

Vad vi gör under veckan (Lv4)

I början av denna vecka avslutar vi behandlingen av styrenheten genom att implementera villkorliga hopp med PC-relativ adressering och fortsätter sedan med att visa hur FLEX-processorn kommunicerar med omvärlden via in- och utportar.

Vi studerar också en normal och en maskinorienterad programflödesplan för ett multiplikationsprogram, lär oss bestämma hur lång tid ett programavsnitt tar att köra, använder verktyg för programutveckling, visar hur man kan adressera data i minnet via indexregistret X och avslutar med att introducera begreppen stack och subrutiner.

Den automatiska styrenhet för FLEX-processorn

Vi visar exempel på hur villkorliga hoppinstruktioner kan implementeras i styrenheten.

FLEX-datorns inportar och utportar

Hur FLEX-datorn kommunicerar med omvärlden.

Normal och maskinorienterad flödesplan

Exempel: Multiplikationsprogram

Exekveringstid för programavsnitt

Hur lång tid tar det för processorn att köra ett programavsnitt som innehåller slingor?

FLEX-datorns utvecklingsmiljö (Eterm för FLEX)

Editor, assembler, laddare och simulator. Anslutning av yttre enheter i simulatorn.

Adressering av data via indexregistret X

Programexempel med adressering via X-registret.

Principen för en stack i minnet

Vi förklarar hur FLEX-processorns stack fungerar och hur den används i vissa instruktioner.

Instruktionsuppsättning för FLEX-processorn (forts.)

Instruktioner som använder stack.

Subrutinexempel

Några olika exempel på hur subrutiner anropas från huvudprogram och hur stackpekaren SP och stacken påverkas.

Under veckan utför vi laboration 2 och förbereder laboration 3

(De uppgifter i arbetsboken och de hemuppgifter i lab-PM som anges på försättsbladet till laboration 3 skall arbetas igenom.)