

Procesorski sistemi v telekomunikacijah
Operacije s celimi števili

(c) Arpad Bűrmen, 2010-2012

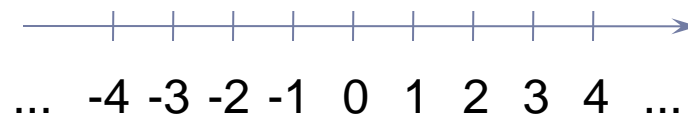
Številski krog za zapis z dvojiškim komplementom

Števila v matematiki so neomejena.

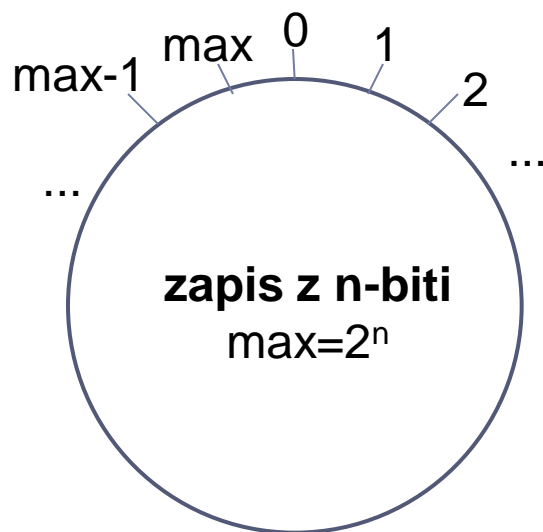
Nepredznačena



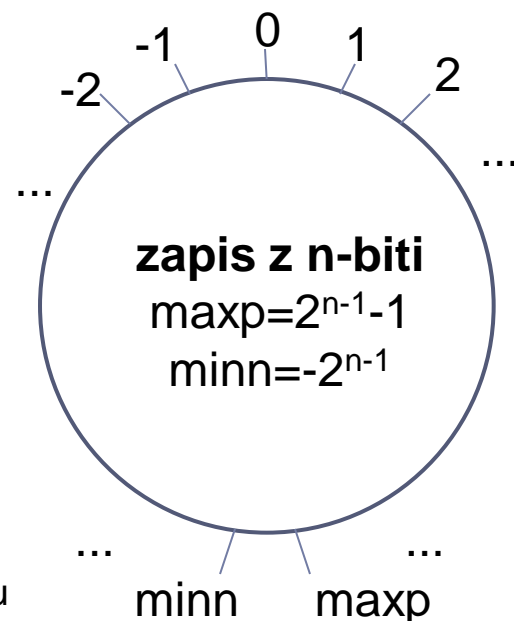
Predznačena



V računalništvu so števila omejena zaradi končne dolžine zapisa.



$x+1 = 1$ korak v smeri urinega kazalca
 $x-1 = 1$ korak nasproti urinemu kazalcu



Zapis števil z dvojiškim komplementom

Nepredznačeno	Zapis (8 bitov)	Predznačeno	Zapis (8 bitov)
0	00000000	0	00000000
1	00000001	1	00000001
2	00000010	2	00000010
...			
127	01111111	127 (maxp)	01111111
128	10000000	-128 (minn)	10000000
129	10000001	-127	10000001
...			
254	11111110	-2	11111110
255 (max)	11111111	-1	11111111

- ▶ n-bitni zapis predznačenega negativnega števila $-x$ je enak n-bitnemu zapisu nepredznačenega števila $2^n - x$

Pa še nekaj o dvojiškem komplementu

- ▶ Dvojiški komplement dobimo tako, da
 1. invertiramo vse bite
 2. k rezultatu invertiranja prištejemo 1
- ▶ **Zakaj?**
- ▶ Zaradi številskega kroga je za n -bitni zapis: $\max+1 = 0$ oziroma $0 = \max+1 = (2^n-1)+1 = 2^n$
- ▶ Torej velja (ampak samo v številskem krogu)
 $-x = 0 - x = 2^n - x = (2^n - 1 - x) + 1$
- ▶ Za n -bitni zapis je $(2^n - 1 - x)$ enako kot invertiranje vseh bitov – operacija $\text{INV}(x)$
- ▶ Torej je: $-x = \text{INV}(x) + 1$
- ▶ Oziroma: **dvojiški komplement $x = \text{INV}(x) + 1$**

Zakaj je zapis z dvojiškim komplementom dober?

