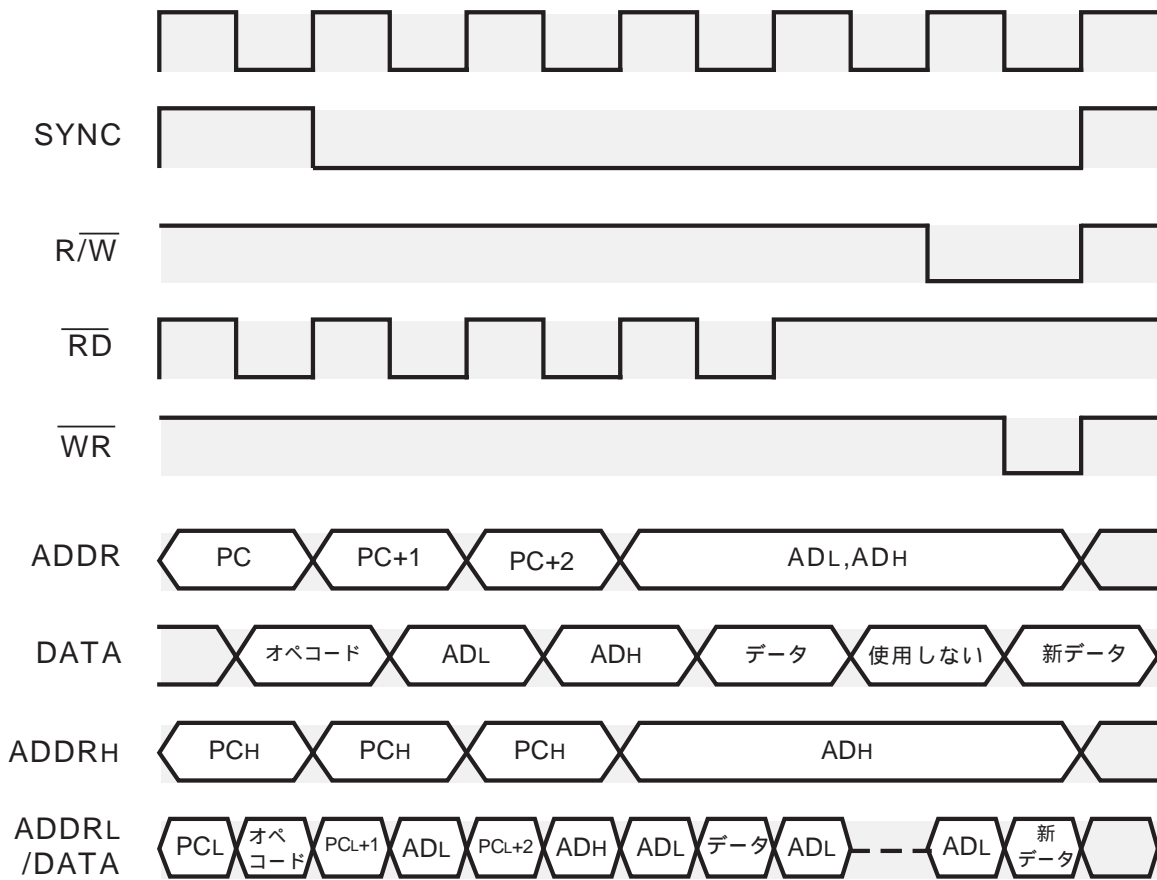
絶対(Absolute)

該当命令 : ASL \$ hhl
 DEC \$ hhl
 INC \$ hhl
 LSR \$ hhl
 ROL \$ hhl
 ROR \$ hhl

必要バイト数 : 3

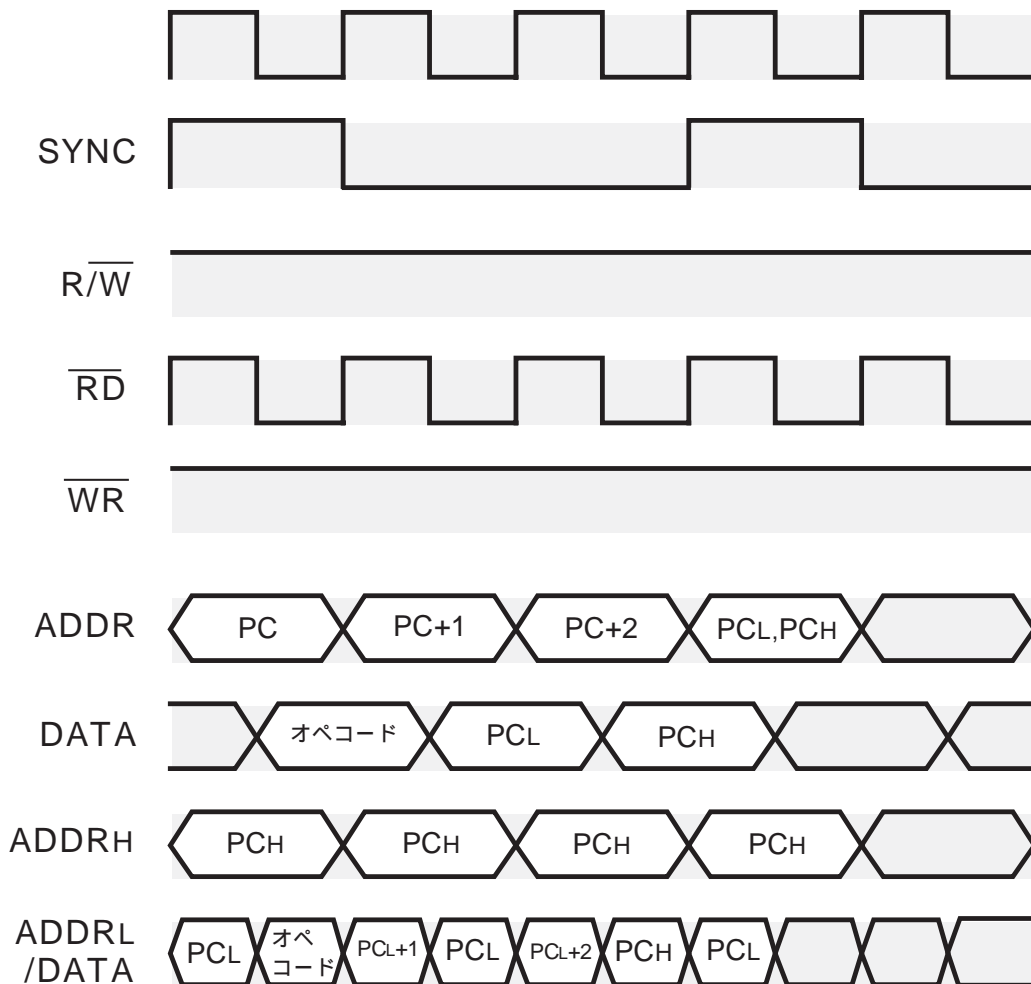
必要サイクル数 : 6

タイミング :



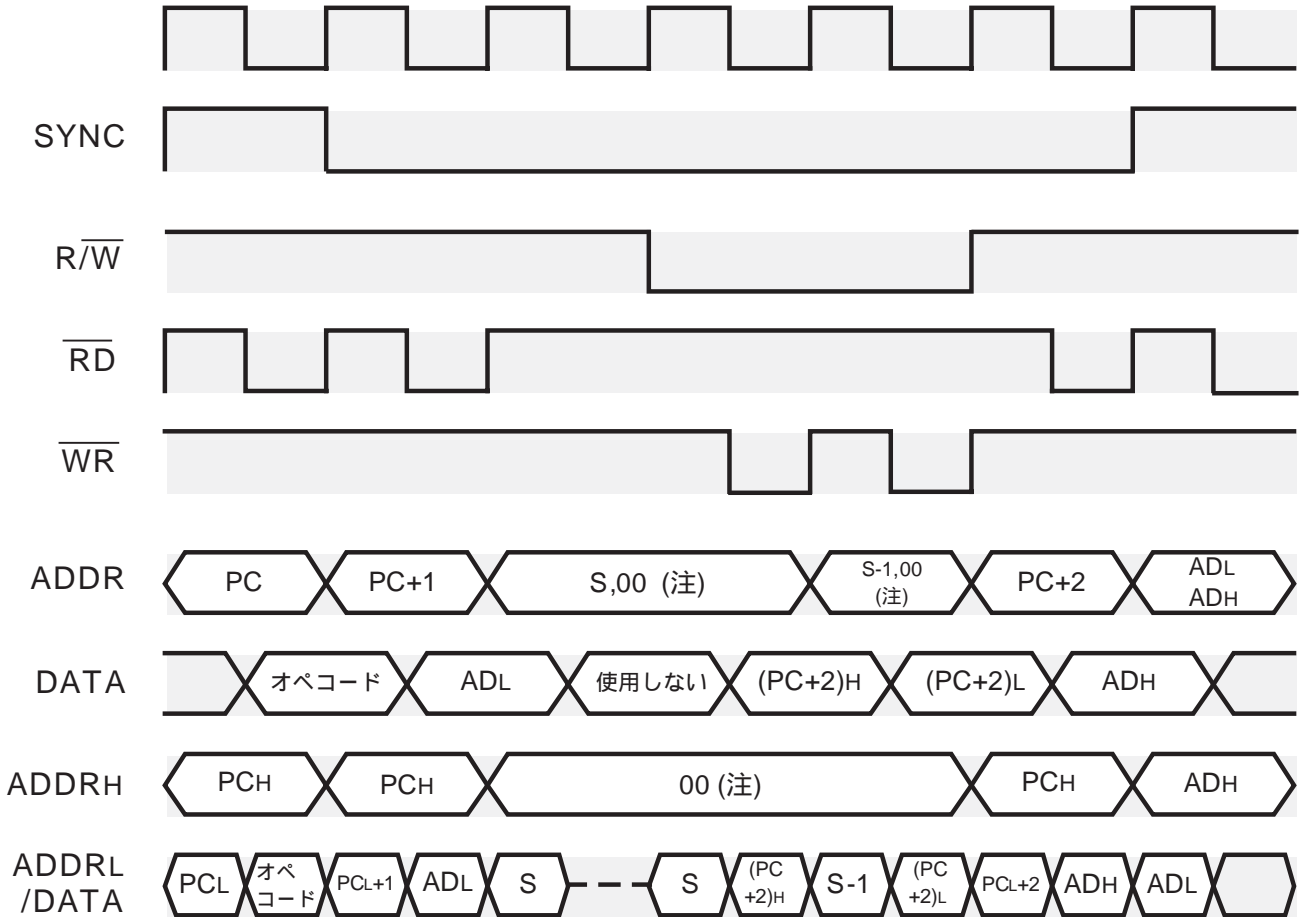
絶対(Absolute)

該当命令 : JMP \$hhll
 必要バイト数 : 3
 必要サイクル数 : 3
 タイミング :



絶対 (Absolute)

該当命令 : JSR \$hhll
 必要バイト数 : 3
 必要サイクル数 : 6
 タイミング :



注. 品種によって、“01”又はSPSフラグの内容となります。

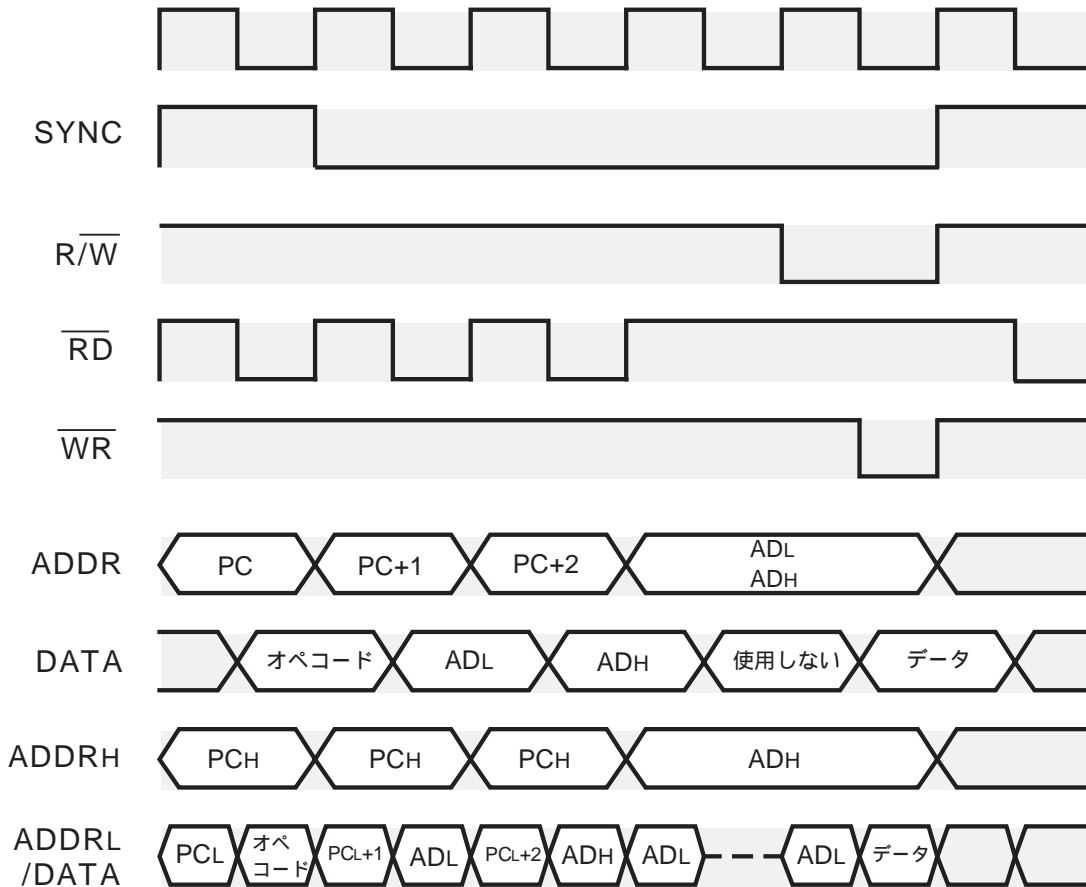
絶対(Absolute)

該当命令 : STA \$hhll
 : STX \$hhll
 : STY \$hhll

必要バイト数 : 3

必要サイクル数 : 5

タイミング :

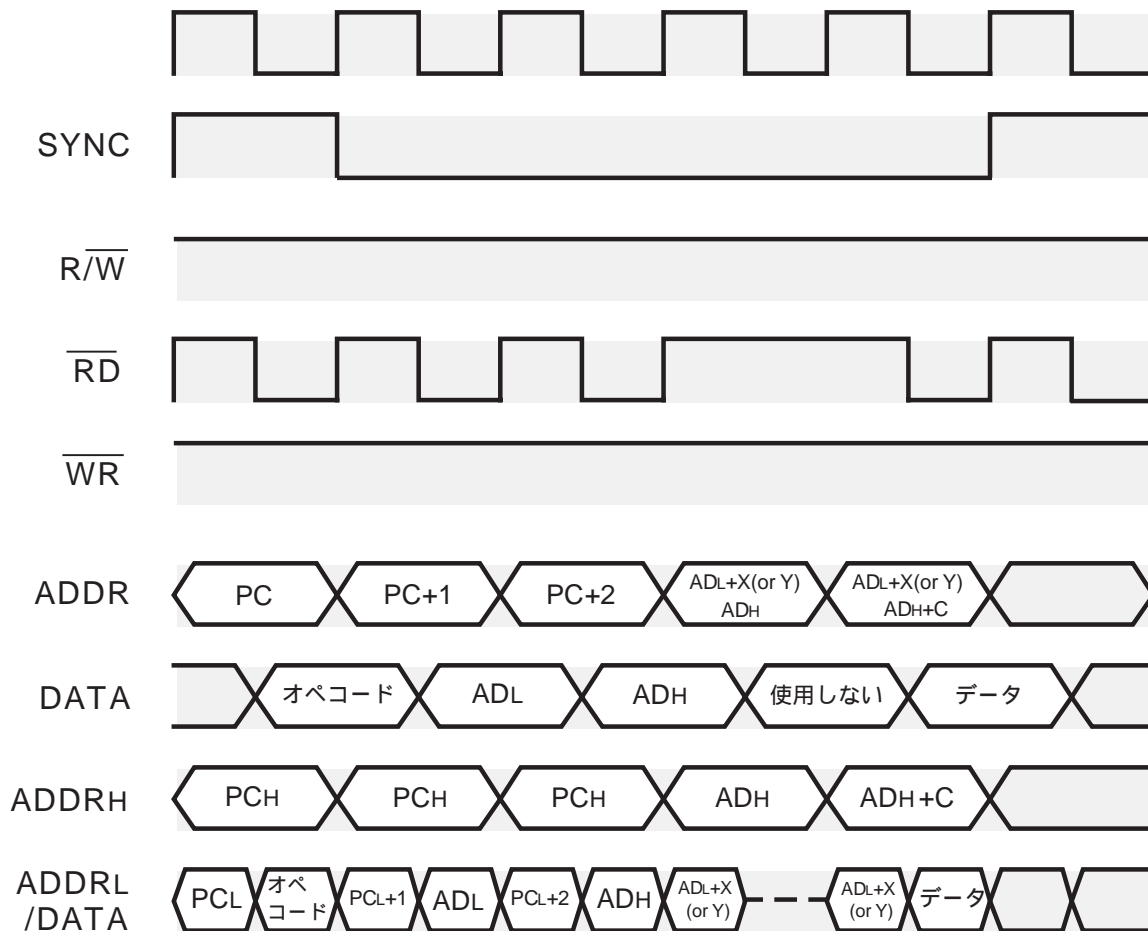


[T=0]

絶対 X(Absolute)
絶対 Y(Absolute)

該当命令 : ADC \$hhll, X 又は Y (T = 0)
AND \$hhll, X 又は Y (T = 0)
CMP \$hhll, X 又は Y (T = 0)
EOR \$hhll, X 又は Y (T = 0)
LDA \$hhll, X 又は Y (T = 0)
LDX \$hhll, Y
LDY \$hhll, X
ORA \$hhll, X 又は Y (T = 0)
SBC \$hhll, X 又は Y (T = 0)

必要バイト数 : 3
必要サイクル数 : 5
タイミング :



C:ADL+X(又は Y)のキャリ

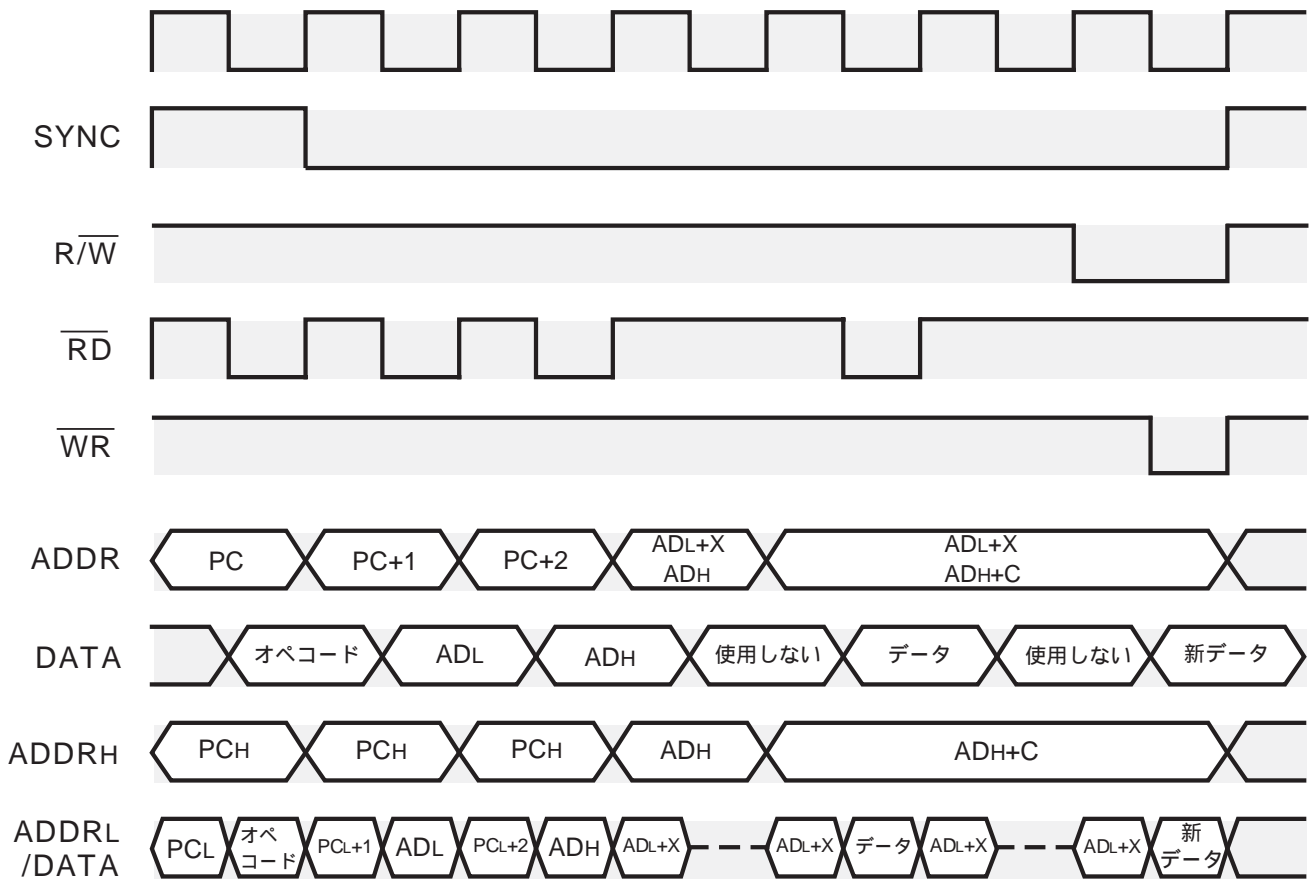
絶対 X(Absolute)

該当命令 : ASL \$hhll, X
 DEC \$hhll, X
 INC \$hhll, X
 LSR \$hhll, X
 ROL \$hhll, X
 ROR \$hhll, X

