8bit CPU WZeta β版向け SHA-1 試作

2021 年 7 日 8 日、WZeta β版 verilog シミュレータで SHA-1 試作プログラムが割込みを受け付けながら正しい結果を出力しました。そのときのプログラムです。整数型仮想コプロ Vicopid のコードも入っています。WZeta の命令セットは WZeta の公式サイトで公開されています。ハードマクロ命令の呼び出しはハードマクロ命令が格納されているプログラム領域のラベルに^を付けた文字列がニモニックになっています。あとは一般的なアセンブラと似たものになっています。雰囲気をつかめるように公開しました。

```
### Vicopoid SHA-1
### 2021/05/24
### Vicopoid mamory
### 0-15   BUFFER
### 16-19  K1-K4
### 20-24  SHAH[5]
### 25-29  A-E
### 30-109 W[80]
### 110    TEMP
### 111    WORK
### 127    STACK   INIT: VICO_SP=128
###
        MEMORY 4
        MODEL TINY
### Vicopoid ALU function #################
@ADD 1
@SUB 2
@MUL 3
@AND 4
@OR  5
@XOR 6
@NOT 7
@MEM 8
@DUP 9
@TOP 10
### #####################################

@SHA_K1 16
@SHA_K2 17
@SHA_K3 18
@SHA_K4 19
@SHA_H0 20
```

```
@SHA_H1 21
@SHA_H2 22
@SHA_H3 23
@SHA_H4 24
@SHA_A 25
@SHA_B 26
@SHA_C 27
@SHA_D 28
@SHA_E 29
@WO   30
@W15  45
@TEMP 110
@WORK 111

        .HMAC
### VICOPOID 1 ############################
VICO_PUSH:
        ST %RO,C
        DEC %VICO_SP
        LD A,%VICO_SP
        ST %R1,A
        LD B,^VICO_COPY.L
        JMP ^VICO_COPY:B
        NOP
        NOP
### VICOPOID 2 ############################
VICO_PUSH%:
        LD A,[0]
        ST %RO,A
        DEC %VICO_SP
        LD A,%VICO_SP
        ST %R1,A
        LD B,^VICO_COPY.L
        JMP ^VICO_COPY:B
        NOP
### VICOPOID 3 ############################
VICO_POP:
        ST %R1,C
        INCX %VICO_SP
        ST %RO,A
        LD B,^VICO_COPY.L
        JMP ^VICO_COPY:B
```

```
VICO_SHIFT8_PART2:
        ST [&VICO_M1:B],A
        LD A,[0]
        +ST [&VICO_M0:B],A
### VICOPOID 4 ############################
VICO_POP%:
        LD A,[0]
        ST %R1,A
        INCX %VICO_SP
        ST %R0,A
        LD B,^VICO_COPY.L
        JMP ^VICO_COPY:B
        NOP
        NOP
### VICOPOID 5 ############################
VICO_PUSH8:
        DEC %VICO_SP
        LD B,%VICO_SP
        ST [&VICO_M0:B],C
        LD A,0
        ST [&VICO_M1:B],A
        ST [&VICO_M2:B],A
        +ST [&VICO_M3:B],A
        NOP
### VICOPOID 5 ############################
VICO_PUSH8%:
        DEC %VICO_SP
        LD B,%VICO_SP
        LD A,[0]
        ST [&VICO_M0:B],A
        LD A,0
        ST [&VICO_M1:B],A
        ST [&VICO_M2:B],A
        +ST [&VICO_M3:B],A
### VICOPOID 6 ############################
VICO_SHIFT8%:
        JR ^VICO_SHIFT8_LBL
        NOP
VICO_READ_PART2:
        LD B,[0]
        STACP [&VICO_M2:B]
        LD B,[0]
```

```
        +STACP [&VICO_M3:B]
        NOP
        NOP
### VICOPOID 6 ############################
VICO_SHIFT8:
        ST [0],C
VICO_SHIFT8_LBL:
        LD B,%VICO_SP
        LD A,[&VICO_M2:B]
        ST [&VICO_M3:B],A
        LD A,[&VICO_M1:B]
        ST [&VICO_M2:B],A
        LD A,[&VICO_M0:B]
        JR ^VICO_SHIFT8_PART2
### VICOPOID 7  ############################
VICO_ALU:
        ST [0],C
        LD B,^VICO_ALU_BODY.L
        JMP ^VICO_ALU_BODY:B
        NOP
        NOP
VICO_READ%_PART2:
        STACP [&VICO_M2:B]
        LD B,[0]
        +STACP [&VICO_M3:B]