- ▶ Seštejmo nepredznačeni števili $y \leq 127$ in $2^n - x$, kjer je $0 < x \leq 128$ (če nepredznačeni zapis teh dveh števil tolmačimo kot predznačeno število, ustreza y pozitivnemu številu y , $2^n - x$ pa negativnemu številu $-x$)
- ▶ $y + (2^n - x) = y - x + 2^n$ pademo na mesto $y - x$ v krogu
- ▶ Od y odštejemo x tako, da k y prištejemo dvojiški komplement x
- ▶ **Eno in isto vezje lahko uporabimo za**
 - seštevanje predznačenih in seštevanje nepredznačenih števil
 - odštevanje predznačenih in nepredznačenih števil (če prej naredimo dvojiški komplement drugega operanda)

$25_{10} = 00011001_2$ prvo predznačeno število
 $-115_{10} = 10001101_2$ drugo predznačeno število

Seštej kot dve nepredznačeni števili

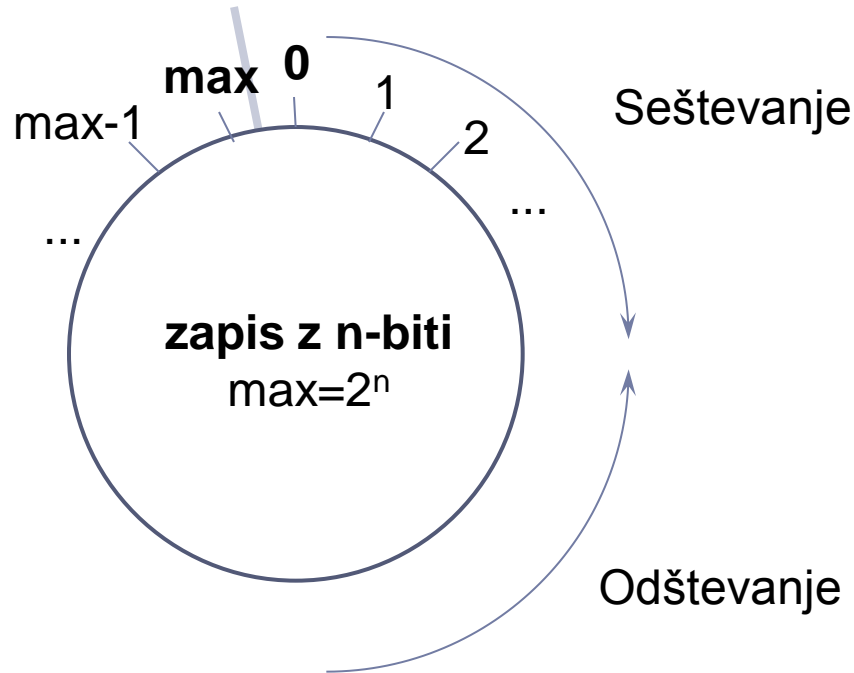
```
00011001
+10001101
-----
10100110
```

$10100110_2 = -128 + 32 + 4 + 2 = -90_{10}$

Seštevanje in odštevanje

Prevelik/premajhen rezultat...

