Адрес	Код команды	Мнемоника	Комментарии
03B	0200	CLA	
03C	A05B	LD LD X_LOW	Load lower part of X into AC
03D	0C00	Push it to (stack-1)	Push X_low onto stack (SP decreases)
03E	A05D	LD Y_LOW	Load lower part of Y into AC
03F	0C00	PUSH it to (SP -1 -1)	Push Y_low onto stack (SP decreases)
040	D65F	CALL 0x65F (sp -3)	Call subroutine. Returns sum_low in AC, Carry set.
041	0800	Pop (sp -2)	Clean Y_low from stack (Pop into dummy R0, or AC if no R0) Clean X_low from stack (Pop into dummy R0, or AC)
042	0800	Pop (SP -1 (result))	
043	E05F	ST R_LOW	ALT x , y TOPLAMI KAYDEDİLDİ
044	A05C	LD X_HIGH	
045	2062	AND SIGN_BIT_MASK	
046	F004	BEQ SKIP_EXT_X	If AC is zero (sign bit was 0, positive), tekrar x yükle
047	A05C	LD X_HIGH	Sign bit was 1 (negative). Reload ORIGINAL X_high from stack
048	3061	OR SIGN_MASK	Apply sign extension mask (FFF0)
049	0C00	PUSH(SP-1)	For Negative push
04A	C04D	JUMP 4D	
04B	A05C	LD X_HIGH	If positive, then it branched here loaded ac again
04C	0C00	PUSH(SP-1)	For Positive push
04D	A05E	LD Y_HIGH	Load higher part of Y
04E	2062	AND SIGN_BIT_MASK	
04F	F004	BEQ SKIP_EXT_X	If AC is zero (sign bit was 0, positive), skip 04A
050	A05E	LD Y_HIGH	Sign bit was 1 (negative). Reload ORIGINAL X_high from stack
051	3061	OR SIGN_MASK	Apply sign extension mask (FFF0)
052	0C00	PUSH(SP-2)	
053	C056	JUMP 4D	
054	A05E	LD Y_HIGH	If positive, then it branched here loaded ac again
055	0C00	PUSH(SP-1)	For Positive push

056	D663	CALL 663	Call subroutine. Returns sum_high in AC.
057	0800	POP	Clean Y_high_ext from stack
058	0800	POP	Clean X_high_ext from stack
059	E060	ST r_HIGH	Store higher 16 bits of result (contains bits 16-19 and sign ext)
05A	0100	HLT	
05B	0x1111		X_LOW
05C	0x0001		X_HIGH
05D	0x2222		Y_LOW
05E	0x000A		Y_HIGH
05F	0x00BB		R_LOW
060	0x00CC		R_HIGH
061	FFF0		SIGN_MASK
062	0008		SIGN_BIT_MASK

SUBPROGRAM:

Адрес	**Код команды**	**Мнемоника**	**Комментарии**
	ADD_LOW		**Input: (SP+1)=X_low, (SP+2)=Y_low**
65F	AC02	LD (SP+2)	AC = MEM(SP+2) 7FF = lower x
660	4C01	ADD (SP +1)	SP +1 = 7FE => Lower x + lower y
661	EC02		
662	0A00	RET	Возврат IP = SP
	ADD_HIGH		**Input: (SP+1)=X_high_ext, (SP+2)=Y_high_ext, Carry flag set from ADD_LOW**
663	AC02	LD SP+2	Load Y_high_ext into AC
664	5C01	ADC (SP+1)	AC = Y_high_ext + X_high_ext + Carry
	EC02		
665	0A00	RET	Возврат IP = SP