### VICOPOID 8  ############################
VICO_SHL%:
        LD B,%VICO_SP
VICO_SHL_FOR:
        SHL  [&VICO_M0:B]
        SHLC [&VICO_M1:B]
        SHLC [&VICO_M2:B]
        SHLC [&VICO_M3:B]
        DEC [0]
        JRZ0 ^VICO_SHL_FOR
        +NOP
### VICOPOID 9  ############################
VICO_SHR%:
        LD B,%VICO_SP
VICO_SHR_FOR:
        SHR  [&VICO_M3:B]
        SHRC [&VICO_M2:B]
```

```
        SHRC [&VICO_M1:B]
        SHRC [&VICO_M0:B]
        DEC [0]
        JRZ0 ^VICO_SHR_FOR
        +NOP
### VICOPOID 10 ############################
VICO_READ:                      # P=A:B
        ST [0],C
        ST %R0,B
        LD C,%R0
        LD B,[0]
        STACP [&VICO_M0:B]
        LD B,[0]
        STACP [&VICO_M1:B]
        JR ^VICO_READ_PART2
### VICOPOID 11 ############################
VICO_READ%:                     # P=A:B
        ST [0],C
        ST %R0,B
        LD C,%R0
        LD B,[0]
        STACP [&VICO_M0:B]
        LD B,[0]
        STACP [&VICO_M1:B]
        LD B,[0]
        JR ^VICO_READ%_PART2
### VICOPOID 12 ############################
VICO_WRITE:                     # P=A:B
        ST [0],C
        ST %R0,B
        ST %R1,A
        LD B,^VICO_WRITE_BODY.L
        JMP ^VICO_WRITE_BODY:B
        NOP
        NOP
        NOP
### VICOPOID 13 ############################
VICO_MOV%:                      # P=A:B
        ST %VICO_PL,B
        ST %VICO_PH,A
        LD B,^VICO_MOV_FOR.L
        JMP ^VICO_MOV_FOR:B
```

```
        NOP
        NOP
        NOP
        NOP
### VICOPOID 14 ############################
VICO_MOVB%:                     # P=A:B
        ST %VICO_PL,B
        ST %VICO_PH,A
        LD B,^VICO_MOVB_FOR.L
        JMP ^VICO_MOVB_FOR:B
        NOP
        NOP
        NOP
        NOP

        .PROG
        LD B,^main.L
        JMP ^main:B

VICO_MOV_FOR:
        INCX  %VICO_PL
        INCXC %VICO_PH
        LD C,[A:B]
        LD B,%VICO_X
        ST [&VICO_M0:B],C
        INCX  %VICO_PL
        INCXC %VICO_PH
        LD C,[A:B]
        LD B,%VICO_X
        ST [&VICO_M1:B],C
        INCX  %VICO_PL
        INCXC %VICO_PH
        LD C,[A:B]
        LD B,%VICO_X
        ST [&VICO_M2:B],C
        INCX  %VICO_PL
        INCXC %VICO_PH
        LD C,[A:B]
        LD B,%VICO_X
        ST [&VICO_M3:B],C
        INC %VICO_X
        DEC [0]
```

```
        JRZ0 ^VIC0_MOV_FOR
        +NOP

VIC0_MOVB_FOR:
        INCX  %VIC0_PL
        INCXC %VIC0_PH
        LD C,[A:B]
        LD A,%VIC0_X
        ST %R0,A
        SHR %R0
        SHR %R0
        LD B,%R0
        AND A,0x03
        JRZ0 ^VIC0_MOVB_LBL1
        ST [&VIC0_M0:B],C
        JR ^VIC0_MOVB_BREAK
VIC0_MOVB_LBL1:
        XOR A,0x01              # 01
        JRZ0 ^VIC0_MOVB_LBL2
        ST [&VIC0_M1:B],C
        JR ^VIC0_MOVB_BREAK
VIC0_MOVB_LBL2:
        XOR A,0x03              # 02
        JRZ0 ^VIC0_MOVB_LBL3
        ST [&VIC0_M2:B],C
        JR ^VIC0_MOVB_BREAK
VIC0_MOVB_LBL3:
        ST [&VIC0_M3:B],C
        JR ^VIC0_MOVB_BREAK
VIC0_MOVB_BREAK:
        INC %VIC0_X