▶ Nepredznačena števila

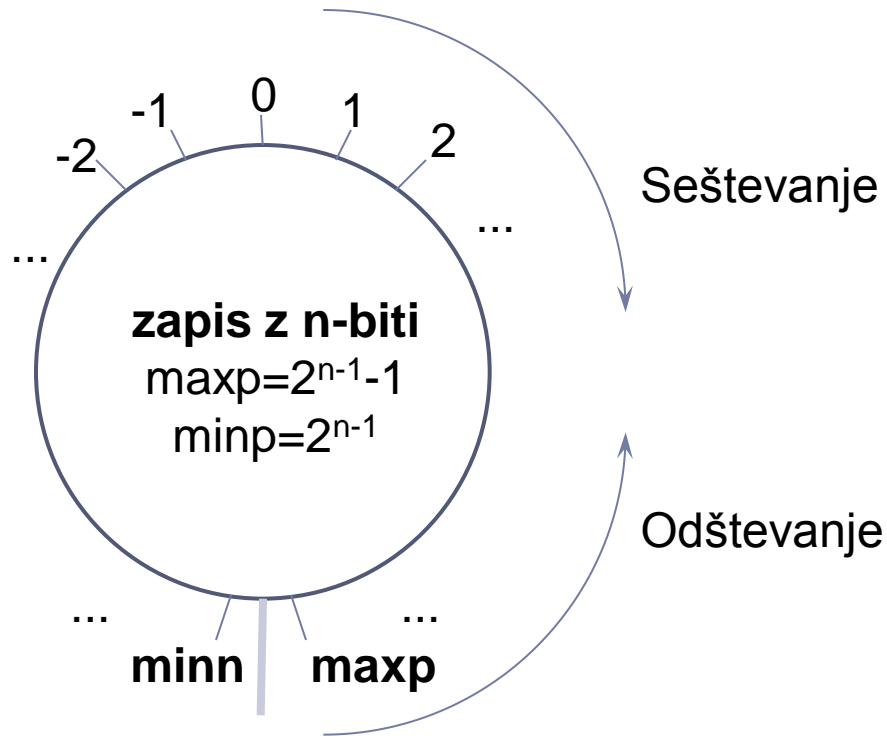


Če prekoračimo mejo med **max** in **0** dobimo napačen rezultat.
Pravimo, da se je zgodil prenos (carry).

Seštevanje in odštevanje

Prevelik/premajhen rezultat...

► Predznačena števila



Če prekoračimo mejo med **maxp** in **maxn** dobimo napačen rezultat. Pravimo, da se je zgodil preliv (overflow).

Zastavice (flags)

