

1. 概要

KL5C16005 は、弊社高速 16 ビット CPU コア「KC160」を搭載したマイコンです。ザイログ社 Z80 CPU 及び弊社 KC80 とオブジェクトレベルで上位互換であり、豊富なソフトウェア資産を変更無しに使用することも可能です。Z80 の全命令に加え、更に各種処理命令を追加することにより、性能的には弊社 KC80 の約 4 倍、つまり Z80 に換算すると 100MHz 以上に相当し、32 ビットマイコンに迫る性能を発揮します。アドレス空間は、16M バイトリニアアドレスをサポートしており、大規模なプログラム及びデータ処理にも対応可能となっています。また、KC80 マイコンシリーズ同様、CMOS 回路と同期バスの採用により低消費電力を実現しています。

KL5C16005 は、その他豊富な周辺機能を搭載しています。汎用タイマ、シリアル I/F、パラレルポート、割り込みコントローラなどを搭載しており、充実した周辺機能の構成となっています。よって、KL5C16005 を使用することにより、コストパフォーマンスに優れたシステムの構築が可能となります。

尚、本チップをベースとしてお客様専用のカスタムマイコンを開発することも可能です。この場合、弊社既存のマクロセルのみならずユーザ回路の搭載も可能です。

特長

CPU コア	: KC160 (高速 16 ビット CPU コア)
ソフトウェア	: ザイログ社製 Z80 とバイナリレベルで上位互換に加え、 16 ビット処理命令を追加
最大クロック周波数	: 16/20MHz
最小命令実行時間	: 62.5/50ns
比較処理能力	: 弊社 KC80 に対し約 4 倍、Z80 換算で 100MHz 相当以上の高性能
バス仕様	: データバス 8/16 ビット切替え可能、命令フェッチ・1クロック、 メモリアクセス・2クロック、16M バイトリニアアドレス空間
シリアル I/F	: 調歩同期式 (1ch.: 全二重、ボーレートジェネレータ内蔵)
パラレルポート	: 32 ビット (1 ビット単位入出力方向設定、ビットセットリセット機能付き)
割り込み	: マスク可能な内外割り込み 8 要因 + NMI、多重割り込み可能
タイマ	: 汎用タイマ (1ch.: イベント、分周、ワンショット、PWM、WDT モード対応)
電源電圧	: 5V 単一電源 (3.3V 動作可能)
パッケージ	: QFP100

【注意】

本文中で記載されております接続例等の回路図は、あくまでもデバイス間の端子接続を示す例としての位置付けのもとに記載しております。よって、それらの図の通りの回路設計をした場合、正常に動作しない場合がありますのでご注意ください。

もし、基板上における端子の処理、またその他の記載内容等ご不明な点がございましたら弊社までご相談ください。

ブロック図

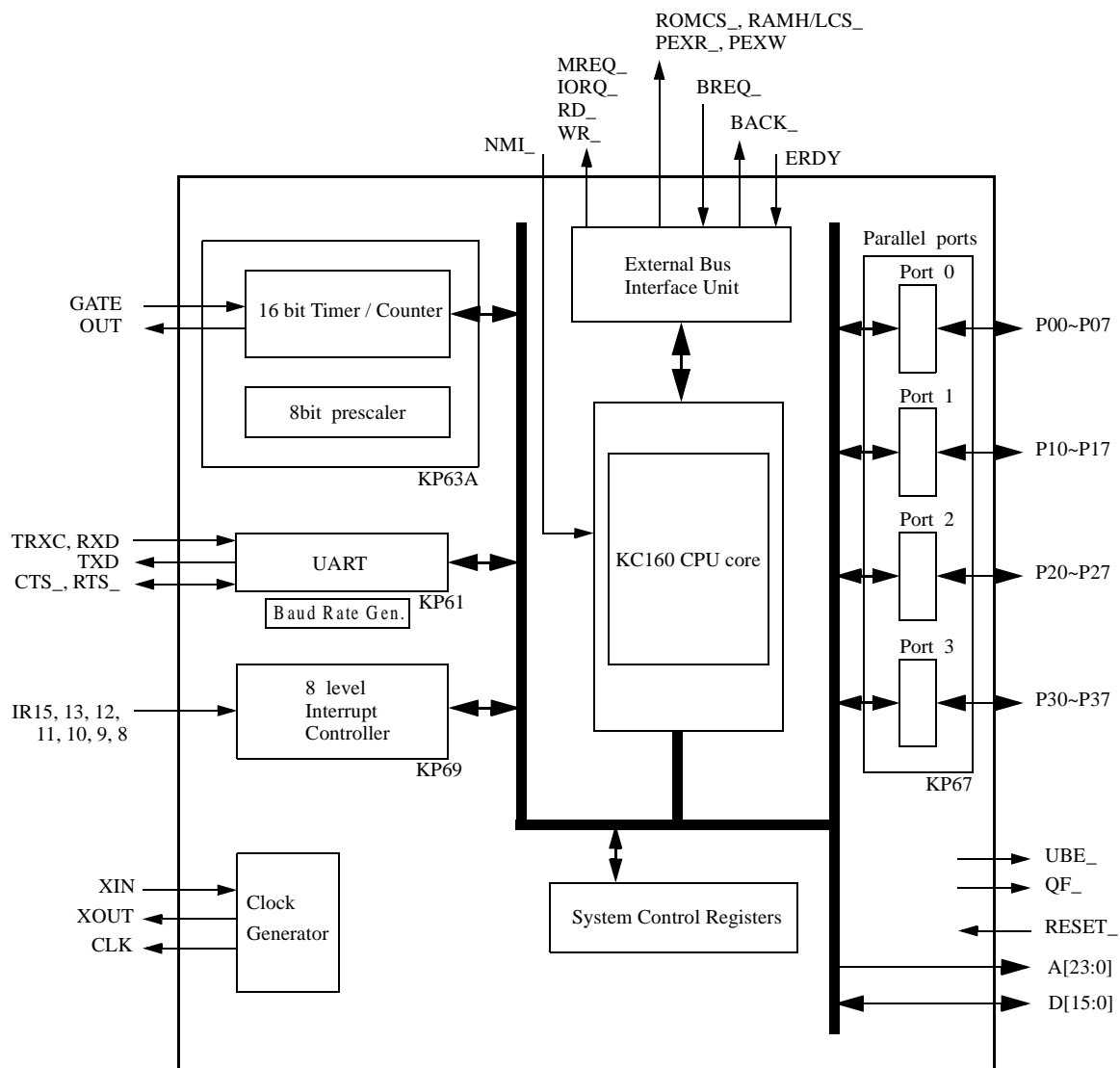


図 1-1 ブロック図

2 . 端子説明

端子配置図及び端子説明表を図 2-1 及び表 2-1a, 2-1b に示します。

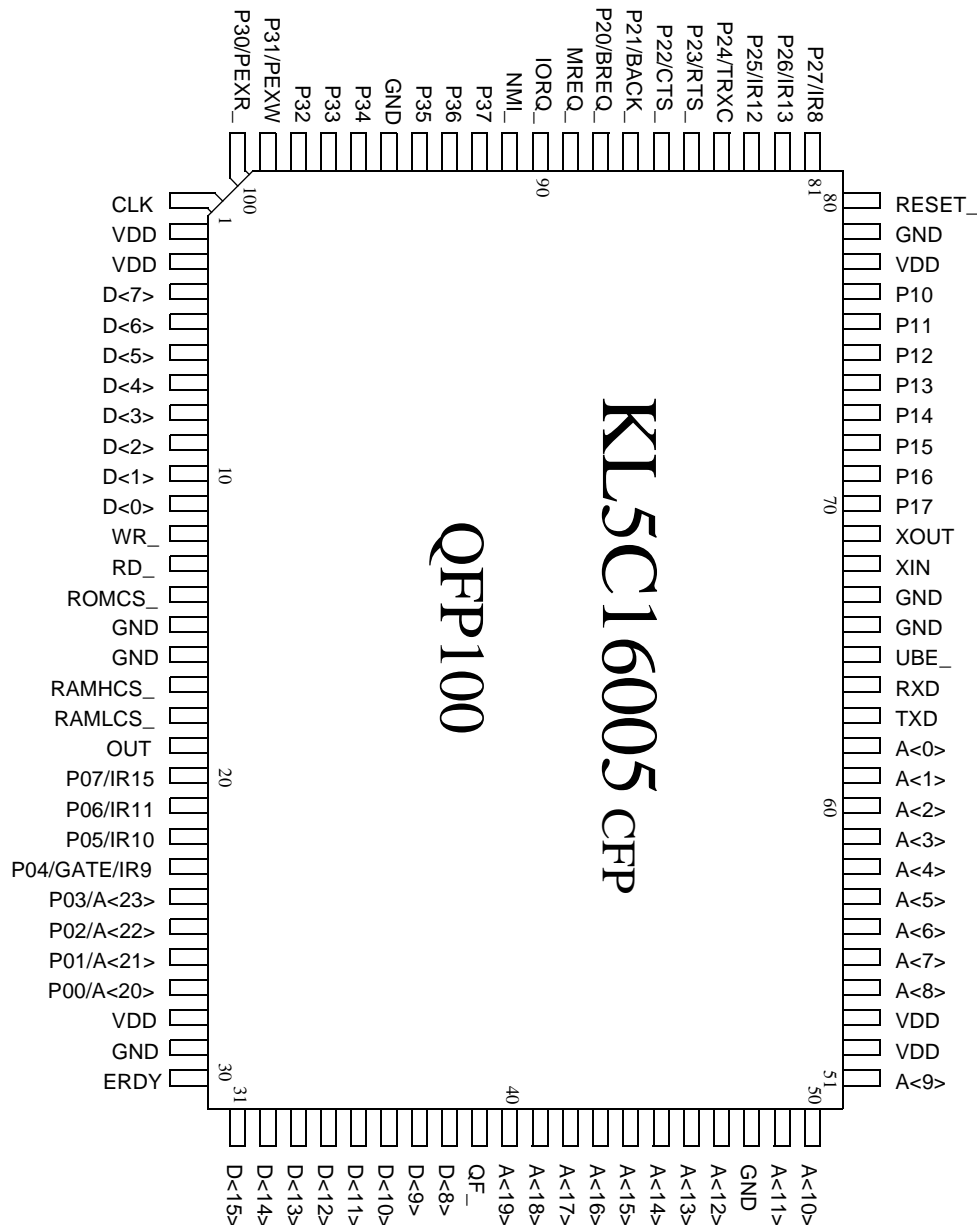


図 2-1 ピン配置図 (上面図)

表 2-1a 端子説明

端子名	I/O	機能説明
VDD	電源	電源を接続してください。
GND	電源	0V を接続してください。
RESET_	I	リセット入力。アクティブ “L”。
CLK	O	クロック出力。 内部システムクロックを出力します。
A<23:0>	O	アドレスバス。
D<15:0>	I/O	データバス。
RD_	O	リード信号。アクティブ “L”。
WR_	O	ライト信号。アクティブ “L”。
MREQ_	O	外部メモリ要求信号。アクティブ “L”。
IORQ_	O	外部 I/O デバイス要求信号。アクティブ “L”。
ERDY	I	ウェイト要求信号。“L” でウェイトを挿入します。
ROMCS_	O	ROM 領域チップセレクト信号。アクティブ “L”。 “000000H ~ 7FFFFFFH” の領域をアクセスすると “L” になります。
RAMLCS_	O	SRAM 領域下位バイト側チップセレクト信号。アクティブ “L”。 主に “800000H ~ FFFFFFFH” の SRAM 領域の下位(偶数)バイト側をアクセスすると “L” になります。
RAMHCS_	O	SRAM 領域上位バイト側チップセレクト信号。アクティブ “L”。 主に “800000H ~ FFFFFFFH” の SRAM 領域の上位(奇数)バイト側をアクセスすると “L” になります。
UBE_	O	上位バイトイネーブル信号。アクティブ “L”。 メモリの上位(奇数)バイト側をアクセスすると “L” になります。
QF_	I/O	キューフラッシュ信号。アクティブ “L”。 分岐(ジャンプ) 命令実行時に “L” になります。 尚、この端子は VDD 側に抵抗(50K ~ 100K 程度) を介してプルアップ処理を行ってください。

表 2-1b 端子説明

端子名	I/O	機能説明
XIN	I	水晶発振子用端子。
XOUT	O	水晶発振子用端子。
BREQ_	I	バスリクエスト信号。アクティブ“L”。
BACK_	O	バスアクノリッジ信号。アクティブ“L”。
IR12, 10, 9, 8	I	外部割り込み入力信号。アクティブ“H”。
IR15, 13, 11	I	外部割り込み入力信号。アクティブ“H”/ アクティブ“L”(切り替え可能)。
TXD	O	UART 送信データ出力。
RXD	I	UART 受信データ入力。
TRXC	I	UART 送受信クロック入力。
CTS_	I	UART クリアトウセンド信号。
RTS_	O	UART リクエストトウセンド信号。
OUT	O	汎用タイマ OUT 出力。
GATE	I	汎用タイマ GATE 入力。
P03 ~ 00	O	パラレルポート 0 出力。
P07 ~ 04	I	パラレルポート 0 入力。
P17 ~ 10	I/O	パラレルポート 1 入出力。
P27 ~ 20	I/O	パラレルポート 2 入出力。
P37 ~ 30	I/O	パラレルポート 3 入出力。
PEXR_	O	外部拡張入力ポート・イネーブル出力。アクティブ“L”。
PEXW	O	外部拡張出力ポート・イネーブル出力。アクティブ“H”。

