

マイクロプログラム方式Brainfuck - メイン

CPU上のBrainfuckコードの対応表

CPU上の値	Brainfuckのコード	マイクロコードの番地
0 >		28 1C
1 <		30 1E
2 +		24 18
3 -		26 1A
4 .		34 22
5 ,		32 20
6 [4 4
6 [4 4
7]		14 E

信号線

selct_reg	0	1	2	3
値	b	c	d	ram[b]
select_im	0	1	2	3
値	0	-1	1	x
target_reg	0	1	2	3
値	b	c	d	ram[b]

マイクロプログラムの命令セット

命令	メモ1	メモ2	7,6,5,4bit	3,2bit	1,0bit	select_reg	select_im	target_reg	use_keyboard	use_display	micro_addr	備考
add	X	Y	0 X	Y	X	X	Y	X	0	0	1	X=X+Y
write	ram[b]		1	3	0	3	0	3	0	1	1	ram[b]をディスプレイに出力
read	ram[b]		2	3	0	3	0	3	1	0	1	キーボードからの入力をram[b]へ
j_ram		0	3	X		0	0	0	0	0	1	ram[b]が0なら次の命令へ
		not 0	3	X		0	0	0	0	0	1	ram[b]が0じゃない場合はジャンプ
j_rom_【	【		4	X		0	0	0	0	0	1	romが 【 なら次の命令へ
		not 【	4	X		0	0	0	0	0	X	romが 【 じゃない場合はジャンプ
j_rom_】	】		5	X		0	0	0	0	0	1	romが 】 なら次の命令へ
		not 】	5	X		0	0	0	0	0	X	romが 】 じゃない場合はジャンプ
j_c		0	6	X		0	0	0	0	0	1	cが0なら次の命令へ
		not 0	6	X		0	0	0	0	0	X	cが0じゃない場合はジャンプ
jmp			7	X		0	0	0	0	0	X	2*X番地へジャンプ
XXXX			8	0	0	0	0	0	0	0	0	命令を取ってきて適当な番地へジャンプ

マイクロプログラム

bfの命令	番地	命令	R1	R2	7,6,5,4bit	3,2bit	1,0bit	コード	コード(hex)
	0	XXXX			8	0	0	128	80
	1	nop			0	0	0	0	0
template	2	add	d	1	0	2	2	10	A
	3	jmp		0	7	0	0	112	70
【	4	j_ram		2	3	0	1	49	31
	5	add	c	1	0	1	2	6	6
	6	add	d	1	0	2	2	10	A
	7	nop			0	0	0	0	0
	8	j_rom_【		10	4	0	5	69	45
	9	add	c	1	0	1	2	6	6
	10	j_rom_】		12	5	0	6	86	56
	11	add	c	-1	0	1	1	5	5
	12	j_c		6	6	0	3	99	63
	13	jmp		2	7	0	1	113	71

マイクロプログラム方式Brainfuck - メイン

]	14	add	c	-1	0	1	1	5	5
	15	nop			0	0	0	0	0
	16	add	d	-1	0	2	1	9	9
	17	nop			0	0	0	0	0
	18	j_rom_		20	4	0	10	74	4A
	19	add	c	1	0	1	2	6	6
	20	j_rom_		22	5	0	11	91	5B
	21	add	c	-1	0	1	1	5	5
	22	j_c		16	6	0	8	104	68
	23	jmp		0	7	0	0	112	70
+	24	add	ram[b]	1	0	3	2	14	E
	25	jmp		2	7	0	1	113	71
-	26	add	ram[b]	-1	0	3	1	13	D
	27	jmp		2	7	0	1	113	71
>	28	add	b	1	0	0	2	2	2
	29	jmp		2	7	0	1	113	71
<	30	add	b	-1	0	0	1	1	1
	31	jmp		2	7	0	1	113	71
,	32	read			2	3	0	44	2C
	33	jmp		2	7	0	1	113	71
.	34	write			1	3	0	28	1C
	35	jmp		2	7	0	1	113	71