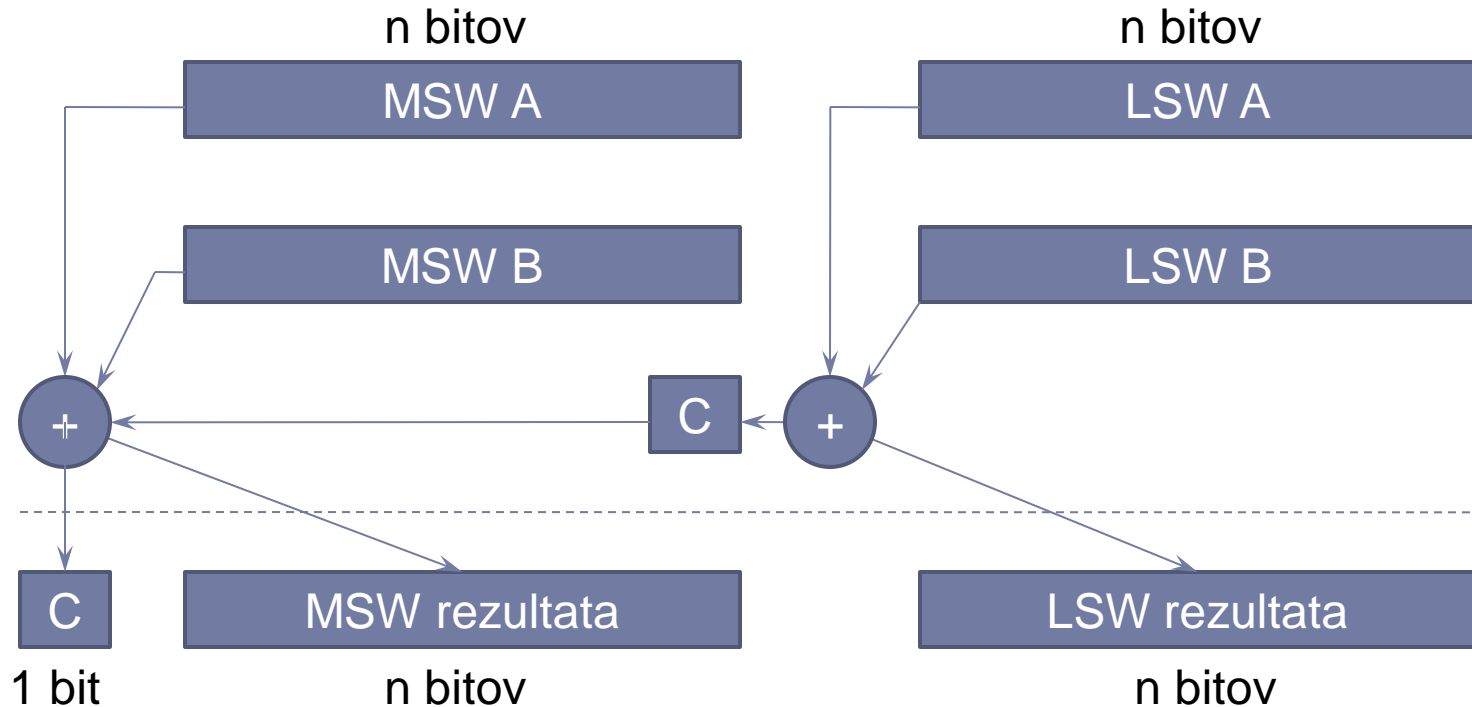
- ▶ Biti v registru stanja – rezultat zadnje operacije.
- ▶ Zastavica Z (Zero)
Z=1, če je rezultat enak 0 oz. Z=0, če ni.
- ▶ Zastavica N (Negative)
N=1, če je najvišji bit rezultata enak 1 oz. N=0, če ni.
- ▶ Zastavica V (oVerflow)
V=1, če se je zgodil preliv, oz. V=0, če se ni.
- ▶ Zastavica C (Carry)
C=1, če se je zgodil prenos, oz. C=0, če se ni.
- ▶ **Pozor! nekateri procesorji (npr. ARM7) pri odštevanju postavijo C=0, če se zgodi prenos in C=1, če se ne zgodi.**

Seštevanje in odštevanje

Zapis rezultata

- ▶ Seštevanje/odštevanje dveh n -bitnih števil
- ▶ Rezultat lahko zmeraj zapišemo z $n+1$ biti
- ▶ Operanda sta nepredznačeni števili
 - rezultat tolmačimo kot nepredznačeno število
- ▶ Operanda sta predznačeni števili
 - rezultat tolmačimo kot predznačeno število
- ▶ C predstavlja $n+1$. bit rezultata
- ▶ **Pozor!**
Pri odštevanju na nekaterih procesorjih (npr. ARM7) je $n+1$. bit rezultata je enak negaciji zastavice C .

Seštevanje dolgih števil



ARM7, 64-bitno seštevanje

```
/* R2 = MSW A,   R1 = LSW A */
```

```
/* R4 = MSW B,   R3 = LSW B */
```

```
/* Rezultat seštevanja: R6 = MSW,   R5 = LSW
```

```
ADDS R5, R3, R1
```

```
/* R5=R3+R1, določi C (statusni register) */
```

```
ADCS R6, R4, R2
```

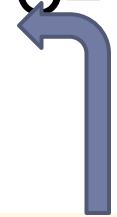
```
/* R6=R4+R2+C, določi C (statusni register) */
```

Odštevanje dolgih števil poteka podobno.

Odštevanje

Primerjava nepredznačenih števil

- ▶ Izračun $A-B$ = primerjava števil A in B
- ▶ Če rezultat pade iz območja nepredzn. števil ... $C=1$
če ostane v območju ... $C=0$
- ▶ Če je rezultat enak 0 ... zastavica $Z=1$
če ni enak 0 ... zastavica $Z=0$
- ▶ Zastavica V nima nobene vloge



Pozor z zastavico C in ARM7!!!
Pri odštevanju na ARM7 dobimo
 $C=0$ če pademo iz območja in
 $C=1$ če ne pademo!

