

FATBYTE 仕様(案)

FATBYTE とは

FATBYTE は CPU の ISA(命令セット アーキテクチャ)のテンプレートです。トランジスタ数が少ない CPU を作る時に互換性を持たせながら新しい命令セットの CPU を作ることができます。またバイトコードのような中間コードを出力するフリーのコンパイラがある言語の移植が簡単になる特長があります。中間コードのバイトコードを実行するにはバイトコード1バイトにつき1バイトをコンパイラが付加した命令コードにしてプログラムメモリに格納。そのプログラムを直接実行させます。8bit CPU ICF3-Z は、この FATBYTE に準拠しているので FATBYTE のプログラムを実行することが可能。

ICF3-Z を普及させることに重点が置かれているので FATBYTE は「優れた技術」ではないかもしれませんが、コンパイラの移植性を良くするためのアプローチとして、やってみて結果を出してみることも面白いかもと考えています。

通常のコマンドコード

bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
0	自作の命令セット						

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
自作の命令セット							

FAT BYTE コード

bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
1	op コード						

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
バイトコード							

op コード命令は通常のコマンドコードを使って自由にプログラムが可能であること。

[ここに入力]

ICF3-Z の op コード実装のメモ

ICF3-Z の命令コードは 1 ワード 32bit であるため、分岐などのアドレスは 2 で割る必要があります。奇数アドレスの場合、奇数アドレスへの分岐であるというマークをして 2 で割ったアドレスに分岐。すべての命令は奇数アドレスへの分岐でないかを判定して、奇数であれば、何も実行せずに RETURN する。この実装には ZF=0 の場合に RETURN する RETURN0 が便利。

FATBYTE 型 CPU の公式実装 ScripDog

自作 8bit CPU WZeta のコードをそのまま流用。現バージョンの WZeta は割込み未実装なので ScripDog も割込み未実装。WZeta は超小型の MCU として作られたので一般的な CPU の命令セットですが貧弱な命令セット。

通常のコマンドコード

bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
0	0						

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
8bit CPU WZeta							

FATBYTE コードの場合、バイトコードがレジスタ 0 番に格納される。

参考

オープンソース 8bit CPU ICF3-Z

<https://icf3z.idletime.tokyo/>

自作 8bit CPU WZeta

<https://wzeta.idletime.tokyo/>

連絡先

説明不足でわかりにくかったように思います。連絡先は ICF3-Z の連絡先と同じです。

[ここに入力]