|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| mulf.asm  accept r11: 0000010010000000% \ 8 = '1 \* 2^4 = 100b = 8.  accept r10: 0000001010000000% \ 2 = '1 \* 2^2 = 10b = 2.  \ Result: 0000010110000000 = '1 \* 2^5 = 10000b = 16.  link l1: ct  \ 1st operand is in R11.  \ 2nd operand is in R10.  \ Result is in R15.  \ Modifies r2, r3, r4, r7.  \ 1. Form result sign.  { and r2, r11, 8000h; }  { and r3, r10, 8000h; }  { xor r2, r2, r3; }  { or r15, r2, z; }  \ 2. Form result order.  { and r2, r11, 7f00h; }  { push nz, 7; }  { rfct; or srl, r2, z; }  { and r3, r10, 7f00h; }  { push nz, 7; }  { rfct; or srl, r3, z; }  { add r2, r2, r3, z; }  { or r7, r2, z; }  \ 3. Mantissa multiplication.  { and r2, r11, 0ffh; } \ X.  { and r3, r10, 0ffh; } \ Y.  { xor r4, r4; }  { push nz, 7; }  { load rn, flags; and nil, r3, 1; }  { cjp rn\_z, m\_1; }  { add r4, r4, r2, z; } | | | | | | | | | | |
|  |  |  |  |  | ІАЛЦ.3.105.634.008 ЛП | | | | | |
|  |  |  |  |  |
| Зм. | Арк | № докум | Підпис | Дата |
| Розроб. | | Гуль О.В. |  | 13.05.12 | Обчислювальна cистема.  Додаток 2. Лістинги програм. | Літ. | | | Арк. | Арк-ів |
| Прев. | | Клятченко Я.М. |  |  | К | - | - | 1 | 5 |
|  | |  |  |  | КПІ, ФПМ  гр. КВ-92 | | | | |
| Н. контр. | |  |  |  |
| Затв. | | Клятченко Я.М. |  |  |

m\_1

{ or srl, r4, r4, z; }

{ or srl, r3, r3, z; }

{ rfct; }

\ 4. Result normalization.

m\_n\_s \ Multiply normalization start.

{ load rn, flags; and nil, r4, 80h; }

{ cjp not rn\_z, m\_n\_e; }

{ or sll, r4, r4, z; }

{ sub r7, r7, 1, nz; }

{ jmap m\_n\_s; }

m\_n\_e \ Multiply normalization end.

\ Write result to r15.

{ push nz, 7; }

{ rfct; or sll, r7, z; }

{ and r7, r7, 7f00h; }

{ or r15, r7, z; }

{ or r15, r15, r4; }

END

{ }

idiv.asm

\accept r11: 2

\accept r10: 0FFFDh \\ -3

accept r11: 8

accept r10: 3

link l1: ct

\ 1st operand is in R11.

\ 2nd operand is in R10.

\ Result is in R15.

\ Modifies r2, r3, r4, r7.

\ 1. Form result sign.

{ and r2, r11, 8000h; }

{ and r3, r10, 8000h; }

{ xor r2, r2, r3; }

{ or r15, r2, z; }

Аркуш

2

Зм.

Арк.

№ докум.

Підпис

Дата

ІАЛЦ.3.105.634.008 ЛП

\ 2.1. Convert operand 1.

{ load rn, flags; and nil, r11, 8000h; }

{ cjp rn\_z, pos\_1; }

{ sub r11, z, r11, nz; }

pos\_1

\ 2.2. Convert operand 2.

{ load rn, flags; and nil, r10, 8000h; }

{ cjp rn\_z, pos\_2; }

{ sub r10, z, r10, nz; }

equ x: r11

equ y: r10

pos\_2

{ load rn, flags; sub nil, x, y, nz; } \ Check if x >= y.

{ cjp not rn\_n, div; }

\ x < y.

{ xor r3, r3; }

{ jmap result; }

div

\ 4. Division.

{ xor r3, r3; } \ r3 is quotient.

div\_loop

{ load rn, flags; sub x, x, y, nz; }

{ cjp rn\_n, result ; }

{ add r3, 1; }

{ jmap div\_loop; }

result

\ 5. Form result.

{ or r15, r15, r3; }

{ or r2, x, z; }

END

{}

conv.asm

\ Int argument is in R11.

\ Real result is in R11.

\ Modifies R11, R2, R3, R4.

Аркуш

3

Зм.

Арк.

№ докум.

Підпис

Дата

ІАЛЦ.3.105.634.008 ЛП

{ load rn, flags; and nil, r11, 8000h; } \ Check sign bit.

{ cjp rn\_z, positive; }

\ Not positive - invert, add 1.

{ xor r11, r11, 7fffh; } \ Invert.

{ add r11, r11, 1; } \ Add 1.

positive

{ and r13, r11, 8000h; } \ Copy sign bit.

{ and r4, r11, 7fffh; } \ Copy of mantiss.

\ Find order.

{ xor r2, r2, r2; }

{ or r2, 15, r2; } \ Order counter - maximum by default.

{ xor r3, r3, r3; }

{ or r3, 4000h, r3; } \ Order mask.

order\_loop

{ load rn, flags; and nil, r11, r3; } \ Test current bit.

{ cjp not rn\_z, order\_end; }

{ sub r2, r2, 1, nz; } \ Dec order.

{ add srl, r3, r3, 0; } \ Shift right mask.

{ jmap order\_loop; }

order\_end

\ Remember order.

{ or r3, r2, 0; }

\ Prepare order.

{ push nz, 7; }

{ add sll, r2, r2, 0; } \ Shift order 1 left (1).

{ rfct; }

\ Write order to format.

{ or r13, r13, r2; }

\ Shift mantiss.

\ Determine mantiss position.

{ load rn, flags; sub nil, 8, r3, nz; }

{ cjp rn\_n, mant\_h\_pt; }

Аркуш

4

Зм.

Арк.

№ докум.

Підпис

Дата

ІАЛЦ.3.105.634.008 ЛП

\ A case, when order <= 8.

{ sub r2, 8, r3, nz; } \ R2 = 8 - R3

m\_l\_1

{ load rn, flags; or nil, r2, r2; }

{ cjp rn\_z, m\_l\_1\_end; }

{ sub r2, r2, 1, nz; }

{ add sll, r4, r4, 0; }

{ jmap m\_l\_1 ; }

m\_l\_1\_end

{ and r4, r4, 0ffh; } \ Write mantiss to result.

{ or r13, r13, r4; }

{ jmap conv\_end; }

mant\_h\_pt

\ A case, when order > 8.

{ sub r2, 16, r3, nz; } \ R2 = 15 - R3

m\_l\_2

{ load rn, flags; or nil, r2, r2; }

{ cjp rn\_z, m\_l\_2\_end; }

{ sub r2, r2, 1, nz; }

{ add sll, r4, r4, 0; }

{ jmap m\_l\_2 ; }

m\_l\_2\_end

{ and r4, r4, 0ff00h; } \ Write mantiss to result.

{ push nz, 7; }

{ add srl, r4, r4, 0; }

{ rfct; }

{ or r13, r13, r4; }

{ or r11, r13, z; }

{ or r15, r13, z; }

conv\_end

{ }

Аркуш

5

Зм.

Арк.

№ докум.

Підпис

Дата

ІАЛЦ.3.105.634.008 ЛП