

高機能 MINICPU

- ・ 配列を処理するための命令を追加
- ・ MINICPU の命令セット

ソースコード集

MINICPU の Verilog HDL 記述

	言語	説明
<u>各種定数宣言</u>	Verilog HDL	MINICPU で用いる各種定数の定義
<u>算術論理演算回路</u>	Verilog HDL	MINICPU の算術論理演算回路 (TINYCPU と同じ)
<u>ステートマシン</u>	Verilog HDL	MINICPU のステートマシン (TINYCPU-F と同じ)
<u>スタック</u>	Verilog HDL	MINICPU のデータスタック
<u>メモリ</u>	Verilog HDL	MINICPU のデュアルポートブ <u>ロック RAM</u>
<u>MINICPU</u>	Verilog HDL	MINICPU 本体

アセンブラとコンパイラ

MINICPU をターゲットとするアセンブラとコンパイラ

	言語	説明
<u>アセンブラ</u>	Perl	MINICPU 用アセンブラ
<u>字句解析</u>	Flex	コンパイラの子句解析 (TINYC と同じ)
<u>構文解析</u>	Bison	MINICPU 用コンパイラの構文解析
<u>メモリ形式への変換</u>	Perl	オブジェクト形式をメモリ形式に変換

C 言語風プログラムのコンパイル・アセンブル例

フィボナッチ数列を配列に求め、それを順に出力するプログラム。

配列の宣言は、サイズを指定するのではなく、要素を並べる。

同じ値が繰り返す場合は、後ろに [] をつけて、繰り返し回数を指定する。

例えば、次の C 言語風プログラムでは、配列 a の要素は、1 が 10 回繰り返したあと 0 が続く大きさ 11 の配列となる。

C 言語風プログラム

```
while(a[i]){  
    a[i]=a[i-1]+a[i-2];  
    i=i+1;
```


*** LABEL LIST ***

```
_001F 01B
_001T 000
_002F 02C
_002T 01D
a      02E
i      02D
```

*** MACHINE PROGRAM ***

```
_001T:
000:102E      PUSHI a
001:202D      PUSH i
002:F000      ADD
003:7000      LD
004:501B      JZ _001F
005:102E      PUSHI a
006:202D      PUSH i
007:F000      ADD
008:102E      PUSHI a
009:202D      PUSH i
00A:1001      PUSHI 1
00B:F001      SUB
00C:F000      ADD
00D:7000      LD
00E:102E      PUSHI a
00F:202D      PUSH i
010:1002      PUSHI 2
011:F001      SUB
012:F000      ADD
013:7000      LD
014:F000      ADD
015:8000      ST
016:202D      PUSH i
017:1001      PUSHI 1
018:F000      ADD
019:302D      POP i
01A:4000      JMP _001T

_001F:
01B:1000      PUSHI 0
01C:302D      POP i

_002T:
01D:102E      PUSHI a
01E:202D      PUSH i
01F:F000      ADD
020:7000      LD
021:502C      JZ _002F
022:102E      PUSHI a
023:202D      PUSH i
024:F000      ADD
025:7000      LD
026:E000      OUT
027:202D      PUSH i
028:1001      PUSHI 1
029:F000      ADD
02A:302D      POP i
02B:401D      JMP _002T

_002F:
02C:0000      HALT
02D:0002      i: 2
02E:0001      a: 1
02F:0001      1
030:0001      1
031:0001      1
032:0001      1
033:0001      1
034:0001      1
035:0001      1
036:0001      1
037:0001      1
038:0000      0
```