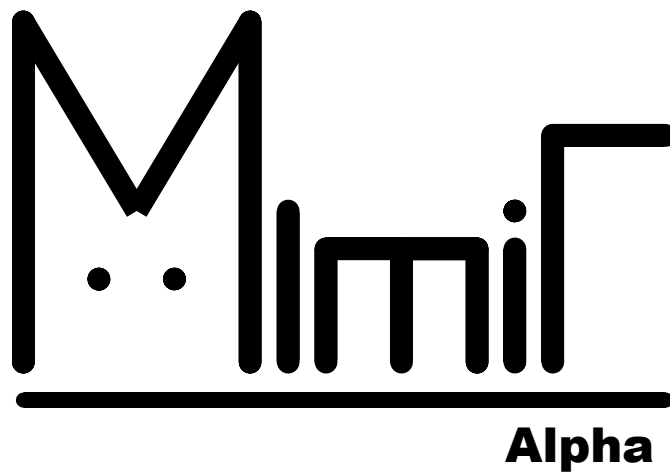


MiMicVM 仕様書 rev 4

2011.11.16 nyatla



本書は、MiMicVM の仕様書である。MiMicVM の機能、言語仕様を定義する。

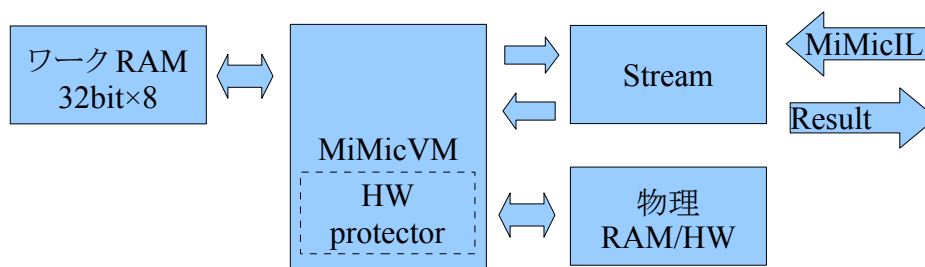
Rev	日付	
1	2011/09/03	新規作成
2	2011/10/04	仕様更新、追記。
3	2011/10/12	命令追加
4	2011.11.16	CALL 命令追加。MiMicVM のリターンコードを更新

目次

1 MiMicVM.....	3
1.1 MiMicVM に実装する機能.....	3
1.2 MiMicVM に実装しない機能.....	3
2 MiMicBC.....	4
2.1 記号定義.....	4
2.2 MiMicInstructionSet.....	5
2.2.1 ビット演算命令.....	5
2.2.2 ビット操作命令.....	5
2.2.3 算術命令.....	6
2.2.4 メモリアクセス命令.....	6
2.2.5 STREAM アクセス命令.....	7
2.2.6 ロード命令.....	7
2.2.7 分岐命令.....	7
2.2.8 制御命令.....	8
2.2.9 定義命令.....	9
3 MiMicIL.....	9
4 MiMicVM の動作仕様.....	10
4.1 MiMicVM の状態図.....	10
4.1.1 サスペンド状態.....	10
4.1.2 ロード状態.....	10
4.1.3 実行状態.....	10
4.2 MiMicVM の入力.....	11
4.3 MiMicVM の出力.....	11
4.4 終了コード.....	12
5 Appendix 1. MiMicVM HTTP Interface.....	14
5.1 入力方式.....	14
5.1.1 REST リクエスト.....	14
5.2 出力方式.....	15
5.2.1 JSON.....	15
5.3 Example. REST+JSON シーケンスのモジュール構成.....	16

1 MiMicVM

MiMicVM は、MiMic に搭載される仮想マシンである。8×32bit のワークメモリ、単一のストリーム、物理メモリに対するインタフェースを持つ。MiMicVM の実行手順は、MiMicIL で記述する。この VM の目的は、小規模な実行手続きを、MCU で一時的に実行することである。大規模なプログラム構造を、定常的に MCU で実行することは、それほど目的にしている。



MiMicVM のブロック図

1.1 MiMicVM に実装する機能

- MCU の管理するハードウェアへのアクセス機能。
- ストリームからのデータ入出力機能。
- 簡単な演算。
- MiMicIL の解釈。

1.2 MiMicVM に実装しない機能

- ハードウェア操作に対する保護機能(変更の可能性有り)
- 排他機能(変更の可能性有り)