必要バイト数 : 3

必要サイクル数 : 7

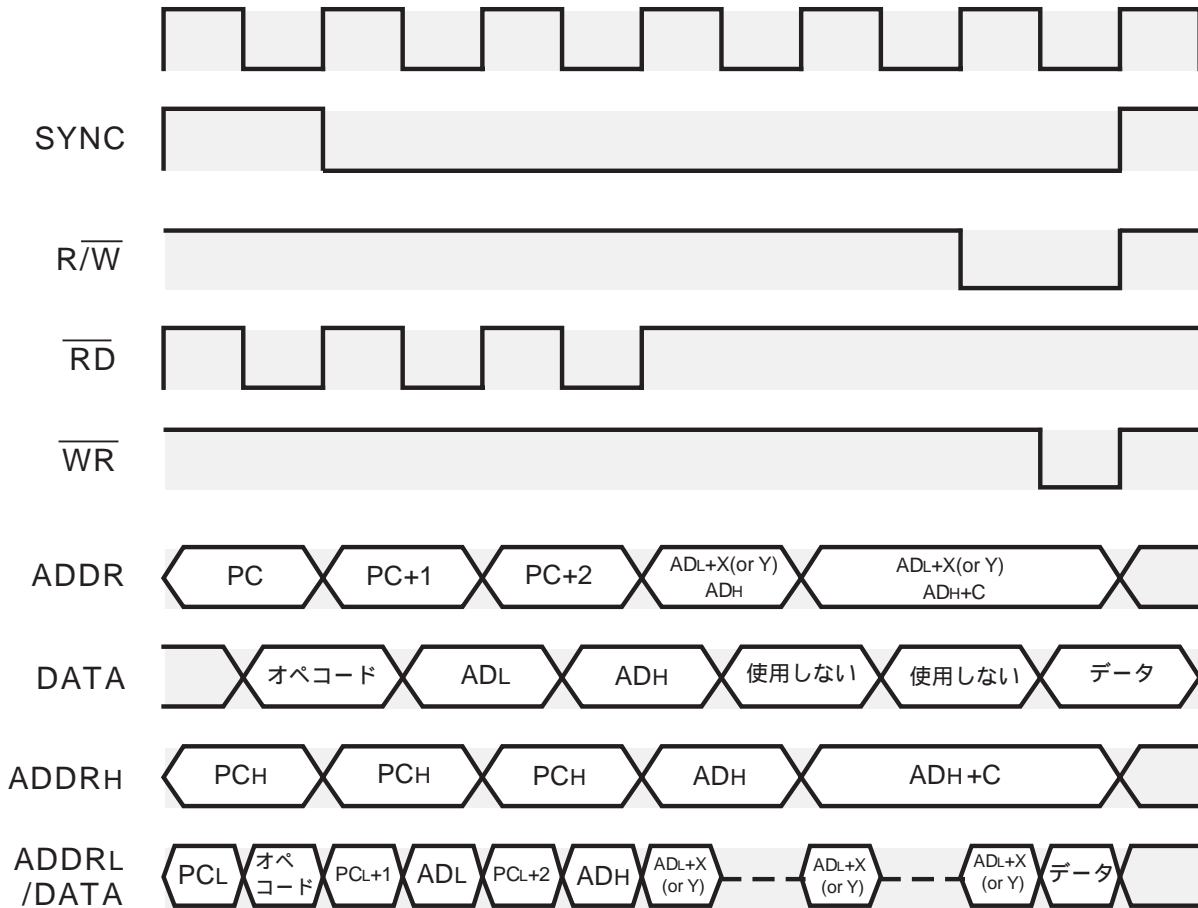
タイミング :



C:ADL+X のキャリ

絶対 X(Absolute)
絶対 Y(Absolute)

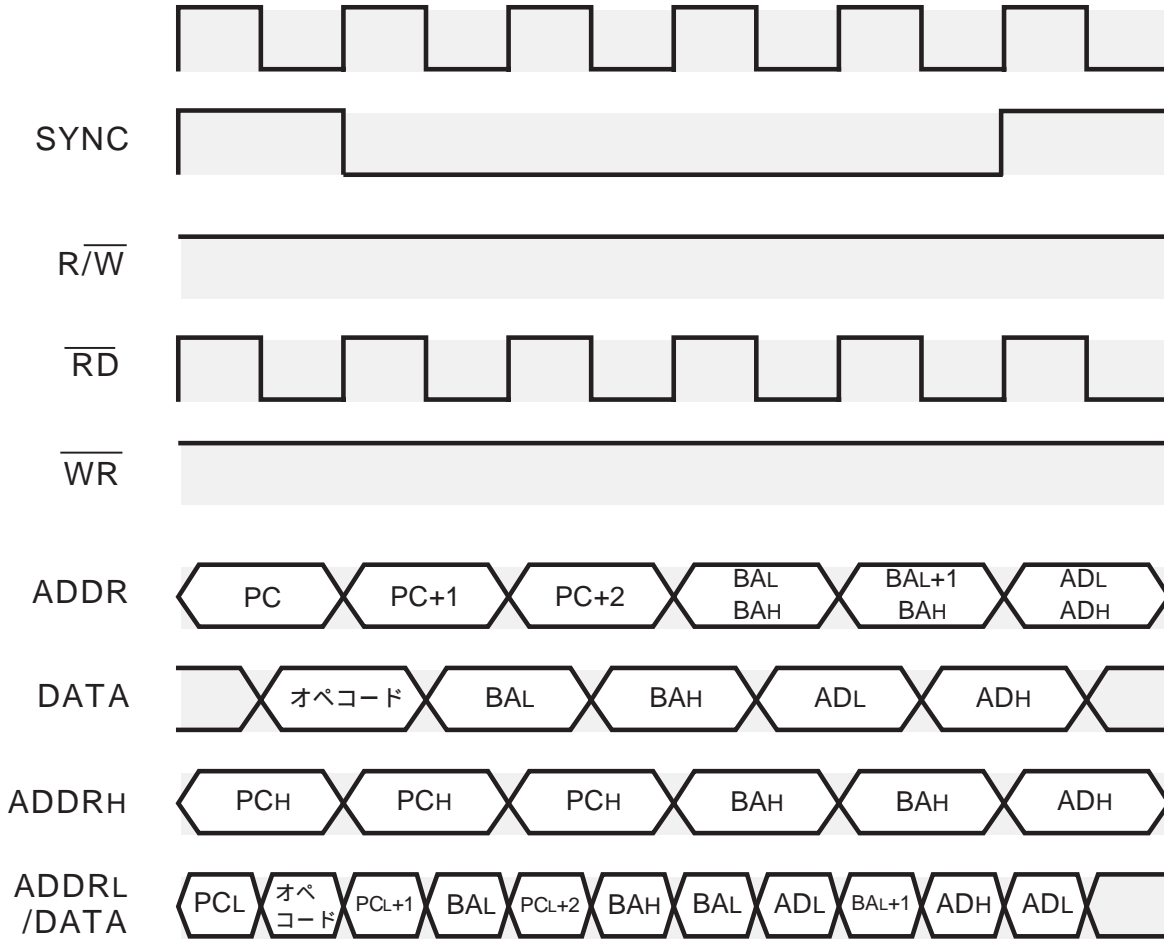
該当命令 : STA \$hhll, X 又は Y
 必要バイト数 : 3
 必要サイクル数 : 6
 タイミング :



C:ADL+X(又は Y)のキャリ

間接 (Indirect)

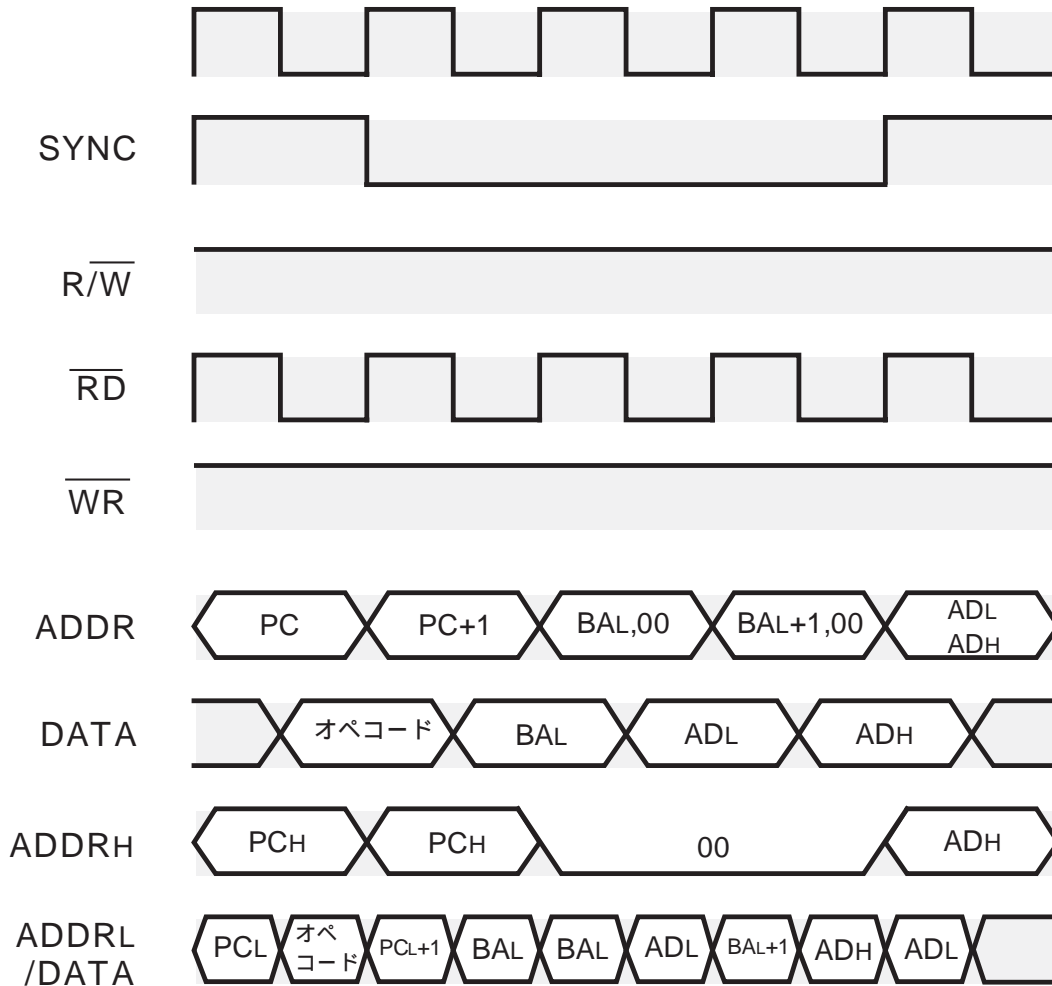
該当命令 : JMP (\$ hhl)
 必要バイト数 : 3
 必要サイクル数 : 5
 タイミング :



BA:ベーシックアドレス

ゼロページ 間接(Zero page Indirect)

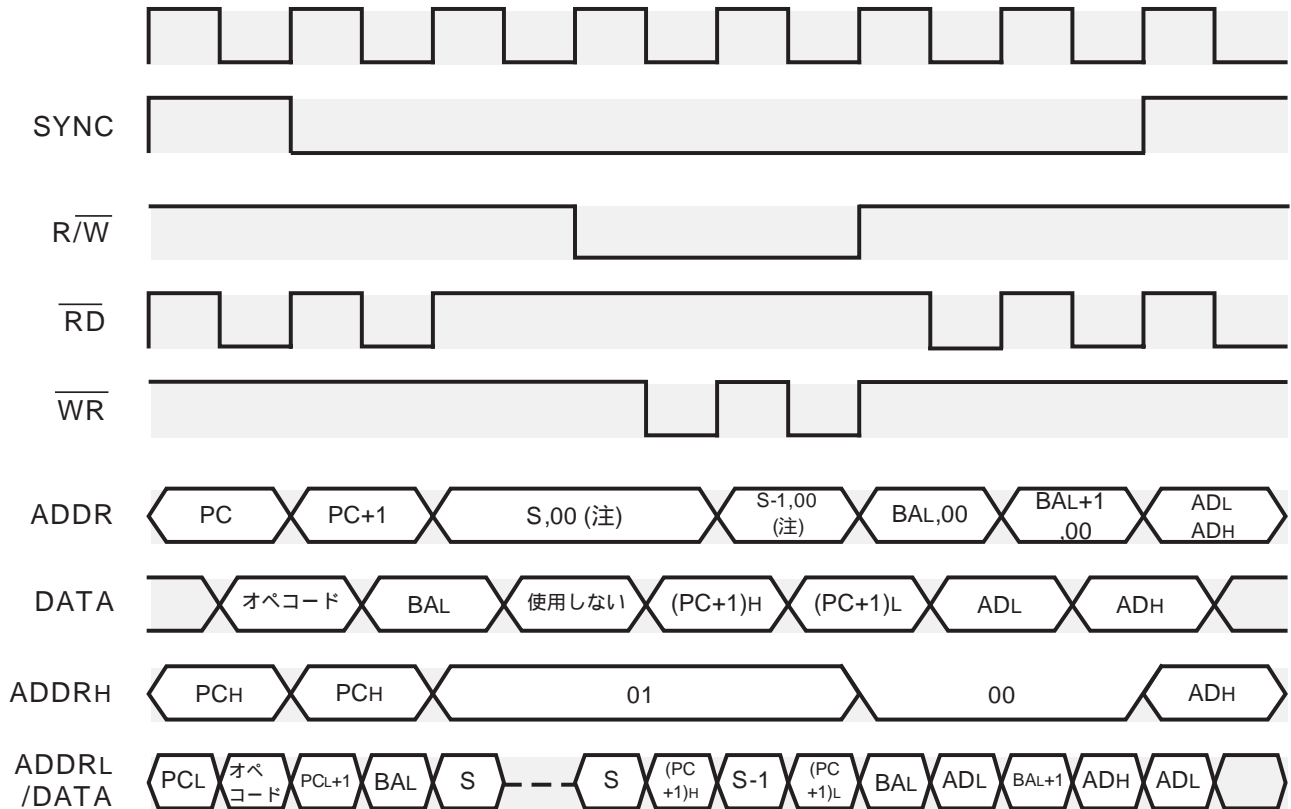
該当命令 : JMP (\$ zz)
 必要バイト数 : 2
 必要サイクル数 : 4
 タイミング :



BA:ベーシックアドレス

ゼロページ 間接(Zero page Indirect)

該当命令 : JSR (\$ zz)
 必要バイト数 : 2
 必要サイクル数 : 7
 タイミング :



BA:ベーシックアドレス

注. 品種によって、“01” 又は SPS フラグの内容となります。

[T=0]

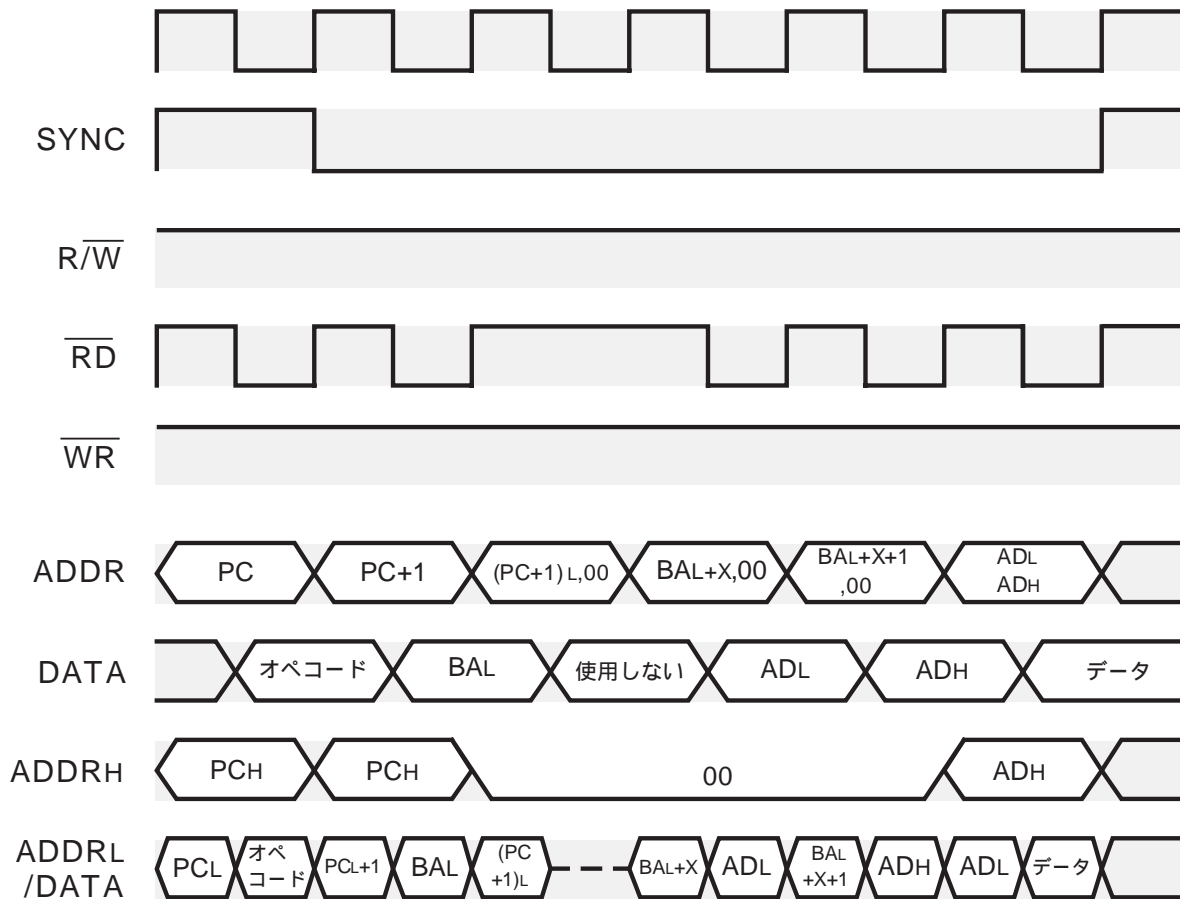
間接 X(Indirect X)

該当命令 : ADC (\$ zz, X) (T = 0)
 AND (\$ zz, X) (T = 0)
 CMP (\$ zz, X) (T = 0)
 EOR (\$ zz, X) (T = 0)
 LDA (\$ zz, X) (T = 0)
 ORA (\$ zz, X) (T = 0)
 SBC (\$ zz, X) (T = 0)

必要バイト数 : 2

必要サイクル数 : 6

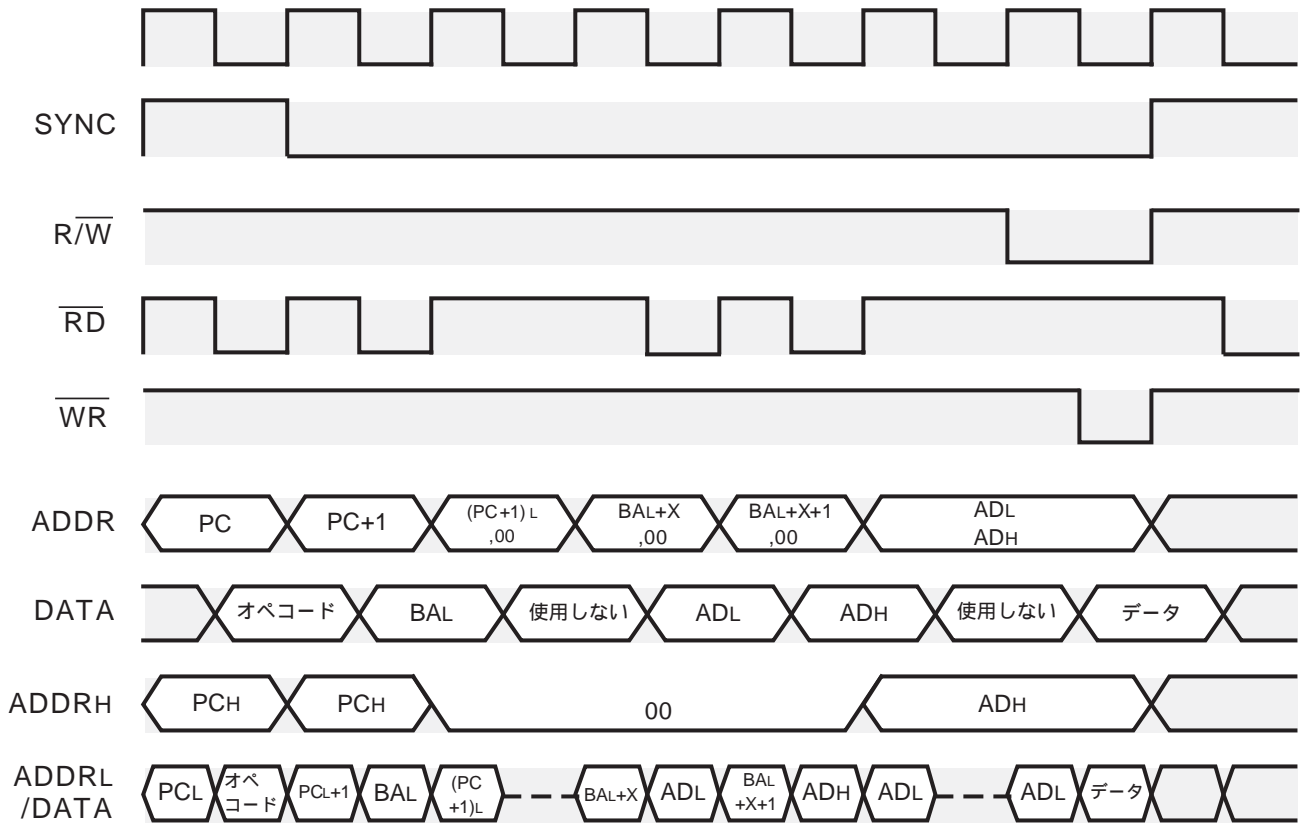
タイミング :



BA:ベーシックアドレス

間接 X(Indirect X)

該当命令 : STA (\$ zz, X)
 必要バイト数 : 2
 必要サイクル数 : 7
 タイミング :



BA:ベーシックアドレス

[T=0]

