Software Tools

Software tools

A number of software tools are provided to help with development.

Assembler

Summary: asm [options] filename.asm >filename.lst

Options:

-h, --help
Print help

These assembly pseudo-ops are provided:

```
                org     0x10            ; Set location to 0x10

                skip    1               ; Reserve one memory location

                data    0x23            ; Emit 8-bit data byte: upper bits
                                        ; are filled in with halt
                                        ; instruction.

                insn    0x4010ff00      ; Emit 32-bit instruction

fred            equ     0x50            ; Set label to value
```

Here is an example run:

```
./asm euc.asm
Pass 1...

0 errors detected in pass 1

Pass 2...
1                       ; Euclid's algorithm
2
3       00                              org     0x00
4       00              a               skip    1               ; First number
5       01              b               skip    1               ; Second number
6       02              tmp             skip    1               ; Tmp variable
7
8       10                              org     0x10
9       10 4800_9000    euclid          st      #144, a         ; Initialize
A
10      11 4800_e901                    st      #233, b         ; Initialize
B
11      12 0062_011a    euclop          jeq     b, eucdon       ; Done ?
12      13 0800_0002                    st      a, tmp
13      14 08e0_0102                    rsbto   b, tmp          ; A - B -> TMP
14      15 0066_0218                    jls     tmp, over       ; A <= B ?
15      16 08e0_0100                    rsbto   b, a            ; A - B -> A
16      17 4018_ff12                    jmp     euclop
17      18 08e0_0001    over            rsbto   a, b            ; B - A -> B
18      19 4018_ff12                    jmp     euclop
19      1a c810_ff00    eucdon          halt
```

```
0 errors detected in pass 2

Symbol table:
a = 0x0
b = 0x1
eucdon = 0x1a
euclid = 0x10
euclop = 0x12
over = 0x18
tmp = 0x2

Memory image:
10: 48009000
11: 4800e901
12: 0062011a
13: 08000002
14: 08e00102
15: 00660218
16: 08e00100
17: 4018ff12
18: 08e00001
19: 4018ff12
1a: c810ff00
```

Simulator

A simulator is provided which produces an instruction trace. The input to the simulator is the assembly listing output from the assembler.

Summary: sim [options] filename.lst

Optoins:

-h, --help
: Print help

-pc xx
: Set initial PC value

Here is an example run:

```
./sim -pc 0x10 euc.lst
10: 48009000
11: 4800e901
12: 0062011a
13: 08000002
14: 08e00102
15: 00660218
16: 08e00100
17: 4018ff12
18: 08e00001
19: 4018ff12
1a: c810ff00
Starting at 10
10: 4800_9000 (C=0)    st      #0x90, 0x00      [00]=90 C=0
11: 4800_e901 (C=0)    st      #0xe9, 0x01      [01]=E9 C=0
12: 0062_011a (C=0)    jeq     0x01, 0x1a       C=0
13: 0800_0002 (C=0)    st      0x00, 0x02       [02]=90 C=0
14: 08e0_0102 (C=0)    rsbto   0x01, 0x02       [02]=A7 C=0
15: 0066_0218 (C=0)    jls     0x02, 0x18       PC=18 C=0
18: 08e0_0001 (C=0)    rsbto   0x00, 0x01       [01]=59 C=1
19: 4018_ff12 (C=1)    jmp     0x12             PC=12 C=1
12: 0062_011a (C=1)    jeq     0x01, 0x1a       C=0
13: 0800_0002 (C=0)    st      0x00, 0x02       [02]=90 C=0
14: 08e0_0102 (C=0)    rsbto   0x01, 0x02       [02]=37 C=1
```

```
15: 0066_0218 (C=1)    jls     0x02, 0x18      C=0
16: 08e0_0100 (C=0)    rsbto   0x01, 0x00      [00]=37 C=1
17: 4018_ff12 (C=1)    jmp     0x12            PC=12 C=1
12: 0062_011a (C=1)    jeq     0x01, 0x1a      C=0
13: 0800_0002 (C=0)    st      0x00, 0x02      [02]=37 C=0
14: 08e0_0102 (C=0)    rsbto   0x01, 0x02      [02]=DE C=0
15: 0066_0218 (C=0)    jls     0x02, 0x18      PC=18 C=0
18: 08e0_0001 (C=0)    rsbto   0x00, 0x01      [01]=22 C=1
19: 4018_ff12 (C=1)    jmp     0x12            PC=12 C=1
12: 0062_011a (C=1)    jeq     0x01, 0x1a      C=0
13: 0800_0002 (C=0)    st      0x00, 0x02      [02]=37 C=0
14: 08e0_0102 (C=0)    rsbto   0x01, 0x02      [02]=15 C=1
15: 0066_0218 (C=1)    jls     0x02, 0x18      C=0
16: 08e0_0100 (C=0)    rsbto   0x01, 0x00      [00]=15 C=1
17: 4018_ff12 (C=1)    jmp     0x12            PC=12 C=1
12: 0062_011a (C=1)    jeq     0x01, 0x1a      C=0
13: 0800_0002 (C=0)    st      0x00, 0x02      [02]=15 C=0
14: 08e0_0102 (C=0)    rsbto   0x01, 0x02      [02]=F3 C=0
15: 0066_0218 (C=0)    jls     0x02, 0x18      PC=18 C=0
18: 08e0_0001 (C=0)    rsbto   0x00, 0x01      [01]=0D C=1
19: 4018_ff12 (C=1)    jmp     0x12            PC=12 C=1
12: 0062_011a (C=1)    jeq     0x01, 0x1a      C=0
13: 0800_0002 (C=0)    st      0x00, 0x02      [02]=15 C=0
14: 08e0_0102 (C=0)    rsbto   0x01, 0x02      [02]=08 C=1
15: 0066_0218 (C=1)    jls     0x02, 0x18      C=0
16: 08e0_0100 (C=0)    rsbto   0x01, 0x00      [00]=08 C=1
17: 4018_ff12 (C=1)    jmp     0x12            PC=12 C=1
12: 0062_011a (C=1)    jeq     0x01, 0x1a      C=0
13: 0800_0002 (C=0)    st      0x00, 0x02      [02]=08 C=0
14: 08e0_0102 (C=0)    rsbto   0x01, 0x02      [02]=FB C=0
15: 0066_0218 (C=0)    jls     0x02, 0x18      PC=18 C=0
18: 08e0_0001 (C=0)    rsbto   0x00, 0x01      [01]=05 C=1
19: 4018_ff12 (C=1)    jmp     0x12            PC=12 C=1
12: 0062_011a (C=1)    jeq     0x01, 0x1a      C=0
13: 0800_0002 (C=0)    st      0x00, 0x02      [02]=08 C=0
14: 08e0_0102 (C=0)    rsbto   0x01, 0x02      [02]=03 C=1
15: 0066_0218 (C=1)    jls     0x02, 0x18      C=0
16: 08e0_0100 (C=0)    rsbto   0x01, 0x00      [00]=03 C=1
17: 4018_ff12 (C=1)    jmp     0x12            PC=12 C=1
12: 0062_011a (C=1)    jeq     0x01, 0x1a      C=0
13: 0800_0002 (C=0)    st      0x00, 0x02      [02]=03 C=0
14: 08e0_0102 (C=0)    rsbto   0x01, 0x02      [02]=FE C=0
15: 0066_0218 (C=0)    jls     0x02, 0x18      PC=18 C=0
18: 08e0_0001 (C=0)    rsbto   0x00, 0x01      [01]=02 C=1
19: 4018_ff12 (C=1)    jmp     0x12            PC=12 C=1
12: 0062_011a (C=1)    jeq     0x01, 0x1a      C=0
13: 0800_0002 (C=0)    st      0x00, 0x02      [02]=03 C=0
14: 08e0_0102 (C=0)    rsbto   0x01, 0x02      [02]=01 C=1
15: 0066_0218 (C=1)    jls     0x02, 0x18      C=0
16: 08e0_0100 (C=0)    rsbto   0x01, 0x00      [00]=01 C=1
17: 4018_ff12 (C=1)    jmp     0x12            PC=12 C=1
12: 0062_011a (C=1)    jeq     0x01, 0x1a      C=0
13: 0800_0002 (C=0)    st      0x00, 0x02      [02]=01 C=0
14: 08e0_0102 (C=0)    rsbto   0x01, 0x02      [02]=FF C=0
15: 0066_0218 (C=0)    jls     0x02, 0x18      PC=18 C=0
18: 08e0_0001 (C=0)    rsbto   0x00, 0x01      [01]=01 C=1
19: 4018_ff12 (C=1)    jmp     0x12            PC=12 C=1
12: 0062_011a (C=1)    jeq     0x01, 0x1a      C=0
13: 0800_0002 (C=0)    st      0x00, 0x02      [02]=01 C=0
14: 08e0_0102 (C=0)    rsbto   0x01, 0x02      [02]=00 C=1
15: 0066_0218 (C=1)    jls     0x02, 0x18      PC=18 C=1
18: 08e0_0001 (C=1)    rsbto   0x00, 0x01      [01]=00 C=1
19: 4018_ff12 (C=1)    jmp     0x12            PC=12 C=1
12: 0062_011a (C=1)    jeq     0x01, 0x1a      PC=1A C=1
1a: c810_ff00 (C=1)    halt                    C=1 halt
```

Note that the right-most column is showing the data transfers which occur as a result of the executed instruction.

If the **inwait** instruction is encountered, the simulator will prompt for a numeric input.