MiMicVM の実行形式は実装依存である。MiMicIL を逐次、または一括変換して、0 番目の命令から、EXIT に到達するまで実行する。

MiMicVM に入力する命令の終端には、EXIT 命令が必要である。終端の EXIT 命令が無い場合、MiMicVM はエラーを返却するか、EXIT を補完して実行する。

2 MiMicBC

MiMicBC は、MiMicVM で実行可能な、バイトコードである。MiMicBC は次に定義する記号の組み合わせであり、テキストストリームを使って、MiMicVM に一方的に入力される。

MiMicBC は、2 つのブロック MiMicTXT (実行コード) MiMicDB (データ) を連結したものである。MiMicTXT は実行前に全て読み込まれるが、MiMicDB は実行時にストリームから読み込まれる。MiMicTXT の長さは設計時に確定するが、MiMicDB の長さは確定する必要が無く、可変長である。

2.1 記号定義

記号	表現	機能
MiMicBC	[[:MiMicTXT:][:MiMicDB:]]	MiMicBC は、実行部の MiMicTXT と、データ部の MiMicDB を連結したものである。
MiMicTXT	[[:MiMicIST:]*[:MiMicIST_END:]]	MiMicTXT は、MiMicBC の実行部である。オペコードとオペランドを 1 個以上格納する。MiMicVM は、MiMicTXT の終端を検出した時点で、実行を開始することが出来る。終端は、".END" インストラクションで定義する。
MiMicDB	[[:HEX8:]]*	バイト単位のデータブロックである。終端は定義されない。
MiMicIST	[[:OP:][:DEF:][:OPR:]]*	命令単位である。1 個の OP (オペコード) か DEF (定義コード) と複数の OPR (オペランド) を連結したものである。OPR の構造は、OP で決定する。組み合わせは、MiMicInstructionSet で定義する。
MiMicIST_END	.E	MiMicBC の .END インストラクションである。
OPR	[[:HEX32:][:WM:]]	オペランドである。
OP	[A-Z]{2}	オペコードである。この命令は、VM の実行コードに変換される。
DEF	\[A-Z]	定義コードである。この命令は、VM の定義値を設定する。
HEX8	[0-9a-f]{2}	8bit の 16 進数である。
HEX16	未定義	未定義
HEX32	[0-9a-f]{8}	32bit の 16 進数である。
WM	[0-9a-f]{2}	ワークメモリの識別子である。1 バイトの 16 進数で、ワークメモリを指定する。

2.2 MiMicInstructionSet

MiMicInstructionSet で定義する、オペコードとオペランドの組み合わせを、以下に示す。
グレーアウトした項目は、現在サポートしない。

2.2.1 ビット演算命令

命令	OP 値	OPR セット	機能
AND	AA	[WM][WM]	2つの[WM]の内容を AND 演算して、先頭の WM に結果を設定する。
	AB	[WM][HEX32]	[WM]の内容と[HEX]の値を AND 演算して、先頭の WM に結果を設定する。
	AC-AD	Undefined	
OR	AE	[WM][WM]	2つの[WM]の内容を OR 演算して、先頭の WM に結果を設定する。
	AF	[WM][HEX32]	[WM]の内容と[HEX]の値を OR 演算して、先頭の WM に結果を設定する。
	AG-AH	Undefined	
XOR	AI	[WM][WM]	2つの[WM]の内容を XOR 演算して、先頭の WM に結果を設定する。
	AJ	[WM][HEX32]	[WM]の内容と[HEX]の値を XOR 演算して、先頭の WM に結果を設定する。
	AK-AL	Undefined	
NOT	AM	[WM]	[WM]のビットを反転する。
	AN-AZ	Undefined	

2.2.2 ビット操作命令

命令	OP 値	OPR セット	機能
SHL	BA	[WM][HEX8]	[WM]の内容を、[HEX8]だけ左にビットシフトする。
	BB	[WM][WM]	1番目の[WM]の内容を、2番目の[WM]の数だけ左にシフトする。
	BC-BD	Undefined	
SHR	BE	[WM][HEX8]	[WM]の内容を、[HEX8]だけ右にビットシフトする。
	BF	[WM][WM]	1番目の[WM]の内容を、2番目の[WM]の数だけ右にシフトする。
	BG-BZ	Undefined	