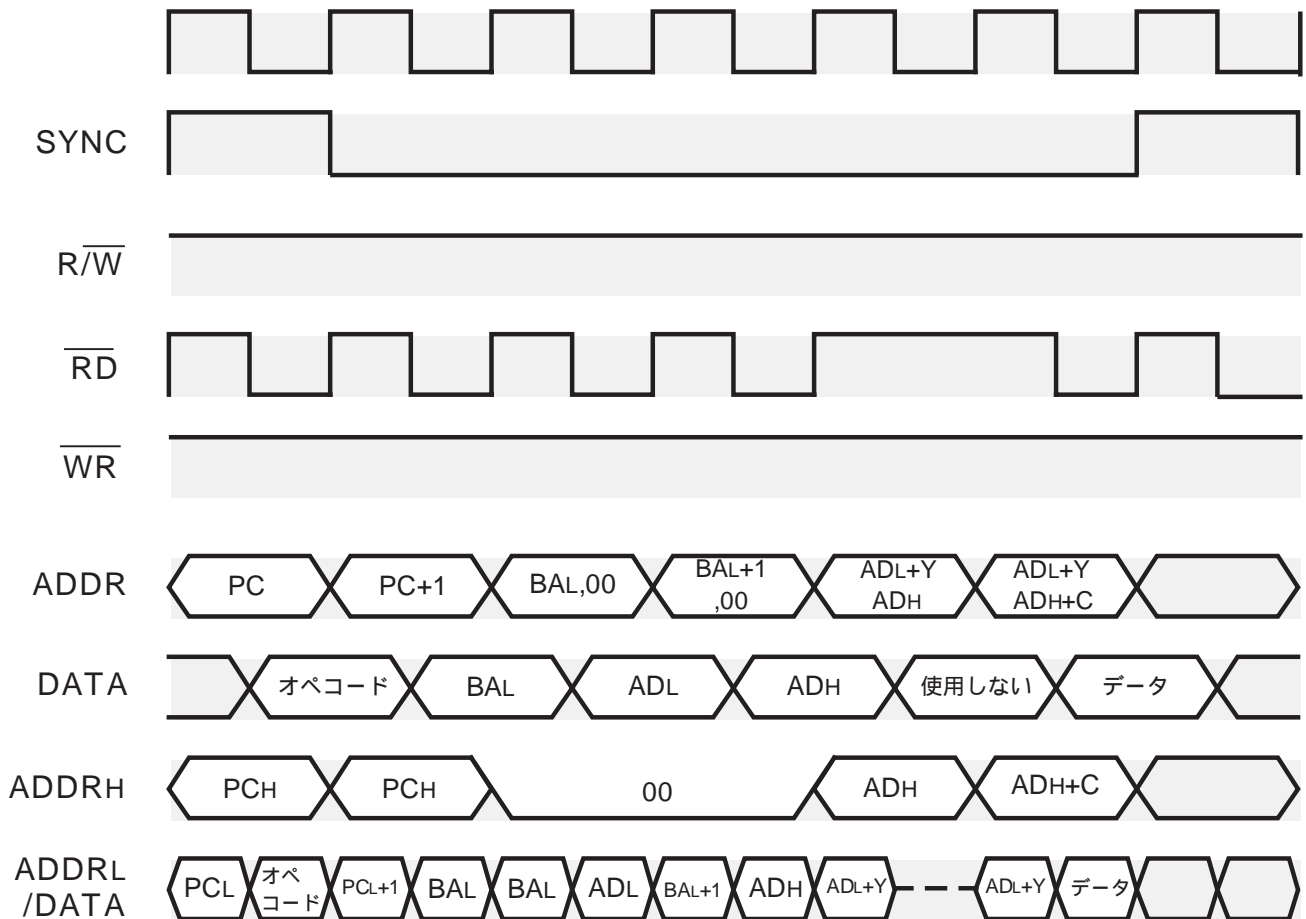
間接 Y(Indirect Y)

該当命令 : ADC (\$ zz), Y (T = 0)
 AND (\$ zz), Y (T = 0)
 CMP (\$ zz), Y (T = 0)
 EOR (\$ zz), Y (T = 0)
 LDA (\$ zz), Y (T = 0)
 ORA (\$ zz), Y (T = 0)
 SBC (\$ zz), Y (T = 0)

必要バイト数 : 2

必要サイクル数 : 6

タイミング :

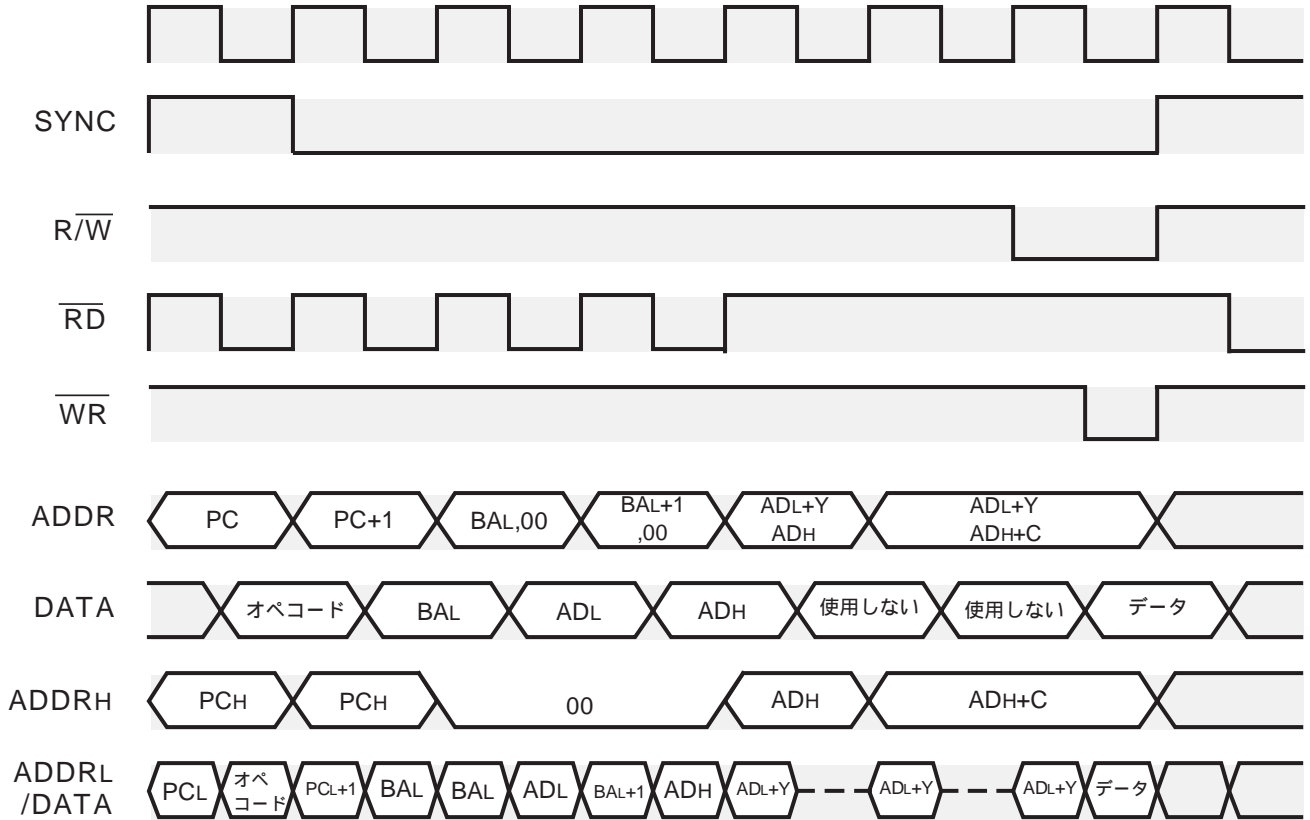


BA:ベーシックアドレス

C:ADL+Y のキャリ

間接 Y(Indirect Y)

該当命令 : STA (\$ zz), Y
 必要バイト数 : 2
 必要サイクル数 : 7
 タイミング :



BA:ベーシックアドレス
 C:ADL+Y のキャリ

相対(Relative)

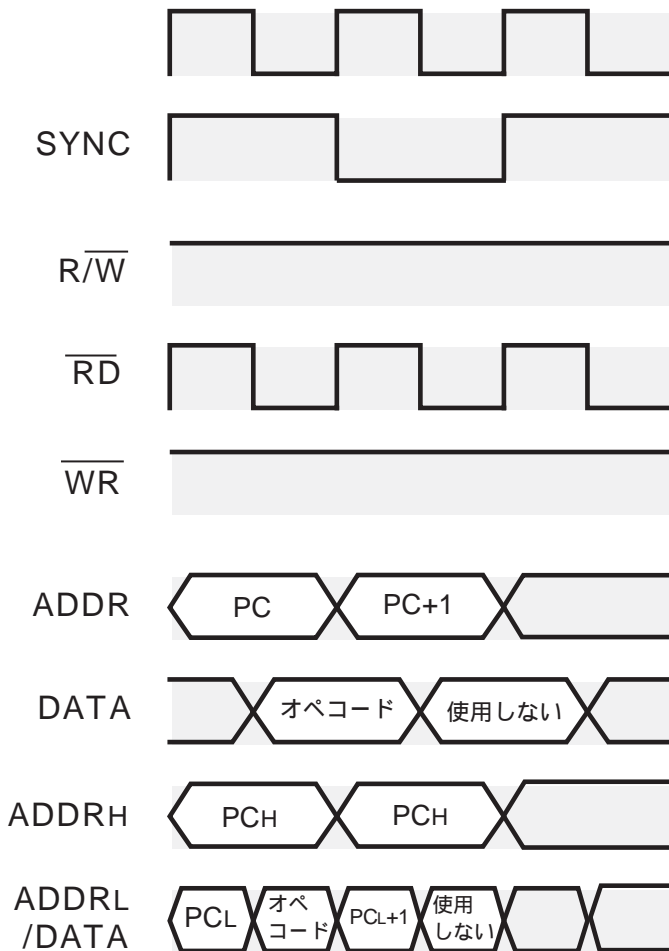
該当命令 : BCC \$hhll
 : BCS \$hhll
 : BEQ \$hhll
 : BMI \$hhll
 : BNE \$hhll
 : BPL \$hhll
 : BVC \$hhll
 : BVS \$hhll

必要バイト数 : 2

(1) ブランチしない場合

必要サイクル数 : 2

タイミング :



相対(Relative)

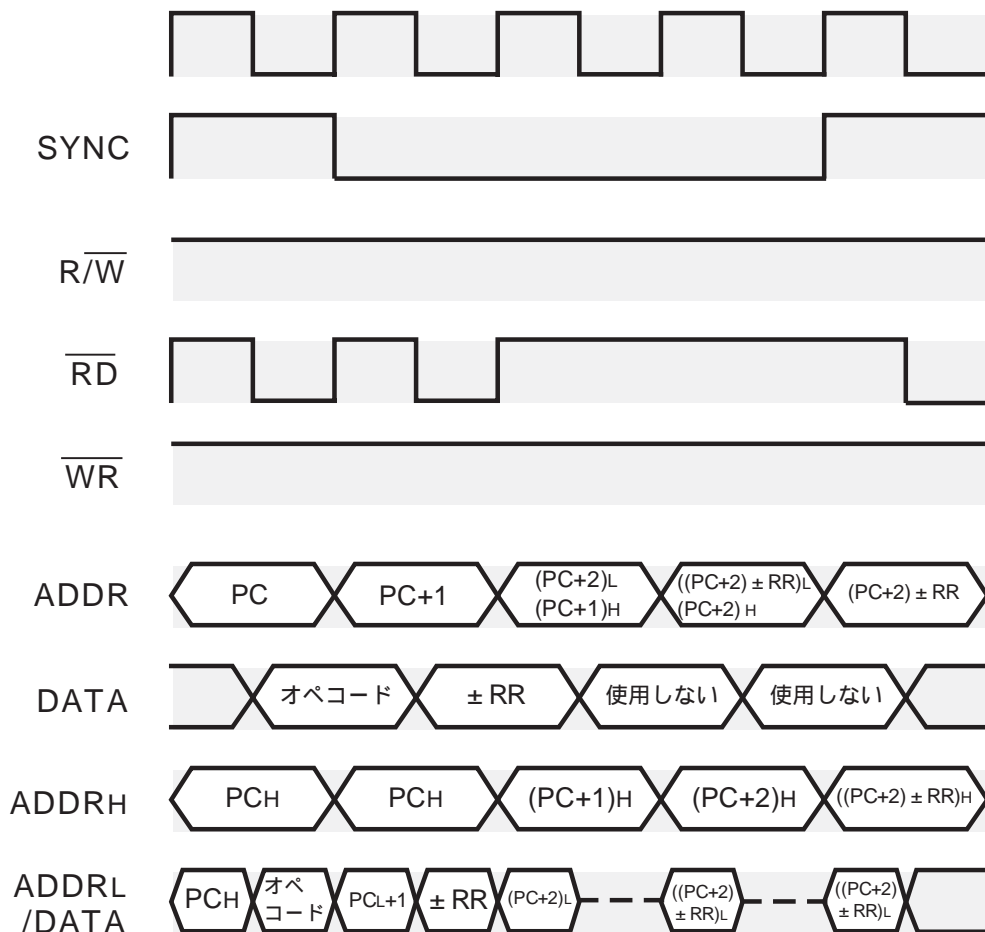
該当命令 : BCC \$ hll
 : BCS \$ hll
 : BEQ \$ hll
 : BMI \$ hll
 : BNE \$ hll
 : BPL \$ hll
 : BVC \$ hll
 : BVS \$ hll

必要バイト数 : 2

(2) ブランチする場合

必要サイクル数 : 4

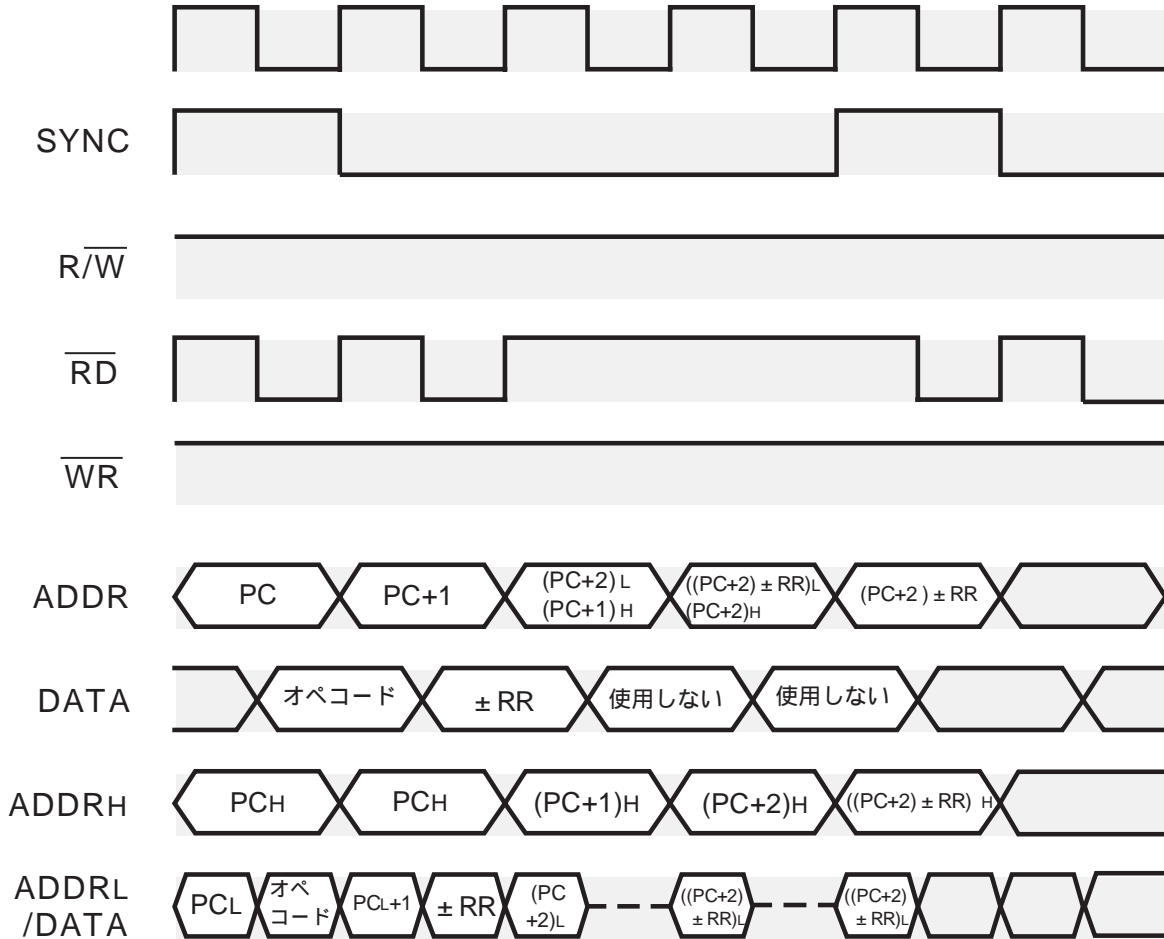
タイミング :



RR:オフセットの値

相対(Relative)

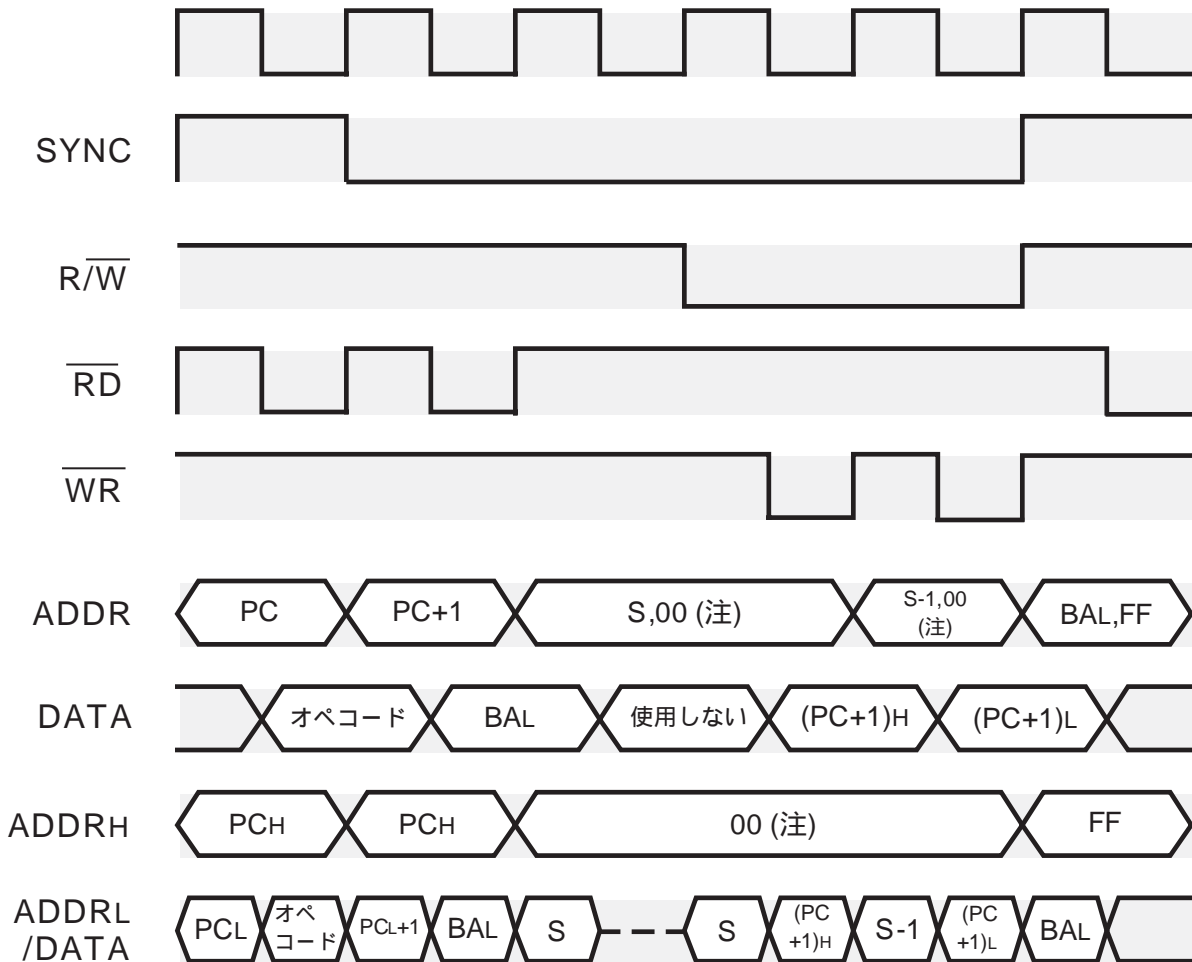
該当命令 : BRA \$hhll
 必要バイト数 : 2
 必要サイクル数 : 4
 タイミング :



RR:オフセットの値

スペシャルページ(Special page)

該当命令 : JSR ¥ \$ hhll
 必要バイト数 : 2
 必要サイクル数 : 5
 タイミング :



BA:ベーシックアドレス

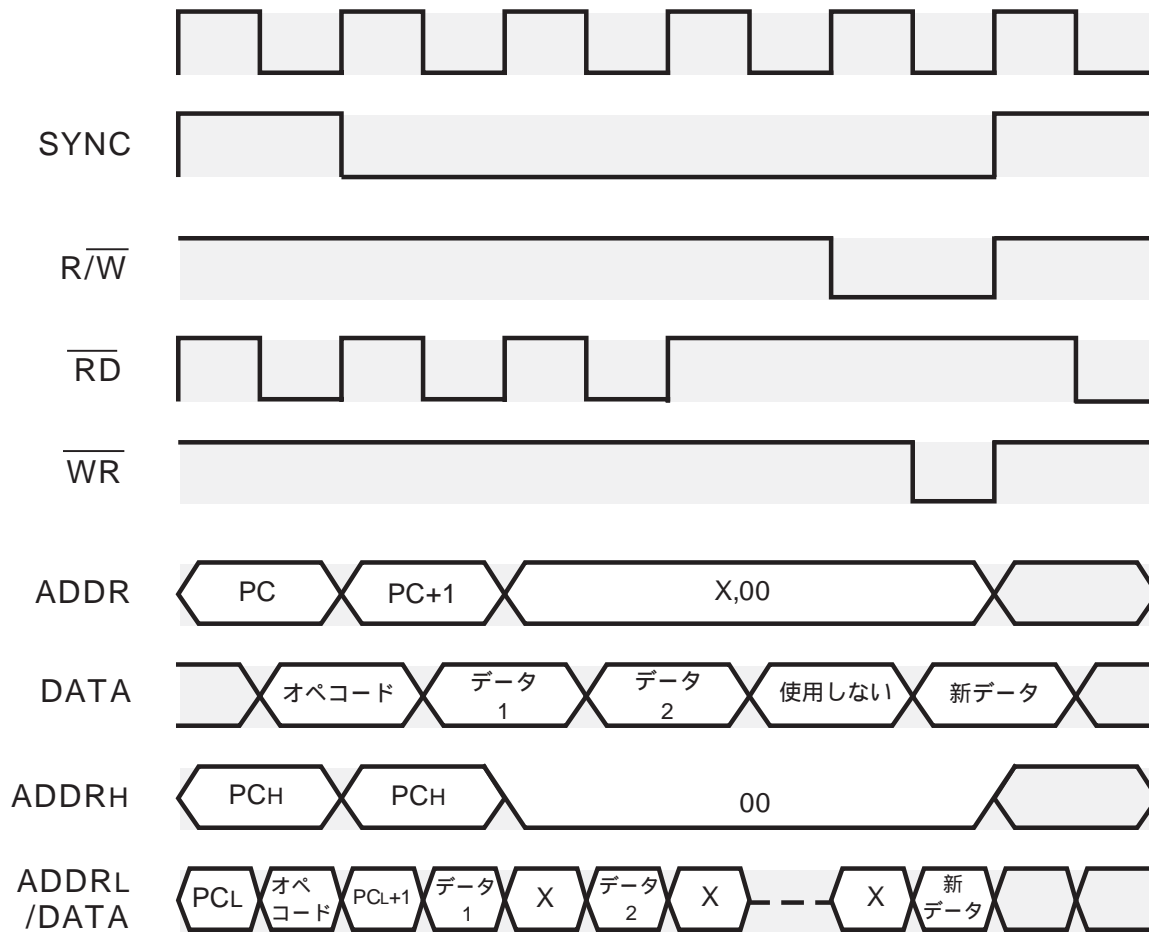
注. 品種によって、“01”又はSPSフラグの内容となります。

該当命令 : ADC # \$ nn (T = 1)
 AND # \$ nn (T = 1)
 EOR # \$ nn (T = 1)
 ORA # \$ nn (T = 1)
 SBC # \$ nn (T = 1)