        DEC [0]
        JRZ0 ^VIC0_MOVB_FOR
        +NOP

VIC0_ALU_BODY:
        LD B,%VIC0_SP
        LD A,[&VIC0_M0:B]
        ST %R0,A
        LD A,[&VIC0_M1:B]
        ST %R1,A
        LD A,[&VIC0_M2:B]
```

```
        ST %R2,A
        LD A,[&VICO_M3:B]
        ST %R3,A
        LD B,[0]
        JR B
        JR ^VICO_ALU_ADD
        JR ^VICO_ALU_SUB
        JR ^VICO_ALU_MUL
        JR ^VICO_ALU_AND
        JR ^VICO_ALU_OR
        JR ^VICO_ALU_XOR
        JR ^VICO_ALU_NOT
        JR ^VICO_ALU_MEM
        JR ^VICO_ALU_DUP
        +NOP                    # TOP
VICO_ALU_ADD:
        INC %VICO_SP
        LD B,%VICO_SP
        LD A,%R0
        ADD [&VICO_M0:B],A
        LD A,%R1
        ADDC [&VICO_M1:B],A
        LD A,%R2
        ADDC [&VICO_M2:B],A
        LD A,%R3
        +ADDC [&VICO_M3:B],A
VICO_ALU_SUB:
        INC %VICO_SP
        LD B,%VICO_SP
        LD A,%R0
        SUB [&VICO_M0:B],A
        LD A,%R1
        SUBC [&VICO_M1:B],A
        LD A,%R2
        SUBC [&VICO_M2:B],A
        LD A,%R3
        +SUBC [&VICO_M3:B],A
VICO_ALU_AND:
        INC %VICO_SP
        LD B,%VICO_SP
        LD A,%R0
        AND [&VICO_M0:B],A
```

```
        LD A,%R1
        AND [&VICO_M1:B],A
        LD A,%R2
        AND [&VICO_M2:B],A
        LD A,%R3
        +AND [&VICO_M3:B],A
VICO_ALU_OR:
        INC %VICO_SP
        LD B,%VICO_SP
        LD A,%R0
        OR [&VICO_M0:B],A
        LD A,%R1
        OR [&VICO_M1:B],A
        LD A,%R2
        OR [&VICO_M2:B],A
        LD A,%R3
        +OR [&VICO_M3:B],A
VICO_ALU_XOR:
        INC %VICO_SP
        LD B,%VICO_SP
        LD A,%R0
        XOR [&VICO_M0:B],A
        LD A,%R1
        XOR [&VICO_M1:B],A
        LD A,%R2
        XOR [&VICO_M2:B],A
        LD A,%R3
        +XOR [&VICO_M3:B],A
VICO_ALU_NOT:
        LD B,%VICO_SP
        LD A,0xFF
        XOR [&VICO_M0:B],A
        XOR [&VICO_M1:B],A
        XOR [&VICO_M2:B],A
        +XOR [&VICO_M3:B],A
VICO_ALU_DUP:
        DEC %VICO_SP
        LD B,%VICO_SP
        LD A,%R0
        ST [&VICO_M0:B],A
        LD A,%R1
        ST [&VICO_M1:B],A
```

```
        LD A,%R2
        ST [&VICO_M2:B],A
        LD A,%R3
        +ST [&VICO_M3:B],A
VICO_ALU_MEM:
        LD A,%VICO_SP
        ST %R1,A
VICO_COPY:                      # {%R1} <- {%R0}
        LD B,%R0
        LD A,[&VICO_M0:B]
        LD C,[&VICO_M1:B]
        LD B,%R1
        ST [&VICO_M0:B],A
        ST [&VICO_M1:B],C
        LD B,%R0
        LD A,[&VICO_M2:B]
        LD C,[&VICO_M3:B]
        LD B,%R1
        ST [&VICO_M2:B],A
        +ST [&VICO_M3:B],C
VICO_ALU_MUL:                   # R7R6R5R4 MUL HIGH
        LD A,0                  # PCODE : A=R3-R0 B=R7-4 C=R11-8
        ST %R4,A
        ST %R5,A
        ST %R6,A
        ST %R7,A
        INC %VICO_SP
        LD B,%VICO_SP
        LD A,[&VICO_M0:B]
        ST %R8,A
        LD A,[&VICO_M1:B]
        ST %R9,A
        LD A,[&VICO_M2:B]
        ST %R10,A
        LD A,[&VICO_M3:B]
        ST %R11,A
        LD C,31
MUL_LBL0:
        LD A,%R8
        AND A,1