2.2.3 算術命令

命令	OP 値	OPR セット	機能
ADD	CA	[WM][WM]	2つの[WM]の内容を加算して、先頭のWMに結果を設定する。
	CB	[WM][HEX32]	[WM]の内容と[HEX]の値を加算して、先頭の[WM]に結果を設定する。
	CC-CD	Undefined	
SUB	CE	[WM][WM]	1番目の[WM]の内容から2番目の[WM]を減算して、1番目の[WM]に結果を設定する。
	CF	[WM][HEX32]	[WM]の内容から[HEX]の値を減算して、先頭の[WM]に結果を設定する。
	CG-CH	Undefined	
MUL	CI	[WM][WM]	1番目の[WM]の内容から2番目の[WM]を減算して、1番目の[WM]に結果を設定する。
	CJ	[WM][HEX32]	[WM]の内容から[HEX]の値を減算して、先頭の[WM]に結果を設定する。
	CK-CZ	Undefined	
INC			
DEC			

2.2.4 メモリアクセス命令

命令	OP 値	OPR セット	機能
MGET	DA	[WM][HEX32]	物理メモリの[HEX32]のアドレスから、32bit 値を読み出して、[WM]に格納する。 ※[HEX32]は、ブラケット[]で囲むことを推奨する。
	DB	[WM][WM]	2番目の[WM]のアドレスから、32bit 値を読み出して、1番目の[WM]に格納する。
	DC-DD	Undefined	
MPUT	DE	[WM][HEX32]	物理メモリの[HEX32]のアドレスに、[WM]の32bit 値を書き込む。
	DF	[WM][WM]	2番目の物理メモリのアドレスへ、1番目の[WM]の32bit 値を書き込む。 ※書き込み方向が→であることに注意
	DG-DZ	Undefined	

2.2.5 STREAM アクセス命令

命令	OP 値	OPR セット	機能
SGET	EA	[WM]	ストリーム(MiMicDB)から、HEX32 のデータブロックを[WM]へ読み込む。
	EB-ED	Undefined	
SPUT	EE	[WM]	[WM]の内容を、HEX32 でストリームに書き込む。 バイトオーダーはビッグエンディアンである。
	EF	[HEX32]	[HEX32]の値を、HEX32 でストリームに書き込む。 バイトオーダーはビッグエンディアンである。
	EG-EZ	Undefined	

2.2.6 ロード命令

命令	OP 値	OPR セット	機能
LD	FA	[WM][WM]	先頭の WM へ後方の WM の内容をコピーする。
LD	FB	[WM][HEX32]	WM へ HEX32 の値をセットする。
	FC-FZ	Undefined	

2.2.7 分岐命令

命令	OP 値	OPR セット	機能
JZ	GA	[WM][HEX8]	[WM]が 0 なら、[HEX8]番目のジャンプテーブルで示される命令へジャンプする。
	GB-GD	Undefined	
JEQ	GE	[WM][WM][HEX8]	WM1 と WM2 の内容が等しければ、[HEX8]番目のジャンプテーブルで示される命令へジャンプする。
	GF-GZ	Undefined	

2.2.8 制御命令

命令	OP 値	OPR セット	機能
NOP	ZA		何もしない。
	ZB	HEX8	HEX8 ミリ秒間、VM の動作を停止する。
	ZC-ZD	Undefined	
CALL	ZE	[WM]	WM の値の組み込み関数をコールする。
	ZF	HEX32	指定番号の組み込み関数をコールする。 組込関数の仕様については、Appendix. native call interface を参照すること。
	ZF-ZX	Undefined	
ABORT	ZY	Undefined	
EXIT	ZZ		終了コード 0 で MiMicIL を終了する。

2.2.9 定義命令

命令	OP 値	OPR セット	機能
.LABEL	.L	[HEX8]	[HEX8]番のラベルを定義する。この値は、ジャンプテーブルに登録される。ラベルは、プログラム全体で8個まで定義できる。
.END	.E		現在のセクションを終了する。この命令は、MiMicIL のセクション終了時にかならず付けること。
.D32	-	[HEX32]	32ビットデータを書き込む。

3 MiMicIL

MiMicBC は、MiMicIL で記述する。MiMicIL は命令、擬似命令、コメントの組み合わせで表現する。命令はインストラクションセットの定義に従い、MiMicVM の状態を更新する。擬似命令は、MiMicIL の構造を記述する。

