

Heimadæmi 6

TÖL309G Tölvutækni og forritun

Hjörvar Sigurðsson

1.

	<u>S</u>	<u>exp</u>	<u>frac</u>			
i.	0	0	0			
ii.	0	0	1			
iii.	0	1	0			
iv.	0	1	1			
v.	1	0	0			
vi.	1	0	1			
vii.	1	1	0			
viii.	1	1	1			

$Bias = 2^{k-1} - 1 = 2^{1-1} - 1$
 $= 2^0 - 1 = 1 - 1 = 0$

i. Sjáum strax að svarið er 0, en reiknum samt.
exp-bitar allir 0, svo iðstæðað, svo
 $(-1)^S (M) (2^E)$
 $= (-1)^0 (0) (2^{1-0})$
 $= 1 \left(\frac{1}{2}\right) (2)$
 $= 1$

ii. exp-bitar allir 0, svo iðstæðað, svo
 $(-1)^S (M) (2^E)$
 $= (-1)^0 \left(\frac{1}{2}\right) (2^{1-0})$
 $= 1 \left(\frac{1}{2}\right) (2)$
 $= 1$

iii. exp-bitar allir 0, svo iðstæðað, svo
 $(-1)^S (M) (2^E)$
 $= (-1)^0 (0) (2^{1-bias})$
 $= 1 (0) (2^{1-0})$
 $= 1 (0) (2)$
 $= +0$

iv. exp-bitar allir 1, svo sérstök tala, frac-bitar ekki allir 0, svo NaN.

v. exp-bitar allir 0, svo iðstæðað, svo
 $(-1)^S (M) (2^E)$
 $= (-1)^1 (0) (2^{1-0})$
 $= -0.$

vi. exp-bitar allir 0, svo iðstæðað, svo
 $(-1)^S (M) (2^E)$
 $= (-1)^1 \left(\frac{1}{2}\right) (2^{1-0})$
 $= (-1) \left(\frac{1}{2}\right) (2)$
 $= (-1) (1)$
 $= -1$

vii. exp-bitar allir 1, svo sérstök tala, frac-bitar ekki allir 0, svo NaN.

viii. exp-bitar allir 1, svo sérstök tala, S-bit = 1, svo $-\infty$

2.

Gefið:

%rbx inniheldur 0×800

%rdx inniheldur $0 \times A$

Formúla:

Segð \rightarrow Viðf. útreikn.

$D(Rb, Ri, S)$ $\text{Mem}[\text{Reg}[Rb] + S * \text{Reg}[Ri] + D]$

i. $(\%rbx, \%rdx)$

Viðfang, samkvamt

formúlu, er:

$$= 0 \times 800 + 0 \times A$$

$$= \underline{\underline{0 \times 80A}}$$

ii. $\$0 \times 24(\%rbx, \%rdx, 2)$

Viðfang, samkvamt

formúlu, er:

$$= 0 \times 800 + 2 * 0 \times A + 0 \times 24$$

$$= 0 \times 800 + 0 \times 14 + 0 \times 24$$

$$= \underline{\underline{0 \times 838}}$$

iii. $\$20(\%rbx, 8)$

Bruggi úr tuzak. yfir

i sext.k.

$$\$0 \times 14(\%rbx, 8)$$

Viðfang, samkvamt

formúlu, er:

$$8 * 0 \times 800 + 0 \times 14$$

$$= 0 \times 4000 + 0 \times 14$$

$$= \underline{\underline{0 \times 4014}}$$

iv. $\$-16(\%rdx, \%rdx, 4)$

Bruggi úr tuzak. yfir

i sext.k.

$$\$-0 \times 10(\%rdx, \%rdx, 4)$$

$$= 0 \times A + 4 * 0 \times A - 0 \times 10$$

$$= 0 \times A + 0 \times 28 - 0 \times 10$$

$$= 0 \times 32 - 0 \times 10$$

$$= \underline{\underline{0 \times 22}}$$

3.

i. **Skipun: movl %edx, (, %rbx, 4)**

Skipunin flytur innihald 4ra bæta gisti, %edx, yfir í einhvert minnisvistfang M, en vistfang M má finna með formúlunni $M = 4 * \text{Reg}[\%rbx]$.

ii. **Skipun: movzbq \$-2, %r8**

Skipunin flytur 1s bæta gildi, -2, yfir í 8 bæta gisti, %r8.

Skipunin flytur minna gildi yfir í stærra hólf, og núllvíkkar því bæti í fjórórð.

iii. **Skipun: movswq %ax, %rax**

Skipunin flytur innihald 2ja bæta gisti, %ax, yfir í 8 bæta gisti, %rax.

Skipunin formerkisvíkkar orð í fjórórð.

iv. **Skipun: movslw (%rsp), %dx**

Skipunin á að flytja innihald 4ra bæta gisti yfir í 2ja bæta gisti, og að formerkisvíkka tvíórð í orð.

Þetta gengur ekki upp – formerkisvíkkun vísar til þess að bæta gildi formerkisbita við gildið, en að stytta gildið með því að bæta við það er mótsögn. Formerkisvíkkun á aðeins við þegar minna gildi er flutt yfir í stærra hólf.

4. a er í gistinu %rdi,

b er í gistinu %rsi,

skilagildi fallsins – köllum það c – er í gistinu %rax,

d er í gistinu %rdx.

a)

aogb:

leaq (%rsi,%rsi,4), %rax # Leggur saman $b + 4 * b$ og setur útkomuna í gistið %rax, eða c.

leaq (%rdi,%rax,8), %rdx # Leggur saman $a + 8 * c$ og setur útkomuna í gistið %rdx – köllum hana d.

leaq 0(%rdx,8), %rax # Leggur saman $0 + 8 * d$ og setur útkomuna í skilagildið c.

subq %rdx, %rax # Dregur d frá c.

ret # Skilar c.

b)

```
long aogb (long a, long b) {  
    long c = b + (4 * b);  
    long d = a + (8 * c);  
    c = 0 + (8 * d);  
    c = c - d;  
    return c;  
}
```

5.

Viðfang fallsins, k , kemur í gistinu `%edi` (og í `%rdi`, en `%rdi` er alltaf fyrsta viðfang fallsins),
skilagildið, r , fer í gistið `%eax`,
breytan a er í gistinu `%edx`.

reikn:

<code>lea 0x0(%rdi,8),%eax</code>	<i># Leggur saman $0 + 8 * k$ og setur niðurstöðuna í r.</i>
<code>sub %edi,%eax</code>	<i># Dregur k frá r.</i>
<code>lea 0xc(%rdi,%rdi,1),%edx</code>	<i># Leggur saman $12 + k + 1 * k$ og setur útkomuna í a.</i>
<code>sub %edx,%eax</code>	<i># Dregur a frá r.</i>
<code>ret</code>	<i># Skilar r.</i>

C kóði:

```
long reikn (long k) {  
    long r = 0 + 8 * k;  
    r = r - k;  
    long a = 12 + k + 1 * k;  
    r = r - a;  
    return r;  
}
```