

ZÁPADOČESKÁ UNIVERZITA V PLZNI
Fakulta aplikovaných věd

Počítačová technika

Násobení dvou 16 bit čísel bez použití instrukce pro násobení

Zadání

Násobení 16 bitů * 16 bitů = 32 bitů (bez použití instrukce násobení). Vstupy budou zadávány z konzole a výstupy zobrazovány také na konzoli, všechny vstupy a výstupu budou hexadecimálně.

Každý program musí obsahovat alespoň jeden kódový a jeden datový segment. Dále musí být řešení každé z úloh navrženo tak, aby obsahovalo alespoň jeden podprogram, který je v programu volán. Proto musí být správně inicializován a použit zásobník.

Popis řešení

Vstup a výstup

Vstup je z konzole přijímán jako jednotlivé ASCII znaky, ty je nutné převést na hexadecimální čísla, případně vypsát chybu že se nejedná o platné znaky pro převod do 16 soustavy.

U výstupu je naopak nutné převést hexadecimální číslo na platné ASCII znaky.

Algoritmus násobení bez instrukce pro násobení

Algoritmus násobení jsem vyřešil nejjednodušším možným způsobem, tedy jejich opakovaným sčítáním.

V „cyklu“ do výsledku opakovaně přičítám 1. operand a to tolikrát jakou hodnotu má 2. operand.

Například pro $2 \cdot 3$ tedy třikrát sečtu číslo dvě.

Existuje ještě jiný algoritmus výpočtu a tím je bitový posun, důvod k volbě mého algoritmu byl subjektivní pocit jednodušší implementace.

Vývojové prostředí

Program byl vyvinut v jazyce symbolický adres ve vývojovém prostředí HEW (High performance Embedded Workshop) vyvinutý společností Renesas. Součástí vývojového prostředí je i simulátor procesoru H8S/2600. Také používáme nástroje GNUH8 v12.02 Windows Tool Chain (ELF).

Definované proměnné a jejich umístění v paměti

Název	Adresa v paměti	Popis
buffer	00FF403C	Buffer pro načtení zadané hodnoty na klávesnici.
citel1	00FF4060	Pole pro 1. operand.
citel1_txt	00FF4000	Textový řetězec pro zadání 1. operandu.
citel2	00FF4062	Pole pro 2. operand.
citel2_txt	00FF400C	Textový řetězec pro zadání 2. operandu.
error	00FF4021	Textový řetězec pro informaci o chybně zadaném vstupu.
par1	00FF4046	Parametrický blok pro 1. operand.
par2	00FF404A	Parametrický blok pro buffer.
par3	00FF404E	Parametrický blok pro 2. operand.

Název	Adresa v paměti	Popis
par4	00FF4052	Parametrický blok pro chybu.
par5	00FF4056	Parametrický blok pro výsledek.
par6	00FF405A	Parametrický blok pro výstup.
soucin_txt	00FF4018	Textový řetězec pro informaci o výsledku.
stack	00FF40A8	Paměťový prostor pro zásobník.
vysledek	00FF4064	Pole pro výsledek výpočtu.
vystup_txt	00FF4033	Pole pro vypsání výsledku na konzoli.

Popis uživatelského ovládání

Ovládání je velmi jednoduché. Po zapnutí programu je uživatel textově vyzván k zadání nejprve 1. operandu a následně 2. operandu. Oba operandy je nutné zadat jako hexadecimální číslo. Povolené jsou tyto znaky 0-9 a A-F, program je case sensitive a přijímá pouze velká písmena.

Po jejich zadání a úspěšném převedení na hexadecimální čísla se provede samotný výpočet, tedy vynásobení zadaných operandů. Výsledek je vypsán zpět na konzoli jako 32 bitové hexadecimální číslo.

V případě, že uživatelský vstup nelze převést na hexadecimální číslo, vypíše se uživateli chybový text a je opětovně vyzván k zadání operandu.

Závěr

Program splňuje zadání úlohy, dokáže vynásobit dvě zadaná čísla bez použití instrukce pro násobení. Výstup je ukládán do paměti a následně vypsán na konzoli. Program by měl taktéž být schopen opakovaného běhu.

Program by šlo řešit i efektivněji, například zvolením jiného algoritmu nebo efektivnějším využitím registrů. Smyslem úlohy ovšem nebylo nalézt nejefektivnější řešení, ale získat zkušenosti s programováním v jazyku symbolických adres. Tento cíl byl podle mého názoru splněn.