

Microcontroller

Selbststudium Semesterwoche 6

6.1 Aufruf einer leeren Funktion, ohne Parameter

```

/*****
void function(void)
/*****
{
    asm NOP;
}

/*****
Hauptprogramm
*****/
void main(void) {

    function();
    for(;;){
    }

};

```

C-Code Debug

Einige Schritte fehlen, da hier nur C-Code debugged wurde. Korrekterweise müssten Assembly Steps debugged werden.

Trace	PC	SP	HX	nächster Befehl	
				C	Asm
BP:	193D	2FF	1900	function()	BSR *-2
1	193B	2FD	1900	{NOP	NOP
2	193C	2FD	1900	}	RTS
3	193F	2FF	1900	for(;;){	BRA *+0

6.2 Aufruf einer Funktion mit lokalen Variablen, ohne Parameterübergabe

```

/*****/
void function(void)
/*****/
{
    unsigned char ucl;
    int il;
    ucl = 1;
    il = (int)ucl;
    ucl++;
}
/*****/
Hauptprogramm
/*****/
void main(void) {
    function();
    for(;;){
    }
};
    
```

Assembly Debug:

Trace	PC	SP	HX	Nächster Befehl	
				C	Ass
BP:	194D	2FF		function()	BSR *-18
1	193B	2FD		function(void)	AIS #-3
2	193D	2FA		ucl = 1	LDA #0x01
3	193F			ucl = 1	TSX
4	1940		02FB	ucl = 1	STA 0,X
5	1941			il = (int)ucl	LDX #01
6	1943		0201	il = (int)ucl	CLRH
7	1944		0001	il = (int)ucl	STHX 2,SP
8	1947			ucl++	INCA
9	1948			ucl++	TSX
10	1949		02FB	ucl++	STA 0,X
11	194A			}	AIS 3
12	194C	2FD		}	RTS
13	194F	2FF		for(;;)	BRA +0

Stack:

		:	SP nach Trace:
	\$02F2	uu	
	\$02F3	uu	
	\$02F4	uu	
	\$02F5	uu	
	\$02F6	uu	
	\$02F7	uu	
	\$02F8	uu	
	\$02F9	uu	
	\$02FA		2
	\$02FB	1 / 02	ucl
	\$02FC	00	il - H
	\$02FD	01	il - L 1, 12
	\$02FE	19	
Bottom	\$02FF	4F	BP, 13
		:	

6.3 Aufruf einer Funktion mit lokalen Variablen, mit Parametern und Rückgabe

```
/******  
char function (unsigned char ucVal, unsigned char *ucRef)  
/******  
{  
    unsigned char ucl1;  
    ucl1 = ucVal;  
    ucl1++;  
    (*ucRef)++; /* Achtung: wenn die Klammern fehlen, wird der Pointer  
    inkrementiert */  
    return ucl1;  
}  
/******  
Hauptprogramm  
*****  
void main(void){  
    unsigned char uc1=0, uc2=0, uc3;  
    uc3 = function(uc1, &uc2);  
    uc3++;  
    for(;;){  
    }  
};
```

Assembly Debug:

Trace	PC	SP	HX	nächster Befehl	
				C	Ass
BP:	1947	2FF		main(void)	AIS #-3
1	1949	2FC		unsigned char uc1 =0;	TSX
2	194A		02FD	unsigned char uc1=0	CLR 2,X
3	194C			unsigned char uc2=0, uc3	CLR 0,X
4	194D			uc3 = function(uc1, &uc2);	CLRA
5	194E			uc3 = function(uc1, &uc2);	BSR *-19
6	193B	2FA		char function (unsigned char ucVal, unsigned char *ucRef)	PSHH
7	193C	2F9		unsigned char uc1; uc1 = ucVal;	STA 1,SP
8	193F			uc1++;	INC 1,SP
9	1942			(*ucRef)++;	INC 0,X
10	1943			return uc1;	TSX
11	1944		02FA	return uc1;	LDA 0,X
12	1945			}	PULH
13	1946	2FA		}	RTS
14	1950	2FC		uc3 = function(uc1, &uc2);	TSX
15	1951		02FD	ur3 = function(uc1, &uc2);	STA 1,X
16	1953			uc3++	INC 1,X
17	1955			for(;;){}	BRA *+0

1 → reserviert Speicher für drei Variablen uc1=0, uc2=0, uc3=0

Stack:

	:	SP nach Trace:
\$02F2	uu	
\$02F3	uu	
\$02F4	uu	
\$02F5	uu	
\$02F6	uu	
\$02F7	uu	
\$02F8	uu	
\$02F9		7
\$02FA	02 / 00 / 01	6, 13 uc1
\$02FB	19	
\$02FC	50	1, 14
\$02FD	00 / 01	uc2
\$02FE	01 / 02	uc3
Bottom \$02FF	00	BP uc1
	:	

Kurzfragen zu 6.3

1. Wie werden die Parameter *ucVal* und *ucRef* beim Funktionsaufruf übergeben?
 → *ucVal* wird über Register übergeben
 → *ucRef* wird über den Stack übergeben
2. Wie heissen diese Arten der Parameterübergabe?
 → *call-by-value*, *call-by-reference*
3. Wie wird das Resultat der Funktion ans Hauptprogramm übergeben?
 → über den Akku A
4. Wofür werden im Programm die Befehle *PSH H* und *PUL H* verwendet? Welcher andere Befehl wäre möglich? Warum wird hier *PSH H* und *PUL H* verwendet?
 → *PSHH: AIS -1* / *PULH AIS +1*

6.4 Aufruf einer Funktion mit Parametern vom Typ struct (optional)

```

/*****
Globale Typ Definition
*****/
struct point
{
int x;
int y;
};
/*****
struct point addpoint(struct point p1, struct point p2)
/*****
{
p1.x += p2.x;
p1.y += p2.y;
return p1;
}
/*****
Hauptprogramm
*****/
void main(void) {
struct point pkt1 = {1, 3};
struct point pkt2 = {2, 4};
struct point pkt3;
pkt3 = addpoint(pkt1, pkt2);
for(;;){}
};

```