↳ odečítání

① A := value - A → převést se dá na: A := A - value

STA ↓ temp1 ↓ LDA A

↳ přepsat pomocí sčítání

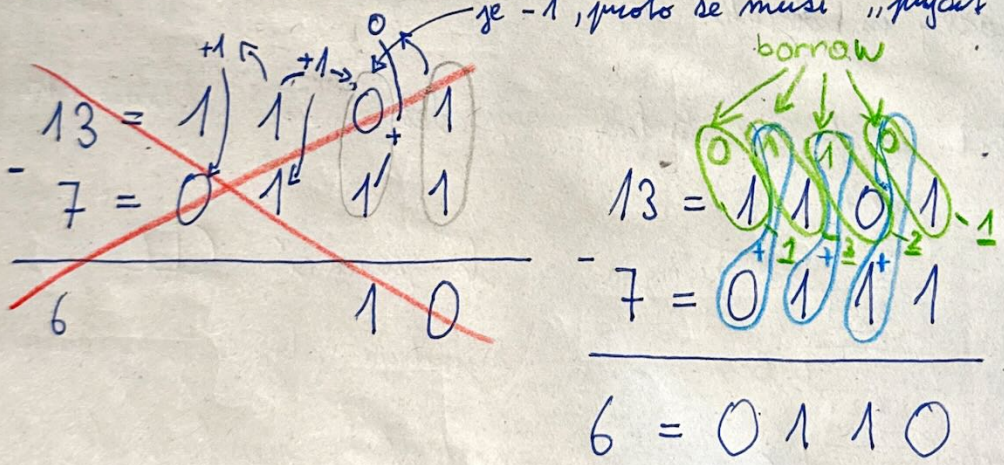
A := value + (-A) nebo-li A := (-A) + value

koncept: NOT A
INC A } do dvojkového doplňku
ADD value

správně: EOR #$FF ← A := (-A) + value
CLC
ADC #1
CLC
ADC value

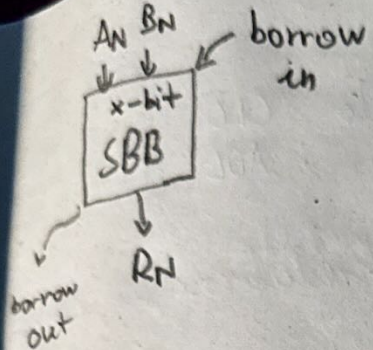
② → protože je časté, tak na instrukce existuje

SBB (subtract with borrow)




↳ propůjčím si číslo z vyššího řádu, abych získal větší číslo, od kterého odečítám. Ten borrow pak přičtu k odečítateli

↳ předpokládáme bezznaménkové odečítání




→ borrow se ukládá do carry příznaku

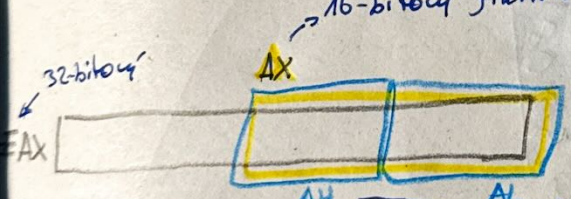
③ SBC (subtract with carry) → dělá SBB interně, ale do carry příznaku uloží negaci borrow

↳ explicitně nastavit carry na 1 → SEC, pak lze SBC

(„u SBB to je CLC")
↳ na 0

x86

↳ více registrů = více svobody → 32-bitové registry, 7 jich má




→ většinou procesory nemají 2 × 8-bitový registr, ale označává se to blíž k

↳ další registry: EBX, ECX, EDX

↳ u 64-bit (x64) → má 4 rozdělení registru: 64, 32, 16, 8

příklady instrukcí

↳ obecný tvar (assembleru) → OP target, source

MOV r, [addr]

↳ 32-bitová adresy do 32-bitového adresového prostoru

↳ lze provést i výpočet a pak to dát jako adresu

↳ ADD/ADC/SBB atd. r reg/imm/addr

↳ mají velké množství možností