3 . KC160 CPU コア

3.1 概要

「KC160」は、ザイログ社 Z80 CPU 及び弊社 KC80 とオブジェクトレベルで上位互換の高速 16 ビット CPU コアです。Z80 の全命令に加え、各種処理命令が追加されています。アドレス空間は、16M バイトをリニアアドレスでサポートしており、大規模なプログラム及びデータ処理にも対応可能となっています。また、互換ボックスの採用により、豊富なソフトウェア資産を変更無しに使用したり、新規プログラムへのスムーズな移行が可能です。

また、「KC160」では命令のプリフェッチを行っています。内部にプリフェッチキューを 4 バイト装備しており、高速動作を実現しています。

3.2 レジスタ構成

KC80 (Z80) レジスタセットに準拠しており、拡張レジスタを追加するにより 24 ビットアドレス (16M バイトのアドレス空間) をサポートしています。図 3-2 に KC160 CPU コアのレジスタ構成を示します。

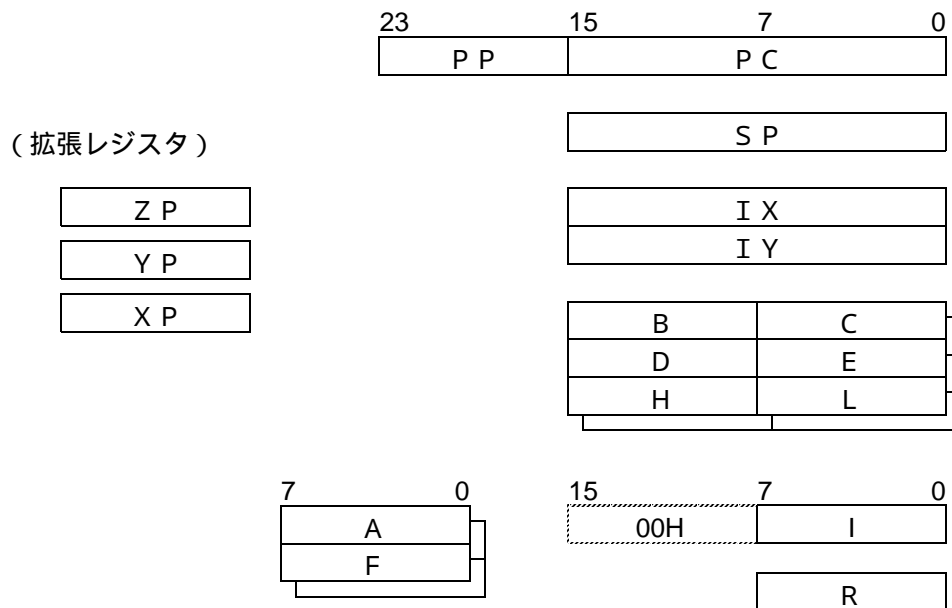


図 3-2 レジスタ構成

各レジスタの機能を以下に示します。

プログラムカウンタ (P P C)

次に実行すべきプログラムのアドレス情報 24 ビットを保持するレジスタです。この PPC (PP:PC) で示されるアドレスから命令をフェッチします。リセットで 000000H に初期化されます。

スタックポインタ (S P)

RAM 上のスタック領域のその時点での先頭アドレス情報を保持するレジスタです。スタック領域のページは拡張レジスタ ZP で指定されるため、ZSP と表記する場合があります。

インデックスレジスタ (I X , I Y)

インデックス・アドレッシングモード用として用いられる 16 ビットのベースアドレスを保持するレジスタで、IX レジスタ、IY レジスタの 2 つがあります。

インタラプトページアドレスレジスタ (I)

割り込みモード 2 及びモード 3 において、間接サブルーチン・ジャンプする為の間接アドレスの上位 16 ビットの値を保持するレジスタです。最上位の 8 ビットは '00H' 固定です。

メモリ・リフレッシュレジスタ (R)

8 ビットレジスタで、その内容は命令フェッチごとに自動的にインクリメントされます。ただし最上位ビットはインクリメントされません。Z80 との互換性を維持するために設けてあります。

アキュムレータ (A , A ') フラグレジスタ (F , F ')

2 組の 8 ビットアキュムレータ (A, A') と、各々対になった 2 組のフラグレジスタ (F, F') があります。アキュムレータは 8 ビットの算術・論理演算の結果を保持するレジスタです。フラグレジスタは演算結果の状態を保持するレジスタです。A, F と A', F' の交換は EX AF, AF' 命令で行います。

汎用レジスタ (B , C , D , E , H , L , B ' , C ' , D ' , E ' , H ' , L ')

2 組の汎用レジスタ群があり、それぞれ単独で 8 ビットレジスタ (B, C, D, E, H, L または B', C', D', E', H', L') として使用可能で、またこれらは 16 ビットのレジスタペア (BC, DE, HL または BC', DE', HL') としても使用できます。B, C, D, E, H, L と B', C', D', E', H', L' の交換は EXX 命令で行います。

拡張レジスタ (P P , Z P , Y P , X P)

24 ビット拡張用に設けたレジスタで、それぞれに以下の位置付けを持っています。

PP: プログラムコード用のページレジスタで、PPC (PP:PC) という形でプログラムカウンタを表します。

ZP: スタック及びローカル変数など主たる作業領域を指定するレジスタです。

YP: グローバル変数処理用のレジスタです。

XP: ZP、YP の補助用としてのレジスタです。

3.3 レジスタの拡張及びアドレッシング

拡張レジスタと既存の 16 ビットレジスタとのデフォルト及びその他の接続について、図 3-3a, 3-3b に示します。

24 ビットアドレスを取り扱うため、Z80 に準拠した 16 ビットレジスタまたは 2 つの 8 ビットレジスタ・ペアに 8 ビットの拡張レジスタを接続して使用します。接続される拡張レジスタはデフォルトで以下のように決まっています。

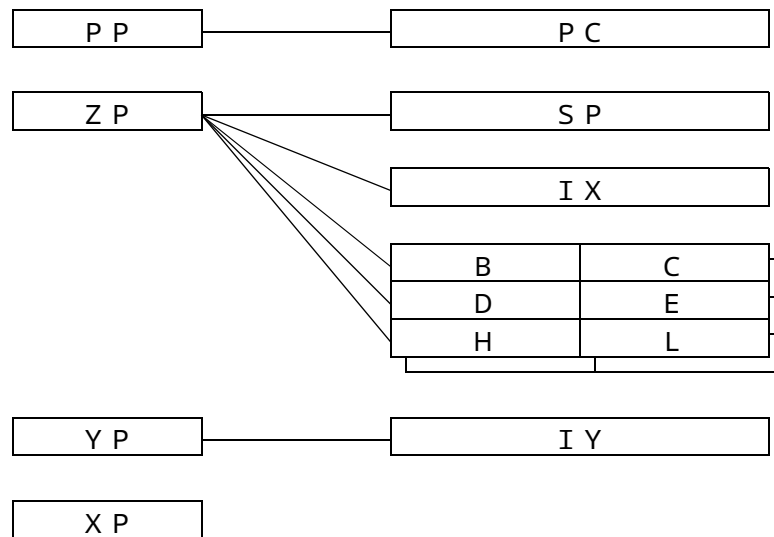


図 3-3a 拡張レジスタのデフォルト接続

レジスタのデフォルト状態は図 3-3a に示すように以下の通りになります。

プログラムカウンタ：PPC (PP:PC)

スタックポインタ：ZSP (ZP:SP)

インデックスレジスタ：ZIX (ZP:IX), YIY (YP:IY)

汎用レジスタ：ZBC (ZP:BC), ZDE (ZP:DE), ZHL (ZP:HL)

更に、KC160 では以下のように、上記デフォルトの組み合わせ以外の拡張レジスタ及び汎用レジスタである A レジスタを接続して使用することができます。この場合、図 3-4a に示したデフォルトの組み合わせ時よりも命令コードが 1 バイト長くなります。

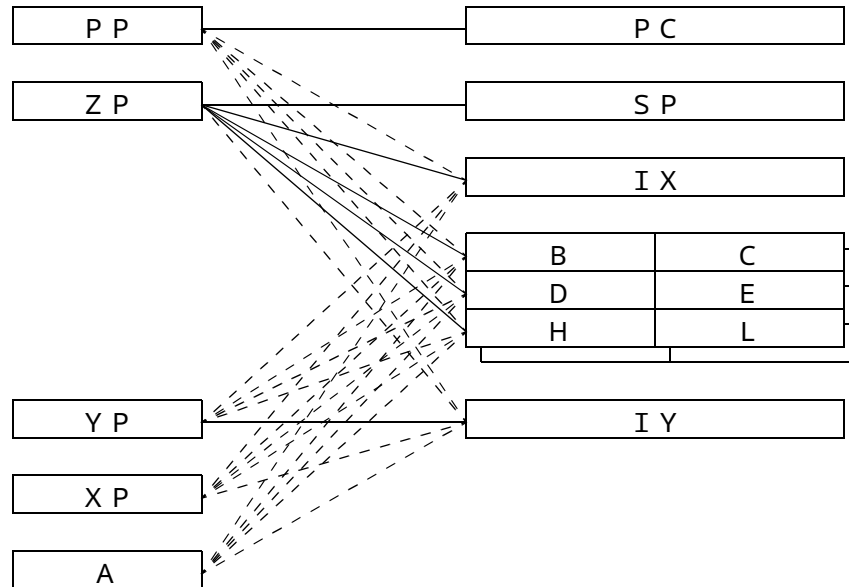


図 3-3b 拡張レジスタと既存の 16 ビットレジスタとの全接続

拡張レジスタを接続した 24 ビットレジスタの記載方法は、それぞれの 16 ビットレジスタの先頭に拡張レジスタの先頭 1 文字を加えた省略形を用いて表します。拡張レジスタの他に A レジスタも設定可能です。

Pxx, Zxx, Yxx, Xxx, Axx

デフォルト接続以外の具体例を以下に示します。

PIX (PP:IX), XIX (XP:IX), ZIY (ZP:IY), AIY (A:IY),

XBC (XP:BC), YDE (YP:DE), PHL (PP:HL), AHL (A:HL) etc.

また、例えば SP という記述を用いてアドレスを指定した場合、デフォルト接続の ZSP と解釈され、24 ビットでアドレッシングします。

拡張レジスタは、なるべく図 3-3a の接続状態で使用することが処理性能やコードサイズの点で有利になります。具体的には、以下のような役割分担による使い方をお勧めします。

3.3.1 ZP レジスタ（スタック及びローカル変数処理）

スタック及びローカル変数処理命令例を以下に示します。ローカル変数とは、C 言語などの世界で使用される表現の 1 つで、ある特定の関数内でのみ使用される変数のことを言います。

ここで使用されている ZP は、スタック及びローカル変数など主たる作業領域を指定するレジスタです。

ex.) PUSH AF	: SP SP-2, (ZSP)	AF	... スタックポインタは 24 ビット
ex.) LD (ZSP+d), HL	: スタック変数処理		
ex.) LD HL, (ZIX+d)	: ローカル変数処理		
ex.) LD B, (ZHL)	: ローカル変数処理		
ex.) LD DE, (nnnn)	: DE (ZP:nnnn)		... デフォルトで先頭 8 ビットに ZP

3.3.2 YP レジスタ（グローバル変数処理）

グローバル変数処理命令例を以下に示します。グローバル変数とは、C 言語などの世界で使用される表現の 1 つで、複数の関数内で使用される変数のことを言います。

ここで使用されている YP は、グローバル変数処理用としてスタック領域とは独立したメモリ領域を指すレジスタです。

ex.) LD (ZIX+d), YIY	: グローバル変数処理
ex.) ADD A, (YIY+d)	: グローバル変数を用いた演算
ex.) PUSH YIY	: (グローバル) 変数へのポインタ指定

3.3.3 XP レジスタ（グローバル変数処理）

XP は、グローバル変数処理用 (YP) の補助として、また、スタック及びローカル変数処理用 (ZP) の補助として用いられます。

ex.) LDF XIX, (nnnnnn)	: グローバル変数処理
ex.) POP XIX	: (グローバル) 変数へのポインタ指定
ex.) LDIR XY	: (YDE) (XHL) ... 24 ビットアドレス指定連続データ転送

3.4 フラグについて

フラグレジスタ (F, F') のビット構成は Z80 に準拠しています。図 3-4 に示します。

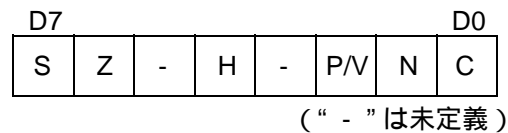


図 3-4 フラグレジスタのビット構成

フラグレジスタの各ビットの機能を以下に示します。

サイン・フラグ (S)

符号付数値による演算で、演算結果が負ならばセットされます。

ゼロ・フラグ (Z)

