Zachodniopomorski Uniwersytet Technologiczny w Szczecinie

EMULACJA PROCESORA Z RODZINY X86 Z WIZUALIZACJĄ WYKONYWANIA ROZKAZÓW MASZYNOWYCH

ALEKSANDRA KRZEMIEŃ

STUDIA I STOPNIA

OPIEKUN: DR INŻ PIOTR BŁASZYŃSKI

KATEDRA INŻYNIERII OPROGRAMOWANIA I CYBERBEZPIECZEŃSTWA

Wydział Informatyki

STRESZCZENIE I CEL

Celem pracy było zaprojektowanie i implementacja systemu, który jest 32-bitowym emulatorem procesora z rodziny x86 oraz wizualizuje proces wykonywania rozkazów maszynowych. Emulator obsługuje kod Assemblera, który można wpisać w dostępnym edytorze tekstu lub wczytać z pliku. W czasie wykonywania instrukcji, wyświetlane są aktualne wartości rejestrów procesora, pamięci oraz flagi. Emulator obługuje liczby całkowite i zmiennoprzecinkowe. Dodatkowo program pełni funckję debugera, umożliwiając wykonywanie kodu linia po linii oraz stawianie breakpointów. Obsługiwane są 3 wywołania systemowe: read, write, exit. Program został napisany w języku C++ z wykorzystaniem frameworku Qt.

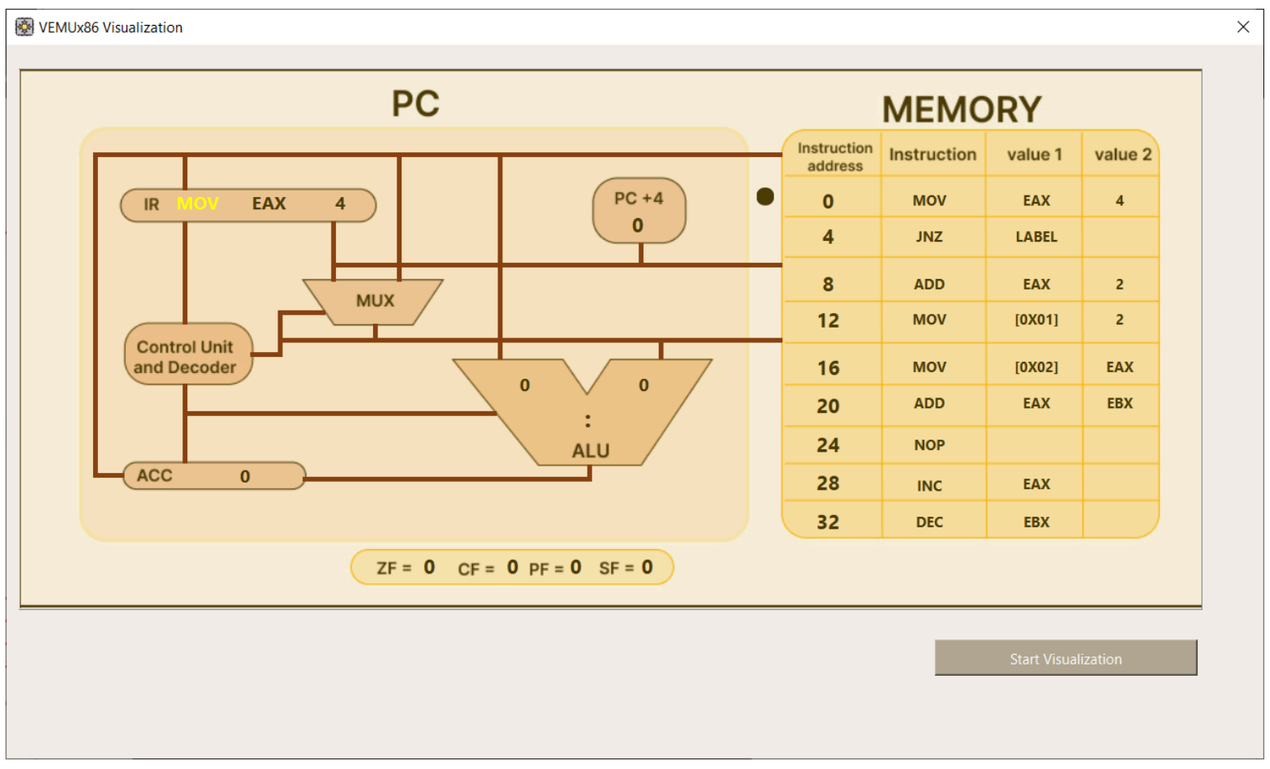
CECHY

Najważniejsze w tworzeniu emulatora było poprawne odzwierciedlenie działania rejestrów, pamięci i przetwarzanie instrukcji. Odtworzone zostały rejestry ogólnego przeznaczenia: EAX, EBX, ECX, EDX, rejestr EFLAGS, rejestry dodatkowe R8 - R15 oraz rejestry zmiennoprzecinkowe XMM0 -XMM7. Możliwe jest przetwarzanie kodu Asemblera w kilku rodzajach adresowania: natychmiastowe, rejestrowe, bezpośrednie, pośrednie, bazowe i bazowo-indeksowe.

