



GRAĐA RAČUNALA

Logika

Binarni brojevi

- Računalo mora imati način zapisivanja brojeva
- Kako se koristi baza 2, svi brojevi moraju biti tome prilagođeni
- Nekoliko problema:
 - Dužina zapisa
 - Preciznost
 - Konverzija

Sabirnice

- Način komunikacije među sklopovima
- Sabirnicu može koristiti samo jedan sklop za pisanje
- Ostali čitaju

Notacija

- Način na koji se zapisuju brojevi
- U bazi 10 brojevi se zapisuju kao primjerice:
 - $346.5 = 3 \times 10^2 + 4 \times 10^1 + 6 \times 10^0 + 5 \times 10^{-1}$
- Isto pravilo vrijedi i za brojeve u bazi 2:
 - $110.1 = 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 + 1 \times 2^{-1}$
- Koriste se još i oktalna (8) i heksadecimalna notacija (16)

Osnovne operacije

- Zbrajanje brojeva
- Isto kao i decimalnih uz poštivanje baze
- Oduzimanje brojeva

Komplement

- Binarni broj koji se dobiva inverzijom bitova
- Primjerice:
- 101001001011
- 010110110100

- Bitan je za zapisivanje negativnih vrijednosti:
- $127 = 0111\ 1111$
- $-127 = 1000\ 0000$

0111 1111	127	127
0111 1110	126	126
0000 0010	2	2
0000 0001	1	1
0000 0000	0	0
1111 1111	255	-0
1111 1110	254	-1
1111 1101	253	-2
1000 0001	129	-126
1000 0000	128	-127

Dvojni Komplement

- Binarni broj koji se dobiva inverzijom bitova + 1
- Primjerice:
 - 101001001011
 - 010110110100 + 1
 - 010110110101
- Bitan je za jednostavnost zbrajanja i oduzimanja

Dvojni komplement

0000 0000	0	0
0000 0001	1	1
0000 0010	2	2
0111 1110	126	126
0111 1111	127	127
1000 0000	128	-128
1000 0001	129	-127
1000 0010	130	-126
1111 1110	254	-2
1111 1111	255	-1

Binarni kalkulatori

- <https://eguruchela.com/math/calculator/twos-complement>
- <https://www.easycalculation.com/2s-complement-addition.php>

Pohrana podataka

- Broj bajtova
- Što znači ako je CPU 32 bitan
- Little endian
- Big endian

32 vs 64 bit

- Adresni prostor procesora
- „koliko adresa je moguće nezavisno prepoznati”
- Sama adresa sadržava vrijednost
- Zamislimo recimo vrijednost (decimalno) 2254
- Binarno je to 1000 1100 1110
- Heksadecimalno je to 08CE

Decimal	Hexadecimal	Binary
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111

Broj bitova

- Broj o kojem ovisi najveći cijeli broj koji je moguće zapisati
- $2^n - 1$

Number of Bits	Maximum possible states
8	256
12	4096 (4 K)
16	65,536 (64 K)
20	1,048,576 (1 M)
24	16,777,216 (16 M)
32	4,294,967,296 (4 G)

Nazivi

1	Bit
4	Nibble
8	Byte
16	Word
32	Double Word
64	Quad Word

Preciznost

- Problem sa zapisom decimalnih brojeva
- Broj moramo preračunati
- `sign 2e m`
- <https://www.h-schmidt.net/FloatConverter/IEEE754.html>

Big vs Little endian

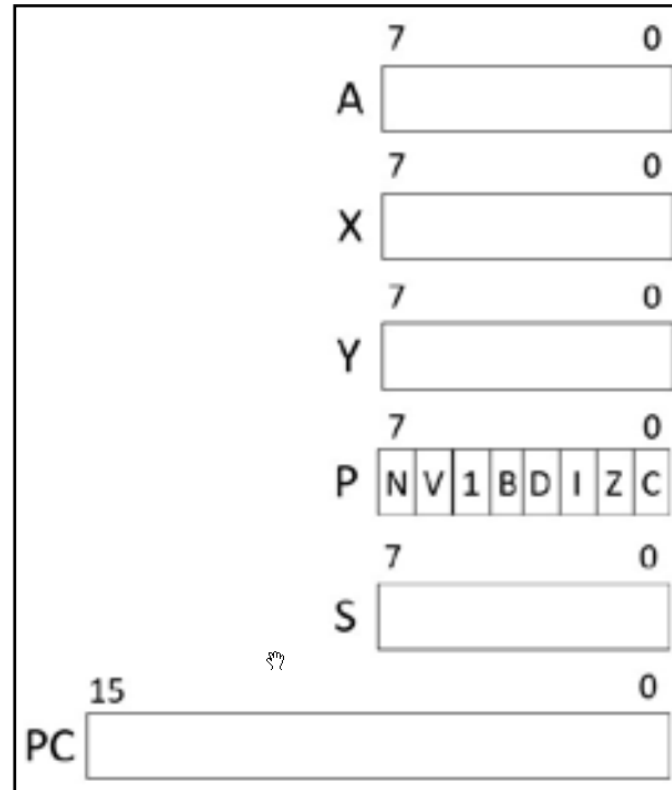
- 08CE
- 0000 1000 1100 1110
- Big endian zapis:
- 08CE
- 0000 1000 1100 1110
- Little endian zapis:
- CE08
- 1100 1110 0000 1000