Če velja	Po odštevanju (A-B) dobimo	
$A=B$	$Z=1$	
$A \neq B$	$Z=0$	
$A \geq B$	$C=0$	
$A > B$	$C=0$ in $Z=0$	($C=1$ in $Z=0$ za ARM7)
$A < B$	$C=1$	($C=0$ za ARM7)
$A \leq B$	$C=1$ ali $Z=1$	($C=0$ ali $Z=1$ za ARM7)

Odštevanje

Primerjava predznačenih števil

- ▶ Izračun $A-B$ = primerjava števil A in B
- ▶ Če je $V=1$, zastavica N laže glede predznaka pravilnega rezultata
- ▶ Pravilna razlika A in B je negativna, če je $N \text{ exor } V = 1$
- ▶ Če je rezultat enak 0 ... zastavica $Z=1$
ni enak 0 ... zastavica $Z=0$
- ▶ Zastavica C nima nobene vloge.

Če velja	Po odštevanju (A-B) dobimo
$A=B$	$Z=1$
$A \neq B$	$Z=0$
$A \geq B$	$(N \text{ exor } V)=0$
$A > B$	$(N \text{ exor } V)=0$ in $Z=0$
$A < B$	$(N \text{ exor } V)=1$
$A \leq B$	$(N \text{ exor } V)=1$ ali $Z=1$

Bitne operacije

Aritmetični/logični pomik bitov v levo

Bit izpade ←

Bit	0	1	0	0	1	1	0	1	
Utež	128	64	32	16	8	4	2	1	= 77_{10}

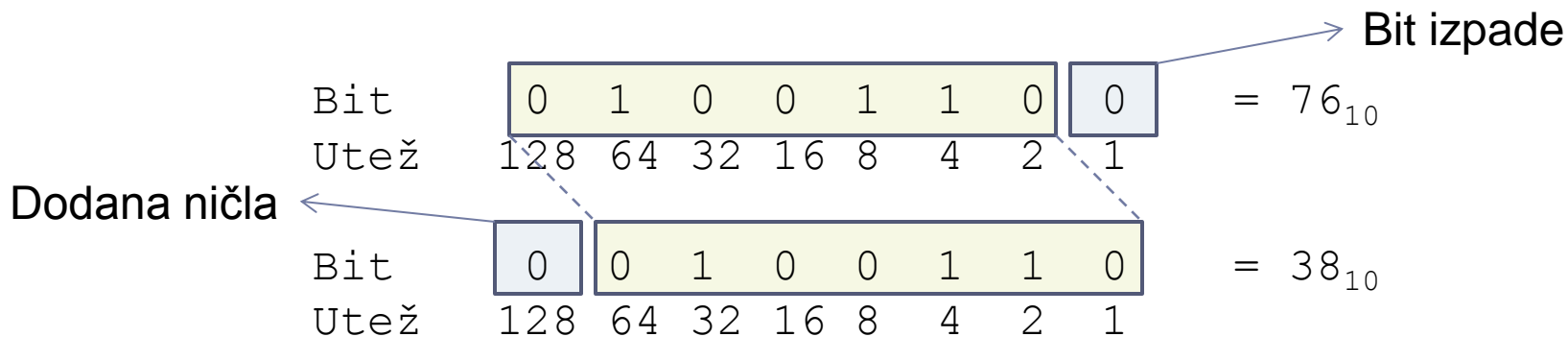
Bit	1	0	0	1	1	0	1	0	
Utež	128	64	32	16	8	4	2	1	= 154_{10}

→ Dodana ničla

- ▶ Biti se pomaknejo za eno mesto v levo.
Na desni (LSB) se doda ničla. Levi bit (MSB) izpade.
- ▶ Biti nepredznačenega števila se pomaknejo k utežem z dvojno vrednostjo.
- ▶ Nepredznačena vrednost se torej podvoji ...
operacija je enaka množenju z 2.
- ▶ Tudi predznačena vrednost se podvoji (množi z 2).