        JRZ1 ^MUL_LBL1
        LD A,%R0
```

```
        ADD %R4,A
        LD A,%R1
        ADDC %R5,A
        LD A,%R2
        ADDC %R6,A
        LD A,%R3
        ADDC %R7,A
MUL_LBL1:
        SHRC %R7
        SHRC %R6
        SHRC %R5
        SHRC %R4
        SHRC %R11
        SHRC %R10
        SHRC %R9
        SHRC %R8
        LOOPINC ^MUL_LBL0
        LD B,%VIC0_SP
        LD A,%R8
        ST [&VIC0_M0:B],A
        LD A,%R9
        ST [&VIC0_M1:B],A
        LD A,%R10
        ST [&VIC0_M2:B],A
        LD A,%R11
        +ST [&VIC0_M3:B],A
VIC0_WRITE_BODY:
        INCX  %R0
        INCXC %R1
        LD A,[A:B]
        LD B,[0]
        ST [&VIC0_M0:B],A
        INCX  %R0
        INCXC %R1
        LD A,[A:B]
        LD B,[0]
        ST [&VIC0_M1:B],A
        INCX  %R0
        INCXC %R1
        LD A,[A:B]
        LD B,[0]
        ST [&VIC0_M2:B],A
```

```
        INCX  %R0
        INCXC %R1
        LD A,[A:B]
        LD B,[0]
        +ST [&VICO_M3:B],A

sha1_ft1:
        ST %SHA_RETA,A
        ST %SHA_RETB,B
        ^VICO_PUSH @SHA_B
        ^VICO_PUSH @SHA_C
        ^VICO_ALU @AND
        ^VICO_PUSH @SHA_B
        ^VICO_ALU @NOT
        ^VICO_PUSH @SHA_D
        ^VICO_ALU @AND
        ^VICO_ALU @OR
        LD A,%SHA_RETA
        LD B,%SHA_RETB
        JMP A:B

sha1_rotate:                    # R9 : n
        ST %R10,A
        ST %R11,B

        ^VICO_ALU @DUP
        ^VICO_SHL% %R9
        ^VICO_POP @WORK

        LD A,32
        ST %R8,A
        LD A,%R9
        SUB %R8,A
        ^VICO_SHR% %R8
        ^VICO_PUSH @WORK
        ^VICO_ALU @OR

        LD A,%R10
        LD B,%R11
        JMP A:B
main:
        ^VICO_PUSH @SHA_H0
```

```
        ^VICO_POP @SHA_A
        ^VICO_PUSH @SHA_H1
        ^VICO_POP @SHA_B
        ^VICO_PUSH @SHA_H2
        ^VICO_POP @SHA_C
        ^VICO_PUSH @SHA_H3
        ^VICO_POP @SHA_D
        ^VICO_PUSH @SHA_H4
        ^VICO_POP @SHA_E

        ^VICO_PUSH8 0x61
        ^VICO_SHIFT8 0x62
        ^VICO_SHIFT8 0x63
        ^VICO_SHIFT8 0x80
        ^VICO_POP @W0
        ^VICO_PUSH8 0x00
        ^VICO_SHIFT8 0x00
        ^VICO_SHIFT8 0x00
        ^VICO_SHIFT8 0x18
        ^VICO_POP @W15

        LD A,16
        ST %SHA_t,A
sha_for1:
        LD A,%SHA_t
        ST %R0,A
        LD A,@W0
        ADD %R0,A
        LD A,3
        SUB %R0,A
        ^VICO_PUSH8% %R0
        ^VICO_ALU @MEM          # W[t-3]

        LD A,%SHA_t
        ST %R0,A
        LD A,@W0
        ADD %R0,A
        LD A,8
        SUB %R0,A
        ^VICO_PUSH8% %R0
        ^VICO_ALU @MEM          # W[t-8]
```

```
        LD A,%SHA_t
        ST %R0,A
        LD A,@W0
        ADD %R0,A
        LD A,14
        SUB %R0,A
        ^VICO_PUSH8% %R0
        ^VICO_ALU @MEM          # W[t-14]

        LD A,%SHA_t
        ST %R0,A
        LD A,@W0
        ADD %R0,A
        LD A,16
        SUB %R0,A
        ^VICO_PUSH8% %R0
        ^VICO_ALU @MEM          # W[t-16]

        ^VICO_ALU @XOR
        ^VICO_ALU @XOR
        ^VICO_ALU @XOR

        LD A,1
        ST %R9,A
        BR ^sha1_rotate