加算、減算、論理演算等において演算結果がゼロである時にセットされます。また、ブロック・サーチ命令時の検索データ検出時、ブロック入出力命令時、ローテート/シフトおよびストリングサーチ、ビットテスト命令等にも用いられます。

ハーフキャリー・フラグ (H)

BCD 演算の結果、下位 4 ビットからのキャリー、ボローを示します。

パリティ/オーバーフロー・フラグ (P / V)

このフラグは 2 つの機能を持ち、論理演算時にはパリティを示し、算術演算ではオーバーフロー が発生したことを示します。

オーバーフローでは、アキュムレータ内の 2 の補数値が + 127 を超えるか、- 128 より小さくなった時にセットされます。また、ブロック・サーチ命令、ブロック入出力命令、LD A, I 命令、LD A, R 命令の実行時にも用いられます。

サブトラクト・フラグ (N)

減算命令の場合セットされます。DAA 命令実行時、CPU はこのフラグを見て、先に実行された命令が加算か減算かを自動判定して処理を行います。

キャリー・フラグ (C)

アキュムレータの最上位ビットからの桁上がりでセットされます (加算命令、減算命令、シフト・ローテイト命令等)。

尚、各命令時のフラグ動作については、命令一覧表を参照してください。

3.5 命令セット

命令体系については、Z80 とバイナリレベルで上位互換であり、Z80 (KC80) の全命令をサポートしています。オペコードについても Z80 のコード体系をベースに新規の拡張命令を追加する形をとり、8 ビット CPU 並みの優れたコード効率を実現します。拡張命令については、16/24 ビットデータ処理サポートや、C コンパイラを用いて作成したプログラムの効率化・コンパクト化を実現しました。

以下に主な拡張命令について説明します。

3.5.1 C コンパイラ対応命令群

主として C コンパイラの性能向上のために追加した命令です。C 言語のローカル変数処理を高速化するためのスタック変数アクセス命令、C 言語の変数 (INT 変数他) を高速にアクセスするための 16 ビットデータ転送命令等です。

- ex.) LD HL, (SP+d) : ディスプレイスメント付きスタック変数アクセス (16 ビット)
- ex.) LD (ZSP+d), AHL : ディスプレイスメント付きスタック変数アクセス (24 ビット)
- ex.) LD HL, (IX+d) : 16 ビットデータ転送

3.5.2 16M バイトアドレス空間対応命令群

アドレス空間を 16M バイトに拡張したことにより、それに対応した命令群を追加しました。24 ビットアドレス指定データ転送命令、アドレスポインタ処理を高速化するための 24 ビットデータ転送命令、3 バイト (24 ビット) 分岐命令等です。

- ex.) LDF A, (nnnnnn) : 24 ビット絶対アドレス指定データ転送 (8 ビットデータ)
- ex.) LDF (nnnnnn), XIX : 24 ビット絶対アドレス指定データ転送 (24 ビットデータ)
- ex.) LD B, (ZHL) : 24 ビット間接アドレス指定データ転送 (8 ビットデータ)
- ex.) ADD A, (YIY+d) : 24 ビットアドレス指定演算 (8 ビットデータ)
- ex.) LDIR XY : 24 ビットアドレス指定連続データ転送
- ex.) LD (ZSP+d), AHL : 24 ビットデータ転送 (間接アドレス指定)
- ex.) PUSH YIY : 24 ビットデータスタック待避
- ex.) CALL3 nnnnnn : 3 バイト Call
- ex.) RET3 : 3 バイト Return
- ex.) JP3 NZ, nnnnnn : 3 バイト条件付き Jump

3.5.3 その他命令群

演算性能向上のため、乗除算命令を追加しました。

- | | | | |
|---------|-----------|--------------------------------|-------------------|
| 8 × 8 | 16 | : MUL H, L / MULS H, L | (Unsigned/Signed) |
| 16 × 16 | 32 | : MUL DE, HL / MULS DE, HL | (Unsigned/Signed) |
| 16 ÷ 8 | 8 ... 8 | : DIV HL, A / DIVS HL, A | (Unsigned/Signed) |
| 32 ÷ 16 | 16 ... 16 | : DIV DEHL, BC / DIVS DEHL, BC | (Unsigned/Signed) |

3.6 バス解放（バスリクエスト/バスアクノレッジ）

通常 CPU がアドレスバス、コントロールバス（MREQ_—, IORQ_—, RD_—, WR_—）の制御権を握っていますが、外部からバス要求（BREQ_— = “L”）があると、CPU はアドレスバス・データバスをハイインピーダンス状態、コントロールバスをインアクティブ状態（“H”）にして、全ての割り込みを禁止します。その後、BACK_— 出力をアクティブ状態（“L”）、コントロールバスをハイインピーダンス状態にして、外部にバスの制御権を譲ります。この機能により、CPU を介さないデータ転送が可能となります。このサイクルは、BREQ_— が “L” の期間ずっと続きます。

バスリクエストの受け付けタイミングは、論理的なバスサイクルの切れ目です。ここで、論理的なバスサイクルの切れ目とは、命令フェッチサイクル、メモリ・I/O データアクセスサイクルの完了時のことを言います。例えば、奇数アドレスの 16 ビットデータ・メモリアクセスにおいては、2 回のアクセスサイクルが必要となりますが、論理的には 1 つの 16 ビットデータアクセスサイクルですから、2 回のサイクルの途中ではなく 2 回目のアクセスサイクル直後（アクセスサイクル完了時）に要求を受け付けます。

外部からのバス要求が解除されると（BREQ_— = “H”）、BACK_— 出力及びコントロールバスをインアクティブ状態（“H”）にして、その後アドレスバス・コントロールバスをドライブして CPU の動作を再開します。

バスリクエスト/バスアクノレッジ・サイクルタイミング例を図 3-6 に示します。

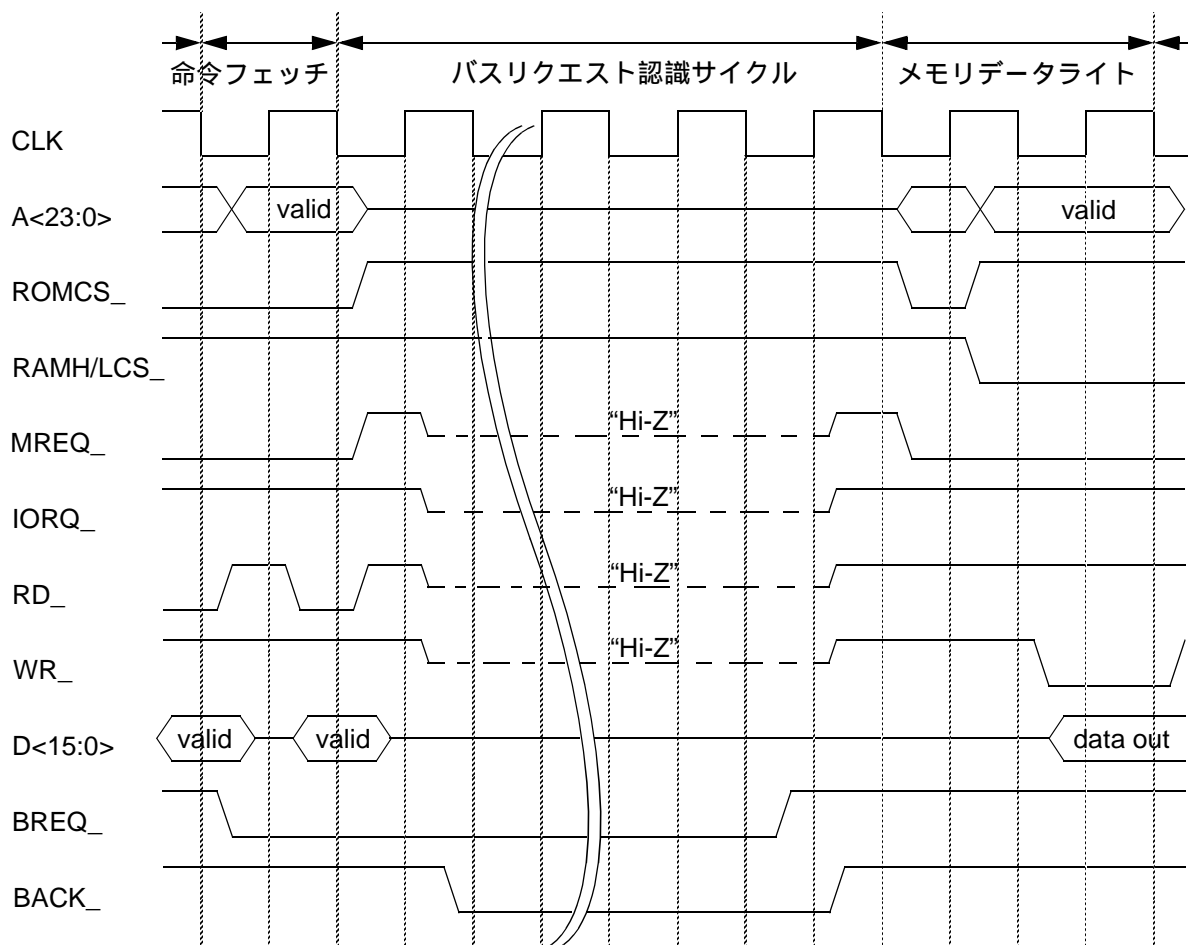


図 3-6 バスリクエスト/バスアクノレッジ・サイクル例

3.7 割り込み

KC160 で受け付けられる割り込み要求には、マスカブル割り込み (INT) とノンマスカブル割り込み (NMI) の 2 通りがあります。マスカブル割り込みよりもノンマスカブル割り込みの方が優先順位が高く、同時に発生した場合は、ノンマスカブル割り込みの方が先に受け付けられます。

3.7.1 マスカブル割り込み

マスカブル割り込みは EI 命令によって割り込みを許可され、DI 命令によって割り込みを禁止されます。この制御は 2 つのイネーブル・フリップフロップ (IFF1, IFF2) によって制御されています。表 3-7-1 はそのフリップフロップの状態遷移とその要因を示したものです。

表 3-7-1 割り込みイネーブルフリップフロップ設定

要因	IFF1	IFF2	
リセット	0	0	
DI 命令	0	0	
EI 命令	1	1	
INT 受付け	0	0	
NMI 受付け	0	-	
RETN 命令受付け	IFF2	-	IFF1 に IFF2 をコピー
LD A, I 命令	-	-	P/V に IFF2 をコピー
LD A, R 命令	-	-	P/V に IFF2 をコピー

(“ - ” は不変を表します)

マスカブル割り込みが受け付けられるのは、以下の条件を全て満たしている時です。

- 1) 2 つのイネーブルフリップフロップ IFF1, IFF2 が共にセットされている。
- 2) BREQ_ 及び BACK_ がインアクティブ状態 (“H”) になっている。
- 3) NMI_ の立ち下がりがエッジを保存していない。

マスカブル割り込みには 4 種類のモードがあり、それぞれに動作が異なります。以下に各モードでの動作を説明します。通常はモード 2 あるいはモード 3 を使用します。

モード 0

リセット時には、自動的にこのモードにセットされます。また IM 0 命令によってこのモードに移ります。このモードは、割り込みサイクル中に読み込んだ命令をそのまま実行します。通常この割り込みで使う命令は、RST 00H ~ 38H 命令です。KC160 では CALL 命令は使用できません。

また、すべてのプログラムが互換ボックス内 (000000H ~ 00FFFFH) に収まらない場合には、割り込みルーチンの先頭に TRA 命令 (後述) を配置し、プログラムカウンタの内容を 3 バイト分スタックに退避 (PUSH) する必要があります。

モード 1

IM 1 命令によってこのモードに移ります。このモードでは、プログラムカウンタの内容をスタックに退避後、割り込みサイクル中に読み込んだデータを無視して、強制的に RST 38H 命令を実行します。モード 0 同様、すべてのプログラムが互換ボックス内に収まらない場合には、割り込みルーチンの先頭に TRA 命令(後述)を配置し、プログラムカウンタ PPC の内容を 3 バイト分スタックに退避(PUSH)する必要があります。

モード 2

IM 2 命令によってこのモードに移ります。このモードでは、割り込みサイクル中に読み込んだ割り込みベクタを下位 8 ビット(但し、最下位ビットは '0')とし、I レジスタの内容を上位 16 ビット(但し、I<15:8> は '00H')とする 24 ビットアドレスで割り込みルーチンのスタートアドレステーブルをアクセスします。そして、読み込んだスタートアドレスから始まる割り込みルーチンを実行します。

尚、モード 2 ではプログラムカウンタの内容は 2 バイト分だけがスタックに待避されるため、すべてのプログラムが互換ボックス内に収まる場合に限られます。

23	15	7	0
00H	I<7:0>	割り込みベクタ	(ビット 0 = '0')

モード 3

24 ビットアドレス空間に対応するため新規に設けた割り込みモードです。IM 3 命令によってこのモードに移ります。このモードでは、割り込みサイクル中に読み込んだ割り込みベクタを下位 8 ビット(但し、最下位ビットは '0')とし、I レジスタの内容を上位 16 ビット(但し、I<15:8> は '00H')とする 24 ビットアドレスで割り込みルーチンのスタートアドレステーブルをアクセスします。そして、読み込んだスタートアドレスから始まる割り込みルーチンを実行します。尚、24 ビットアドレスのうち最上位の 8 ビットは '00H' 固定のためスタートアドレステーブルは互換ボックス内(000000H ~ 00FFFFH)に配置する必要があります。