Bitne operacije

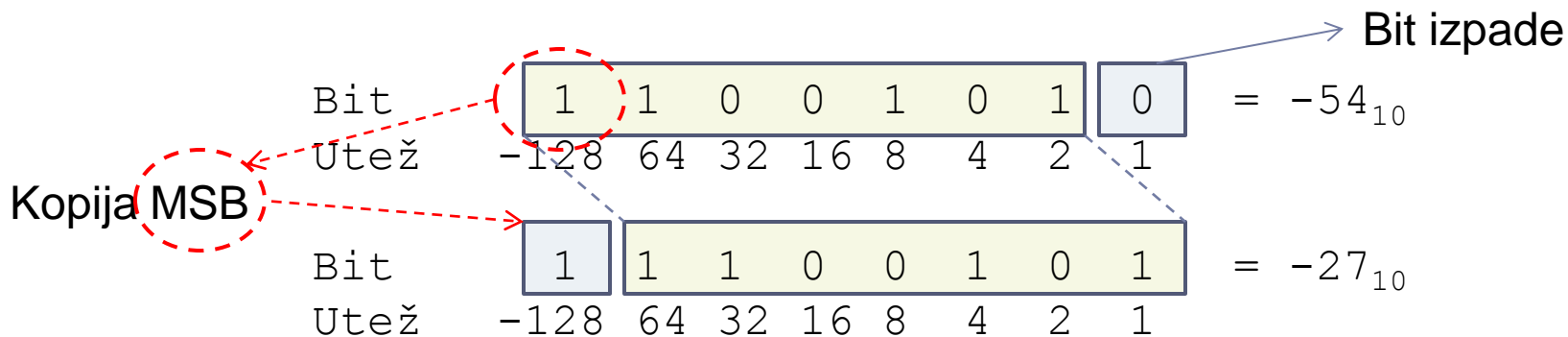
Logični pomik v desno



- ▶ Biti se pomaknejo za eno mesto v desno. Na levi (MSB) se doda ničla. Desni bit (LSB) izpade.
- ▶ Biti nepredznačenega števila se pomaknejo k utežem s polovično vrednostjo.
- ▶ Nepredznačena vrednost se prepolovi ... operacija je enaka deljenju z 2.
- ▶ Operacija nima pomena za predznačena števila.

Bitne operacije

Aritmetični pomik v desno



- ▶ Biti se pomaknejo za eno mesto v desno. Na levi (MSB) se ponovi MSB prvotne vrednosti. Desni bit (LSB) izpade.
- ▶ Biti predzn. števila se pomaknejo k utežem s polovično vrednostjo.
- ▶ Po pomiku sta leva dva bita enaka 11 ($-128+64=-64_{10}$) ali 00 (0_{10})
- ▶ Za predznačena števila je operacija enakovredna deljenju z 2.
- ▶ Operacija nima pomena za nepredznačena števila.

Množenje celih števil

- ▶ Dve n-bitni nepredznačeni števili ... 2n-biten rezultat
- ▶ Dve n-bitni predznačeni števili ... 2n-biten rezultat
- ▶ Nepredznačeno in predznačeno množenje nista enaki operaciji!
- ▶ Dvojiško množenje = pomikanje v levo + seštevanje

Desetiško nepredznačeno

$$235 \times 153$$

$$\begin{array}{r}
 23500 \\
 +11750 \\
 +00705 \\
 \hline
 035955
 \end{array}
 \quad
 \begin{array}{r}
 = 235 \times 100 \\
 = 235 \times 50 \\
 = 235 \times 3 \\
 \hline
 153
 \end{array}$$

