Univerzitet Džemal Bijedić Mostar

Fakultet informacijskih tehnologija

**Taylorov red**

Predmet: Ugrađeni sistemi i sistemi za rad u realnom vremenu

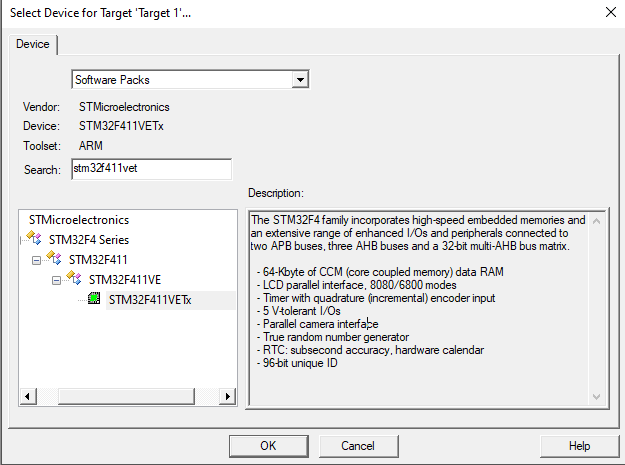
**Mentor: Student:**

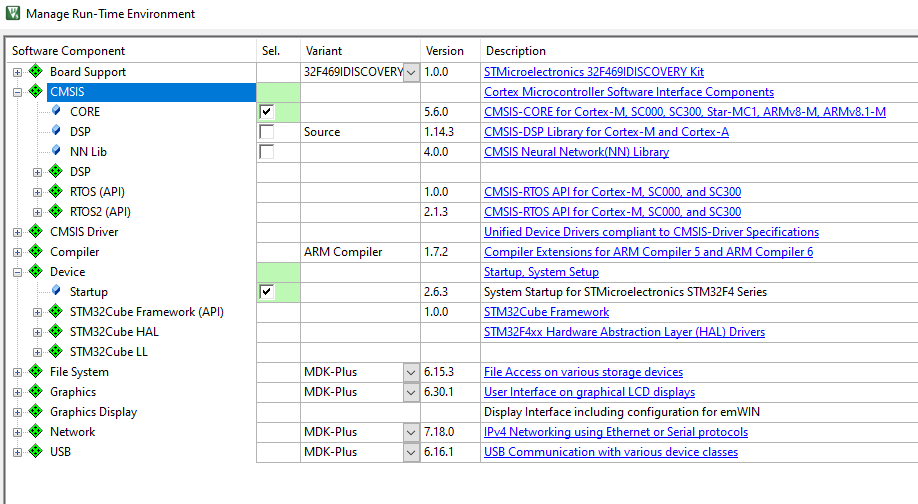
Prof. dr. Elmir Babović Zorka Kunovac

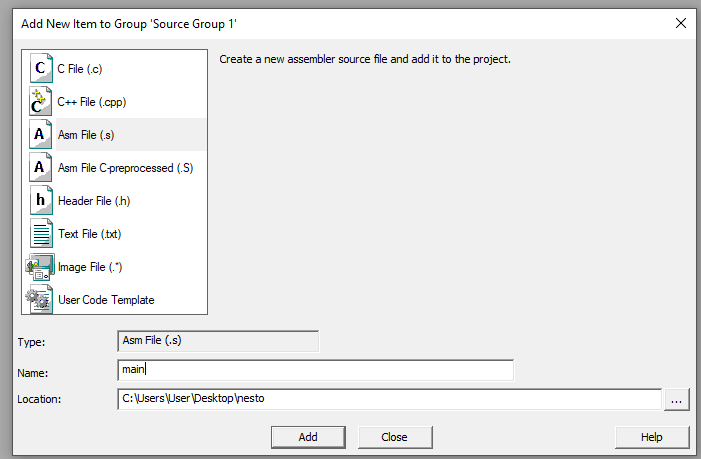
Mostar, mart 2023. godine

Za izradu projekta u asembleru korišteno je razvojno okruženje Keil Micro Vision 5, STM32F411VETx i simulator uz čiju pomoć se vrši debagiranje.

Za kreiranje koda u Keli µVision-u, u ovom slučaju, potrebno je prilikom kreiranje projekta izabrati STM32F411VETx. Nakon toga na Device označiti Startup a na CMSIS označiti Core, zatim u Source Group dodati „new item“, i odabrati Asm File(.s). Naposljetku, na options for target se oznaci Use MicroLIB a na Debug – Use Simulator.







U ovom projektu se radi na Taylorovoj seriji odnosno Taylorovom redu. Uprošteno govoreći, Taylorov niz je proširenje neke funkcije u beskonačan zbir članova, gdje svaki član ima veći eksponent poput x, x2, x3, itd.

Taylorova serija također uključuje računanje faktorijela ali zbog pojednostavljivanja koda odnosno prikaza same Taylorove serije računanje faktorijela je prikazano kao odvojeni zadatak razdvojen u dva tipa a to su: primjer faktorijela sa IT blokom i primer rekurzivnog faktorijela.

Najprije će biti objašnjena Taylorova serija a zatim dva primjera sa faktorijelom.

Dakle, u ovom slučaju radi se o Taylovoj seriji sa 4 uslova: (x = x - x^3/3! + x^5/5! - x^7/7! ) gdje formula funkcioniše za svako x u rangu [0, pi/2].

