

計算機HOGE-II用アセンブリ言語命令集

計算機システム II 補足資料

2014.09.24

★1 この命令集の見方

例えば, HOGE という命令の説明が

HOGE

ラベル	オペコード	オペランド
[label]	HOGE	GR, addr

のようになっていた場合, この命令には

- オペランドは必ず2つ指定しなければならない
- ラベルは指定してもしなくてもよい

ということの意味している. [fuga] というのは, fuga はあってもなくてもよいということ. また, GR というオペランドには汎用レジスタの名前 (GR0 から GR7 までのいずれか) を指定し, addr というオペランドには番地 (ラベルで表すことが多い) を指定する.

★2 アセンブラ命令

アセンブラに対する指示. 実際の機械語命令には対応しない.

START

ラベル	オペコード	オペランド
label	START	

「ここからプログラムがはじまるよ」

END

ラベル	オペコード	オペランド
	END	

「ここでプログラムおわり」

DC

DC: Define Constant

ラベル	オペコード	オペランド
[label]	DC	const

メモリに1語分の領域を確保して, そこに定数 const を入れておく

DS

DS: Define Storage

ラベル	オペコード	オペランド
[label]	DS	n

メモリに n 語分の領域を確保

★3 機械語命令

実際の機械語命令 (1語または2語) に対応したもの.

LD

LD: ロード, LoaD. 1語 (2語のLDもあるので注意)

ラベル	オペコード	オペランド
[label]	LD	GR1, GR2

レジスタ GR2 の内容をレジスタ GR1 にセットする (コピーする). $GR1 \leftarrow (GR2)$

LD

LD: ロード, LoaD. 2語 (1語のLDもあるので注意)

ラベル	オペコード	オペランド
[label]	LD	GR, addr

番地 addr の内容をレジスタ GR にセットする (メモリの値をレジスタにコピーする)

ST

ST:ストア, SToRe. 2語

ラベル	オペコード	オペランド
[label]	ST	GR, addr

レジスタ GR の内容を番地 addr に転送する (レジスタの値をメモリにコピーする)

ADDA

ADDA:算術加算, ADD Arithmetic. 1語

ラベル	オペコード	オペランド
[label]	ADDA	GR1, GR2

GR1 と GR2 の値を用いて算術加算 (-32768 から 32767 までの符号付き整数とみなして加算) を行い, 結果を GR1 にセットする. $GR1 \leftarrow (GR1) + (GR2)$

ADDL

ADDL:論理加算, ADD Logical. 1語

ラベル	オペコード	オペランド
[label]	ADDL	GR1, GR2

GR1 と GR2 の値を用いて論理加算 (0 から 65535 までの符号なし整数とみなして加算) を行い, 結果を GR1 にセットする

SUBA

SUBA:算術減算, SUBtract Arithmetic. 1語

ラベル	オペコード	オペランド
[label]	SUBA	GR1, GR2

GR1 と GR2 の値を用いて算術減算を行い, 結果を GR1 にセットする. $GR1 \leftarrow (GR1) - (GR2)$

SUBL

SUBL:論理減算, SUBtract Logical. 1語

ラベル	オペコード	オペランド
[label]	SUBL	GR1, GR2

GR1 と GR2 の値を用いて論理減算を行い, 結果を GR1 にセットする

JUMP

JUMP:無条件分岐, 2語

ラベル	オペコード	オペランド
[label]	JUMP	addr

プログラムレジスタに番地 addr をセットする

JZE

JZE:ゼロ分岐, Jump on ZERo. 2語

ラベル	オペコード	オペランド
[label]	JZE	addr

ゼロフラグが1なら番地 addr へ分岐 (直前の演算結果が0なら分岐)

JNZ

JNZ:非ゼロ分岐, Jump on Non Zero. 2語

ラベル	オペコード	オペランド
[label]	JNZ	addr

ゼロフラグが0なら番地 addr へ分岐 (直前の演算結果が0でなければ分岐)

JPL

JPL:正分岐, Jump on PLus. 2語

ラベル	オペコード	オペランド
[label]	JPL	addr

サインフラグが0かつゼロフラグが0なら番地 addr へ分岐 (直前の演算結果が正なら分岐. 0なら分岐しないことに注意)

JMI

JMI:負分岐, Jump on MInus. 2語

ラベル	オペコード	オペランド
[label]	JMI	addr

サインフラグが1なら番地 addr へ分岐 (直前の演算結果が負なら分岐)

JOV

JOV:オーバーフロー分岐, Jump on OVERflow. 2 語

ラベル	オペコード	オペランド
[label]	JOV	addr

オーバーフローフラグが1なら番地 addr へ分岐

CPA

CPA:算術比較, ComPare Arithmetic. 1 語

ラベル	オペコード	オペランド
[label]	CPA	GR1, GR2

2つのレジスタの内容を算術的に比較し, フラグレジスタをセット. SUBA と異なり, 2つのレジスタの値は変化しない.

	ZF	SF	OF
(GR1) > (GR2)	0	0	0
(GR1) = (GR2)	1	0	0
(GR1) < (GR2)	0	1	0

CPL

CPL:論理比較, ComPare Logical. 1 語

ラベル	オペコード	オペランド
[label]	CPL	GR1, GR2

2つのレジスタの内容を論理比較し, フラグレジスタをセット.

CALL

CALL:呼び出し, CALL. 2 語

ラベル	オペコード	オペランド
[label]	CALL	addr

この命令の次の命令の番地をスタックにプッシュして, 番地 addr へ実行を移す. より具体的には以下の通り.

1. スタックポインタ (SP) の値を 1 減らす
2. プログラムレジスタ (PR) の値を SP が指す番地に格納
3. PR に番地 addr をセットする

サブルーチン呼び出しなどに用いられる.

RET

RET:復帰, RETurn. 1 語

ラベル	オペコード	オペランド
[label]	RET	

スタックからポップした番地へ実行を移す (プログラムの呼び出し元へ戻る). より具体的には以下の通り.

1. SP が指す番地の内容を PR に設定
2. SP の値を 1 増やす

サブルーチン呼び出しからの復帰などに用いられる.

PUSH

PUSH:プッシュ. 2 語

ラベル	オペコード	オペランド
[label]	PUSH	x, GR

GR の値をスタックにプッシュする. より具体的には以下の通り.

1. スタックポインタ (SP) の値を 1 減らす
2. GR の値を SP が指す番地に格納

このアセンブリ言語では, 参考になっている CASLII の仕様にあわせて, PUSH 命令には二つのオペランドを書くようにしてあります. ただし, この講義で解説したような使い方では, 一つ目のオペランド x は常に 0 です. CASLII ではここに 0 以外の値を指定することもできますが, そのような使い方はこの講義の範囲を超えるので, 解説しません.

POP

POP:ポップ. 1 語

ラベル	オペコード	オペランド
[label]	POP	GR

スタックからポップした内容を GR にセットする. より具体的には以下の通り.

1. SP が指す番地の内容を GR にセット
2. SP の値を 1 増やす