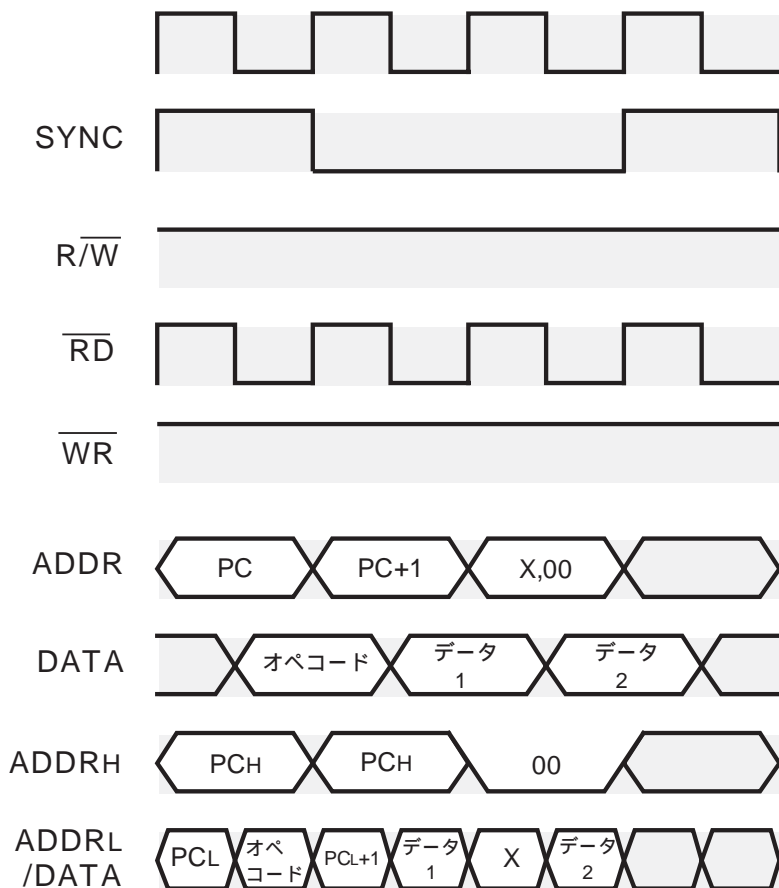
必要バイト数 : 2

必要サイクル数 : 5

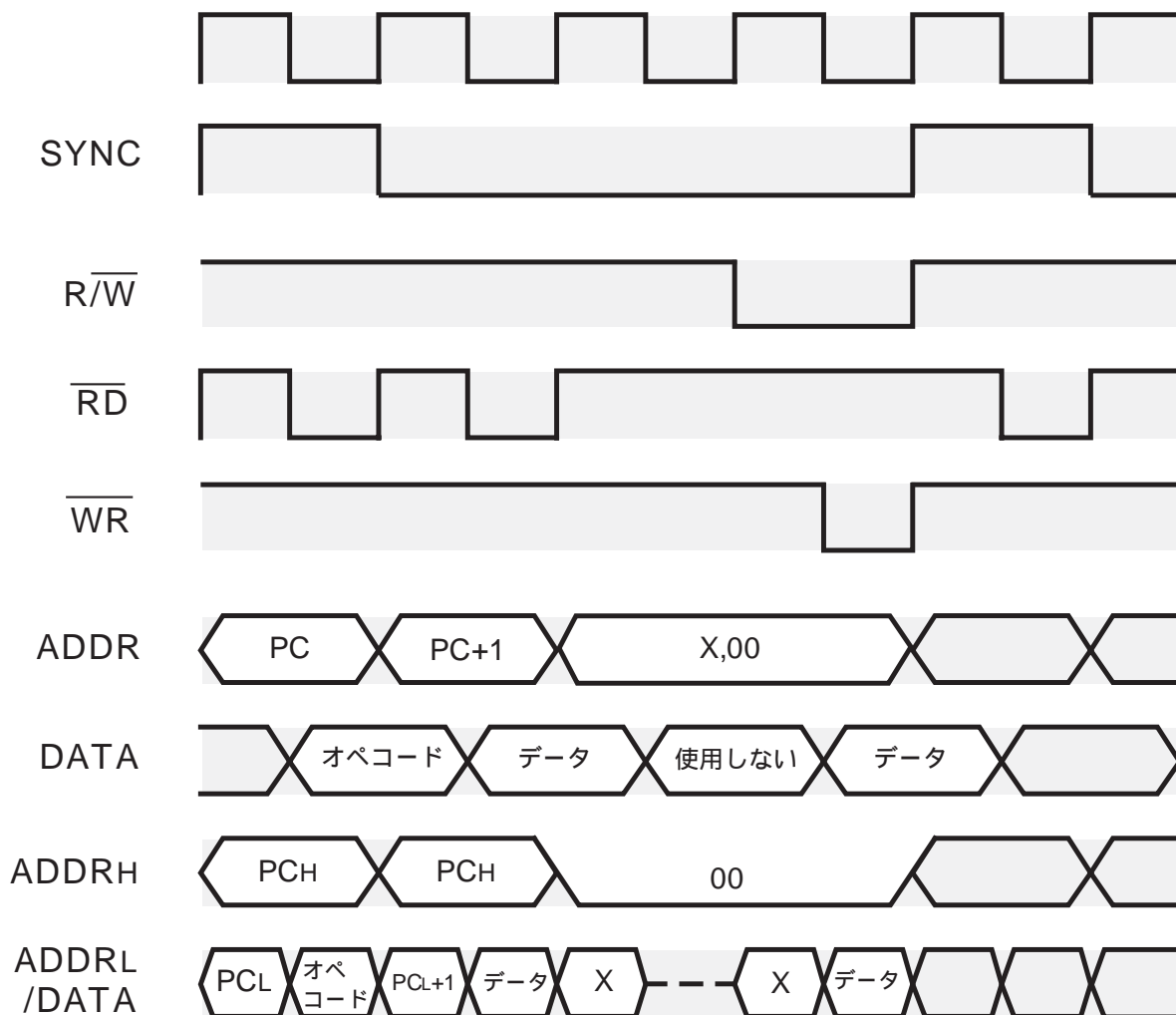
タイミング :



該当命令 : CMP # \$ nn (T = 1)
 必要バイト数 : 2
 必要サイクル数 : 3
 タイミング :

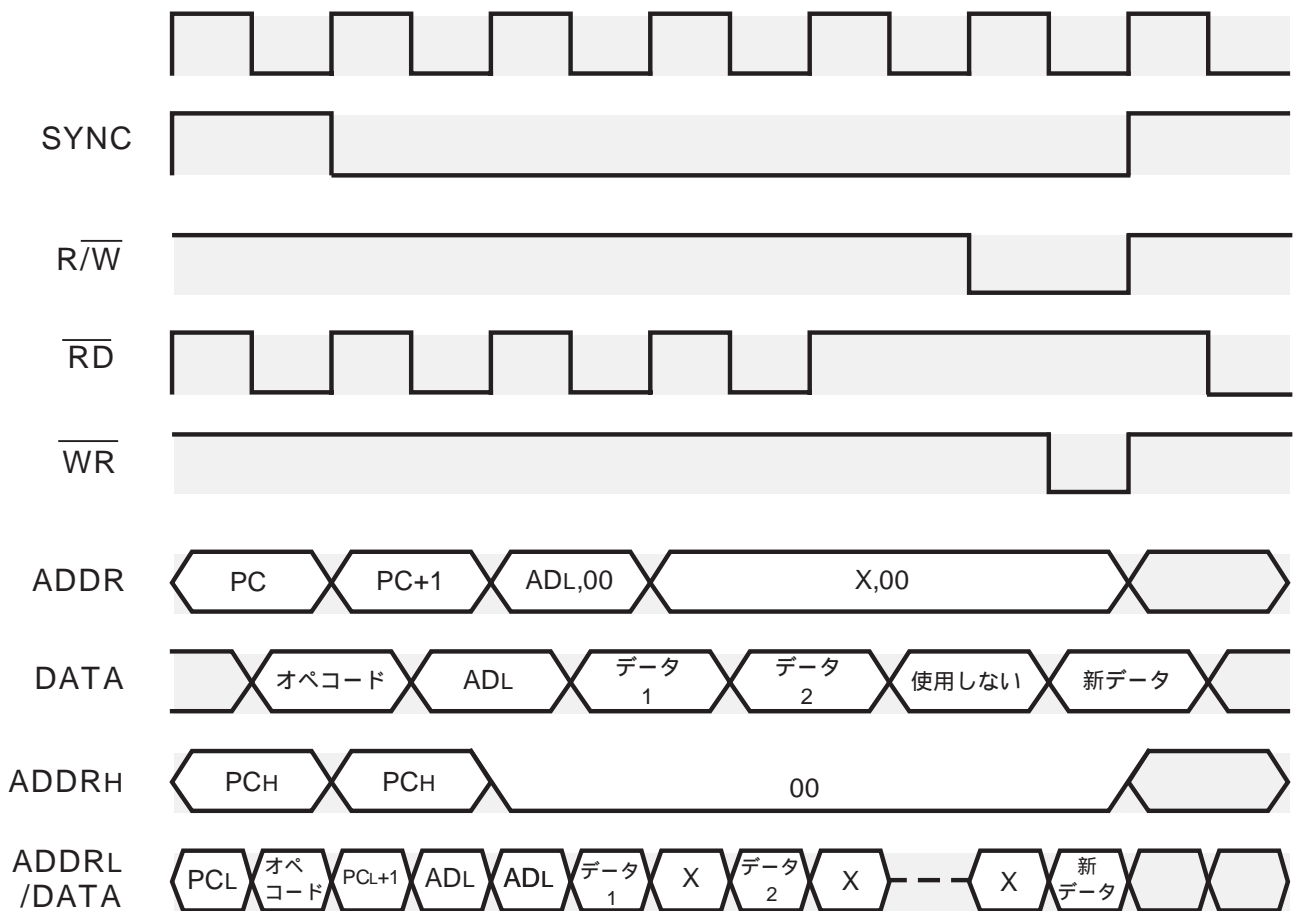


該当命令 : LDA # \$nn (T = 1)
 必要バイト数 : 2
 必要サイクル数 : 4
 タイミング :

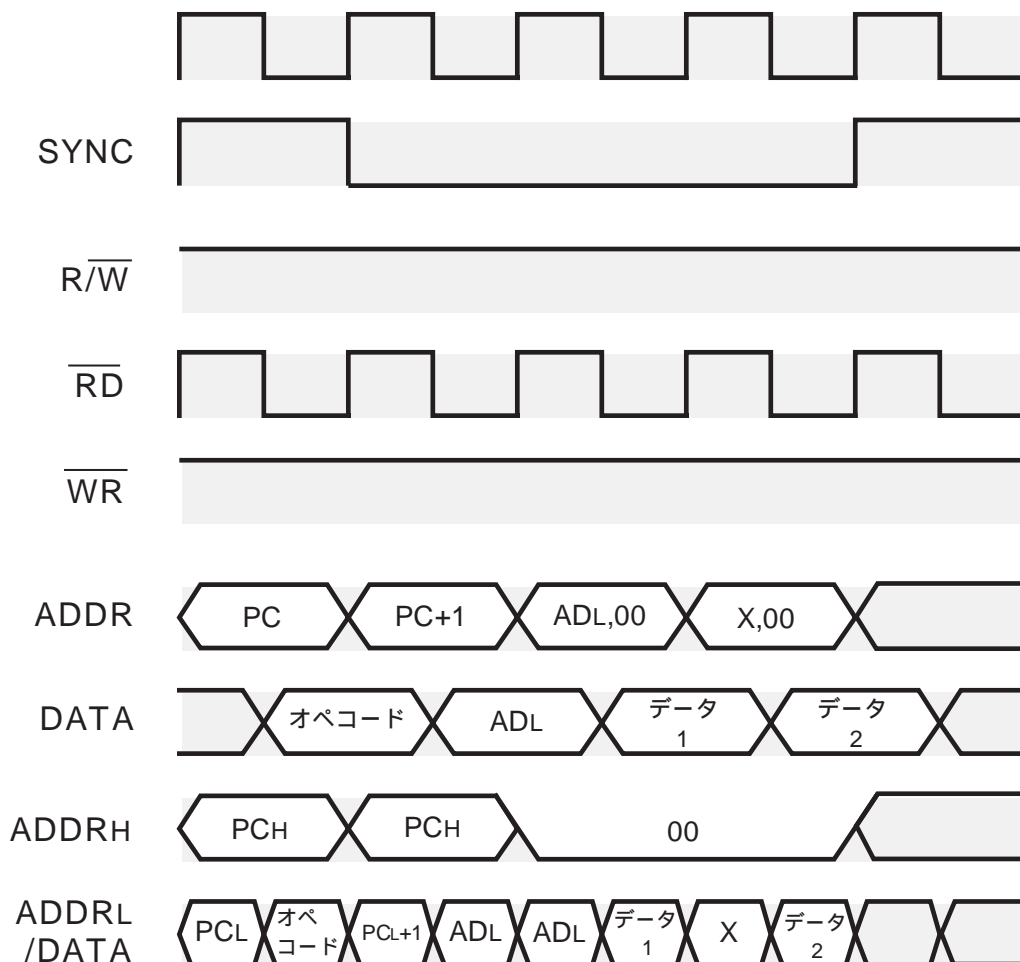


該当命令 : ADC \$zz (T = 1)
 AND \$zz (T = 1)
 EOR \$zz (T = 1)
 ORA \$zz (T = 1)
 SBC \$zz (T = 1)

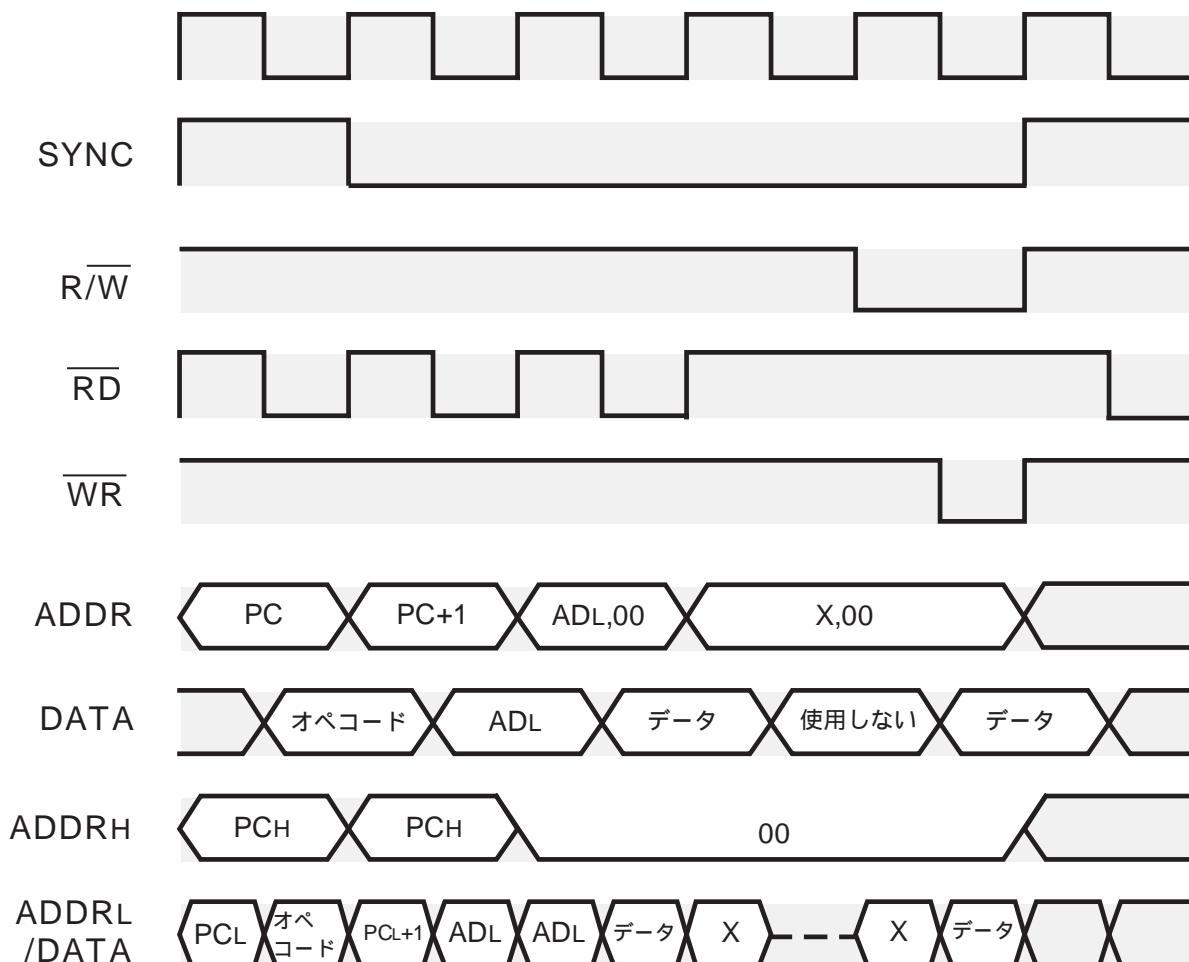
必要バイト数 : 2
 必要サイクル数 : 6
 タイミング :



該当命令 : CMP \$zz (T = 1)
 必要バイト数 : 2
 必要サイクル数 : 4
 タイミング :

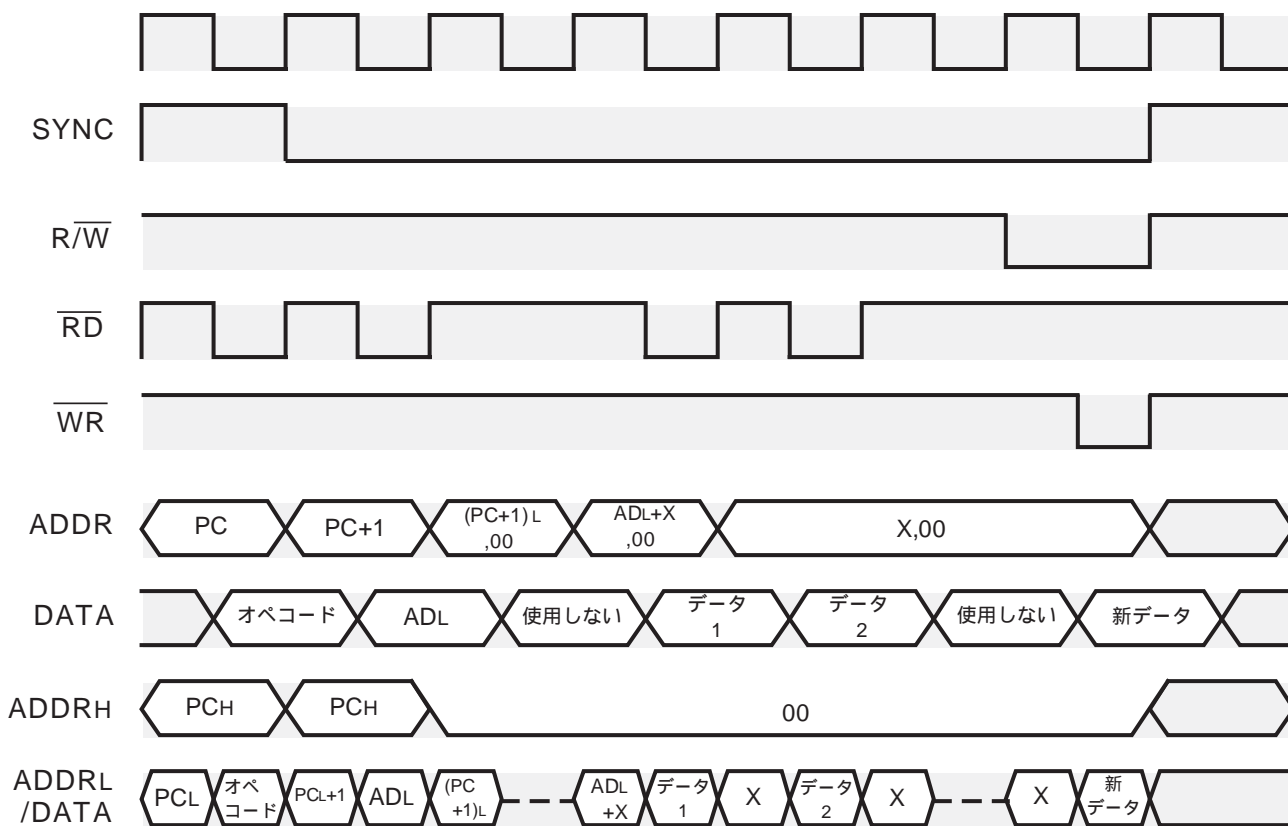


該当命令 : LDA \$zz (T = 1)
 必要バイト数 : 2
 必要サイクル数 : 5
 タイミング :