        LD A,%SHA_t
        ST %R0,A
        LD A,@W0
        ADD %R0,A       # R0=t
        ^VICO_POP% %R0  # W[t] = rotate()

        INCX %SHA_t
        XOR A,79
        JRZ0 ^sha_for1
### ########################################################################
        LD A,0
        ST %SHA_t,A
sha_for2:
### ########################################################################
        LD A,5
        ST %R9,A
```

```
        ^VICO_PUSH @SHA_A
        BR ^sha1_rotate
        BR ^sha1_ft1
        ^VICO_ALU @ADD
        ^VICO_PUSH @SHA_E
        ^VICO_ALU @ADD

        LD A,%SHA_t
        ST %R0,A
        LD A,@W0
        ADD %R0,A        # R0=t
        ^VICO_PUSH% %R0  # W[t]
        ^VICO_ALU @ADD

        ^VICO_PUSH @SHA_K1
        ^VICO_ALU @ADD          # Top: TEMP

        ^VICO_POP @TEMP

        ^VICO_PUSH @SHA_D
        ^VICO_POP  @SHA_E

        ^VICO_PUSH @SHA_C
        ^VICO_POP  @SHA_D

        LD A,30
        ST %R9,A
        ^VICO_PUSH @SHA_B
        BR ^sha1_rotate
        ^VICO_POP  @SHA_C

        ^VICO_PUSH @SHA_A
        ^VICO_POP  @SHA_B

        ^VICO_PUSH @TEMP
        ^VICO_POP  @SHA_A

        INCX %SHA_t
        XOR A,19
        JRZ0 ^sha_for2

### ######################################################################
```

```
        LD A,20
        ST %SHA_t,A
sha_for3:
        LD A,5
        ST %R9,A
        ^VICO_PUSH @SHA_A

        LD B,^sha1_rotate.L
        LD A,^sha1_rotate.H
        BAL A:B

        LD B,^sha1_ft2.L
        LD A,^sha1_ft2.H
        BAL A:B

        ^VICO_ALU @ADD
        ^VICO_PUSH @SHA_E
        ^VICO_ALU @ADD

        LD A,%SHA_t
        ST %R0,A
        LD A,@W0
        ADD %R0,A         # R0=t
        ^VICO_PUSH% %R0   # W[t]
        ^VICO_ALU @ADD

        ^VICO_PUSH @SHA_K2
        ^VICO_ALU @ADD          # Top: TEMP

        ^VICO_POP @TEMP

        ^VICO_PUSH @SHA_D
        ^VICO_POP  @SHA_E

        ^VICO_PUSH @SHA_C
        ^VICO_POP  @SHA_D

        LD A,30
        ST %R9,A
        ^VICO_PUSH @SHA_B

        LD B,^sha1_rotate.L
```

```
        LD A,^sha1_rotate.H
        BAL A:B

        ^VICO_POP  @SHA_C

        ^VICO_PUSH @SHA_A
        ^VICO_POP  @SHA_B

        ^VICO_PUSH @TEMP
        ^VICO_POP  @SHA_A

        INCX %SHA_t
        XOR A,39
        JRZO ^sha_for3
### ########################################################################
        LD A,40
        ST %SHA_t,A
sha_for4:
        LD A,5
        ST %R9,A
        ^VICO_PUSH @SHA_A

        LD B,^sha1_rotate.L
        LD A,^sha1_rotate.H
        BAL A:B

        LD B,^sha1_ft3.L
        LD A,^sha1_ft3.H
        BAL A:B

        ^VICO_ALU @ADD
        ^VICO_PUSH @SHA_E
        ^VICO_ALU @ADD

        LD A,%SHA_t
        ST %RO,A
        LD A,@WO
        ADD %RO,A        # RO=t
        ^VICO_PUSH% %RO  # W[t]
        ^VICO_ALU @ADD

        ^VICO_PUSH @SHA_K3
```

```
        ^VICO_ALU @ADD          # Top: TEMP

        ^VICO_POP @TEMP

        ^VICO_PUSH @SHA_D
        ^VICO_POP  @SHA_E

        ^VICO_PUSH @SHA_C
