

A ROM-ban tárolt aritmetikai rutinok használata

TVC

Előző számunkban bemutattuk a TV-Computer ROM-jának adatmozgató szubrutinjait. A gép az adatmozgató rutinokhoz hasonlóan a négy alpműveletet is a veremben elhelyezett számok között végzi el a következő módon.

A veremben egymás fölé kell tenni a két operandust (OP1, OP2), amelyek között elvégzi a műveletet. Az eredmény az elsőnek betett (OP1) szám helyén keletkezik, az OP2 pedig elvész (definiálatlan).

A szubrutinok hívása az előzőekben már ismertetett módon, az RST 18H utasítást követően, a szubrutin sorszámanak megadásával történhet. Ebben az esetben is igaz, hogy ha a sorszám legnagyobb helyiértékű (128-as súlyú) bitje 1, akkor ez az utolsó végrehajtandó szubrutin sorszáma; az ezt követő kódot a processzor már utasításként értelmezi és végrehajtja.

Az 1. táblázat a szubrutinokat mutatja be. IY a veremmutató, a verem pedig a memóriában fentről lefelé terjed. Az utolsó sorban a 04-es sorszámú szubrutin a veremben levő szám előjelét változtatja ellenkezőjére.

A gép beépített BASIC függvénykészlete hívható a felhasználói gépi kódú programokból is, bár itt meg kell jegyezni, hogy a függvények használatával a gépi kódú programozás legfőbb előnye, a futási idő jelentős csökkenése nem érvényesül.

Ez a probléma a beépített függvények ki-kerülésével oldható meg úgy, hogy a nem használt függvényt a felhasználó által megírt szubrutinnal pótoljuk. Természetesen ehhez gyorsabb algoritmusra van szükség.

A gépi kódú felhasználói függvények a ROM ismertetett adatmozgató és aritmetikai szubrutinjainak alkalmazásával készülhetnek, de ha a futási idő nem elsődleges, akkor minden további nélkül hívhatják a 2. táblázatban szereplő függvényeket. Hívás előtt az argumentumot a verembe kell helyezni, az eredmény ugyanitt keletkezik.

A 3. táblázatban összefoglalásképpen megadjuk az RST 18H utáni sorszámkhoz tartozó szubrutinok memóriabeli címét, így azok CALL-lal gyorsabban hívhatók. Ekkor a program hossza nő. A „!”-et csak az RST 18H-val célszerű hívni.

A függvények egy numerikus értéket szolgáltatnak eredményül, és attól függően, hogy egy vagy több bemenő paraméterük van, felépítésükben lehet némi eltérés.

A függvényeket megvalósító gépi kódú rutinok szerkezete:

— az első két bájtt az előző BASIC válto-

Szubrutin hívása	OP1 helye	OP2 helye	Művelet	Eredmény helye
RST 18H 00H	IY	IY-9	+	IY
RST 18H 01H	IY	IY-9	/	IY
RST 18H 02H	IY	IY-9	*	IY
RST 18H 03H	IY	IY-9	-	IY
RST 18H 04H	IY	nincs	CHS(X)	IY

1. táblázat

Függvény hívása	A függvény BASIC megfelelője
CALL EAECH	ABS(X)
CALL EBA7H	COS(X)
CALL EBB8H	EXP(X)
CALL ECAFH	INT(X)
CALL ED21H	LOG(X)
CALL EDE3H	PI
CALL EE3EH	SGN(X)
CALL EE5EH	SIN(X)
CALL EECaH	SQR(X)
CALL EEFBH	TAN(X)

2. táblázat

3. táblázat

RST 18H utáni sorszám	CALL-lal hívható cím	Funkció	Megjegyzés
00H	F493H	OP1=OP1+OP2	IY=IY+9
01H	F5FBH	OP1=OP1/OP2	IY=IY+9
02H	F512H	OP1=OP1*OP2	IY=IY+9
03H	F48EH	OP1=OP1-OP2	IY=IY+9
04H	F726H	OP1=CHS(OP1)	IY=IY
05H	EA82H	(IY)=N. KONST.	IY=IY-9
06H	EA9FH	(IY)=X reg.	IY=IY-9
07H	EA9AH	(IY)=Y reg.	IY=IY-9
08H	EAD2H	X reg.=(IY)	IY=IY
09H	EACDH	Y reg.=(IY)	IY=IY
0AH	EAC3H	X reg.=(IY)	IY=IY+9
0BH	EABEH	Y reg.=(IY)	IY=IY+9
0CH	FA92H	(IY-9)=(IY)	IY=IY-9
0DH	FB28H	(HL)=(IY)	IY=IY+9
0EH	EA68H	Lásd gépi kódú felhasználói függvények.	