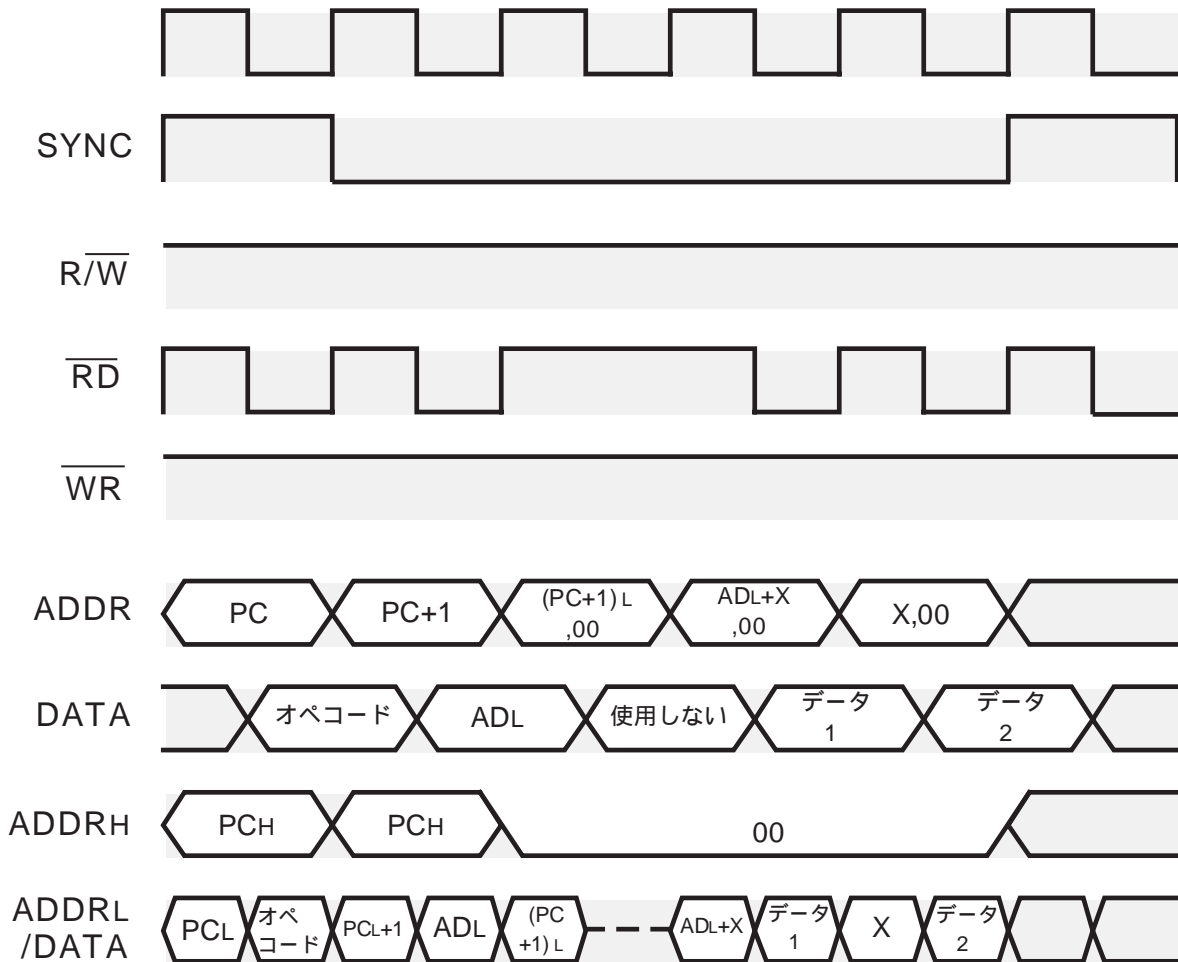


該当命令 : ADC \$ZZ, X (T = 1)
 AND \$ZZ, X (T = 1)
 EOR \$ZZ, X (T = 1)
 ORA \$ZZ, X (T = 1)
 SBC \$ZZ, X (T = 1)

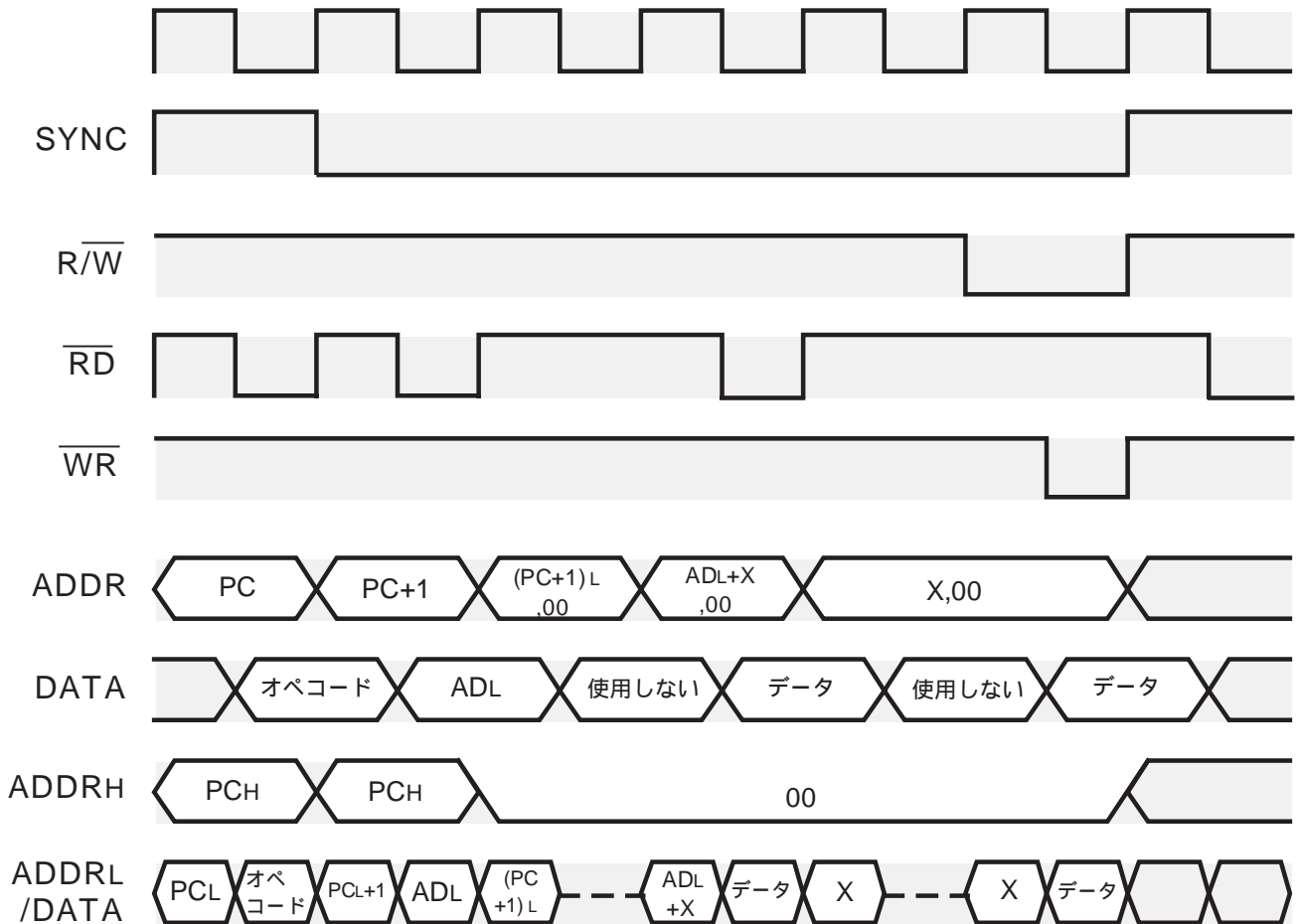
必要バイト数 : 2
 必要サイクル数 : 7
 タイミング :



該当命令 : CMP \$zz, X(T = 1)
 必要バイト数 : 2
 必要サイクル数 : 5
 タイミング :



該当命令 : LDA \$zz, X (T = 1)
 必要バイト数 : 2
 必要サイクル数 : 6
 タイミング :

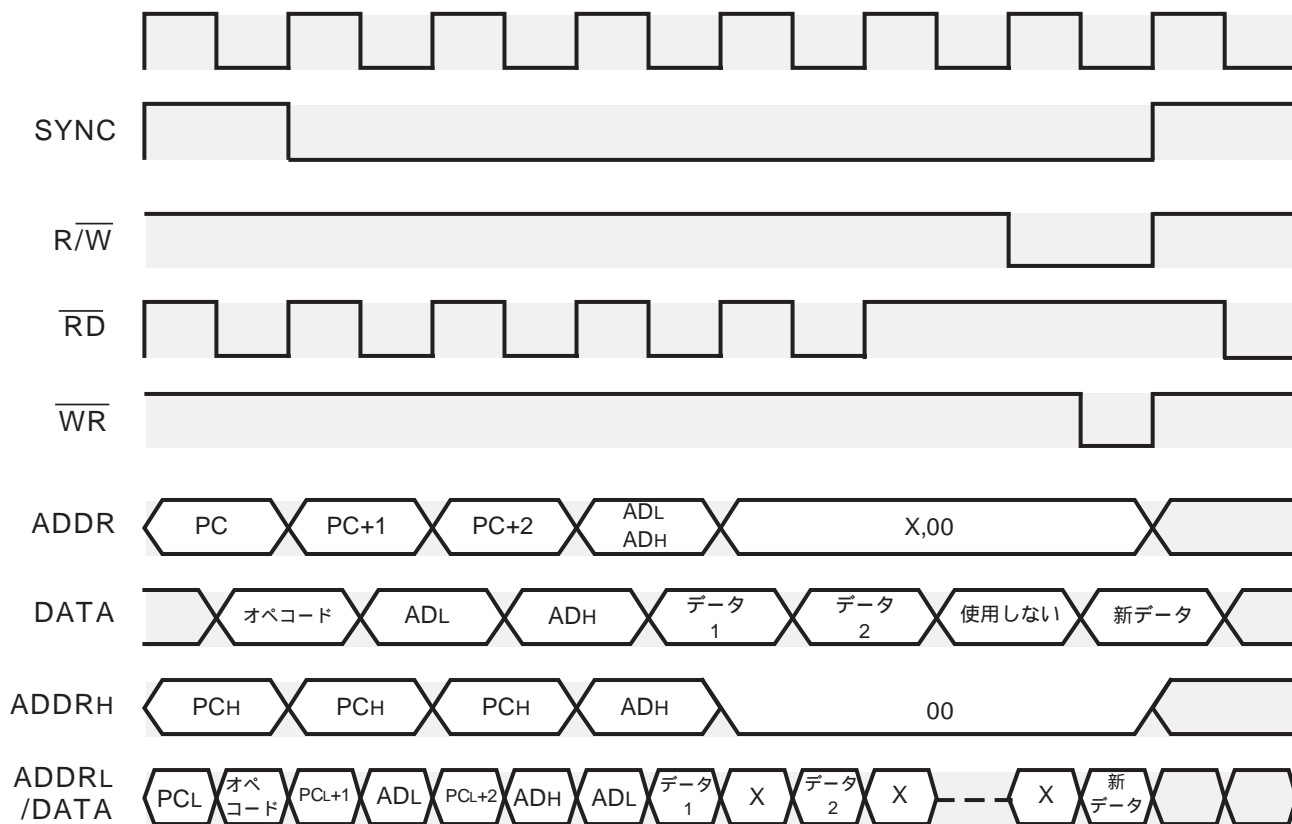


該当命令 : ADC \$hhll (T = 1)
 AND \$hhll (T = 1)
 EOR \$hhll (T = 1)
 ORA \$hhll (T = 1)
 SBC \$hhll (T = 1)

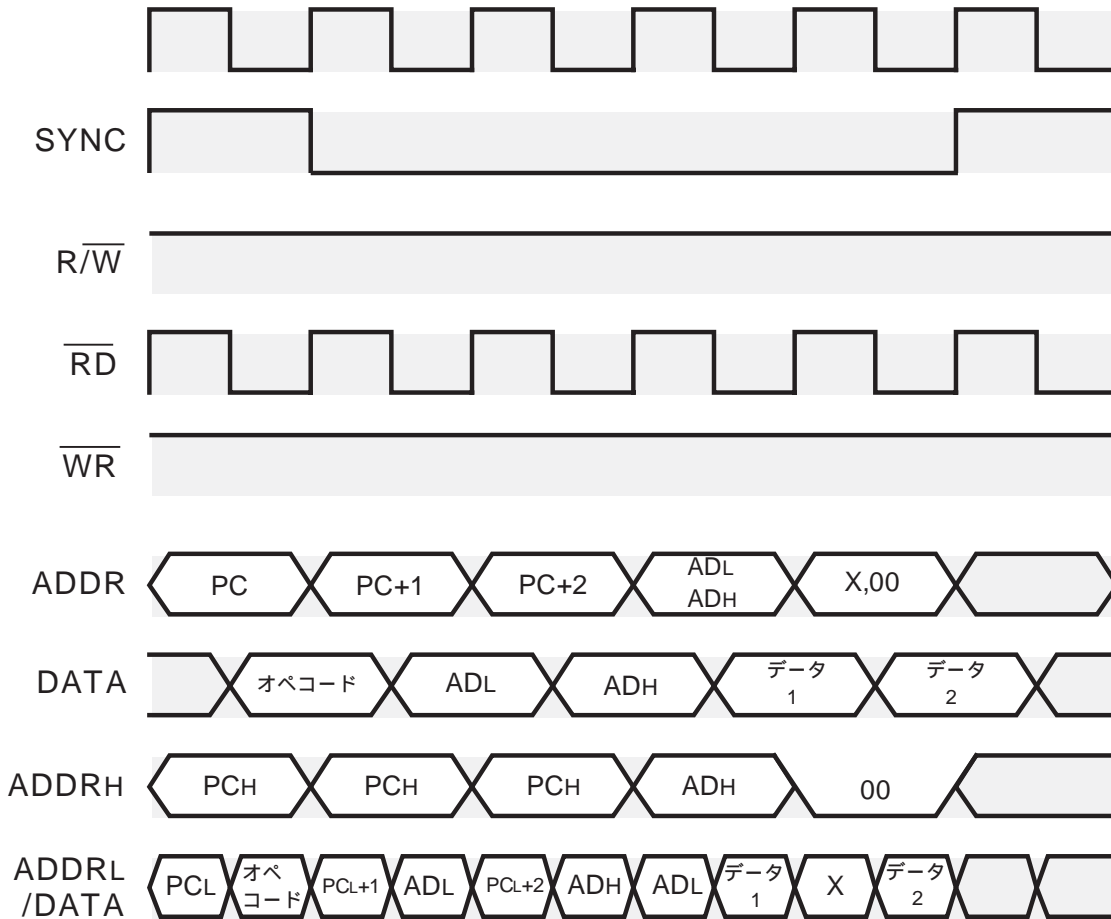
必要バイト数 : 3

必要サイクル数 : 7

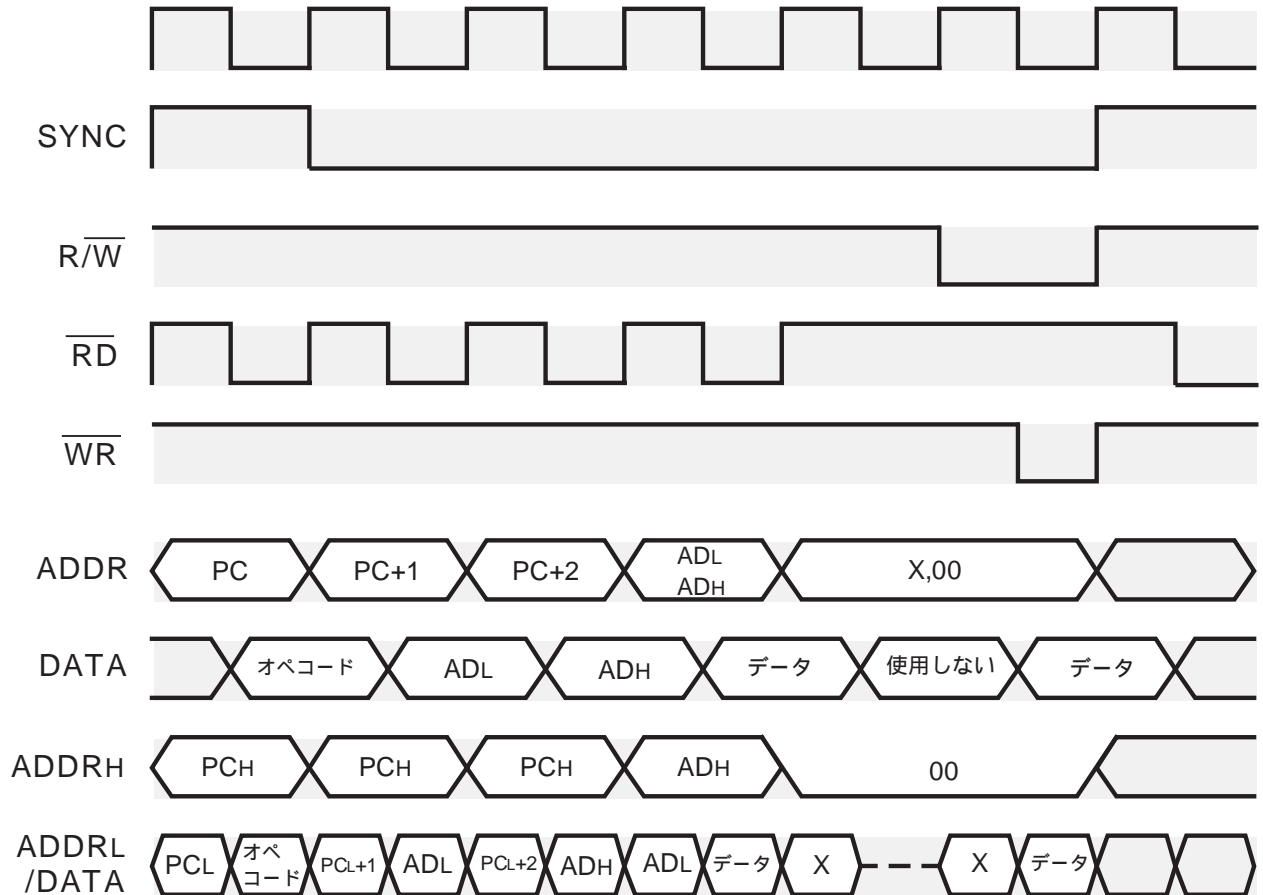
タイミング :



該当命令 : CMP \$hhll (T = 1)
 必要バイト数 : 3
 必要サイクル数 : 5
 タイミング :



該当命令 : LDA \$hhll(T = 1)
 必要バイト数 : 3
 必要サイクル数 : 6
 タイミング :



[T = 1]

絶対 X(Absolute X)

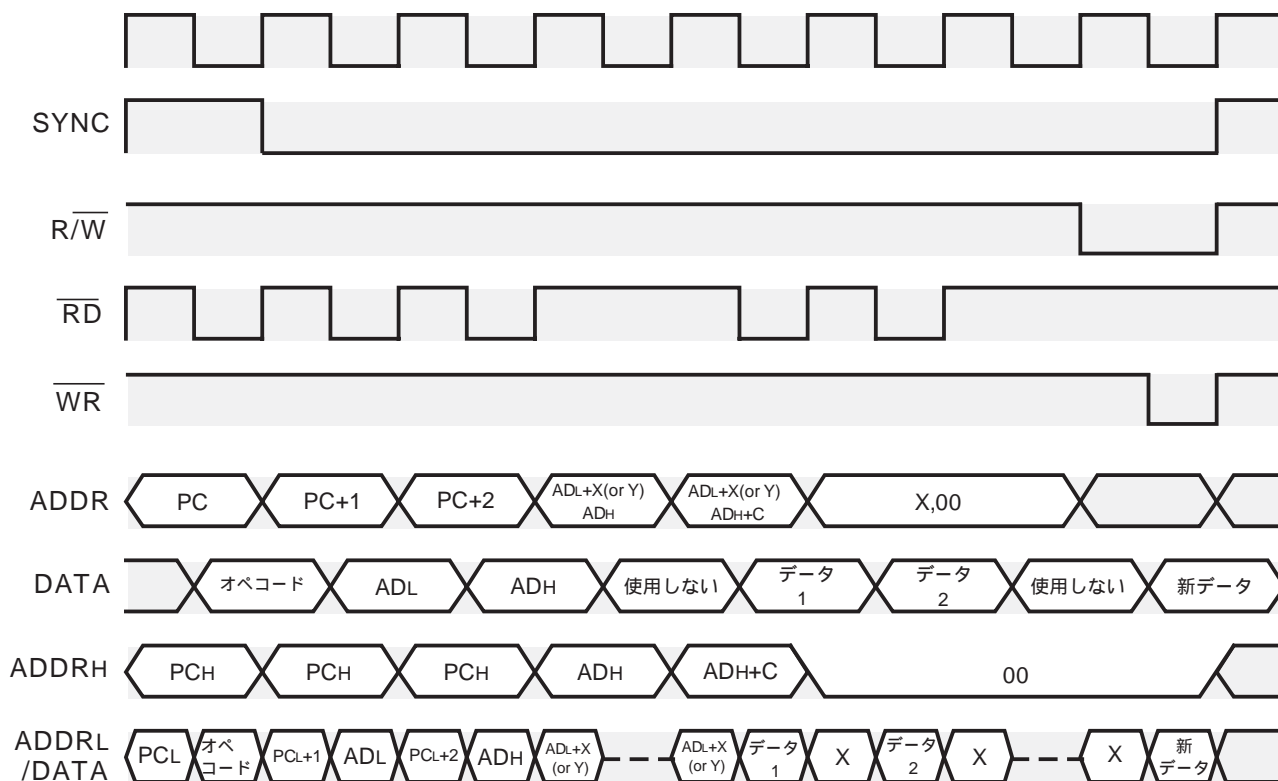
絶対 Y(Absolute Y)

該当命令 : ADC \$hhll, X 又は Y (T = 1)
 AND \$hhll, X 又は Y (T = 1)
 EOR \$hhll, X 又は Y (T = 1)
 ORA \$hhll, X 又は Y (T = 1)
 SBC \$hhll, X 又は Y (T = 1)