Dvojiško nepredznačeno

$$110001 \times 110110$$

$$\begin{array}{r}
 11000100000 \\
 +01100010000 \\
 +00000000000 \\
 +00011000100 \\
 +00001100010 \\
 +00000000000 \\
 \hline
 101001010110
 \end{array}
 \quad
 \begin{array}{r}
 = 110001 \times 100000 \\
 = 110001 \times 010000 \\
 = 110001 \times 000000 \\
 = 110001 \times 000100 \\
 = 110001 \times 000010 \\
 = 110001 \times 000000 \\
 \hline
 110110
 \end{array}$$

Deljenje celih števil

- ▶ n-bitno nepredznačeno deljenje ... n-biten rezultat
- ▶ n-bitno predznačeno deljenje ... n+1 – biten rezultat
- ▶ Dvojiško deljenje = pomikanje v desno + odštevanje

Desetiško nepredznačeno

$$18763 : 121$$

$$121 \leq 18763$$

$$1210 \leq 18763$$

$$12100 \leq 18763 \quad \text{tu začnemo}$$

$$121000 > 18763$$

$$18763 = 12100 \times 1 + 6663$$

$$6663 = 1210 \times 5 + 613$$

$$613 = 121 \times 5 + 8$$

$$18763 = 121 \times 155 + 8$$

Dvojiško nepredznačeno

$$110111 : 101$$

$$101 \leq 110111$$

$$1010 \leq 110111$$

$$10100 \leq 110111$$

$$101000 \leq 110111 \quad \text{tu začnemo}$$

$$1010000 > 110111$$

$$110111 = 101000 \times 1 + 1111$$

$$1111 = 10100 \times 0 + 1111$$

$$1111 = 1010 \times 1 + 101$$

$$101 = 101 \times 1 + 0$$

$$110111 = 101 \times 1011 + 0$$

Množenje in deljenje v praksi

- ▶ Običajno je izvedeno v obliki vgrajenega programa.
- ▶ Čeprav ima procesor ukaz za množenje/deljenje, traja izvajanje tega ukaza več urinih ciklov.
- ▶ Primer: ARM7, 68HC11

- ▶ V signalnih procesorjih je množenje ponavadi izvedeno s temu namenjenim vezjem – množilnikom.
- ▶ Rezultat dobimo v enem urinem ciklu.

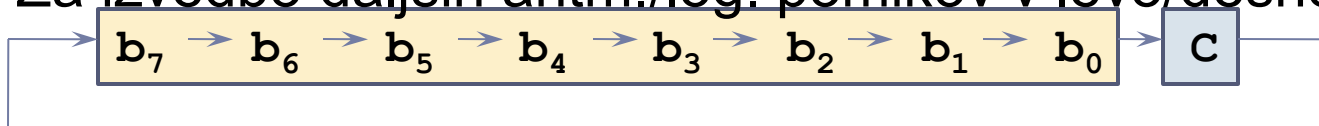
Bitne operacije

- ▶ Delajo s soležnimi biti operandov.

	<u>x</u>	<u>y</u>	<u>x AND y</u>	<u>x OR y</u>	<u>x EXOR y</u>
xxxxxxx	0	0	0	0	0
op yyyyyyy	0	1	0	1	1
-----	1	0	0	1	1
zzzzzzzz	0	1	1	1	0