MiMicIL の書式は、以下の通りである。

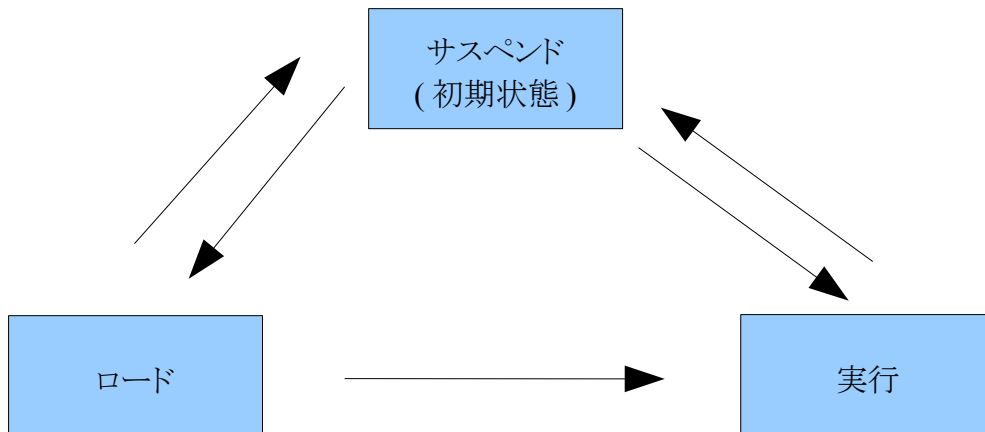
記号	表現	機能
TEXT	[LINE:]+	MiMicIL を記述したテキスト。1 行以上の[LINE:]からなる。
LINE	([:SP:]*[:VOP:][:OP:]+[:SP:]*;[:COMMENT:])?[\r\n]+	LINE は、命令、または擬似命令と、セミコロンで区切られたコメントである。コメント行だけの場合もある。
OP	[A-Z]+[:SP:]+[:OPR:]	命令語である。命令は、オペコードとオペランドを[:SP:]で区切って連結したものである。オペコードの種類は、MiMicInstructionSet の命令に一致する。
VOP	\[A-Z]+[:SP:]+[:OPRS:]	擬似命令後である。擬似命令は、先頭に”.”をもち、その他の構造は[:OP:]と同じである。
OPRS	([:VALUE:][:SP:])*[:SP:][:VALUE:]	オペランドである。オペランドは、値を、で連結したものである。
VALUE	[:HEX32:][:WM:][:UINT]	オペランドの値である。32bit の 16 進数、10 進数、ワークメモリ番号のいずれかである。
HEX32	0x[0-9a-f]{1,8}	16 進数である。
WM	#[0-9a-f]{1,2}	プレフィクスに#を付けた、2 桁未満の 16 進数である。
UINT	[0-9]+	最大値が INT_MAX の整数値である。

4 MiMicVM の動作仕様

MiMicVM の動作について説明する。

4.1 MiMicVM の状態図

MiMicVM には、サスペンド/ロード/実行の 3 状態がある。状態間の遷移を、図に示す。



4.1.1 サスペンド状態

この状態は、MiMicVM が停止している状態である。初期状態である。ストリームから MiMicIL を受信すると、ロード状態に遷移する。

4.1.2 ロード状態

この状態は、ストリームから MiMicIL を読み出している状態である。MiMicVM は、MiMicIL を命令単位でコンパイルして、内部に蓄積する。MiMicIL の EXIT インストラクションを検出すると、実行状態に遷移する。何らかのエラーが発生した場合には、コンパイルしたコードを破棄して、サスペンド状態に戻る。外部アクセスは、ストリームからの読み出しのみを行う。

4.1.3 実行状態

この状態は、MiMicIL のインストラクションを実行している状態である。MiMicVM は、蓄積済みの MiMicIL を先頭から実行する。エラーが発生するか、MiMicIL の EXIT インストラクションを検出すると、サスペンド状態に戻る。外部アクセスは、ストリームからの読み出し/書込み、物理メモリからの読み出し/書込み、ワークメモリの操作、すべてを行う。ストリームからの入力、MiMicIL のインストラクションが実行された時点で行われる。ストリームにデータが無い場合には、一定時間、その入力を待つことがある。

4.2 MiMicVM の入力

MiMicVM に入力するストリームは、MiMicIL に一致していなければならない。他のプロトコルでラップした場合は、入力前に構造文章を取り除く必要がある。

4.3 MiMicVM の出力

MiMicVM の出力は、MiMicIL の出力値である。出力値は、32bit 単位である。

MiMicIL の出力形式は、入力した MiMicIL により定義されるので、出力ストリーム解釈の責務は受信側にあるものとする。

他のプロトコルでラップする場合には、32bit 毎のデータとして処理する。

4.4 終了コード

MiMicVM は、バイトコードの実行後に終了コードを返す。

エラーコードは、ビットフィールド毎に意味を持つ。

31:エラーフラグ。1 でエラーである。

30-24:予約

23-16:モジュール ID (0x01:MiMicVM,0x02:MiMicJson)