必要バイト数 : 3

必要サイクル数 : 8

タイミング :



C:ADL+X(又は Y)のキャリ

絶対 X(Absolute X)

絶対 Y(Absolute Y)

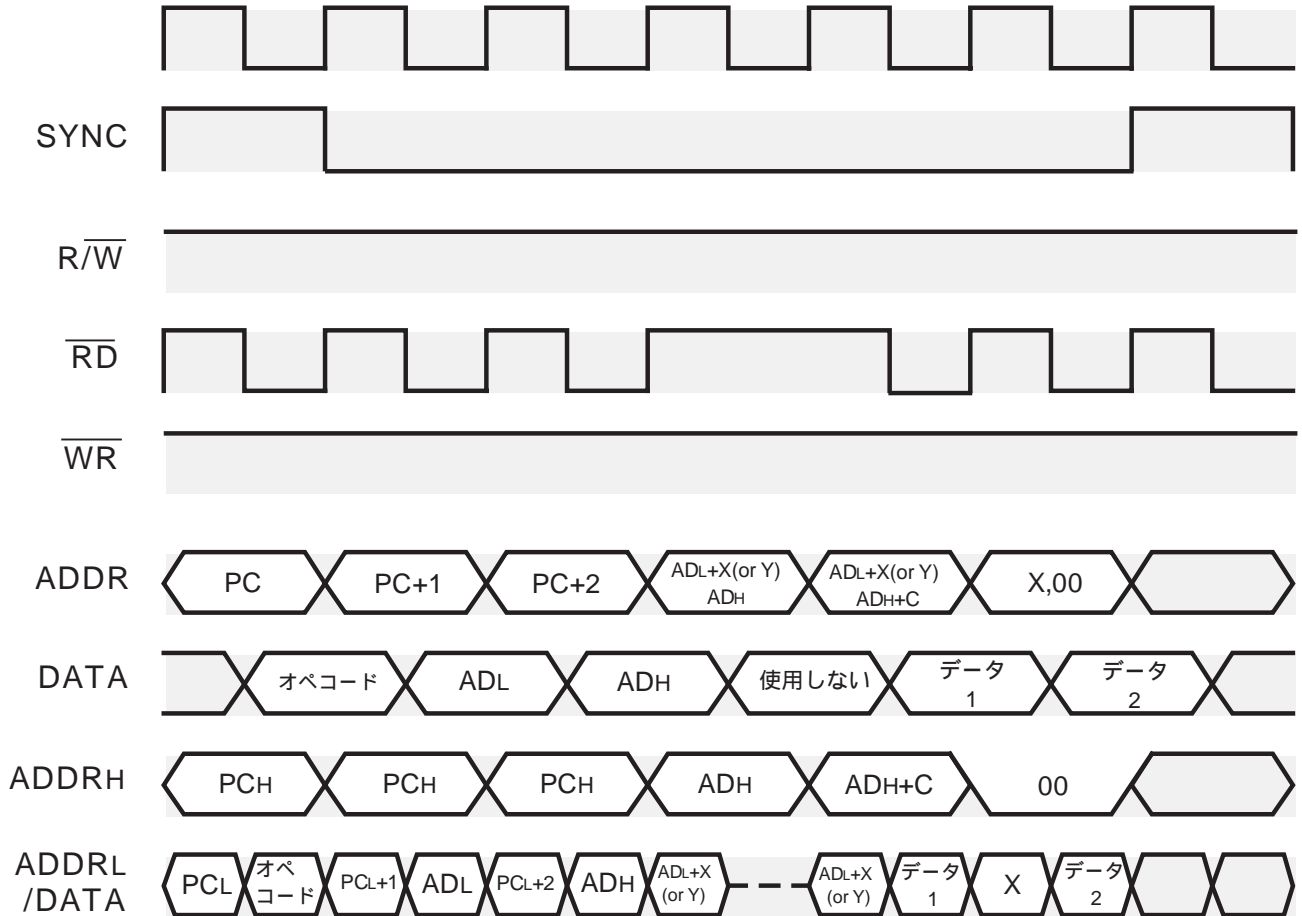
[T = 1]

該当命令 : CMP \$hhll, X 又は Y (T = 1)

必要バイト数 : 3

必要サイクル数 : 6

タイミング :



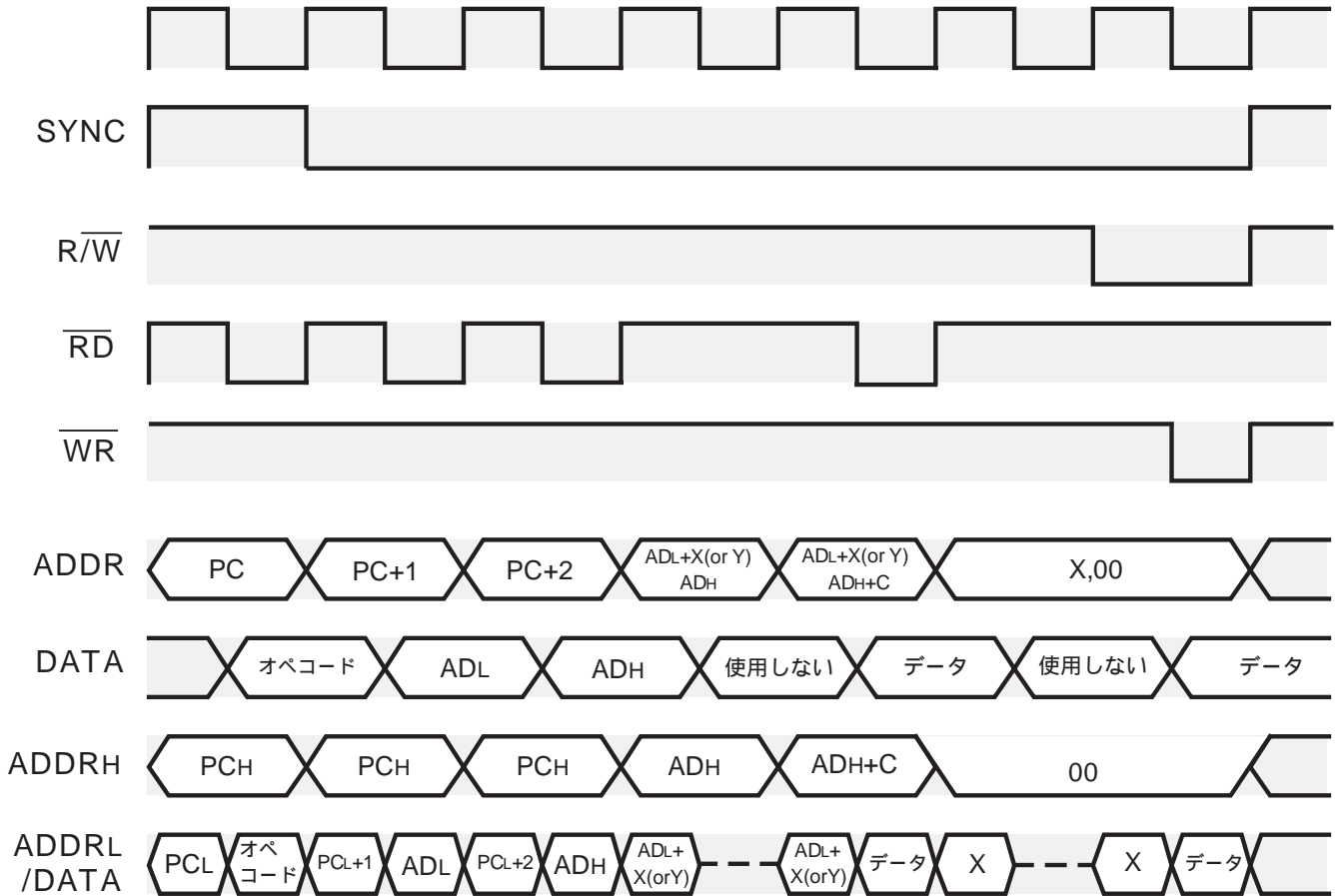
C:ADL+X(又はY)のキャリ

絶対 X(Absolute X)

[T = 1]

絶対 Y(Absolute Y)

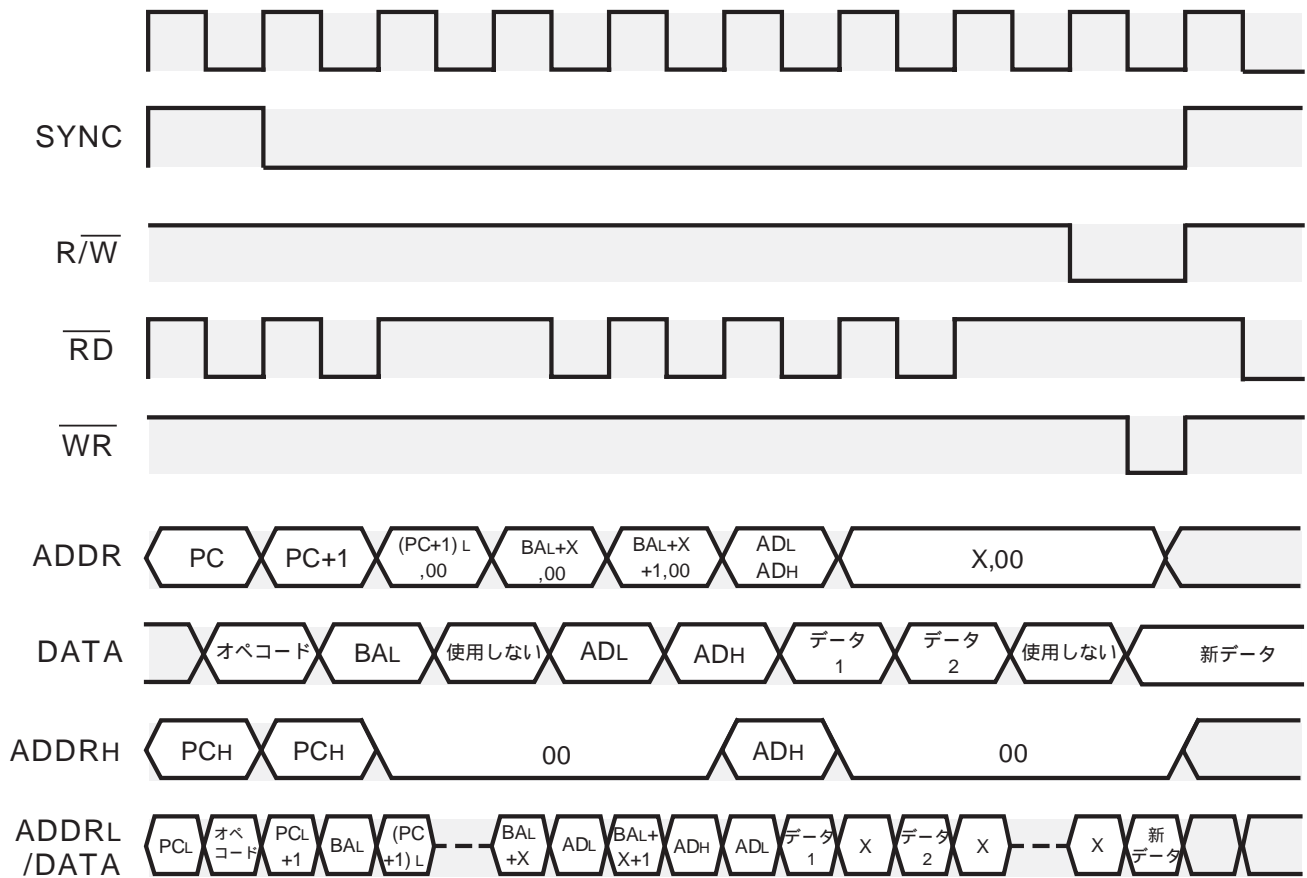
該当命令 : LDA \$hhl, X 又は Y (T = 1)
必要バイト数 : 3
必要サイクル数 : 7
タイミング :



C:ADL+X(又はY)のキャリ

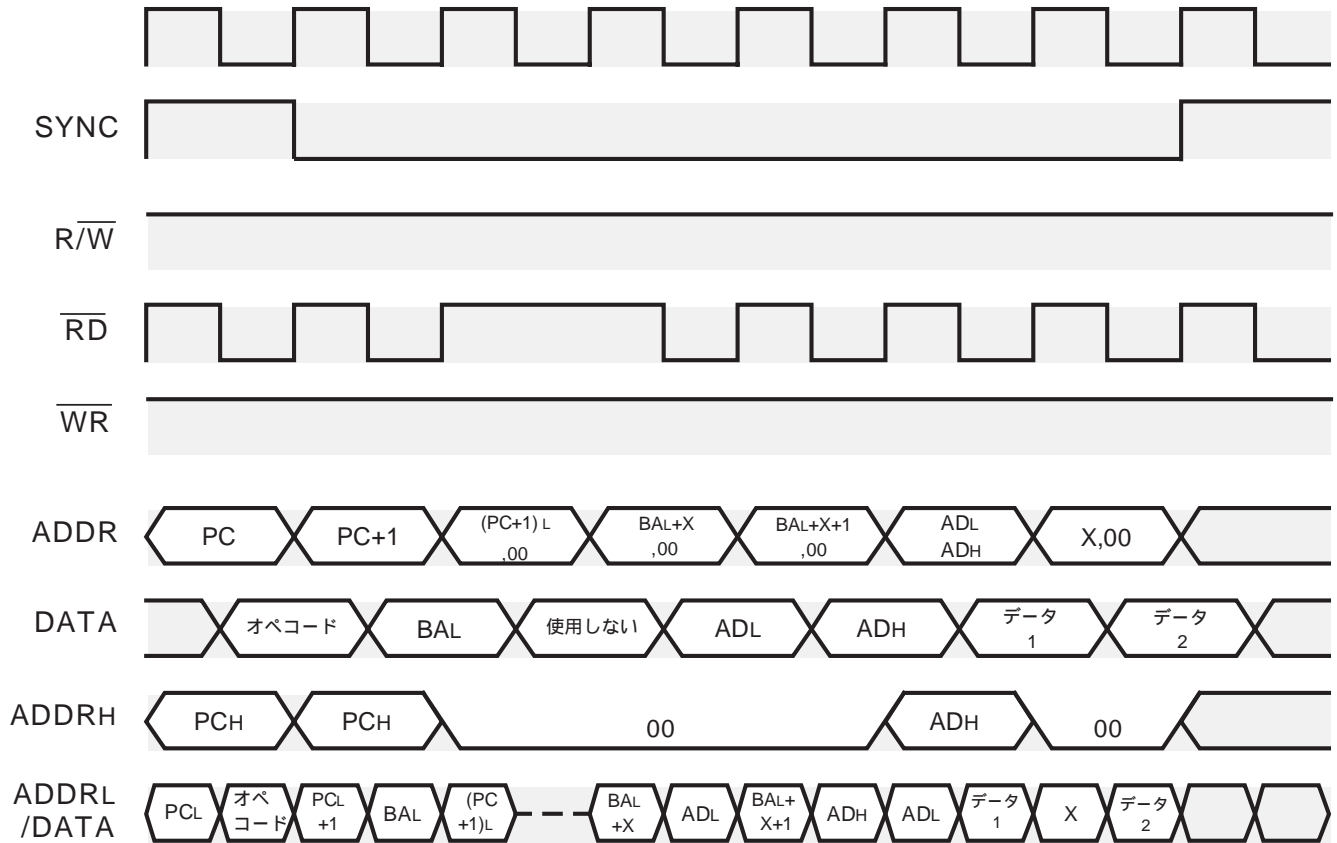
該当命令 : ADC (\$zz, X) (T = 1)
 AND (\$zz, X) (T = 1)
 EOR (\$zz, X) (T = 1)
 ORA (\$zz, X) (T = 1)
 SBC (\$zz, X) (T = 1)

必要バイト数 : 2
 必要サイクル数 : 9
 タイミング :



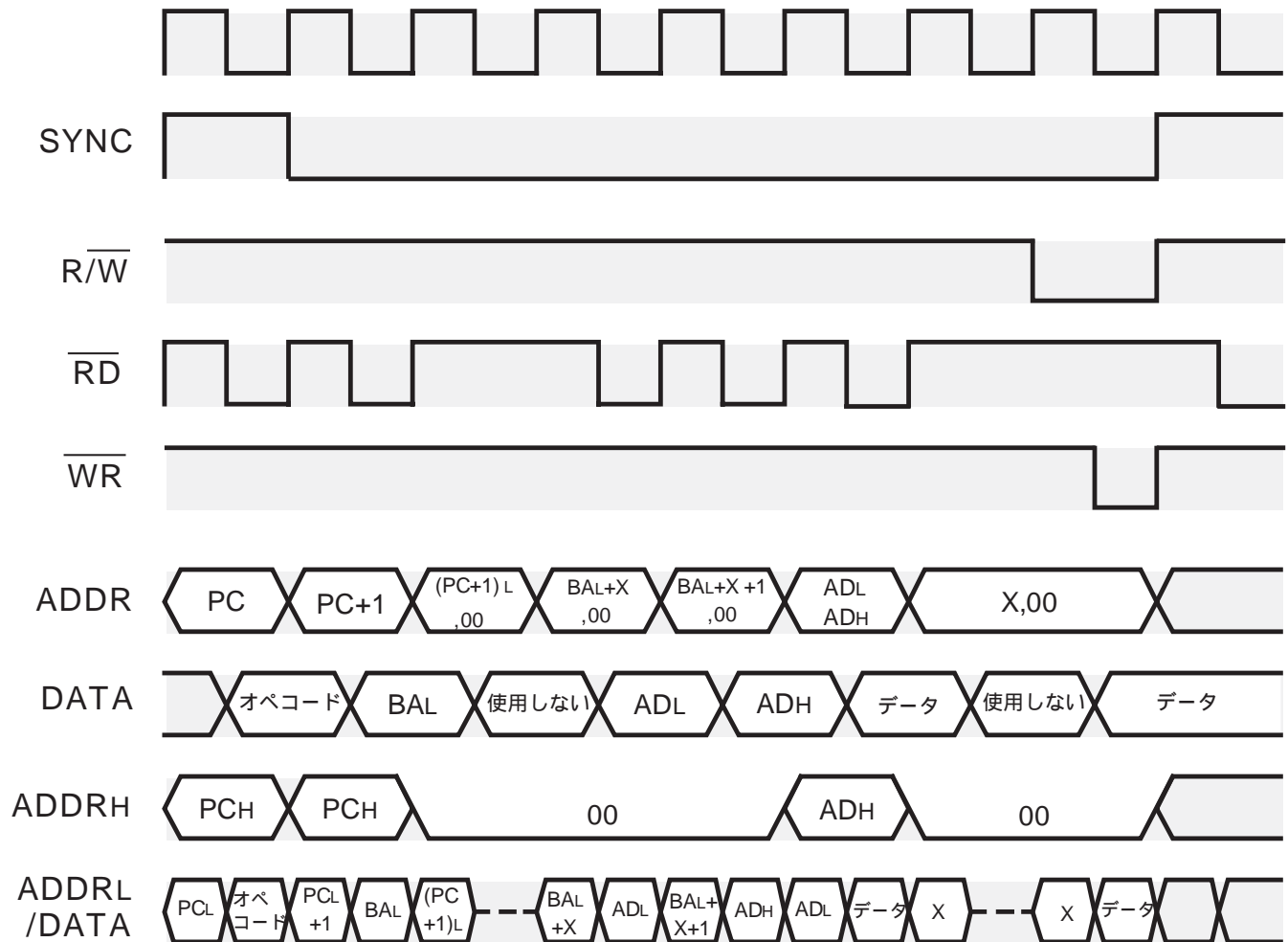
BA:ベーシックアドレス

該当命令 : CMP (\$zz, X) (T = 1)
 必要バイト数 : 2
 必要サイクル数 : 7
 タイミング :



BA:ベーシックアドレス

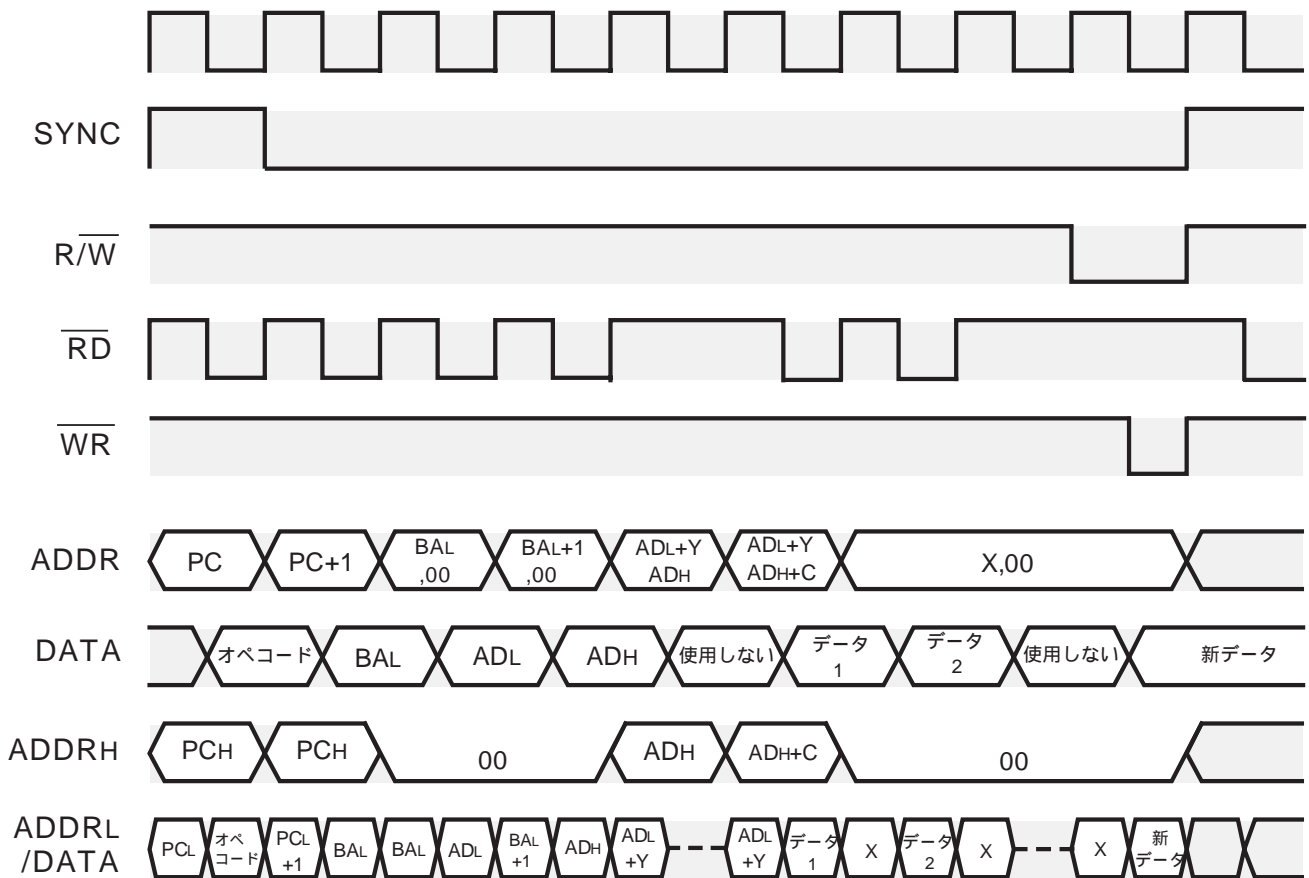
該当命令 : LDA (\$zz, X) (T = 1)
 必要バイト数 : 2
 必要サイクル数 : 8
 タイミング :