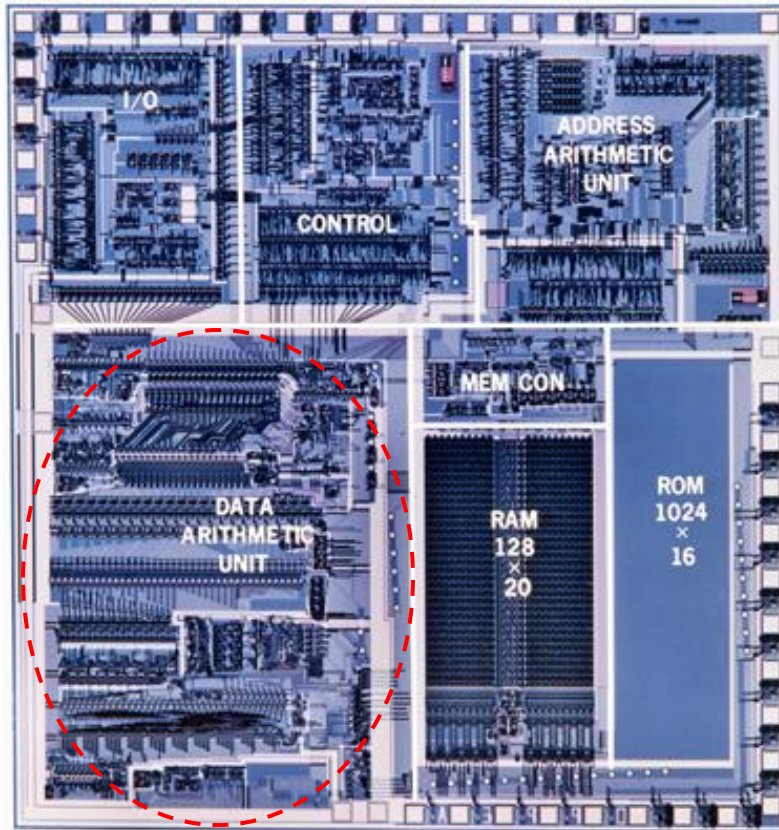
- ▶ Drugemu operandu (y) včasih pravimo tudi **maska**
- ▶ Invertiranje vseh bitov = eniški komplement (one's complement)
- ▶ Sprememba predznaka za predznačena števila =
= dvojiški komplement (two's complement) = eniški komplement + 1
- ▶ Rotacije – krožni pomik bitov preko bita C

Za izvedbo daljših aritm./log. pomikov v levo/desno.

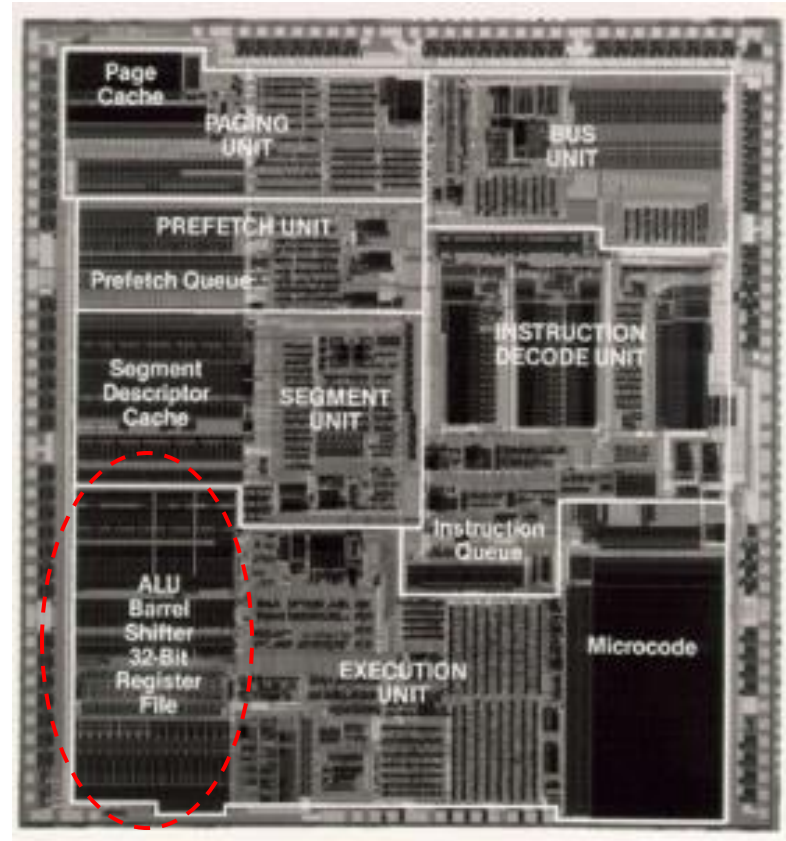


Rotacija bitov v desno

... v siliciju



Bell labs DSP-1



Intel 80386

ALU = Arithmetic Logic Unit – Aritmetično-Logična Enota