またモード 3 では、無条件にプログラムカウンタの内容を 3 バイト分スタックに退避(PUSH)するので、16M バイトのアドレス空間全領域での実行が可能です。

RETI 命令

KL5C16005 では外部に Z80 ペリフェラル LSI を使用できませんので、RETI 命令は内蔵の割り込みコントローラ KP69 に EOI 信号を出力するための命令です。

但し、モード 3 専用の RETI 命令(3 バイト RETI 命令)は用意していませんので、割り込みルーチンの最後では RETI 命令を配置する代わりに以下の処理が必要となります。

```
OUT (1DH), A      : 内部 I/O (1DH) にライト動作を行い EOI を発行、データ値は任意
EI                 :
RET3               :
```

内部 I/O アドレス 1DH 番地は、書き込みが行われると、内蔵の割り込みコントローラ KP69 に EOI を発行する I/O レジスタです。この場合、書き込むデータは任意です。

TRA 命令

モード 0、モード 1 割り込みでは、プログラムカウンタの内容を 2 バイト分スタックに退避し RST 命令等を実行します。全プログラムが互換ボックス内に収まっていれば問題ありませんが、16M バイトのアドレス空間全領域を考慮した場合、プログラムカウンタ PPC の内容を 3 バイト分スタックに退避する必要があります。KC160 では、飛び先の先頭アドレス(RST xxH 命令では 0000xxH 番地)に TRA 命令を配置することで、3 バイト PUSH を行います。つまり、TRA 命令はプログラムカウンタの内容をスタックに退避するための 2 バイト PUSH を 3 バイト PUSH に変更する命令です。

3.7.2 ノンマスカブル割り込み

ノンマスカブル割り込みはソフトウェアでマスクできない割り込みです。NMI_‾ の立ち下がリエッジを検出し、命令の切れ目でその割り込みを受け付けます。ノンマスカブル割り込みは BREQ_‾ がインアクティブ状態 (“H”) になっている時に有効になります。

ノンマスカブル割り込みを認識すると、プログラムカウンタの内容をスタックにセーブして、000066H 番地へサブルーチンコールします。マスカブル割り込みのモード 0、1 と同様に、すべてのプログラムが互換ボックス内に収まらない場合には、割り込みルーチンの先頭に TRA 命令を配置し、プログラムカウンタの内容を 3 バイト分スタックに退避 (PUSH) する必要があります。

この割り込みルーチンからの復帰は、互換ボックス内のみを使用時 (TRA 命令を配置しない時) には RETN 命令、すべてのアドレス空間を使用する場合 (TRA 命令を配置する場合) には RETN3 命令 (3 バイト POP) の実行により実現します。ノンマスカブル割り込みでは、終了時に割り込みイネーブルフリップフロップの IFF1 の値を IFF2 にコピーすることが必要なため、RET3 命令ではなく RETN3 命令を使用してください。

3.8 HALT の実行と解除

HALT 命令を実行すると、次にアクセスするアドレスを出力した状態を保持し、コントロールバスについては MREQ_‾ のみアクティブ状態 (“L”) に、IORQ_‾, RD_‾, WR_‾ をインアクティブ状態 (“H”) にします。

ここで、リセット入力や割り込み入力 (ノンマスカブル割り込みあるいは割り込みイネーブルフリップフロップ IFF1, IFF2 がセットされている状態でのマスカブル割り込み) があれば、この状態は解除されます。

3.9 リセット

RESET_ 入力を内部システムクロックの最小 3 クロック期間アクティブ状態 (“L”) にすることによって、CPU はリセットされます。リセット期間中は、アドレスバス・データバスはハイインピーダンス状態となります。システムクロックの立ち下がりエッジで RESET_ 入力がいんアクティブ状態 (“H”) になったことを確認すると、000000H 番地の命令のフェッチ、実行を順次開始します。また、割り込みモードはモード 0 にセットされ、割り込みイネーブルフリップフロップ IFF1, IFF2 はリセットされます。

タイミング波形を図 3-9 に示します。

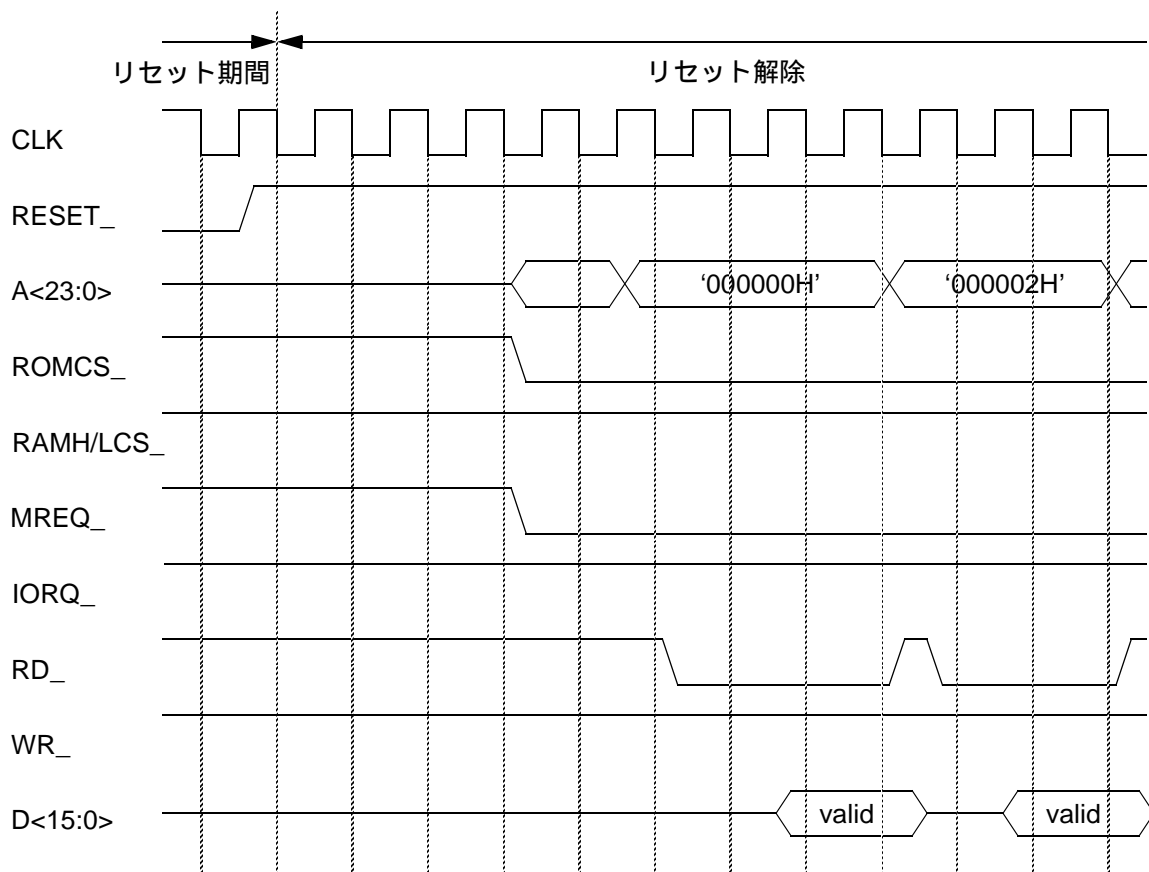


図 3-9 リセット・タイミング(ROM 領域 2 ウェイト)

4．バス接続

4.1 メモリ接続

使用するメモリの構成により、2つのモード（接続モード A，B）からバスの接続方法を選択することができます。

4.1.1 接続モード A

ROM 領域(000000H ~ 7FFFFFFH)をデータバス幅 16 ビット、SRAM 領域(800000H ~ FFFFFFFH)をデータバス幅 16 ビットとして使用します。モードの設定方法については、第 1 2 章（動作モード設定）を参照ください。

ROM 領域のチップセレクト信号として ROMCS_ が出力されます。また、SRAM 領域のチップセレクト信号として RAMHCS_ 及び RAMLCS_ が出力され、データバスの上位（奇数）バイト側と下位（偶数）バイト側をアクセスしたときにそれぞれアクティブ状態になります。

接続例として、データバス 8 ビット幅の SRAM 2 個を接続した場合を図 4-1-1 に示します。また、このモード時に使用するアドレスバス、データバス、チップセレクト信号の各端子を表 4-1-1 に示します。

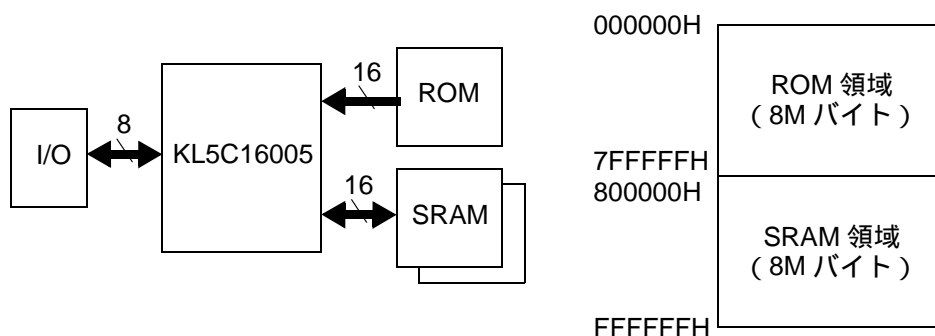


図 4-1-1 接続モード A(ROM 領域 16 ビット、SRAM 領域 16 ビット)

表 4-1-1 接続モード A のバス端子

	端子番号	マイコン端子名	ROM 端子名 (参考)	SRAM 端子名 (参考)
アドレスバス	24	A<23>	A<22>	A<22>
	25	A<22>	A<21>	A<21>
	26	A<21>	A<20>	A<20>
	27	A<20>	A<19>	A<19>
	40	A<19>	A<18>	A<18>
	41	A<18>	A<17>	A<17>
	42	A<17>	A<16>	A<16>
	43	A<16>	A<15>	A<15>
	44	A<15>	A<14>	A<14>
	45	A<14>	A<13>	A<13>
	46	A<13>	A<12>	A<12>
	47	A<12>	A<11>	A<11>
	49	A<11>	A<10>	A<10>
	50	A<10>	A<9>	A<9>
	51	A<9>	A<8>	A<8>
	54	A<8>	A<7>	A<7>
	55	A<7>	A<6>	A<6>
	56	A<6>	A<5>	A<5>
	57	A<5>	A<4>	A<4>
	58	A<4>	A<3>	A<3>
	59	A<3>	A<2>	A<2>
	60	A<2>	A<1>	A<1>
	61	A<1>	A<0>	A<0>
	62	A<0>	-	-
データバス	31	D<15>	O<15>	I/O<7>
	32	D<14>	O<14>	I/O<6>
	33	D<13>	O<13>	I/O<5>
	34	D<12>	O<12>	I/O<4>
	35	D<11>	O<11>	I/O<3>
	36	D<10>	O<10>	I/O<2>
	37	D<9>	O<9>	I/O<1>
	38	D<8>	O<8>	I/O<0>
	4	D<7>	O<7>	I/O<7>
	5	D<6>	O<6>	I/O<6>
	6	D<5>	O<5>	I/O<5>
	7	D<4>	O<4>	I/O<4>
	8	D<3>	O<3>	I/O<3>
	9	D<2>	O<2>	I/O<2>
	10	D<1>	O<1>	I/O<1>
	11	D<0>	O<0>	I/O<0>
				上位バス制御チップ
				下位バス制御チップ
制御信号	14	ROMCS_	CE	-
	17	RAMHCS_	-	CE (上位側)
	18	RAMLCS_	-	CE (下位側)

4.1.2 接続モード B

ROM 領域(000000H ~ 7FFFFFFH)をデータバス幅 16 ビット、SRAM 領域(800000H ~ FFFFFFFH)をデータバス幅 8 ビットとして使用します。モードの設定方法については、第 1 2 章(動作モード設定)を参照ください。

ROM 領域のチップセレクト信号として ROMCS₀ が出力され、SRAM 領域のチップセレクト信号として RAMHCS が出力されます。SRAM 領域のデータバスとして、D<15:8> を使用します。

接続例として、データバス 8 ビット幅の SRAM 1 個を接続した場合を図 4-1-2 に示します。また、このモード時に使用するアドレスバス、データバス、チップセレクト信号の各端子を表 4-1-2 に示します。

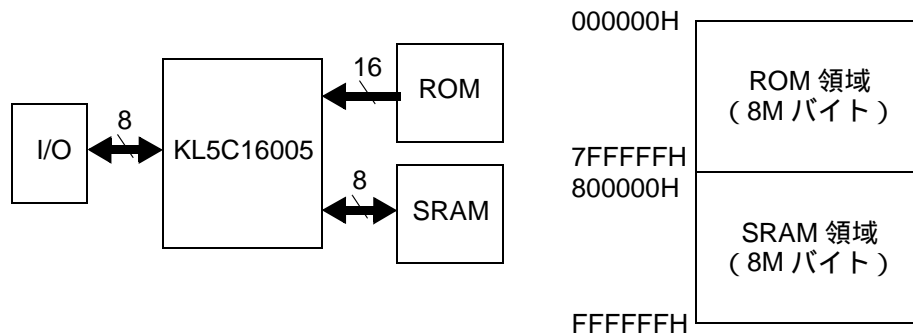


図 4-1-2 接続モード B(ROM 領域 16 ビット、SRAM 領域 8 ビット)