BA:ベーシックアドレス

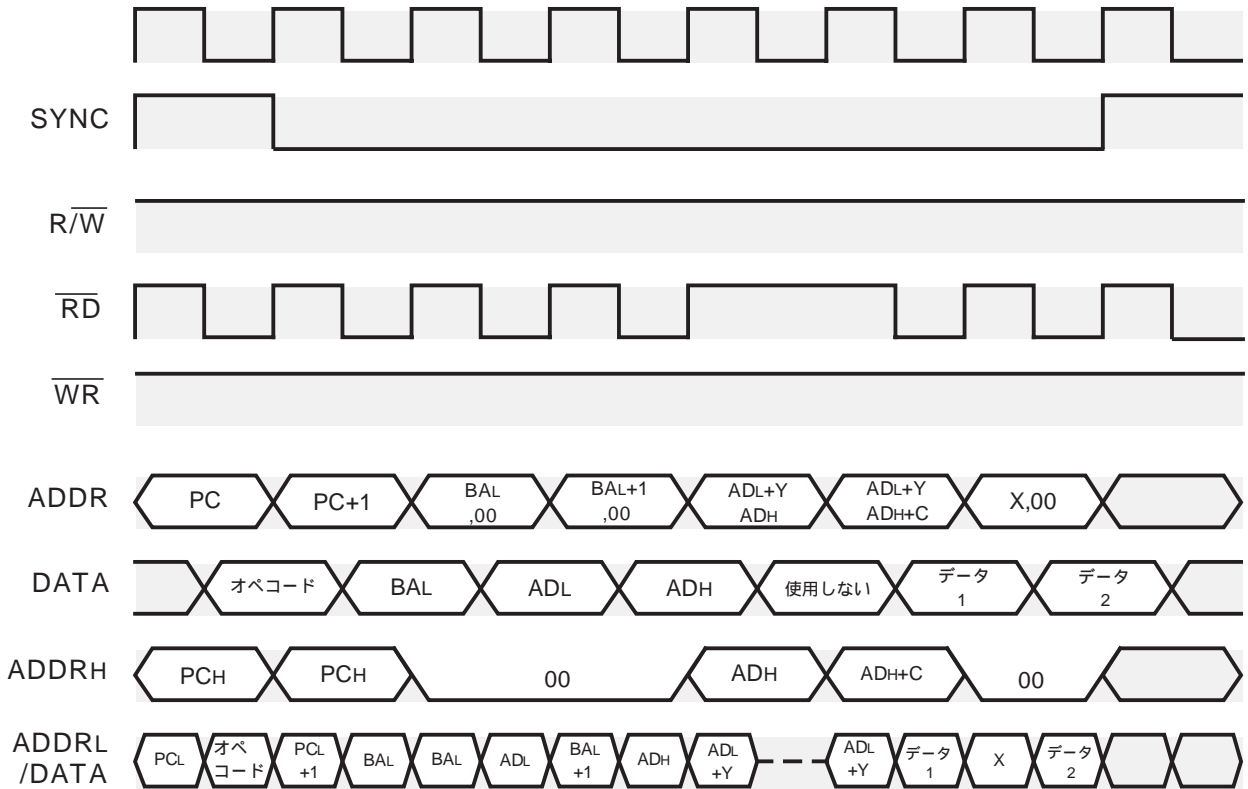
該当命令 : ADC (\$zz), Y (T = 1)
 AND (\$zz), Y (T = 1)
 EOR (\$zz), Y (T = 1)
 ORA (\$zz), Y (T = 1)
 SBC (\$zz), Y (T = 1)

必要バイト数 : 2
 必要サイクル数 : 9
 タイミング :



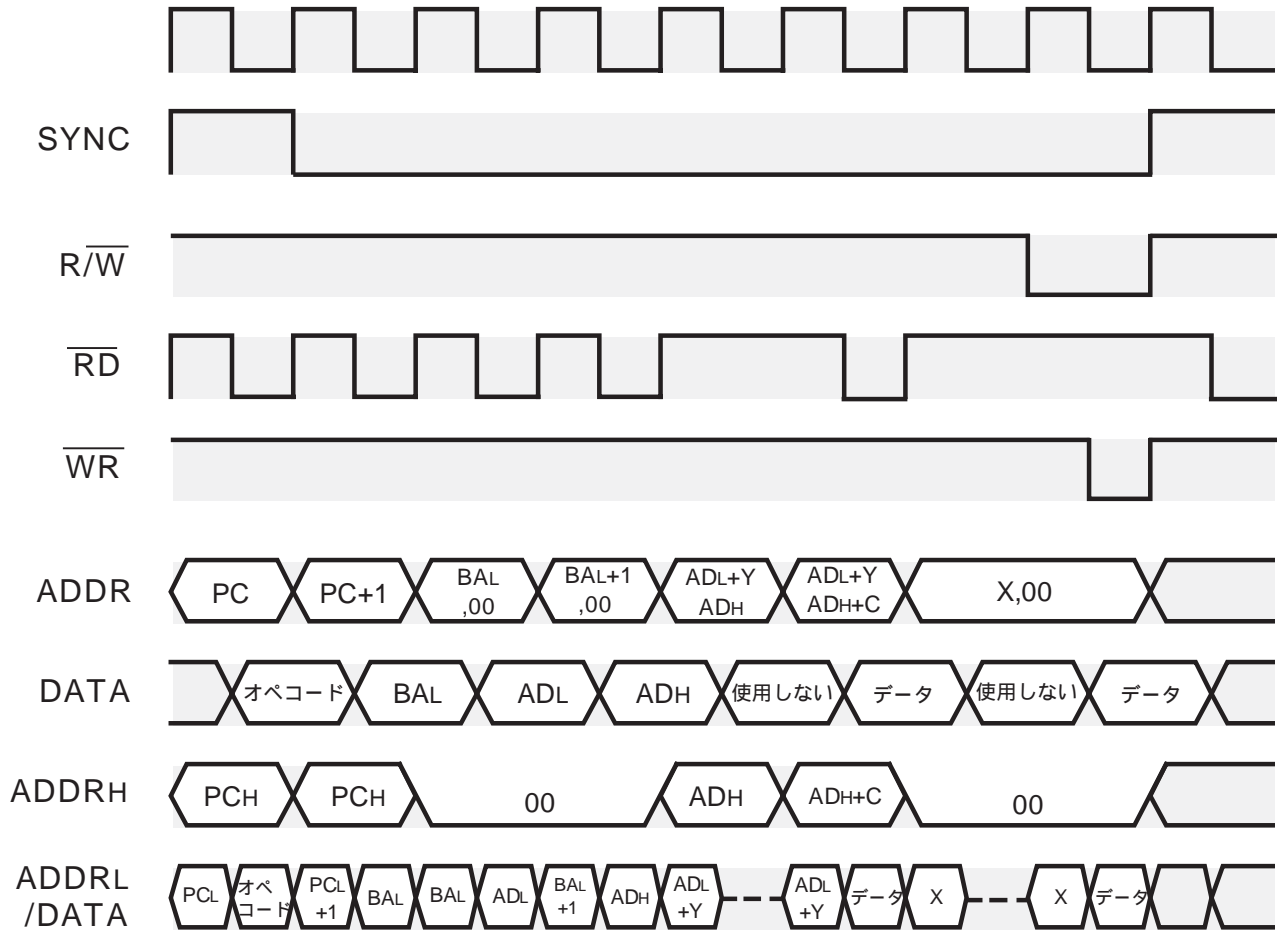
BA:ベーシックアドレス
 C:ADL+Yのキャリ

該当命令 : CMP (\$zz), Y (T = 1)
 必要バイト数 : 2
 必要サイクル数 : 7
 タイミング :



BA:ベースアドレス
 C:ADL+Yのキャリ

該当命令 : LDA (\$zz), Y (T = 1)
 必要バイト数 : 2
 必要サイクル数 : 8
 タイミング :



BA:ベーシックアドレス
 C:ADL+Y のキャリ

付録 2 . 740 ファミリ機械語命令一覧表

項目 分類	命令記号	機能	フラグ		命令コード			バイト 数	サイクル 数	注記				
			N	V	T	B	D				I	Z	C	D ₇ D ₆ D ₅ D ₄
命令 送 受 ト ア 命 令 転 送 命 令 操 作 タ ク ソ ク 命 令	LDA # \$nn	(A) nn	x	x	x	x	x	x	1 0 1 0	1 0 0 1	A9	2	2	2
	LDA \$zz	(A) (M) ただし M = (zz)	x	x	x	x	x	x	1 0 1 0	0 1 0 1	A5	2	3	2
	LDA \$zz, X	(A) (M) ただし M = (zz + (X))	x	x	x	x	x	x	1 0 1 1	0 1 0 1	B5	2	4	2
	LDA \$hhll	(A) (M) ただし M = (hhll)	x	x	x	x	x	x	1 0 1 0	1 1 0 1	AD	3	4	2
	LDA \$hhll, X	(A) (M) ただし M = (hhll + (X))	x	x	x	x	x	x	1 0 1 1	1 1 0 1	BD	3	5	2
	LDA \$hhll, Y	(A) (M) ただし M = (hhll + (Y))	x	x	x	x	x	x	1 0 1 1	1 0 0 1	B9	3	5	2
	LDA (\$zz, X)	(A) (M) ただし M = ((zz + (X) + 1)(zz + (X)))	x	x	x	x	x	x	1 0 1 0	0 0 0 1	A1	2	6	2
	LDA (\$zz, Y)	(A) (M) ただし M = ((zz + 1)(zz) + (Y))	x	x	x	x	x	x	1 0 1 1	0 0 0 1	B1	2	6	2
	LDX # \$nn	(X) nn	x	x	x	x	x	x	1 0 1 0	0 0 1 0	A2	2	2	
	LDX \$zz	(X) (M) ただし M = (zz)	x	x	x	x	x	x	1 0 1 0	0 1 1 0	A6	2	3	
	LDX \$zz, Y	(X) (M) ただし M = (zz + (Y))	x	x	x	x	x	x	1 0 1 1	0 1 1 0	B6	2	4	
	LDX \$hhll	(X) (M) ただし M = (hhll)	x	x	x	x	x	x	1 0 1 0	1 1 1 0	AE	3	4	
	LDX \$hhll, Y	(X) (M) ただし M = (hhll + (Y))	x	x	x	x	x	x	1 0 1 1	1 1 1 0	BE	3	5	
	LDY # \$nn	(Y) nn	x	x	x	x	x	x	1 0 1 0	0 0 0 0	A0	2	2	
	LDY \$zz	(Y) (M) ただし M = (zz)	x	x	x	x	x	x	1 0 1 0	0 1 0 0	A4	2	3	
	LDY \$zz, X	(Y) (M) ただし M = (zz + (X))	x	x	x	x	x	x	1 0 1 1	0 1 0 0	B4	2	4	
	LDY \$hhll	(Y) (M) ただし M = (hhll)	x	x	x	x	x	x	1 0 1 0	1 1 0 0	AC	3	4	
	LDY \$hhll, X	(Y) (M) ただし M = (hhll + (X))	x	x	x	x	x	x	1 0 1 1	1 1 0 0	BC	3	5	
	LDM # \$nn, \$zz	(M) nn ただし M = (zz)	x	x	x	x	x	x	0 0 1 1	1 1 0 0	3C	3	4	
	STA \$zz	(M) (A) ただし M = (zz)	x	x	x	x	x	x	1 0 0 0	0 1 0 1	85	2	4	
	STA \$zz, X	(M) (A) ただし M = (zz + (X))	x	x	x	x	x	x	1 0 0 1	0 1 0 1	95	2	5	
	STA \$hhll	(M) (A) ただし M = (hhll)	x	x	x	x	x	x	1 0 0 0	1 1 0 1	8D	3	5	
	STA \$hhll, X	(M) (A) ただし M = (hhll + (X))	x	x	x	x	x	x	1 0 0 1	1 1 0 1	9D	3	6	
	STA \$hhll, Y	(M) (A) ただし M = (hhll + (Y))	x	x	x	x	x	x	1 0 0 1	1 0 0 1	99	3	6	
	STA (\$zz, X)	(M) (A) ただし M = ((zz + (X) + 1)(zz + (X)))	x	x	x	x	x	x	1 0 0 0	0 0 0 1	81	2	7	
STA (\$zz, Y)	(M) (A) ただし M = ((zz + 1)(zz) + (Y))	x	x	x	x	x	x	1 0 0 1	0 0 0 1	91	2	7		
STX \$zz	(M) (X) ただし M = (zz)	x	x	x	x	x	x	1 0 0 0	0 1 1 0	86	2	4		
STX \$zz, Y	(M) (X) ただし M = (zz + (Y))	x	x	x	x	x	x	1 0 0 1	0 1 1 0	96	2	5		
STX \$hhll	(M) (X) ただし M = (hhll)	x	x	x	x	x	x	1 0 0 0	1 1 1 0	8E	3	5		
STY \$zz	(M) (Y) ただし M = (zz)	x	x	x	x	x	x	1 0 0 0	0 1 0 0	84	2	4		
STY \$zz, X	(M) (Y) ただし M = (zz + (X))	x	x	x	x	x	x	1 0 0 1	0 1 0 0	94	2	5		
STY \$hhll	(M) (Y) ただし M = (hhll)	x	x	x	x	x	x	1 0 0 0	1 1 0 0	8C	3	5		
TAX	(X) (A)	x	x	x	x	x	x	1 0 1 0	1 0 1 0	AA	1	2		
TXA	(A) (X)	x	x	x	x	x	x	1 0 0 0	1 0 1 0	8A	1	2		
TAY	(Y) (A)	x	x	x	x	x	x	1 0 1 0	1 0 0 0	A8	1	2		
TYA	(A) (Y)	x	x	x	x	x	x	1 0 0 1	1 0 0 0	98	1	2		
TSX	(X) (S)	x	x	x	x	x	x	1 0 1 1	1 0 1 0	BA	1	2		
TXS	(S) (X)	x	x	x	x	x	x	1 0 0 1	1 0 1 0	9A	1	2		
PHA	(M(S)) (A), (S) (S) - 1	x	x	x	x	x	x	0 1 0 0	1 0 0 0	48	1	3		
PHP	(M(S)) (PS), (S) (S) - 1	x	x	x	x	x	x	0 0 0 0	1 0 0 0	08	1	3		
PLA	(S) (S) + 1, (A) (M(S))	x	x	x	x	x	x	0 1 1 0	1 0 0 0	68	1	4		
PLP	(S) (S) + 1, (PS) (M(S))	(スタックに退避していた値)						0 0 1 0	1 0 0 0	28	1	4		

740 ファミリ機械語命令一覧表

項目 分類	命令記号	機能	フラグ	命令コード	16進表記	バイト 数	サイクル 数	注記		
			N V T B D I Z C	D7D6D5D4 D3D2D1D0						
演算命令	ADC # \$nn	(A) (A) + nn + (C)	x x x x	0 1 1 0 < B2 >	1 0 0 1	69	2	2	1	
	ADC \$zz	(A) (A) + (M) + (C) ただし M = (zz)	x x x x	0 1 1 0 < B2 >	0 1 0 1	65	2	3	1	
	ADC \$zz, X	(A) (A) + (M) + (C) ただし M = (zz + (X))	x x x x	0 1 1 1 < B2 >	0 1 0 1	75	2	4	1	
	ADC \$hhll	(A) (A) + (M) + (C) ただし M = (hhll)	x x x x	0 1 1 0 < B2 >	1 1 0 1	6D	3	4	1	
	ADC \$hhll, X	(A) (A) + (M) + (C) ただし M = (hhll + (X))	x x x x	0 1 1 1 < B2 >	1 1 0 1	7D	3	5	1	
	ADC \$hhll Y	(A) (A) + (M) + (C) ただし M = (hhll + (Y))	x x x x	0 1 1 1 < B2 >	1 0 0 1	79	3	5	1	
	ADC (\$zz, X)	(A) (A) + (M) + (C) ただし M = ((zz + (X) + 1)(zz + (X)))	x x x x	0 1 1 0 < B2 >	0 0 0 1	61	2	6	1	
	ADC (\$zz, Y)	(A) (A) + (M) + (C) ただし M = ((zz + 1)(zz) + (Y))	x x x x	0 1 1 1 < B2 >	0 0 0 1	71	2	6	1	
	減算命令	SBC # \$nn	(A) (A) - nn - (C)	x x x x	1 1 1 0 < B2 >	1 0 0 1	E9	2	2	1
		SBC \$zz	(A) (A) - (M) - (C) ただし M = (zz)	x x x x	1 1 1 0 < B2 >	0 1 0 1	E5	2	3	1
		SBC \$zz, X	(A) (A) - (M) - (C) ただし M = (zz + (X))	x x x x	1 1 1 1 < B2 >	0 1 0 1	F5	2	4	1
		SBC \$hhll	(A) (A) - (M) - (C) ただし M = (hhll)	x x x x	1 1 1 0 < B2 >	1 1 0 1	ED	3	4	1
		SBC \$hhll, X	(A) (A) - (M) - (C) ただし M = (hhll + (X))	x x x x	1 1 1 1 < B2 >	1 1 0 1	FD	3	5	1
		SBC \$hhll, Y	(A) (A) - (M) - (C) ただし M = (hhll + (Y))	x x x x	1 1 1 1 < B2 >	1 0 0 1	F9	3	5	1
		SBC (\$zz, X)	(A) (A) - (M) - (C) ただし M = ((zz + (X) + 1)(zz + (X)))	x x x x	1 1 1 0 < B2 >	0 0 0 1	E1	2	6	1
		SBC (\$zz, Y)	(A) (A) - (M) - (C) ただし M = ((zz + 1)(zz) + (Y))	x x x x	1 1 1 1 < B2 >	0 0 0 1	F1	2	6	1
	命令	INC A	(A) (A) + 1	x x x x x x	0 0 1 1	1 0 1 0	3A	1	2	
		INC \$zz	(M) (M) + 1 ただし M = (zz)	x x x x x x	1 1 1 0 < B2 >	0 1 1 0	E6	2	5	
		INC \$zz, X	(M) (M) + 1 ただし M = (zz + (X))	x x x x x x	1 1 1 1 < B2 >	0 1 1 0	F6	2	6	
		INC \$hhll	(M) (M) + 1 ただし M = (hhll)	x x x x x x	1 1 1 0 < B2 >	1 1 1 0	EE	3	6	
		INC \$hhll, X	(M) (M) + 1 ただし M = (hhll + (X))	x x x x x x	1 1 1 1 < B2 >	1 1 1 0	FE	3	7	
		命令	DEC A	(A) (A) - 1	x x x x x x	0 0 0 1	1 0 1 0	1A	1	2
	DEC \$zz		(M) (M) - 1 ただし M = (zz)	x x x x x x	1 1 0 0 < B2 >	0 1 1 0	C6	2	5	
	DEC \$zz, X		(M) (M) - 1 ただし M = (zz + (X))	x x x x x x	1 1 0 1 < B2 >	0 1 1 0	D6	2	6	
	DEC \$hhll		(M) (M) - 1 ただし M = (hhll)	x x x x x x	1 1 0 0 < B2 >	1 1 1 0	CE	3	6	
	DEC \$hhll, X		(M) (M) - 1 ただし M = (hhll + (X))	x x x x x x	1 1 0 1 < B2 >	1 1 1 0	DE	3	7	
	INX		(X) (X) + 1	x x x x x x	1 1 1 0	1 0 0 0	E8	1	2	
	DEX		(X) (X) - 1	x x x x x x	1 1 0 0	1 0 1 0	CA	1	2	
INY	(Y) (Y) + 1		x x x x x x	1 1 0 0	1 0 0 0	C8	1	2		
DEY	(Y) (Y) - 1		x x x x x x	1 0 0 0	1 0 0 0	88	1	2		
乗除算命令	MUL \$zz, X		M(S), (A) (A) × M(zz + (X)) (S) (S) - 1	x x x x x x x x	0 1 1 0	0 0 1 0	62	2	15	6
	DIV \$zz, X	(A) (M(zz + (X) + 1), M(zz + (X))) ÷ (A) M(S) 剰余の1の補数 (S) (S) - 1	x x x x x x x x	1 1 1 0	0 0 1 0	E2	2	16	6	