Dużą częścią pracy było stworzenie interpretera języka Assembler NASM o składni Intel, by umożliwić przetwarzanie instrukcji. W ramach stworzonego interpretera realizowane są 3 sekcje: .text, .data i .bss. Sekcje .data i .bss umożliwiają korzystanie z pamięci oraz rezerwację pamięci.

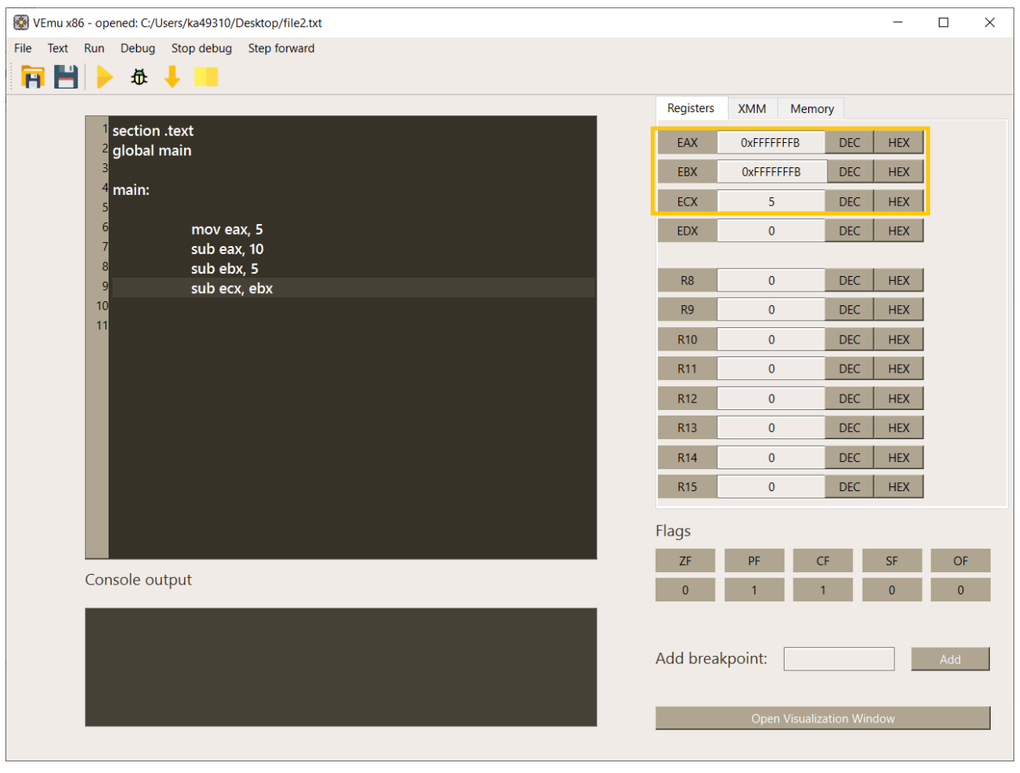
WIZUALIZACJA

Wizualizacja ma pełnić charakter edukacyjny, obrazujący kolejność i zasadę działania procesora oraz sposób w jaki przetwarza instrukcje. Kolejno podświetlają się odpowiednie rozkazy, operandy oraz połączenia elementów procesora pokazując przepływ danych. Wartości poszczególnych części procesora jak licznik rozkazów - PC, rejestr rozkazów - IR zmieniają swoje wartości. Dodatkowo wyświetlane są wartości rejestru flag (ZF, CF, PF, SF).



OKNO GŁÓWNE

Okno główne programu z widocznym edytorem tekstowym, rejestrami z prawej strony oraz oknem wyjścia konsoli gdzie pojawiają się komunikaty o błędach podczas interpretacji lub wartości wywołań systemowych. W pasku zakładek są 3 opcje bloków: rejestry ogólnego przezanczenia i R8 - R15, rejestry zmiennoprzecinkowe XMM i bloki pamięci. Pułpaki można ustawić klikając na pasek z lewej strony edytora lub ręcznie z prawej strony okna wpisując linijkę gdzie ma być postawiona pułapka.



PODSUMOWANIE

Udało się zrealizować założenia pracy i stworzyć program, który jest emulatorem procesora z rodziny x86 oraz wizualizuje wykonywanie rozkazów maszynowych. Interpreter prawidłowo wykonuje instrukcje języka Asembler w wielu rodzajach adresowania, a emulator właściwie manipuluje rejestrami oraz pamięcią. Zosotało zaimplementowane wiele instrukcji różnego rodzaju: transferu danych (np. MOV, LEA) , arytmetyczno-logiczne (np. ADD, DEC, MUL, OR, XOR, SHL) i przepływu sterowania (CMP, JMP, JMP IF). Wizualizacja pełni funkcję edukacyjną pokazując proces przetwarzania instrukcji krok po kroku poszczególnych elementów procesora, ich wartości i aktualne flagi.

Dalsze prace mogłyby dotyczyć rozszerzenia emulatora o większą ilość rejestrów czy umożliwienie wyboru tempa wyświetlanej wizualziacji.

BIBLIOGRAFIA

[1] Tomasz Wojtowicz. “Visualizing CPU Microarchitecture”. W: 24.6 (2015), s. 197–210.
[2] Gynvael Coldwind. Zrozumieć programowanie. PWN.
[3] x86 Instruction Set Reference. URL: https://c9x.me/x86/