        ^VICO_POP  @SHA_D

        LD A,30
        ST %R9,A
        ^VICO_PUSH @SHA_B

        LD B,^sha1_rotate.L
        LD A,^sha1_rotate.H
        BAL A:B

        ^VICO_POP  @SHA_C

        ^VICO_PUSH @SHA_A
        ^VICO_POP  @SHA_B

        ^VICO_PUSH @TEMP
        ^VICO_POP  @SHA_A

        INCX %SHA_t
        XOR A,59
        JRZO ^sha_for4
### ######################################################################
        LD A,60
        ST %SHA_t,A
sha_for5:
### ######################################################################
        LD A,5
        ST %R9,A
        ^VICO_PUSH @SHA_A

        LD B,^sha1_rotate.L
        LD A,^sha1_rotate.H
        BAL A:B
```

```
BR ^sha1_ft2
^VICO_ALU @ADD
^VICO_PUSH @SHA_E
^VICO_ALU @ADD

LD A,%SHA_t
ST %R0,A
LD A,@W0
ADD %R0,A         # R0=t
^VICO_PUSH% %R0   # W[t]
^VICO_ALU @ADD

^VICO_PUSH @SHA_K4
^VICO_ALU @ADD           # Top: TEMP

^VICO_POP @TEMP

^VICO_PUSH @SHA_D
^VICO_POP  @SHA_E

^VICO_PUSH @SHA_C
^VICO_POP  @SHA_D

LD A,30
ST %R9,A
^VICO_PUSH @SHA_B

LD B,^sha1_rotate.L
LD A,^sha1_rotate.H
BAL A:B

^VICO_POP  @SHA_C

^VICO_PUSH @SHA_A
^VICO_POP  @SHA_B

^VICO_PUSH @TEMP
^VICO_POP  @SHA_A

INCX %SHA_t
XOR A,79
JRZ0 ^sha_for5
```

```
### ####################################################################

        ^VICO_PUSH @SHA_A
        ^VICO_PUSH @SHA_H0
        ^VICO_ALU @ADD

        ^VICO_ALU @TOP
        LD B,&ANSDATA.L
        LD A,&ANSDATA.H
        STABP %R3
        STABP %R2
        STABP %R1
        STABP %R0

        ^VICO_PUSH @SHA_B
        ^VICO_PUSH @SHA_H1
        ^VICO_ALU @ADD

        ^VICO_ALU @TOP
        LD A,4
        ST %R4,A
        LD A,&ANSDATA.L
        ADD A,%R4
        LD B,&ANSDATA.H
        SWAP
        STABP %R3
        STABP %R2
        STABP %R1
        STABP %R0

        ^VICO_PUSH @SHA_C
        ^VICO_PUSH @SHA_H2
        ^VICO_ALU @ADD

        ^VICO_ALU @TOP
        LD A,8
        ST %R4,A
        LD A,&ANSDATA.L
        ADD A,%R4
        LD B,&ANSDATA.H
        SWAP
        STABP %R3
```

```
        STABP %R2
        STABP %R1
        STABP %R0

        ^VICO_PUSH @SHA_D
        ^VICO_PUSH @SHA_H3
        ^VICO_ALU @ADD

        ^VICO_ALU @TOP
        LD A,12
        ST %R4,A
        LD A,&ANSDATA.L
        ADD A,%R4
        LD B,&ANSDATA.H
        SWAP
        STABP %R3
        STABP %R2
        STABP %R1
        STABP %R0

        ^VICO_PUSH @SHA_E
        ^VICO_PUSH @SHA_H4
        ^VICO_ALU @ADD

        ^VICO_ALU @TOP
        LD A,16
        ST %R4,A
        LD A,&ANSDATA.L
        ADD A,%R4
        LD B,&ANSDATA.H
        SWAP
        STABP %R3
        STABP %R2
        STABP %R1
        STABP %R0
        $PRINT &ANSDATA 20

        EXIT

sha1_ft2:
        ST %SHA_RETA,A
        ST %SHA_RETB,B
```

```
        ^VICO_PUSH @SHA_B
        ^VICO_PUSH @SHA_C
        ^VICO_PUSH @SHA_D
        ^VICO_ALU @XOR
        ^VICO_ALU @XOR
        LD A,%SHA_RETA
        LD B,%SHA_RETB
        JMP A:B
sha1_ft3:
        ST %SHA_RETA,A
        ST %SHA_RETB,B