740 ファミリ機械語命令一覧表

項目 分類	命令記号	機能	フラグ	命令コード			バイト 数	サイクル 数	注記
			N V T B D I Z C	D7D6D5D4	D3D2D1D0	16進表記			
演算命令	AND # \$nn	(A) (A) nn	x x x x x	0 0 1 0	1 0 0 1	29	2	2	1
	AND \$zz	(A) (A) (M) ただし M = (zz)	x x x x x	0 0 1 0	0 1 0 1	25	2	3	1
	AND \$zz, X	(A) (A) (M) ただし M = (zz + (X))	x x x x x	0 0 1 1	0 1 0 1	35	2	4	1
	AND \$hhll	(A) (A) (M) ただし M = (hhll)	x x x x x	0 0 1 0	1 1 0 1	2D	3	4	1
	AND \$hhll, X	(A) (A) (M) ただし M = (hhll + (X))	x x x x x	0 0 1 1	1 1 0 1	3D	3	5	1
	AND \$hhll, Y	(A) (A) (M) ただし M = (hhll + (Y))	x x x x x	0 0 1 1	1 0 0 1	39	3	5	1
	AND (\$zz, X)	(A) (A) (M) ただし M = ((zz + (X) + 1)(zz + (X)))	x x x x x	0 0 1 0	0 0 0 1	21	2	6	1
	AND (\$zz, Y)	(A) (A) (M) ただし M = ((zz + 1)(zz) + (Y))	x x x x x	0 0 1 1	0 0 0 1	31	2	6	1
	ORA # \$nn	(A) (A) nn	x x x x x	0 0 0 0	1 0 0 1	09	2	2	1
	ORA \$zz	(A) (A) (M) ただし M = (zz)	x x x x x	0 0 0 0	0 1 0 1	05	2	3	1
	ORA \$zz, X	(A) (A) (M) ただし M = (zz + (X))	x x x x x	0 0 0 1	0 1 0 1	15	2	4	1
	ORA \$hhll	(A) (A) (M) ただし M = (hhll)	x x x x x	0 0 0 0	1 1 0 1	0D	3	4	1
	ORA \$hhll, X	(A) (A) (M) ただし M = (hhll + (X))	x x x x x	0 0 0 1	1 1 0 1	1D	3	5	1
	ORA \$hhll, Y	(A) (A) (M) ただし M = (hhll + (Y))	x x x x x	0 0 0 1	1 0 0 1	19	3	5	1
	ORA (\$zz, X)	(A) (A) (M) ただし M = ((zz + (X) + 1)(zz + (X)))	x x x x x	0 0 0 0	0 0 0 1	01	2	6	1
	ORA (\$zz, Y)	(A) (A) (M) ただし M = ((zz + 1)(zz) + (Y))	x x x x x	0 0 0 1	0 0 0 1	11	2	6	1
	EOR # \$nn	(A) (A) nn	x x x x x	0 1 0 0	1 0 0 1	49	2	2	1
	EOR \$zz	(A) (A) (M) ただし M = (zz)	x x x x x	0 1 0 0	0 1 0 1	45	2	3	1
	EOR \$zz, X	(A) (A) (M) ただし M = (zz + (X))	x x x x x	0 1 0 1	0 1 0 1	55	2	4	1
	EOR \$hhll	(A) (A) (M) ただし M = (hhll)	x x x x x	0 1 0 0	1 1 0 1	4D	3	4	1
	EOR \$hhll, X	(A) (A) (M) ただし M = (hhll + (X))	x x x x x	0 1 0 1	1 1 0 1	5D	3	5	1
	EOR \$hhll, Y	(A) (A) (M) ただし M = (hhll + (Y))	x x x x x	0 1 0 1	1 0 0 1	59	3	5	1
	EOR (\$zz, X)	(A) (A) (M) ただし M = ((zz + (X) + 1)(zz + (X)))	x x x x x	0 1 0 0	0 0 0 1	41	2	6	1
	EOR (\$zz, Y)	(A) (A) (M) ただし M = ((zz + 1)(zz) + (Y))	x x x x x	0 1 0 1	0 0 0 1	51	2	6	1
	COM \$zz	(M) (M) ただし M = (zz)	x x x x x	0 1 0 0	0 1 0 0	44	2	5	
	BIT \$zz	(A) (M) ただし M = (zz)	M7M6 x x x x	0 0 1 0	0 1 0 0	24	2	3	
	BIT \$hhll	(A) (M) ただし M = (hhll)	M7M6 x x x x	0 0 1 0	1 1 0 0	2C	3	4	
	TST \$zz	(M) = 0? ただし M = (zz)	x x x x x	0 1 1 0	0 1 0 0	64	2	3	
比較命令	CMP # \$nn	(A) - nn	x x x x x	1 1 0 0	1 0 0 1	C9	2	2	3
	CMP \$zz	(A) - (M) 大きさの比較	x x x x x	1 1 0 0	0 1 0 1	C5	2	3	3
	CMP \$zz, X	(A) - (M) 大きさの比較	x x x x x	1 1 0 1	0 1 0 1	D5	2	4	3
	CMP \$hhll	(A) - (M) 大きさの比較	x x x x x	1 1 0 0	1 1 0 1	CD	3	4	3
	CMP \$hhll, X	(A) - (M) 大きさの比較	x x x x x	1 1 0 1	1 1 0 1	DD	3	5	3
	CMP \$hhll, Y	(A) - (M) 大きさの比較	x x x x x	1 1 0 1	1 0 0 1	D9	3	5	3
	CMP (\$zz, X)	(A) - (M) 大きさの比較	x x x x x	1 1 0 0	0 0 0 1	C1	2	6	3
	CMP (\$zz, Y)	(A) - (M) 大きさの比較	x x x x x	1 1 0 1	0 0 0 1	D1	2	6	3
	CPX # \$nn	(X) - nn	x x x x x	1 1 1 0	0 0 0 0	E0	2	2	
	CPX \$zz	(X) - (M) 大きさの比較	x x x x x	1 1 1 0	0 1 0 0	E4	2	3	
	CPX \$hhll	(X) - (M) 大きさの比較	x x x x x	1 1 1 0	1 1 0 0	EC	3	4	
	CPY # \$nn	(Y) - nn	x x x x x	1 1 0 0	0 0 0 0	C0	2	2	
CPY \$zz	(Y) - (M) 大きさの比較	x x x x x	1 1 0 0	0 1 0 0	C4	2	3		
CPY \$hhll	(Y) - (M) 大きさの比較	x x x x x	1 1 0 0	1 1 0 0	CC	3	4		

740 ファミリ機械語命令一覧表

項目 分類	命令記号	機能	フラグ	命令コード			バイト 数	サイクル 数	注記	
			N V T B D I Z C	D7D6D5D4	D3D2D1D0	16進表記				
演算命令	ASL A	左シフト C $\overline{\text{A7A6}} \text{A1A0}$ 0 ただし M = (zz)	x x x x x	0 0 0 0	1 0 1 0	0A	1	2		
	ASL \$zz		x x x x x	0 0 0 0	0 1 1 0	06	2	5		
	ASL \$zz, X		ただし M = (zz + (X))	x x x x x	0 0 0 1	0 1 1 0	16	2	6	
	ASL \$hhll		左シフト C $\overline{\text{M7M6}} \text{M1M0}$ 0 0 0 0 ただし M = (hhll)	x x x x x	0 0 0 0	1 1 1 0	0E	3	6	
	ASL \$hhll, X	ただし M = (hhll + (X))		x x x x x	0 0 0 1	1 1 1 0	1E	3	7	
	回転命令	LSR A	右シフト 0 $\overline{\text{A7A6}} \text{A1A0}$ C ただし M = (zz)	0 x x x x	0 1 0 0	1 0 1 0	4A	1	2	
		LSR \$zz		0 x x x x	0 1 0 0	0 1 1 0	46	2	5	
		LSR \$zz, X	ただし M = (zz + (X))	0 x x x x	0 1 0 1	0 1 1 0	56	2	6	
		LSR \$hhll	右シフト 0 $\overline{\text{M7M6}} \text{M1M0}$ C ただし M = (hhll)	0 x x x x	0 1 0 0	1 1 1 0	4E	3	6	
	LSR \$hhll, X	ただし M = (hhll + (X))		0 x x x x	0 1 0 1	1 1 1 0	5E	3	7	
	ビット命令	ROL A	左シフト $\overline{\text{A7A6}} \text{A1A0}$ C ただし M = (zz)	x x x x x	0 0 1 0	1 0 1 0	2A	1	2	
		ROL \$zz		x x x x x	0 0 1 0	0 1 1 0	26	2	5	
		ROL \$zz, X	ただし M = (zz + (X))	x x x x x	0 0 1 1	0 1 1 0	36	2	6	
		ROL \$hhll	左シフト $\overline{\text{M7M6}} \text{M1M0}$ C ただし M = (hhll)	x x x x x	0 0 1 0	1 1 1 0	2E	3	6	
		ROL \$hhll, X		ただし M = (hhll + (X))	x x x x x	0 0 1 1	1 1 1 0	3E	3	7
	ビット処理	ROR A	右シフト C $\overline{\text{A7A6}} \text{A1A0}$ ただし M = (zz)	x x x x x	0 1 1 0	1 0 1 0	6A	1	2	
		ROR \$zz		x x x x x	0 1 1 0	0 1 1 0	66	2	5	
		ROR \$zz, X	ただし M = (zz + (X))	x x x x x	0 1 1 1	0 1 1 0	76	2	6	
ROR \$hhll		右シフト C $\overline{\text{M7M6}} \text{M1M0}$ ただし M = (hhll)	x x x x x	0 1 1 0	1 1 1 0	6E	3	6		
ROR \$hhll, X			ただし M = (hhll + (X))	x x x x x	0 1 1 1	1 1 1 0	7E	3	7	
RRF \$zz	$\overline{\text{M7}} \text{M4} \text{M3} \text{M0}$ 0 0 0 0 ただし M = (zz)	x x x x x x x x	1 0 0 0	0 0 1 0	82	2	8			
フラグ設定命令	CLB i, A	(Ai) 0 0 0 0 ただし i = 0~7	x x x x x x x x	i i i 1	1 0 1 1	(1+2i) × 10 + B	1	2		
	CLB i, \$zz	(Mi) 0 0 0 0 ただし i = 0~7, M = (zz)	x x x x x x x x	i i i 1	1 1 1 1	(1+2i) × 10 + F	2	5		
	SEB i, A	(Ai) 1 0 0 0 ただし i = 0~7	x x x x x x x x	i i i 0	1 0 1 1	2i × 10 + B	1	2		
	SEB i, \$zz	(Mi) 1 0 0 0 ただし i = 0~7, M = (zz)	x x x x x x x x	i i i 0	1 1 1 1	2i × 10 + F	2	5		
フラグ設定命令	CLC	(C) 0	x x x x x x x 0	0 0 0 1	1 0 0 0	18	1	2		
	SEC	(C) 1	x x x x x x x 1	0 0 1 1	1 0 0 0	38	1	2		
	CLD	(D) 0	x x x x 0 x x x	1 1 0 1	1 0 0 0	D8	1	2		
	SED	(D) 1	x x x x 1 x x x	1 1 1 1	1 0 0 0	F8	1	2		
	CLI	(I) 0	x x x x 0 x x x	0 1 0 1	1 0 0 0	58	1	2		
	SEI	(I) 1	x x x x 1 x x x	0 1 1 1	1 0 0 0	78	1	2		
	CLT	(T) 0	x x 0 x x x x x	0 0 0 1	0 0 1 0	12	1	2		
	SET	(T) 1	x x 1 x x x x x	0 0 1 1	0 0 1 0	32	1	2		
CLV	(V) 0	x 0 x x x x x x	1 0 1 1	1 0 0 0	B8	1	2			

740 ファミリ機械語命令一覧表

項目 分類	命令記号	機能	フラグ	命令コード		バイト 数	サイクル 数	注記	
			N V T B D I Z C	D7D6D5D4 D3D2D1D0	16進表記				
分岐・リターン命令	BRA \$hhll	(PC) (PC)+2+Rel	xxxxxxx	1000	0000	80	2	4	
	JMP \$hhll	(PC) hhll	xxxxxxx	0100	1100	4C	3	3	
	JMP (\$hhll)	(PCL) (hhll), (PCH) (hhll+1)	xxxxxxx	0110	1100	6C	3	5	
	JMP (\$zz)	(PCL) (zz), (PCH) (zz+1)	xxxxxxx	1011	0010	B2	2	4	
	JSR \$hhll	(M(S)) (PCH), (S) (S)-1, (M(S)) (PCL), (S) (S)-1を順に行い、次に(PC) hhll	xxxxxxx	0010	0000	20	3	6	
	JSR (\$zz)	(M(S)) (PCH), (S) (S)-1, (M(S)) (PCL), (S) (S)-1を順に行い、次に(PCL) (zz), (PCH) (zz+1)	xxxxxxx	0000	0010	02	2	7	
	JSR ¥\$hhll	(M(S)) (PCH), (S) (S)-1, (M(S)) (PCL), (S) (S)-1を順に行い、次に(PCL) ll(アドレス下位バイト), (PCH) FF	xxxxxxx	0010	0010	22	2	5	
	BBC i, A, \$hhll	(Ai)=0のとき (PC) (PC)+2+Rel ただし i=0-7 (Ai)=1のとき (PC) (PC)+2	xxxxxxx	iii1	0011	(l+2i)×10+3	2	4	4
	BBC i, \$zz, \$hhll	(Mi)=0のとき (PC) (PC)+3+Rel ただし i=0-7 (Mi)=1のとき (PC) (PC)+3	xxxxxxx	iii1	0111	(l+2i)×10+7	3	5	4
	BBS i, A, \$hhll	(Ai)=1のとき (PC) (PC)+2+Rel ただし i=0-7 (Ai)=0のとき (PC) (PC)+2	xxxxxxx	iii0	0011	2i×10+3	2	4	4
BBS i, \$zz, \$hhll	(Mi)=1のとき (PC) (PC)+3+Rel ただし i=0-7 (Mi)=0のとき (PC) (PC)+3	xxxxxxx	iii0	0111	2i×10+7	3	5	4	
BCC \$hhll	(C)=0のとき (PC) (PC)+2+Rel (C)=1のとき (PC) (PC)+2	xxxxxxx	1001	0000	90	2	2	4	
BCS \$hhll	(C)=1のとき (PC) (PC)+2+Rel (C)=0のとき (PC) (PC)+2	xxxxxxx	1011	0000	B0	2	2	4	
BNE \$hhll	(Z)=0のとき (PC) (PC)+2+Rel (Z)=1のとき (PC) (PC)+2	xxxxxxx	1101	0000	D0	2	2	4	
BEQ \$hhll	(Z)=1のとき (PC) (PC)+2+Rel (Z)=0のとき (PC) (PC)+2	xxxxxxx	1111	0000	F0	2	2	4	
BPL \$hhll	(N)=0のとき (PC) (PC)+2+Rel (N)=1のとき (PC) (PC)+2	xxxxxxx	0001	0000	10	2	2	4	
BMI \$hhll	(N)=1のとき (PC) (PC)+2+Rel (N)=0のとき (PC) (PC)+2	xxxxxxx	0011	0000	30	2	2	4	
BVC \$hhll	(V)=0のとき (PC) (PC)+2+Rel (V)=1のとき (PC) (PC)+2	xxxxxxx	0101	0000	50	2	2	4	
BVS \$hhll	(V)=1のとき (PC) (PC)+2+Rel (V)=0のとき (PC) (PC)+2	xxxxxxx	0111	0000	70	2	2	4	
リターン	RTI	(S) (S)+1, (PS) (M(S)), (S) (S)+1, (PCL) (M(S)), (S) (S)+1, (PCH) (M(S))	(スタックに退避していた値)	0100	0000	40	1	6	
RTS	(S) (S)+1, (PCL) (M(S)), (S) (S)+1, (PCH) (M(S)), (PC) (PC)+1	xxxxxxx	0110	0000	60	1	6		
割り込み	BRK	(B) 1, (PC) (PC)+2, (M(S)) (PCH), (S) (S)-1, (M(S)) (PCL), (S) (S)-1, (M(S)) (PS), (S) (S)-1, (I) 1, (PC) BADRS	xxx1x1xx	0000	0000	00	1	7	
その他	NOP	(PC) (PC)+1	xxxxxxx	1110	1010	EA	1	2	
特殊	WIT	内部クロックが停止する。	xxxxxxx	1100	0010	C2	1	2	
STP	発振が停止する。	xxxxxxx	0100	0010	42	1	2		

記号	内容	記号	内容
A	アキュムレータ	hh	アドレス上位バイト 0-255のデータ
Ai	アキュムレータのビット i	ll	アドレス下位バイト 0-255のデータ
X	インデックスレジスタX	zz	ゼロページアドレス 0-255のデータ
Y	インデックスレジスタY	nn	0-255のデータ
M	メモリ	i	0-7のデータ
Mi	メモリのビットi	iii	0-7のデータ
PS	プロセッサステータスレジスタ	<B2>	命令の第2バイト
S	スタックポインタ	<B3>	命令の第3バイト
PC	プログラムカウンタ	Rel	相対アドレス
PCL	プログラムカウンタの下位バイト	BADRS	ブレイクアドレス
PCH	プログラムカウンタの上位バイト		データの移動する方向を示す
N	ネガティブフラグ	()	レジスタ、メモリなどの内容を示す
V	オーバーフローフラグ	+	加算
T	X修飾演算モードフラグ	-	減算
B	ブレイクフラグ	x	乗算
D	10進モードフラグ	÷	除算
I	割り込み禁止フラグ		論理和
Z	ゼロフラグ		論理積
C	キャリフラグ		排他的論理和
#	イミディエイトモード	—	否定
\$	16進表記	x	命令実行後フラグが不変
¥	スペシャルページモード		命令実行後フラグが変化

- 注1. 表に書いてある機能は(T)=0のときのものです。
(T)=1のときは、(A)の代わりに(M(X))が入り、サイクル数を+3します。
2. 同上。ただし、サイクル数は+2します。
3. 同上。ただし、サイクル数は+1します。
4. 分岐が生じたとき、サイクル数を+2します。
5. 品種により STP 命令を無効にできます。STP 命令が無効の場合、NOP 命令の2回連続実行と同一の動作でサイクル数は2サイクルとなります。
6. 一部の品種では、乗除算命令が使用できません。

付録 3 . 740 ファミリ命令コード対応表

D7 - D4 \ D3 - D0	Hexadecimal notation																
	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	ORA IND, X	JSR ZP, IND	BBS 0, A	—	ORA ZP	ASL ZP	BBS 0, ZP	PHP	ORA IMM	ASL A	SEB 0, A	—	ORA ABS	ASL ABS	SEB 0, ZP
0001	1	BPL	ORA IND, Y	CLT	BBC 0, A	—	ORA ZP, X	ASL ZP, X	BBC 0, ZP	CLC	ORA ABS, Y	DEC A	CLB 0, A	—	ORA ABS, X	ASL ABS, X	CLB 0, ZP
0010	2	JSR ABS	AND IND, X	JSR SP	BBS 1, A	BIT ZP	AND ZP	ROL ZP	BBS 1, ZP	PLP	AND IMM	ROL A	SEB 1, A	BIT ABS	AND ABS	ROL ABS	SEB 1, ZP
0011	3	BMI	AND IND, Y	SET	BBC 1, A	—	AND ZP, X	ROL ZP, X	BBC 1, ZP	SEC	AND ABS, Y	INC A	CLB 1, A	LDM ZP	AND ABS, X	ROL ABS, X	CLB 1, ZP
0100	4	RTI	EOR IND, X	STP (注)	BBS 2, A	COM ZP	EOR ZP	LSR ZP	BBS 2, ZP	PHA	EOR IMM	LSR A	SEB 2, A	JMP ABS	EOR ABS	LSR ABS	SEB 2, ZP
0101	5	BVC	EOR IND, Y	—	BBC 2, A	—	EOR ZP, X	LSR ZP, X	BBC 2, ZP	CLI	EOR ABS, Y	—	CLB 2, A	—	EOR ABS, X	LSR ABS, X	CLB 2, ZP
0110	6	RTS	ADC IND, X	MUL ZP, X (注)	BBS 3, A	TST ZP	ADC ZP	ROR ZP	BBS 3, ZP	PLA	ADC IMM	ROR A	SEB 3, A	JMP IND	ADC ABS	ROR ABS	SEB 3, ZP
0111	7	BVS	ADC IND, Y	—	BBC 3, A	—	ADC ZP, X	ROR ZP, X	BBC 3, ZP	SEI	ADC ABS, Y	—	CLB 3, A	—	ADC ABS, X	ROR ABS, X	CLB 3, ZP
1000	8	BRA	STA IND, X	RRF ZP	BBS 4, A	STY ZP	STA ZP	STX ZP	BBS 4, ZP	DEY	—	TXA	SEB 4, A	STY ABS	STA ABS	STX ABS	SEB 4, ZP
1001	9	BCC	STA IND, Y	—	BBC 4, A	STY ZP, X	STA ZP, X	STX ZP, Y	BBC 4, ZP	TYA	STA ABS, Y	TXS	CLB 4, A	—	STA ABS, X	—	CLB 4, ZP
1010	A	LDY IMM	LDA IND, X	LDX IMM	BBS 5, A	LDY ZP	LDA ZP	LDX ZP	BBS 5, ZP	TAY	LDA IMM	TAX	SEB 5, A	LDY ABS	LDA ABS	LDX ABS	SEB 5, ZP
1011	B	BCS	LDA IND, Y	JMP ZP, IND	BBC 5, A	LDY ZP, X	LDA ZP, X	LDX ZP, Y	BBC 5, ZP	CLV	LDA ABS, Y	TSX	CLB 5, A	LDY ABS, X	LDA ABS, X	LDX ABS, Y	CLB 5, ZP
1100	C	CPY IMM	CMP IND, X	WIT	BBS 6, A	CPY ZP	CMP ZP	DEC ZP	BBS 6, ZP	INY	CMP IMM	DEX	SEB 6, A	CPY ABS	CMP ABS	DEC ABS	SEB 6, ZP
1101	D	BNE	CMP IND, Y	—	BBC 6, A	—	CMP ZP, X	DEC ZP, X	BBC 6, ZP	CLD	CMP ABS, Y	—	CLB 6, A	—	CMP ABS, X	DEC ABS, X	CLB 6, ZP
1110	E	CPX IMM	SBC IND, X	DIV ZP, X (注)	BBS 7, A	CPX ZP	SBC ZP	INC ZP	BBS 7, ZP	INX	SBC IMM	NOP	SEB 7, A	CPX ABS	SBC ABS	INC ABS	SEB 7, ZP
1111	F	BEQ	SBC IND, Y	—	BBC 7, A	—	SBC ZP, X	INC ZP, X	BBC 7, ZP	SED	SBC ABS, Y	—	CLB 7, A	—	SBC ABS, X	INC ABS, X	CLB 7, ZP

注 . これらの命令の有無は品種により異なります。

- 3バイト命令
- 2バイト命令
- 1バイト命令

クロック制御命令及び乗除算命令は品種に依存しますので、各品種の機能説明を参照してください。

740ファミリソフトウェアマニュアル

発行年月日 2001年1月6日 Rev. 1.00
2006年8月1日 Rev. 2.00

発行 株式会社 ルネサス テクノロジ 営業企画統括部
〒100-0004 東京都千代田区大手町2-6-2

© 2006. Renesas Technology Corp., All rights reserved. Printed in Japan.

740 ファミリ
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

RJ09B0352-0200