表 4-1-2 接続モード B のバス端子

	端子番号	マイコン端子名	ROM 端子名 (参考)	SRAM 端子名 (参考)
アドレスバス	24	A<23>	A<22>	A<23>
	25	A<22>	A<21>	A<22>
	26	A<21>	A<20>	A<21>
	27	A<20>	A<19>	A<20>
	40	A<19>	A<18>	A<19>
	41	A<18>	A<17>	A<18>
	42	A<17>	A<16>	A<17>
	43	A<16>	A<15>	A<16>
	44	A<15>	A<14>	A<15>
	45	A<14>	A<13>	A<14>
	46	A<13>	A<12>	A<13>
	47	A<12>	A<11>	A<12>
	49	A<11>	A<10>	A<11>
	50	A<10>	A<9>	A<10>
	51	A<9>	A<8>	A<9>
	54	A<8>	A<7>	A<8>
	55	A<7>	A<6>	A<7>
	56	A<6>	A<5>	A<6>
	57	A<5>	A<4>	A<5>
	58	A<4>	A<3>	A<4>
	59	A<3>	A<2>	A<3>
	60	A<2>	A<1>	A<2>
	61	A<1>	A<0>	A<1>
	62	A<0>	-	A<0>
データバス	31	D<15>	O<15>	IO<7>
	32	D<14>	O<14>	IO<6>
	33	D<13>	O<13>	IO<5>
	34	D<12>	O<12>	IO<4>
	35	D<11>	O<11>	IO<3>
	36	D<10>	O<10>	IO<2>
	37	D<9>	O<9>	IO<1>
	38	D<8>	O<8>	IO<0>
	4	D<7>	O<7>	-
	5	D<6>	O<6>	-
	6	D<5>	O<5>	-
	7	D<4>	O<4>	-
	8	D<3>	O<3>	-
	9	D<2>	O<2>	-
	10	D<1>	O<1>	-
	11	D<0>	O<0>	-
制御信号	14	ROMCS_	CE	-
	17	RAMHCS_	-	CE
	18	RAMLCS_	-	-

4.2 I/O 接続

外部 I/O のバス接続方法を以下に示します。

データバスとしては、16 ビットデータバス D<15:0> のうち I/O 接続用 8 ビットデータバスとして上位側の D<15:8> を使用します。

また、アドレスバスは、基本的に下位 8 ビット A<7:0> を使用しますが、C レジスタを用いたレジスタ間接による外部 I/O アクセスの場合には、B レジスタの値が A<15:8> に出力されるため、16 ビットアドレスによるアクセスが可能となります。但し、内部 I/O はアドレスバス下位 8 ビット A<7:0> のみのデコードで上位アドレスは無視して指定されますのでご注意ください。例えば I/O アドレス 001FH, 011FH, 021FH, ... , FF1FH をアクセスすると、すべて内部 I/O アドレス 1FH がアクセス対象になります。内部 I/O アドレスとして、00H ~ 3FH が既に予約されています。

接続例及び接続端子については、図 4-2 及び表 4-2 を参照してください。

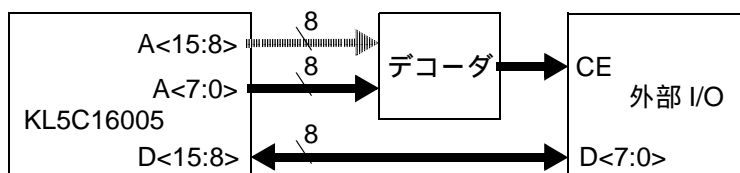


図 4-2 外部 I/O 接続例

表 4-2 外部 I/O のバス端子 (全接続モード共通)

	端子番号	マイコン端子名	外部 I/O 端子名
アドレスバス	44	A<15>	-
	45	A<14>	-
	46	A<13>	-
	47	A<12>	-
	49	A<11>	-
	50	A<10>	-
	51	A<9>	-
	54	A<8>	-
	55	A<7>	-
	56	A<6>	-
	57	A<5>	-
	58	A<4>	-
	59	A<3>	-
	60	A<2>	-
	61	A<1>	-
	62	A<0>	-
データバス	31	D<15>	D<7>
	32	D<14>	D<6>
	33	D<13>	D<5>
	34	D<12>	D<4>
	35	D<11>	D<3>
	36	D<10>	D<2>
	37	D<9>	D<1>
	38	D<8>	D<0>

5 . アドレス空間

5.1 メモリアドレス空間

メモリアドレス空間として、16M バイト・リニアアクセスを可能としました。

また、000000H ~ 00FFFFH の 64K バイトのアドレス空間については、互換ボックスの採用により Z80 との完全な互換性を実現しています。互換ボックス内には、4K バイト (00F000H ~ 00FFFFH) あるいは 32K バイト (008000H ~ 00FFFFH) の RAM 領域が存在し、SRAM 領域内の 80F000H ~ 80FFFFH あるいは 808000H ~ 80FFFFH に配置された RAM をイメージとしてアクセスできる構造にしました。よって、ROM 領域内に SRAM を配置する必要がありません。領域サイズは、システムコントロールレジスタにより設定可能です。詳細については、第 1 2 章 (動作モード設定) を参照してください。

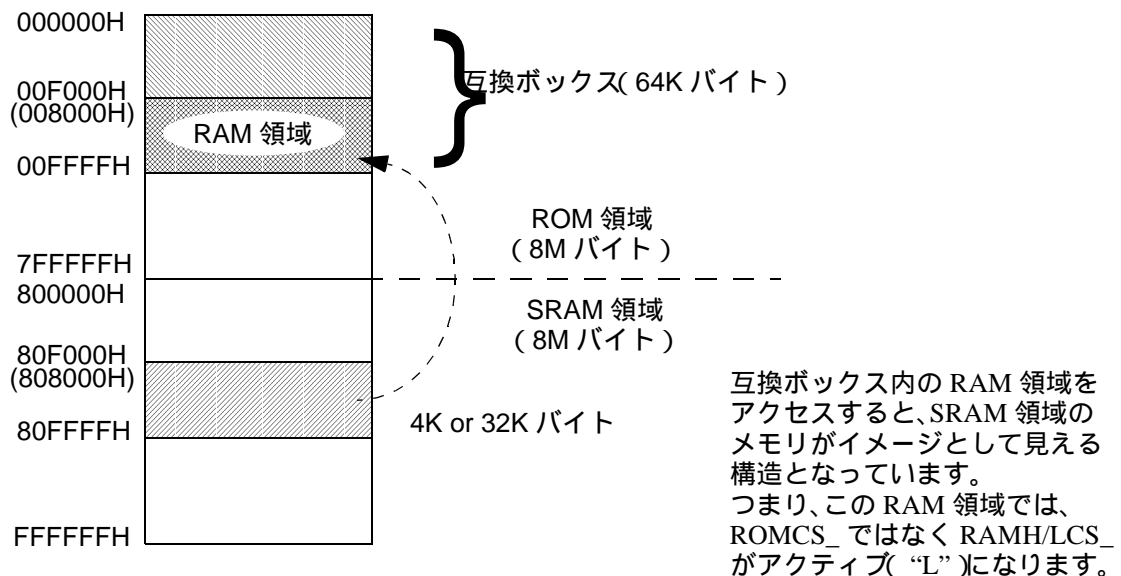


図 5-1 アドレス空間

5.2 互換ボックス

バイナリレベルでの Z80 上位互換性を実現するため、Z80 アドレス空間領域 (64K バイト) に相当する 000000H ~ 00FFFFH においては互換性を最優先とした仕様になっています。これにより、Z80 の命令セットをそのまま使用できるだけでなく、Z80 で作成したプログラム資産をそのまま移植することも可能です。

詳細については、第 7 章 (互換ボックス) を参照してください。

5.3 I/O アドレスマッピング

I/O アドレスとしては、基本的にアドレスバス A<7:0> による 00H ~ FFH を使用します。内部 I/O には 00H ~ 3FH が割り当てられており、外部 I/O として使用できる領域は 40H ~ FFH となります。但し、C レジスタを用いたレジスタ間接による外部 I/O アクセスの場合には、B レジスタの値が A<15:8> に出力されるため、16 ビットアドレスによるアクセスが可能となります。しかし、内部 I/O アドレスのデコードは A<7:0> のみの 8 ビットデコードで、上位アドレスは無視して指定されます。つまり、アドレス下位 8 ビットが 00H ~ 3FH の領域においては自動的に内部 I/O アクセスとなり、外部 I/O アクセスできませんのでご注意ください。

内部 I/O アドレスマッピングについては、表 5-3 に示します。

表 5-3 内部 I/O アドレスマッピング

ブロック名	I/O アドレス	ライト時	リード時
川崎製鉄使用予約	00H ~ 17H	-	-
システム コントロール レジスタ	18H 19H	SCR0 SCR1	SCR0 SCR1
川崎製鉄使用予約	1AH ~ 1CH	-	-
割り込み処理用	1DH	マニュアル EOI (* 3)	-
外部 R/W ポート	1EH	(* 1)	(* 2)
川崎製鉄使用予約	1FH	-	-
汎用タイマ	20H 21H	カウンタ コマンド	カウンタ ステータス
川崎製鉄使用予約	22H ~ 27H	-	-
UART 非同期シリアル I/F	28H 29H 2AH 2BH	BRG レート 川崎製鉄使用予約 送信データ コマンド	BRG レート 川崎製鉄使用予約 受信データ ステータス
川崎製鉄使用予約	2CH ~ 34H	-	-
割り込み コントローラ	35H 36H 37H	LER / PGR 川崎製鉄使用予約 IVR / IMR	ISR 川崎製鉄使用予約 IMR
パラレルポート	38H 39H 3AH 3BH 3CH 3DH 3EH 3FH	ポート 0 データ ビット操作コマンド ポート 1 データ ポート 1 方向制御 ポート 2 データ ポート 2 方向制御 ポート 3 データ ポート 3 方向制御	ポート 0 データ - ポート 1 データ ポート 1 方向制御 ポート 2 データ ポート 2 方向制御 ポート 3 データ ポート 3 方向制御

(* 1)

I/O アドレス 1EH にデータ・ライトを行うと、バスサイクルに同期して外部拡張出力ポート・イネーブル信号 PEXW に“H”パルスが出力されます。外部でデコードすることなく、データラッチ回路等のゲート端子に直接接続すれば、パラレル出力ポートを拡張することができます。

(* 2)

I/O アドレス 1EH にデータ・リードを行うと、バスサイクルに同期して外部拡張入力ポート・イネーブル信号 PEXR₁ に“L”パルスが出力されます。外部でデコードすることなく、3ステートバッファ回路等の出力イネーブル信号端子に直接接続すれば、パラレル入力ポートを拡張することができます。

(* 3)

詳細内容については、3.7.1 項（マスカブル割り込み）を参照ください。

6 . タイミング波形

この章では、命令フェッチ及びメモリ・I/O アクセスのタイミング波形について説明します。

尚、命令フェッチを含めたメモリアクセスについては、それぞれのバス接続モードごとの参照先を表 6-0 にまとめました。

また、内蔵のウェイトステートコントローラによるウェイトの挿入が可能であり、その設定方法については第 1 2 章（動作モード設定）を参照ください。

表 6-0 各接続モードにおけるメモリアクセス参照先一覧

バス接続モード	ROM 領域 (000000H ~ 7FFFFFFH)	SRAM 領域 (800000H ~ FFFFFFH)
接続モード A ROM 領域 16 ビット SRAM 領域 16 ビット	fetch : 6.1 data : 6.2.1 ~ 6.2.3	data : 6.2.1 ~ 6.2.3
接続モード B ROM 領域 16 ビット SRAM 領域 8 ビット	fetch : 6.1 data : 6.2.1 ~ 6.2.3	data : 6.2.4

図中で、fetch は命令フェッチ時、data はデータアクセス時の説明を記載した項目を示していますので、使用する接続モードに応じて必要な項目を参照してください。

6.1 命令フェッチタイミング (ROM 領域)

通常の ROM 16 ビット幅における命令フェッチサイクルを図 6-1a ~ 6-1c に示します。

基本サイクルは、1 クロックです。ジャンプ時には 1 クロック追加されます。尚、リセット直後は 2 ウェイトの設定になっています。

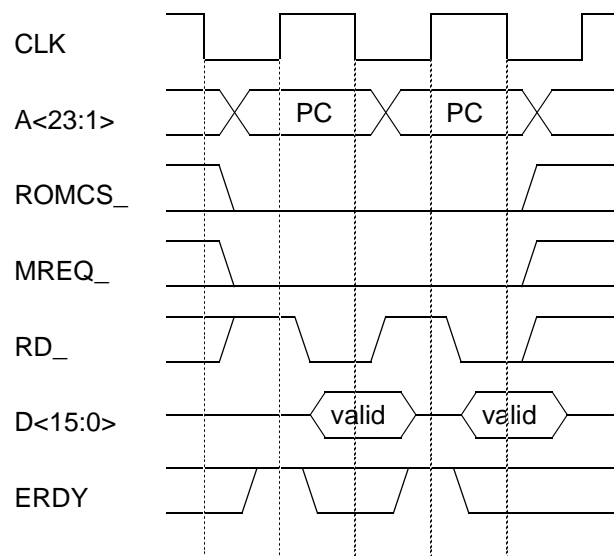


図 6-1a 命令フェッチ・サイクル (ROM、0 ウェイト)