15-8:予約

7-0:エラー ID

最低でも 0x00000000 と 0x80000000 の 2 種類を返すこと。それ以上の詳細は、実装による。

設定値	意味
0x00000000	MVM_OK MiMicBC の実行に成功した。
0x80000000	MVM_NG 原因不明のエラー
0x8001xxxx	MiMicVM
0x80010000	MVM_RUNTIME_NG MiMicVM の実行時エラー。
0x80010001	MVM_RUNTIME_NG_CONVERT MiMicBC の解釈エラー。
0x80010002	MVM_RUNTIME_NG_BAD_SECTION セクション区切り異常 .END が見つからないなど
0x80010003	MVM_RUNTIME_NG_BC_TOO_LONG バイトコードが長すぎる。(領域不足)
0x80010004	MVM_RUNTIME_NG_WM_RANGE 範囲外の WM 指定
0x80010011	MVM_RUNTIME_NG_UNKNOWN_LABEL 未知のジャンプ先
0x80010021	MVM_RUNTIME_NG_UNKNOWN_CALL コールしようとしたネイティブ関数が見つからない。
0x80010022	MVM_RUNTIME_NG_CALL ネイティブ関数が異常終了した。
0x8002xxxx	MiMic JSON
0x80020000	MVM_JSON_NG MiMicVM JSON モジュール内のエラー
0x80010001	MVM_JSON_NG_VERSION JSON バージョンに対応していない。

5 Appendix 1. MiMicVM HTTP Interface

MiMicVM のストリームを CGI の HTTP ストリームに接続することで、MiMicVM のリモート実行環境を作ることが出来る。ここでは、HTTP ストリームとの接続方法について、説明する。

5.1 入力方式

入力方式には2種類がある。1 つは REST による接続、2 つめは XMLRPC による接続である。REST による接続は、送信可能な MiMicIL の長さに制限がある。XMLRPC による送信は、alpha1 では未対応である。

どちらの場合でも、MiMicVM は入力された MiMicBC の内容を信頼しては”ならない”。実行前に、不正な構文をチェックし、出力方式で定める方式により、エラーコードを送信しなければならない。

5.1.1 REST リクエスト

URL パラメータ	M/O/R	設定値	意味
v	M	1	MiMicBC のバージョンを指定する。alpha1 では 1 を指定する。
o	O	[jx]	出力形式を指定する。現在は、j のみが指定できる。
		j	JSON 形式の出力を返す。
		x	予約値
bc	M	[:MiMicBC:]	MiMicBC の内容を指定する。

REST リクエストについて、以下の制約を定める。

- URL パラメータの順序については、v パラメータは必ず先頭である必要がある。
- v=1 の場合は、URL パラメータの順序は変更できない。
- REST リクエストの全長は、1024 バイトまでは保障する。それ以上は実装依存である。
- v パラメータが想定している値と異なる場合には、エラーを返却する。
- 不明な URL パラメータを、無視する。

5.2 出力方式

出力方式は、JSONとXMLが選択できる。どちらの形式も、MiMicVMの出力コードを、短い構造文章でラップしたものである。alpha1では、JSON形式のみに対応する。

構造文章は、バージョン、エラーコード、出力データの3文章で構成する。

5.2.1 JSON

JSONの構造は以下の通りである。出力するJSONは、`eval("(" + output + ")");`でJavascriptオブジェクトに変換できる。JSONに拡張メンバを追加する場合は、プレフィクス"x_"をメンバ名に含めること。

メンバ	type	設定値	意味
version	string	MiMicVM/1.0;Json/1.0	処理系の識別子である。
result	int	MiMicVMの終了コード。	
stream	array(int)		MiMicVMの出力したストリームである。32Bit HEX表現の配列である。ResultがMVM_OKのときに有効な値がある。

5.3 Example. REST+JSON シーケンスのモジュール構成

HTTPClient 側のモジュールは、Javascript (又は任意の言語), HTTPServer 側のモジュールは C 言語で実装する。MiMicIL Assembler は Javascript で実装したライブラリーとして提供する。