6502

- Stari procesor
- Nastao 1975
- Koristi 5V napajanje
- 0-0.8V – logička 0
- 2-5V – logičko 1
- 8 bita
- 16 bit adresna sabirnica (65536 lokacija)

Registri

- A, X, Y, P, S, PC



General purpose (opća namjena)

- A, X, Y
- A jedini za aritmetiku
- X, Y mogu se koristiti kao indeksi za memoriju

Zastavice

- Registar P
- Bit 1 se ne koristi
- N – negativna vrijednost (ako je operacija postavila bit 7 rezultata)
- V – Overflow - ako operacija izađe iz raspona -128 do 127
- B – Break – izvršena je operacija BRK. Služi razlikovanju softverskog i hardverskog prekida
- D – Decimal mode – označava da procesor koristi BCD zapis brojeva

Zastavice

- I – Interrupt disable – procesor neće obrađivati prekide (osim NMI)
- Z – Zero – Rezultat operacije je 0
- C – Carry – operacija je uzrokovala carry

Stack pointer

- Registar S
- Odnosi se na memoriju \$100 - \$1FF
- Govorimo o 256 bajtova
- Postavlja se na vrh
- Svako spremanje na vrh smanjuje S
- Ofset je fiksnih \$100
- LIFO
- Korisnik mora paziti

Program counter

- PC
- 16 bita
- Svaka instrukcija je 1 bajt opcode iza čega je do dva operanda, ovisno o instrukciji

Instrukcije

- Mnemonici za opkodove
- Postoje različiti načini adresiranja
- Direktno adresiranje - #
- Indirektno adresiranje

Pravila pisanja

- 24 znači 24 decimalno
- \$24 znači da koristimo heksadecimalni zapis (36 decimalno)
- #24 znači izravna vrijednost 24
- #\$24 znači izravno 36 u decimalnom zapisu

Primjer iz knjige

- LDA – Load A with value
- ADC – ADD using carry
- SBC – Subtract using Carry
- SEC – Set carry directly
- CLC – Clear carry directly

Carry

- Mora biti podešen prije operacije
- ADC će ga koristiti i dodati ukupnoj vrijednosti
- SBC će ga koristiti kako bi ga dodao prije oduzimanja

- <https://skilldrick.github.io/easy6502/>

Instruction Sequence	Description	Results				
		A	N	V	Z	C
CLC LDA #1 ADC #1	8-bit addition with no Carry input: Clear the Carry flag, then load an immediate value of 1 into the A register and add 1 to it.	\$02	0	0	0	0
SEC LDA #1 ADC #1	8-bit addition with a Carry input: Set the Carry flag, then load an immediate value of 1 into the A register and add 1 to it.	\$03	0	0	0	0
SEC LDA #1 SBC #1	8-bit subtraction with no Borrow input: Set the Carry flag, then load an immediate value of 1 into the A register and subtract 1 from it. C = 1 indicates no borrow occurred.	\$00	0	0	1	1
CLC LDA #1 SBC #1	8-bit subtraction with a Borrow input: Clear the Carry flag, then load an immediate value of 1 into the A register and subtract 1 from it. C = 0 indicates a borrow occurred.	\$FF	1	0	0	0
CLC LDA #\$FF ADC #1	Unsigned overflow: Add 1 to \$FF. C = 1 indicates a carry occurred.	\$00	0	0	1	1
SEC LDA #0 SBC #1	Unsigned underflow: Subtract 1 from 0. C = 0 indicates a borrow occurred.	\$FF	1	0	0	0
CLC LDA #\$7F ADC #1	Signed overflow: Add 1 to \$7F. V = 1 indicates signed overflow occurred.	\$80	1	1	0	0
SEC LDA #\$80 SBC #1	Signed underflow: Subtract 1 from \$80. V = 1 indicates signed underflow occurred.	\$7F	0	1	0	1

Branching

- Grananje je jedan od osnovnih načina kontrole rada
- JMP – skače na neku lokaciju
- Direktno se upisuje u JMP
- BCC i BCS – skaču ako je Carry Clear ili Set
- BNE BEQ ako je Z clear ili set
- BPL BMI Ako je N clear ili set
- BVC BVS ako je V clear ili set
- Koristi se relativno adresiranje -128 do 127



**Hvala na
pažnji!**