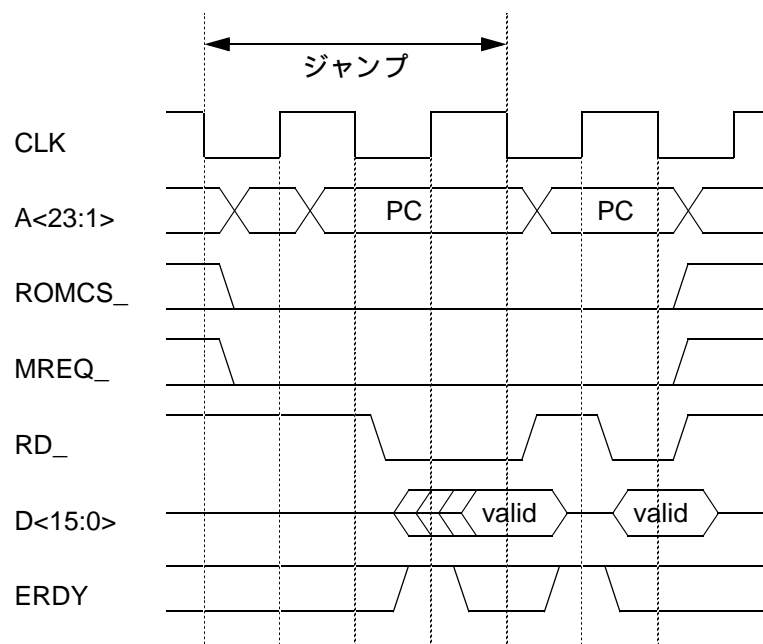


図 6-1b 命令フェッチ・サイクル (ROM、ジャンプ時、0 ウェイト)

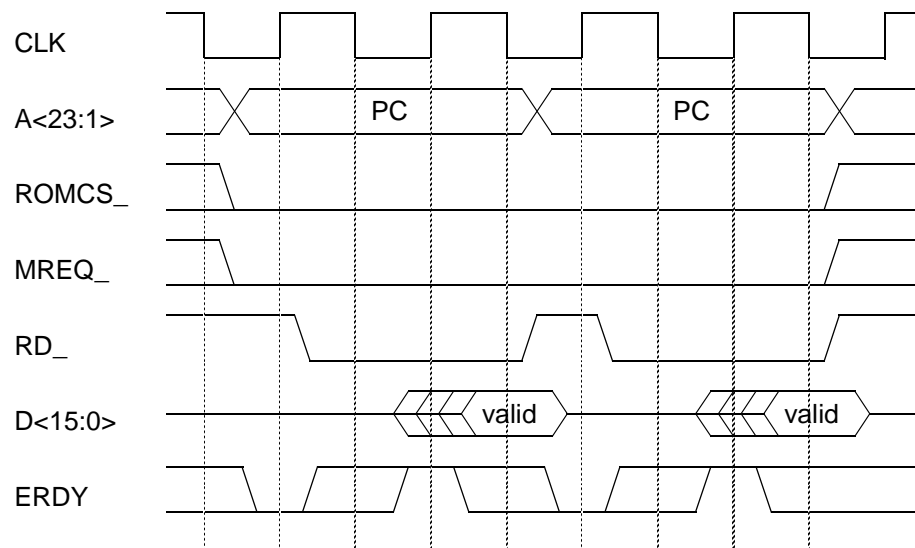


図 6-1c 命令フェッチ・サイクル(ROM、1 ウェイト)

6.2 メモリデータアクセスタイミング

6.2.1 ROM/SRAM 領域 16 ビットデータアクセス (ROM/SRAM 16 ビットモード時)

ROM/SRAM 16 ビットモード時の ROM/SRAM 領域における 16 ビットデータのリード / ライト・サイクルを図 6-2-1a, 6-2-1b に示します。

基本サイクルは、2 クロックです。但し、奇数アドレスの 16 ビットデータアクセスは、2 回のアクセスサイクルにより実現します。尚、リセット直後は 2 ウェイトの設定になっています。

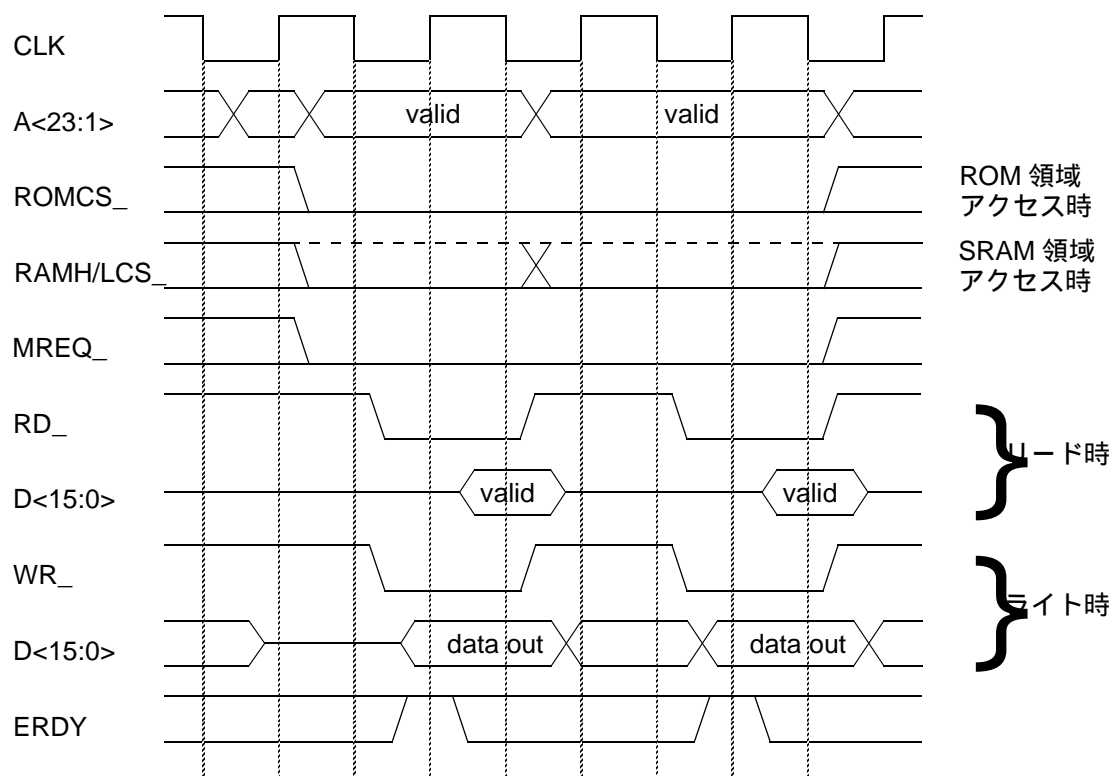


図 6-2-1a 16 ビットデータ・リード / ライト・サイクル (ROM/SRAM 領域 16 ビットモード、0 ウェイト)

SRAM 領域アクセス時には、RAMHCS_ 及び RAMLCS_ 信号は表 6-2-1 に示した値を出力します。

表 6-2-1 16 ビットデータアクセス時の RAMH/LCS_ 出力信号

データアドレス	メモリアドレス	RAMHCS_	RAMLCS_
偶数アドレス (MA)	MA	0	0
奇数アドレス (MA)	1 回目 : MA	0	1
	2 回目 : MA+1	1	0

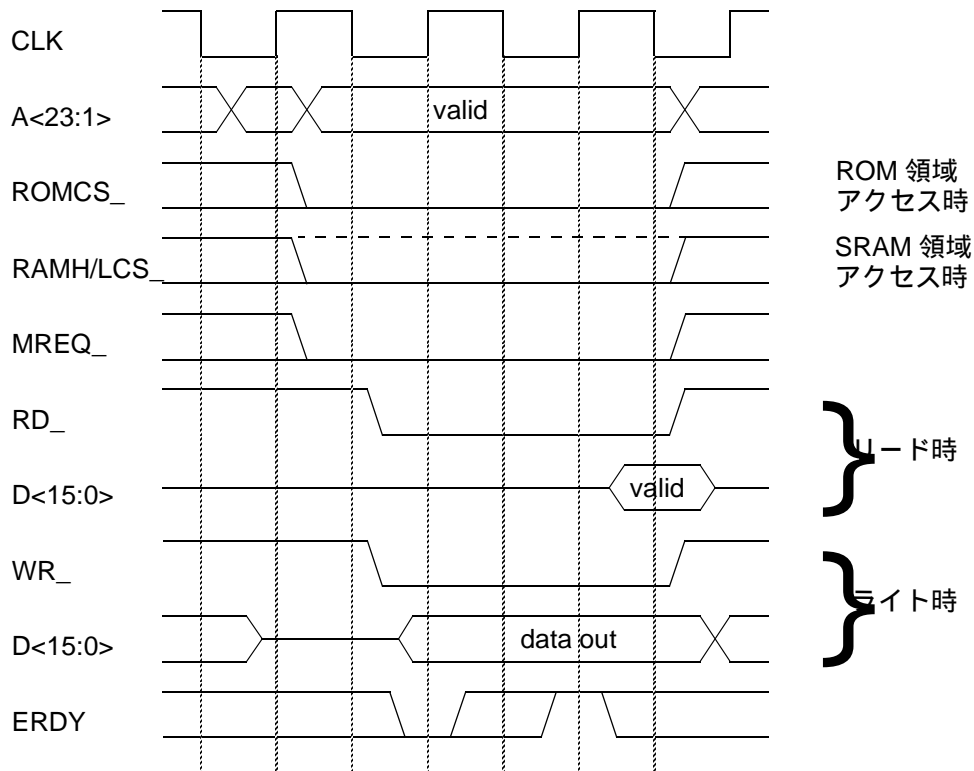


図 6-2-1b 16 ビットデータ・リード/ライト・サイクル(ROM/SRAM 領域 16 ビットモード、1 ウェイト)

6.2.2 ROM/SRAM 領域 8 ビットデータアクセス (ROM/SRAM 16 ビットモード時)

ROM/SRAM 16 ビットモード時の ROM/SRAM 領域における 8 ビットデータのリード/ライト・サイクルを図 6-2-2 に示します。

基本サイクルは、2 クロックです。尚、リセット直後は 2 ウェイトの設定になっています。

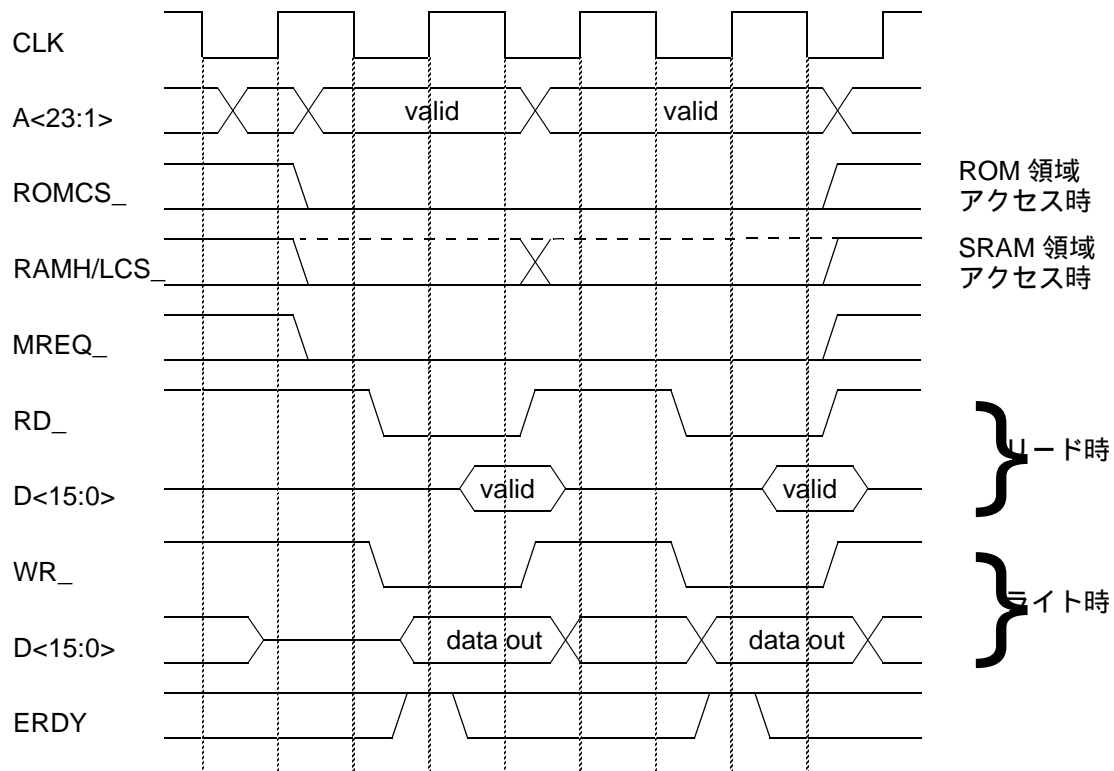


図 6-2-2 8 ビットデータ・リード/ライト・サイクル(ROM/SRAM 領域 16 ビットモード、0 ウェイト)