Registri koji se koriste za računanje su:

S0 - ulazni operand i povratni rezultat

S1 - 1/3! (invfact3)

S2 - 1/5! (invfact5)

S3 - 1/7! (invfact7)

S4 - x \* S1 (xdiv3), zatim S4\*S7 (x^2 \* xdiv3) (x3div3)

S5 - x \* S2 (xdiv5), zatim S5\*S8 (x^4 \* xdiv5) (x5div5)

S6 - x \* S3 (xdiv7), zatim S6\*S9 (x^6 \* xdiv7) (x7div7)

S7 - x ^ 2

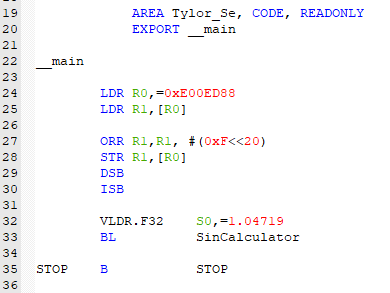
S8 - x ^ 4

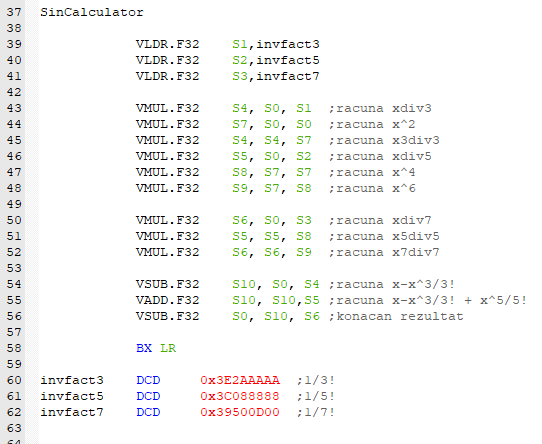
S9 - x ^ 6

S10 - pomoćni registar

U projektu se izbjegava dijeljenje računanjem 1/3! , 1/5! itd.

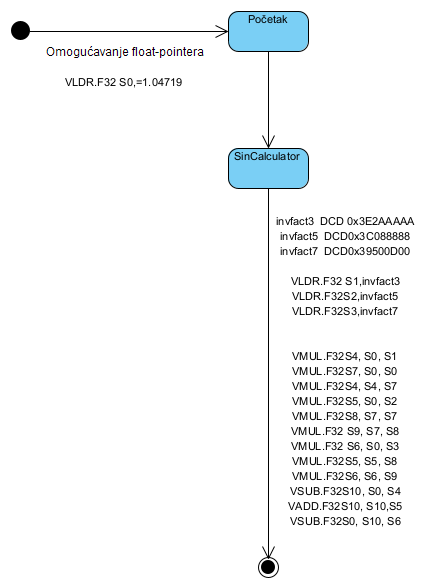
Na sljedećoj slici je prikazan kod glavne funkcije a zatim SinCalculator.



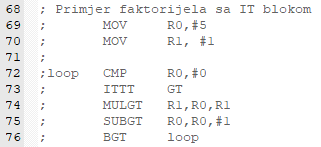


Radi lakšeg razumijevanja koda unutar SinCalculator dodani su komentari. Što se tiče koda u main dijelu linije koda od 24 do 30 se koriste za omogućavanje float pointera a nakon toga se postavlja ulazni operand koji se obračunava u sinus kalkulatoru.

Na sljedećoj slici je prikazan FMS dijagram ovog dijela programa.

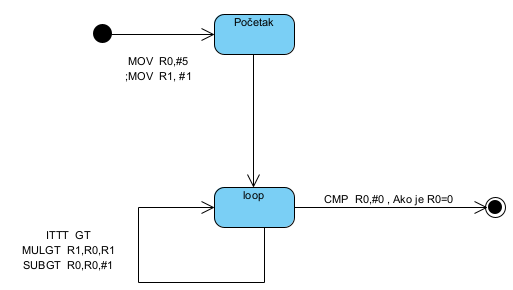


U nastavku je prikazan primjer faktorijela sa IT blokom.

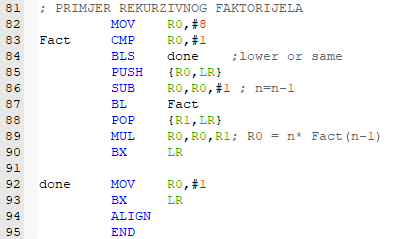


U registar R0 se smješta broj 5 a u registar R1 broj 1. Nakon toga slijedi loop gdje se prvo vrši komparacija vrijednosti registra R0 sa 0. Ukoliko R0 ima vrijednost veću od 0, množi se R0 i R1 te se rezultat tog množenja smješta u R1, te se vrijednost R0 umanjuje. Petlja se ponavlja sve dok je R0 veće od 0. ITTT predstavlja IT blok i ima tri slova T jer se uspostavlja uslov GT (Greater than) za naredne 3 linije koda.

Na sljedećoj slici je prikazan FMS dijagram faktorijela sa IT blokom.



Na narednoj slici je prikazan kod rekurzivnog faktorijela.



U ovom slučaju u registar R0 smješta se broj 8. Nakon toga slijedi branch Fact. U Fact se prvo vrši komparacija da li je R0 jednako 1. Ako je manje ili jednako prelazi se na branch done koji dodjeluje R0 vrijednost 1 i tu završava.

Ukoliko to nije slučaj vrijednost R0 se stavlja na stek i umanjuje za 1. Petlja se ponavlja sve dok uslov ne bude ispunjen tj. dok R0 ne bude manje ili jednako od 1. Kad se uslov ostvari onda se sa steka uzimaju vrijednosti i smještaju u registar R1 i zatim se množi R0 sa R1 i ta vrijednost se smješta u R0.