        ^VICO_PUSH @SHA_B
        ^VICO_PUSH @SHA_C
        ^VICO_ALU @AND
        ^VICO_PUSH @SHA_B
        ^VICO_PUSH @SHA_D
        ^VICO_ALU @AND
        ^VICO_ALU @OR
        ^VICO_PUSH @SHA_C
        ^VICO_PUSH @SHA_D
        ^VICO_ALU @AND
        ^VICO_ALU @OR
        LD A,%SHA_RETA
        LD B,%SHA_RETB
        JMP A:B

        .DATA
        REG 1   R0
        REG 1   R1
        REG 1   R2
        REG 1   R3
        REG 1   R4
        REG 1   R5
        REG 1   R6
        REG 1   R7
        REG 1   R8
        REG 1   R9
        REG 1   R10
        REG 1   R11
        REG 1   SHA_RETA
        REG 1   SHA_RETB
        REG 1   SHA_t
```

```
REG 1   VICO_SP 0x00
REG 1   VICO_X          # MOV 4BYTE/1   MOVB 1BYTE/1
REG 1   VICO_PH
REG 1   VICO_PL
MEM 128 VICO_M0 !0x100 ¥
        0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
        0x99 0xA1 0xDC 0xD6 0x01 0x89 0xFE 0x76 0xF0 0x00 0x00 0x00 0x00 0x00 0x00 0x00
        0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
        0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
        0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
        0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
        0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
        0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
MEM 128 VICO_M1 !0x100 ¥
        0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
        0x79 0xEB 0xBC 0xC1 0x23 0xAB 0xDC 0x54 0xE1 0x00 0x00 0x00 0x00 0x00 0x00 0x00
        0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
        0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
        0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
        0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
        0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
        0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
MEM 128 VICO_M2 !0x100 ¥
        0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
        0x82 0xD9 0x1B 0x62 0x45 0xCD 0xBA 0x32 0xD2 0x00 0x00 0x00 0x00 0x00 0x00 0x00
        0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
        0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
        0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
        0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
        0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
        0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
MEM 128 VICO_M3 !0x100 ¥
        0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
        0x5A 0x6E 0x8F 0xCA 0x67 0xEF 0x98 0x10 0xC3 0x00 0x00 0x00 0x00 0x00 0x00 0x00
        0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
        0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
        0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
        0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
        0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
        0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
MEM 4  DWORD !4
MEM 32 ANSDATA &0xF00
```