SRAM 領域アクセス時には、RAMHCS_ 及び RAMLCS_ 信号は表 6-2-2 に示した値を出力します。

表 6-2-2 8 ビットデータアクセス時の RAMH/LCS_ 出力信号

データアドレス	メモリアドレス	RAMHCS_	RAMLCS_
偶数アドレス (MA)	MA	1	0
奇数アドレス (MA)	MA	0	1

6.2.3 ROM/SRAM 領域 24 ビットデータアクセス (ROM/SRAM 16 ビットモード時)

ROM/SRAM 16 ビットモード時の ROM/SRAM 領域における 24 ビットデータのリード / ライト・サイクルについては、6.2.1 及び 6.2.2 項で示された 16 ビットデータアクセスと 8 ビットデータアクセスを連続して行う方法で実現します。表 6-2-3 に示します。

表 6-2-3 24 ビットデータアクセス

データアドレス	メモリアドレス	アクセスデータ幅
偶数アドレス (MA)	1 回目 : MA	16 ビット
	2 回目 : MA+2	8 ビット
奇数アドレス (MA)	1 回目 : MA	8 ビット
	2 回目 : MA+1	16 ビット

各アクセスサイクルのタイミング波形については、6.2.1 及び 6.2.2 項を参照してください。

6.2.4 ROM/SRAM 領域データアクセス (ROM/SRAM 8 ビットモード時)

ROM/SRAM 8 ビットモード時の ROM/SRAM 領域におけるデータのリード/ライト・サイクルを図 6-2-4a, 6-2-4b に示します。データバスは上位側の D<15:8> を使用します。よって、SRAM 領域ではチップセレクト信号として RAMHCS_ を使用します。

基本サイクルは、2 クロックです。尚、リセット直後は 2 ウェイトの設定になっています。

16/24 ビットデータアクセスについては、16 ビットの場合は 2 回に、24 ビットの場合は 3 回に分けて、いずれの場合にもメモリアドレスが増える方向でアクセスが行われます。

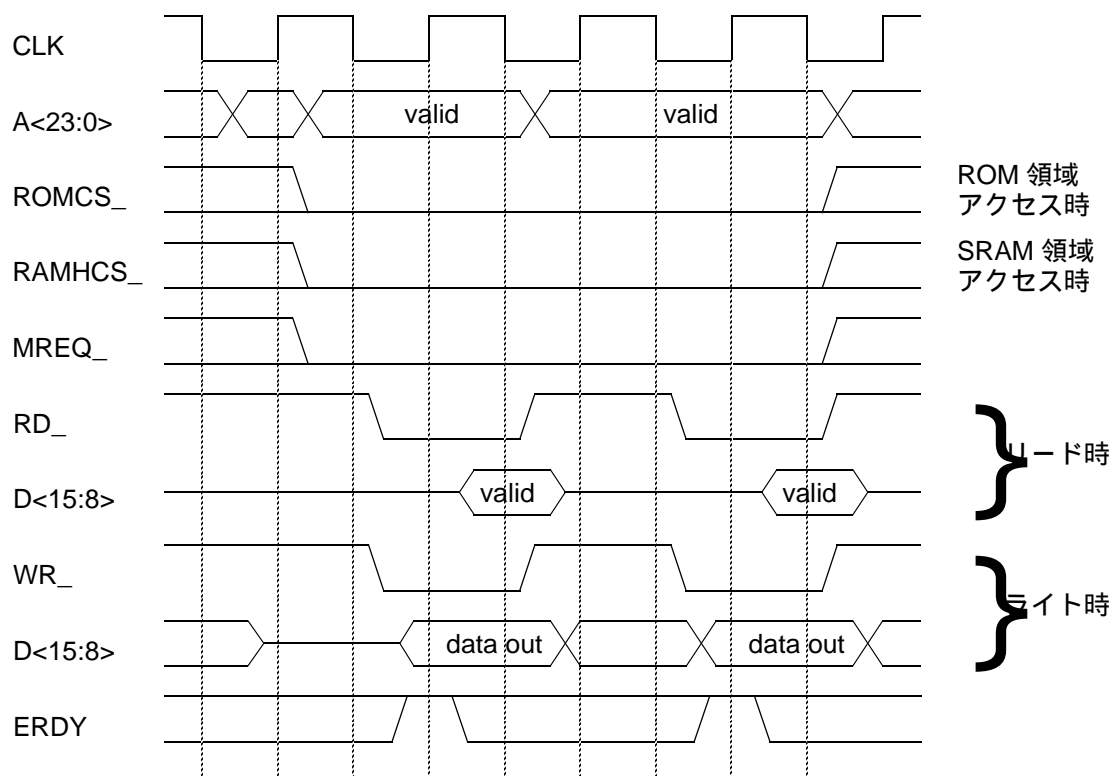


図 6-2-4a データ・リード/ライト・サイクル(ROM/SRAM 領域 8 ビットモード、0 ウェイト)

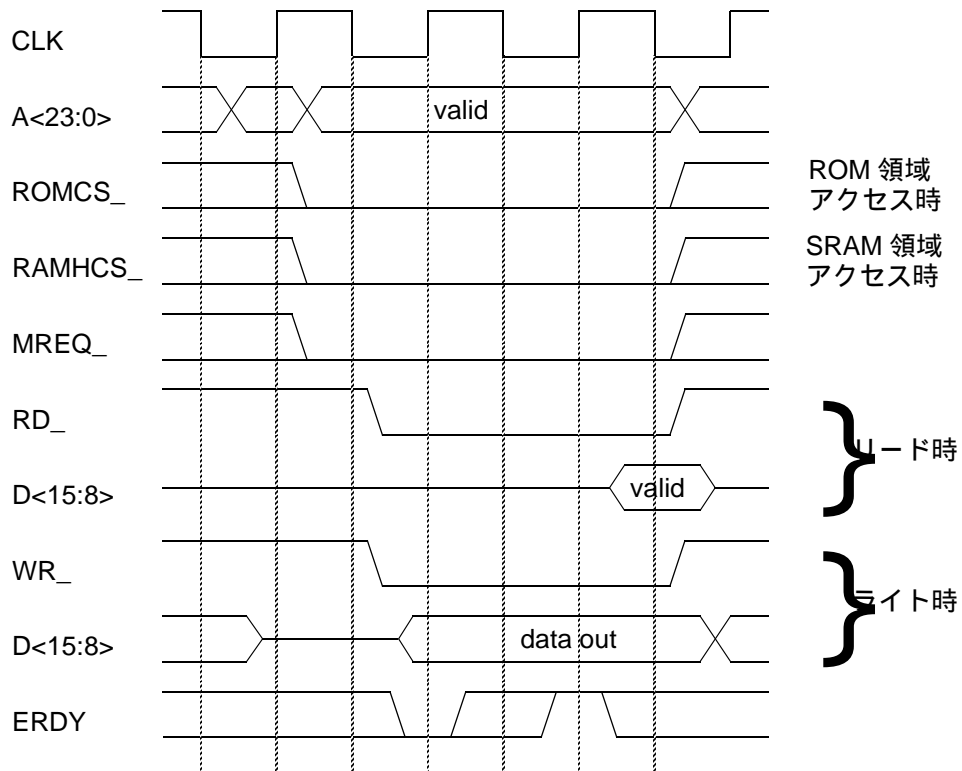


図 6-2-4b 8 ビットデータ・リード/ライト・サイクル(ROM/SRAM 領域 8 ビットモード、1 ウェイト)

6.3 I/O アクセスタイミング

I/O 領域におけるデータのリード/ライト・サイクルを図 6-3a, 6-3b に示します。

基本サイクルは、3 クロックです。データバスは 8 ビット幅で、上位バイト側の D<15:8> を使用します。尚、リセット直後は 4 ウェイトの設定になっています。

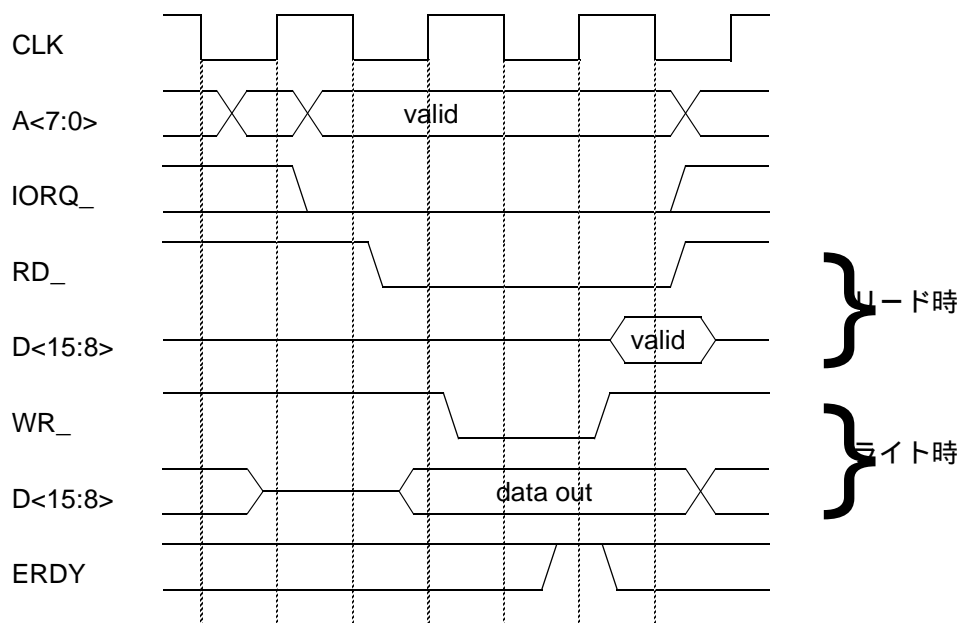


図 6-3a I/O・リード/ライト・サイクル(0 ウェイト)

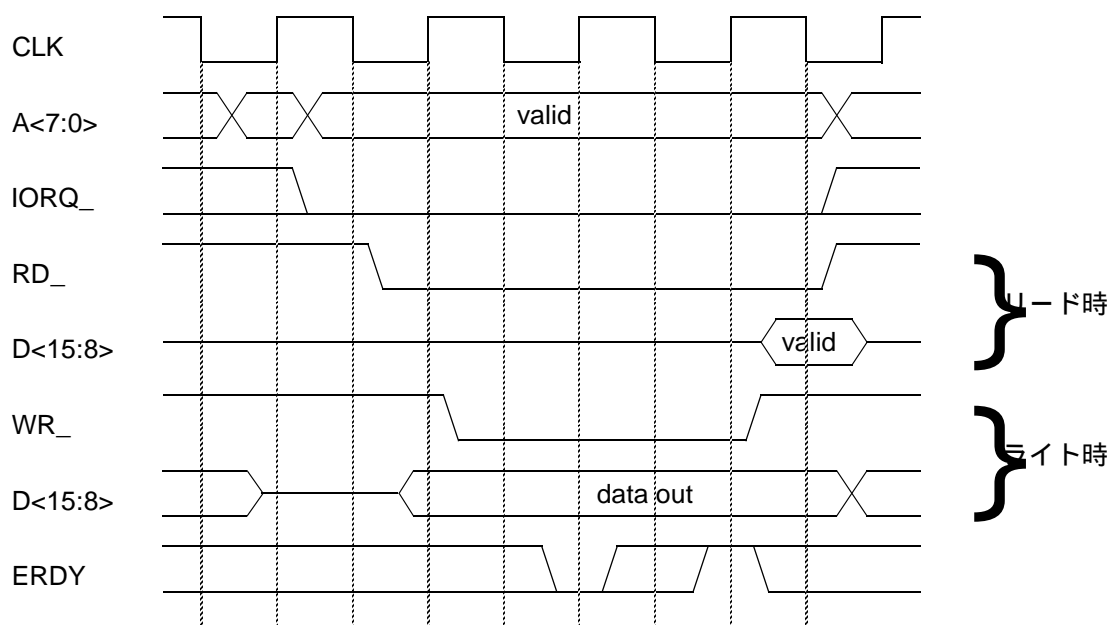


図 6-3b I/O・リード/ライト・サイクル(1 ウェイト)

7. 互換ボックス

7.1 概要

バイナリレベルでの Z80 上位互換性を実現するため、Z80 アドレス空間領域（64K バイト）に相当する 000000H ~ 00FFFFH においては互換性を最優先とした仕様になっています。これにより、Z80 ソフトウェア資産の継承とアドレス空間の拡大を同時に実現しました。Z80 の命令セットをそのまま使用できるだけでなく、Z80 で作成したプログラム資産をそのまま移植することも可能です。また、互換性と無関係にすべてのアドレス空間で自由にプログラム上の行き来が可能ですので、新規プログラムへの移行をスムーズに行うことができます。

