P1 Tachyon Dictionary Structure

Freitag, 6. August 2021 10:03

Word Codes

PASM WORDCODE ADDRESS - $0000..$01FF

0	0	0	0	0	0	0	C	C	C	C	C	C	C	C	C

HUB WORDCODE CALL ADDRESS - $0200..$7DFF - If J is set then a jump rather than a call is made.

0	H	H	H	H	H	H	H	H	H	H	H	H	H	H	J

15-BIT LITERAL - $0000..$7FFF

1	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L

???? 8-BIT TASK REGISTER ADDRESS - $00..$FF ????

0	1	1	1	1	1	1	0	R	R	R	R	R	R	R	R

Jumpop 7EFF
7-BIT "IF" FORWARD BRANCH ADDRESS - 0..+127 words

0	1	1	1	1	1	1	1	0	F	F	F	F	F	F	F

7-bit "UNTIL" REVERSE BRANCH ADDRESS - 0..-127 words

0	1	1	1	1	1	1	1	1	U	U	U	U	U	U	U

If code>decops=73FF:
Direct execution of pasm instruction with par2,TOS

0	1	1	1	0	0	1	i	i	i	i	i	i	z	c	r

79BC(0075) 25 EE BC 58 | DECODE2 movi pasm2_1,X ' write instruction code to instruction field of target. Insert S[8..0] into D[31..23]

79C0(0076) 25 32 3C 85 | cmp regops,X wc
79C4(0077) 1E 3E 8C 80 | PASM2_1 if_nc add par2,tos ' always execute selected instruction with two parameters. Add will be overwritten

79C8(0078) 8A 00 4C 5C | if_nc jmp #DROP ' and drop one

##

TF5> HERE DUP . .LONG --- 16156 0000.3F1C ok
TF5> @NAMES DUP . .LONG --- 22776 0000.58F8 ok
TF5>
TF5> : 1add 1 + ;
TF5>
TF5> HERE DUP . .LONG --- 16162 0000.3F22 ok
TF5> @NAMES DUP . .LONG --- 22768 0000.58F0 ok
TF5>
TF5> : 2add 1add 1add ;
TF5>
TF5> HERE DUP . .LONG --- 16166 0000.3F26 ok
TF5> @NAMES DUP . .LONG --- 22760 0000.58E8 ok
TF5>

TF5> HELP 1add
58F0 3F1C 2 1add
ADDR: DATA NAME
3F1C: 8001 1

```
3F1E: 7D01  +
3F20: 005A  EXIT

HELP 2add
58E8 3F22 2 2add
ADDR: DATA  NAME
3F22: 3F1C  1add
3F24: 3F1D  1add >>>LSB set for branch instead of CALL
3F26: 005A  EXIT

TF5> @NAMES 10 DUMP ---
0000.58E8:   44 32 61 64  64 00 22 3F  44 31 61 64  64 00 1C 3F    D2add."?D1add..? ok


: 3_add 1add 1add 1add ;


TF5> HELP 3_add
58E0 3F26 2 3_add
ADDR: DATA  NAME
3F26: 3F1C  1add
3F28: 3F1C  1add
3F2A: 3F1D  1add  >>>LSB set for branch instead of CALL
3F2C: 005A  EXIT


@NAMES 10 DUMP ---
0000.58E0:   45 33 5F 61  64 64 26 3F  44 32 61 64  64 00 22 3F    E3_add&?D2add."? ok
TF5>


795C(005D) 23 4A BC 04 | doNEXT              rdword  X,IP          ' read word code instruction
7960(005E) 02 46 FC 81 |                    add     IP,#2 wc      ' advance IP to next wordcode (clears the
carry too!)
7964(005F) 09 4A 7C 2A | EXECX               shr     X,#9 nr,wz    ' cog or hub?
7968(0060) 25 00 28 5C |              if_z   jmp     X             ' execute the code by directly indexing the
first 512 longs in cog
796C(0061) 25 30 3C 85 |                    cmp     decops,X wc   ' addresses above this value should be
decoded
7970(0062) 6A 00 70 5C |              if_c   jmp     #DECODE       ' decode further
7974(0063)           | WORDCODE                                   ' CALL - must be a wordcode call/jump
7974(0063) 01 4A 7C 62 |                    test    X,#1 wz       ' skip pushing the IP if bit 0 is set
7978(0064) 01 4A D4 64 |              if_nz  andn    X,#1          ' and fixup
797C(0065) 5C CE A8 54 | ENTER              if_z   movd    _enter_,retptr  ' setup the stack ptr
7980(0066) 01 B8 E8 80 |              if_z   add     retptr,#1     ' update the stack ptr
7984(0067) 23 7E AB A0 | _enter_           if_z   mov     retstk,IP     ' save it on the stack (dest modified)
7988(0068) 25 46 BC A0 |                    mov     IP,X          ' CALL/JUMP - so after saving the IP, load it
with new address
798C(0069) 5D 00 7C 5C |                    jmp     #doNEXT
7990(006A)           | DECODE              ' literal?
7990(006A) 0F 4A 7C 2A |                    shr     X,#15 nr,wz   ' embedded 15-bit literal? (b15=1)
7994(006B) 1C 4A 94 60 | PUSH15             if_nz   and     X,mask15      ' mask out b15
7998(006C) 3B 00 54 5C |              if_nz  jmp     #PUSHX        ' push this literal without having to do
a call/return
799C(006D) 25 34 3C 85 |                    cmp     jumpop,X wc   ' test for jump + disp
79A0(006E) 75 00 4C 5C |              if_nc  jmp     #decode2
79A4(006F) 8A 3C 7C E8 | doJUMP              tjnz    tos,#DROP     ' discard flag and continue w/o
jumping
79A8(0070) 80 4A 7C 62 |                    test    X,#$80 wz     ' reverse jump? nz
79AC(0071) 7F 4A FC 60 |                    and     X,#$7F        ' mask displacement
79B0(0072) 01 4A FC 2C |                    shl     X,#1          ' index as words
79B4(0073) 25 46 BC 9C |                    sumnz   IP,X          ' +/- jump
79B8(0074) 8A 00 7C 5C |                    jmp     #DROP         ' discard condition flag
79BC(0075) 25 EE BC 58 | DECODE2             movi    pasm2_1,X     ' write instruction code to
instruction field of target
79C0(0076) 25 32 3C 85 |                    cmp     regops,X wc
79C4(0077) 1E 3E 8C 80 | PASM2_1            if_nc   add     par2,tos      ' always execute selected
instruction with two parameters
79C8(0078) 8A 00 4C 5C |              if_nc  jmp     #DROP         ' and drop one
79CC(0079) FF 4A FC 60 | doREG               and     X,#$FF
79D0(007A) 08 4A BC 80 |                    add     X,regptr
```

```
79D4(007B) 3B 00 7C 5C |                jmp     #PUSHX
79D8(007C) 23 FC BC 08 | LMM                 rdlong  instr,IP
79DC(007D) 04 46 FC 80 |                add     IP,#4
79E0(007E) 00 00 00 00 | instr               nop
79E4(007F) 7C 00 7C 5C |                jmp     #LMM

79E8(0080) 2C 68 FC 5C | AEXEC               call    #POPX           ' Exec wordcode from tos
79EC(0081) 5F 00 7C 5C |                jmp     #EXECX
```