7.2 互換ボックスの定義

アドレス領域 000000H ~ 00FFFFH の 64K バイト・アドレス空間を「互換ボックス」と言います。命令コードの先頭バイトが 000000H ~ 00FFFFH に存在すれば互換ボックスの中に居ると判断します。この領域内においては従来の Z80 プログラムを変更無しに使用することが可能です。

一方、互換ボックス外の領域（以下、「拡張領域」と呼びます）においては、特別な制約無しに自由にプログラムを作成することが可能です。尚、従来命令についてはアドレス上位 8 ビット $A<23:16>$ にデフォルト接続された拡張レジスタの値を出力します。詳細については、第 3 章（KC160 CPU コア）を参照してください。

また、モード 2、モード 3 割り込みのスタートアドレステーブルは必ず互換ボックス内に配置する必要がありますのでご注意ください。ちなみに、モード 0、モード 1 割り込み及びノンマスカブル割り込み（NMI）のルーチンのスタート番地も互換ボックス内に配置されます。

7.3 互換ボックスの必要性

Z80 では、64K バイトのアドレス空間の中にプログラム領域とデータ領域を同居させていました。しかし、KC160 においては 16M バイトアドレス空間となり、拡張した上位 8 ビットによりそれらの領域を区別して指定することができます。PP レジスタがプログラム領域、ZP レジスタがデータ領域を指定します。

このように、アドレス $A<23:16>$ を新たに定義したため、従来命令に関しては、個々の命令動作は Z80 と互換でも命令列（プログラム）として同一の結果にならない場合が発生します。その例を図 7-3 に示します。

このプログラムは Z80 用に作成されたものです。よって、プログラム領域とデータ領域が同一ページ内に同居した形式になっています。拡張領域に配置した場合、Z80 上では問題なく動作するプログラムが KC160 上では正常に動作しません。プログラムカウンタの上位 8 ビットはプログラム領域を示す PP レジスタの値ですが、HL レジスタの上位 8 ビットにはデフォルトでデータ領域を示す ZP レジスタの値が出力されるからです。結果的に目的のアドレスをアクセスできずに、誤ったアドレスに格納されているデータを読み込むことになり、誤動作の原因になります。ここで、目的のアドレスがプログラム領域を指定するように LD A, (HL) を LD A, (PHL) と書き換えれば正常に動作しますが、そのためにはプログラムの書き換えが必要になります。

これに対応するためには、プログラムカウンタの指しているアドレスと HL レジスタで指定しようとしているアドレスが一致している必要があります。すなわち、プログラム領域を指定する ZP とデータ領域を指定する PP とが同一である必要があります。よって、最も確実な方法として、特定の状態において $A<23:16>$ に出力される値を同一値 00H とすることにしました。これが互換ボックスです。図中の互換ボックス側での動作に示すように、互換ボックス内においては同じプログラムが問題なく動作します。アドレスの上位 8 ビットに固定値 00H が出力されるため、HL レジスタの値が実際にアクセスすべきアドレス（TBL1：000123H 番地）を指しています。

以上、互換ボックスの採用により、既存の Z80 ソフトウェア資産のルーチンをそのままに、新規に C 言語等で作成したルーチンによる機能追加が容易に可能となります。

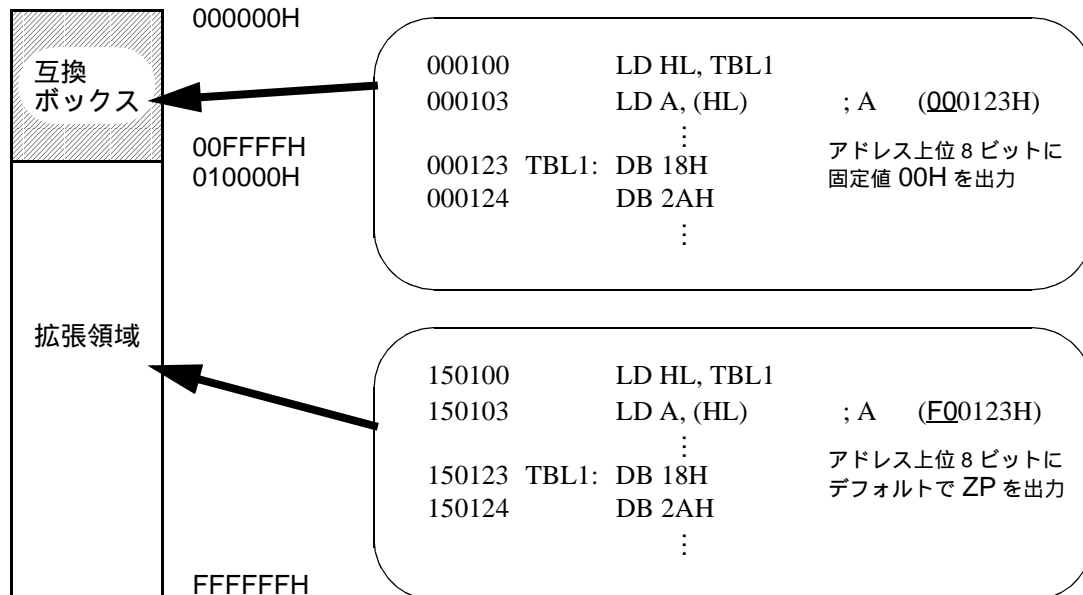


図 7-3 プログラム動作例

7.4 命令動作

各命令の動作については、前述したように、互換ボックス内においては、Z80 で定義されていた命令（従来命令）はアドレス A<23:16> に 00H を出力します。また、KC160 で追加定義された命令（拡張命令）は影響を受けず互換ボックス内外で一部を除き同じ動作を行います。

詳細については、以下の内容を参照してください。

従来命令との関係

互換ボックス内では、A<23:16> を 00H にします。

互換ボックス外では、それぞれの命令の右側に示した内容がデフォルト値として A<23:16> に出力されます。

8bit LOAD

LD (HL), nn	: ZP
LD (HL), r	: ZP
LD (IX+d), nn	: ZP, ZP ± 1
LD (IY+d), nn	: YP, YP ± 1
LD (IX+d), r	: ZP, ZP ± 1
LD (IY+d), r	: YP, YP ± 1
LD (nnnn), A	: ZP
LD (BC/DE), A	: ZP
LD A, (nnnn)	: ZP
LD A, (BC/DE)	: ZP
LD r, (HL)	: ZP
LD r, (IX+d)	: ZP, ZP ± 1
LD r, (IY+d)	: YP, YP ± 1

16bit LOAD

LD (nnnn), HL/IX/IY : ZP
 LD (nnnn), ss : ZP
 LD HL/IX/IY, (nnnn) : ZP
 LD ss, (nnnn) : ZP
 POP ss/IX/IY : ZP
 PUSH ss/IX/IY : ZP

EXCHANGE, BLOCK TRANS, BLOCK SEARCH

EX (SP), HL/IX/IY : ZP
 LDD/LDDR : ZP
 LDI/LDIR : ZP
 CPD/CPDR : ZP
 CPI/CPIR : ZP

ADD, SUBTRACT, COMPARE

ADD/ADC A, (HL) : ZP
 ADD/ADC A, (IX+d) : ZP, ZP \pm 1
 ADD/ADC A, (IY+d) : YP, YP \pm 1
 SBC A, (HL) : ZP
 SBC A, (IX+d) : ZP, ZP \pm 1
 SBC A, (IY+d) : YP, YP \pm 1
 AND/OR/XOR (HL) : ZP
 AND/OR/XOR (IX+d) : ZP, ZP \pm 1
 AND/OR/XOR (IY+d) : YP, YP \pm 1
 INC/DEC (HL) : ZP
 INC/DEC (IX+d) : ZP, ZP \pm 1
 INC/DEC (IY+d) : YP, YP \pm 1
 SUB/CP (HL) : ZP
 SUB/CP (IX+d) : ZP, ZP \pm 1
 SUB/CP (IY+d) : YP, YP \pm 1

ROTATE, SHIFT

RLD/RRD : ZP
 RL/RLC (HL) : ZP
 RL/RLC (IX+d) : ZP, ZP \pm 1
 RL/RLC (IY+d) : YP, YP \pm 1
 RR/RRC (HL) : ZP
 RR/RRC (IX+d) : ZP, ZP \pm 1
 RR/RRC (IY+d) : YP, YP \pm 1
 SLA/SRA/SRL (HL) : ZP
 SLA/SRA/SRL (IX+d) : ZP, ZP \pm 1
 SLA/SRA/SRL (IY+d) : YP, YP \pm 1

BIT SET/RESET

BIT/SET/RES b, (HL) : ZP
 BIT/SET/RES b, (IX+d) : ZP, ZP \pm 1
 BIT/SET/RES b, (IY+d) : YP, YP \pm 1

JUMP

JP (HL)	: ZP
JP (IX)	: ZP
JP (IY)	: YP
JP cc, nnnn	: 配置上の次命令のアドレス A<23:16>
JP nnnn	: 配置上の次命令のアドレス A<23:16>
CALL cc, nnnn	: 配置上の次命令のアドレス A<23:16>
CALL nnnn	: 配置上の次命令のアドレス A<23:16>
RST nn	: 00H

拡張命令との関係

新規定義命令は互換ボックス内外で一部を除き同じ動作をします。
それぞれの命令の右側に示した内容が A<23:16> に出力されます。

8bit LOAD

LDF A, (nnnnnn)	: nn
LDF (nnnnnn), A	: nn

16bit LOAD (ss = BC, DE, HL, SP, IX, IY)

LDF ss, (nnnnnn)	: nn
LDF (nnnnnn), ss	: nn

24bit LOAD

LDF (nnnnnn), XIX/YIY/AHL	: nn
LDF XIX/YIY/AHL, (nnnnnn)	: nn
PUSH XIX/YIY/AHL	: ZP (SP アドレス)
POP XIX/YIY/AHL	: ZP (SP アドレス)

EXCHANGE, BLOCK TRANS, BLOCK SEARCH

CPD/CPDR X	: XP, XP - 1
CPI/CPIR X	: XP, XP + 1
IND/INDR X	: XP, XP - 1
INI/INIR X	: XP, XP + 1
OUTD/OTDR X	: XP, XP - 1
OUTI/OTIR X	: XP, XP + 1
LDD/LDDR XY	: XP, XP - 1 (ソース側アドレス)
	: YP, YP - 1 (ディスティネーション側アドレス)
LDI/LDIR XY	: XP, XP + 1 (ソース側アドレス)
	: YP, YP + 1 (ディスティネーション側アドレス)

JUMP

CALL3 nnnnnn	: ZP (SP アドレス)
	: nn (飛び先アドレス)
RET3	: ZP
RETN3	: ZP

拡張命令の例外

新規定義命令のうち、下記命令については互換ボックス内外で異なる動作をします。

互換ボックス内では、A<23:16> に 00H が出力されます。

互換ボックス外では、それぞれの命令の右側に示した内容が A<23:16> に出力されます。

16bit LOAD (ss = BC, DE, HL, SP, IX, IY)

LD ss, (*IX+d)	: *, * ± 1
LD ss, (*IY+d)	: *, * ± 1
LD (*IX+d), ss	: *, * ± 1
LD (*IY+d), ss	: *, * ± 1
LD ss, (SP+d)	: ZP, ZP ± 1
LD (SP+d), ss	: ZP, ZP ± 1

アドレス以外に影響する命令

以下の演算命令は、互換ボックス内ではページレジスタへのキャリーは常に伝わず、互換ボックス外ではページレジスタへのキャリーが伝わります。

ADD *HL/*IX/*IY, ss (ss = BC, DE, HL, SP, IX, IY)

ADC/SBC *HL, ss (ss = BC, DE, HL, SP)

INC/DEC *ss (ss = BC, DE, HL, SP, IX, IY)

その他

互換ボックス内では、以下の各命令コードは単独の命令として定義されます。動作については NOP 命令と同じです。したがって、その命令の実行直後に割り込みを受け付けることができます。

互換ボックス外では、以下の各命令コードは単独の命令としては使用禁止です。

尚、それぞれの命令の右側にはオペコード (HEX 表示) を示しました。

LD A, A	: 7F
LD B, B	: 40
LD C, C	: 49
LD D, D	: 52
LD E, E	: 5B
LD H, H	: 64
LD L, L	: 6D

尚、上記以外の命令は互換ボックス内外で同じ動作となります。

7.5 互換ボックス内での制約事項

互換ボックスについての制約事項を以下にまとめましたので、充分考慮の上ご使用ください。

リセット解除後は、互換ボックス内の 000000H 番地からプログラムが動作開始する。

RST 命令のルーチンの先頭アドレスは互換ボックス内になる。

例えば、RST 38H 命令の場合、飛び先アドレスは 000038H となる。

ノンマスカブル割り込み (NMI) の割り込みルーチンの先頭アドレスは 000066H となる。

マスカブル割り込みのスタートアドレステーブルは互換ボックス内に配置される。

(I<15:8> = 00H のため固定)

その他制約事項ではありませんが、SRAM/DRAM 領域に配置された 4K バイトあるいは 32K バイト分の RAM をイメージとしてアクセス可能な構造となっています。詳細については、第 5 章 (アドレス空間) 及び第 12 章 (動作モード設定) を参照してください。