University of Florida	EEL 4744	Drs. E. M. Schwartz &A. Antonio Arroyo
Department of Electrical & Computer Engineering		Professors in ECE
Page 1/5	PARAMETER PASSING	15-Mar-01 11:29 AM

## PARAMETER PASSING METHODS

How do you pass parameters between the subroutines (or interrupts) and the main routine or other subroutines?

- 1. Pass the parameter(s) (data or pointer) in the internal registers.
- 2. Pass the parameter(s) immediately after the call instruction, i.e. in the program memory space. (This requires that the parameter(s) be fixed at assembler time.)
- 3. Pass a pointer to the location of the parameter(s) immediately after the call instruction.
- 4. Pass the parameter(s) on the stack prior to the call. (PSH)
- 5. Pass a pointer to parameter(s) on the stack prior to the call. (PSH)

The Problem: Find the average of two numbers

Solution 1	: ]	Pass the parameter(s) in the internal registers.			
Solution 1a	Solution 1a: Pass the parameter <i>data</i> in the internal registers.				
START: LDS LDAA LDAB			<pre>;Start program at \$B600 ;initialize stack pointer ;Load data in the A and B ; registers ;Call the subroutine AVG to ; get average</pre>		
* * * * * * *	* * * * *	* * * * * * * * * * * * * * * * *	* * * * * * * * * * * * * * * * * * * *		
		e of inputs in a accumulator A	accumulators A and B		
AVG:	ABA		;A=A+B		
	ASRA		<pre>;Shift A right by 1 bit ; (divide by 2) keeping ; bit 7 (for sign ; extension)</pre>		
	RTS				
Solution 11	<u>o</u> : ]	Pass the parameters add	dresses in internal pointer registers.		
	ORG	\$0000			
DATA:	FCB	\$37, \$A3			
	ORG	\$B600	;Start program at \$B600		
START:	LDS	#\$0041	;initialize stack pointer		
	LDX	#DATA	;Load X index reg. with		
			; address of data		
	JSR	AVG1	;Call the subroutine AVG1 ; to get average		
* * * * * * *	••• *****	* * * * * * * * * * * * * * * * *	******		
* Get average of two data bytes in successive memory					
* starting at location pointed to by X; Output in A					

University	of Florida		EEL 4744	Drs. E. M. Schwartz &A. Antonio Arroyo
Department	of Electrical	& Computer Engi	neering	Professors in ECE
Page 2/5			PARAMETER PASSING	15-Mar-01 11:29 AM
AVG1:	LDAA	0,X	;A = 1st piece of data	
	ADDA	1,X	;A=A + 2nd piece of data	
	ASRA		;Divide by 2	
	RTS			

## Solution 2

Pass the parameter(s) immediately after the call instruction, i.e., in the program memory space. (This requires that the parameter(s) be fixed at assemble time.)

Since data follows the call, the return address pushed on the top of the stack by the subroutine call must be corrected before returning from the subroutine.

START: DATA1: DATA2:	JSR FCB	\$B600 #\$0041 AVG2 \$37 \$A3	;Start program at \$B600 ;initialize stack pointer ;Call the subroutine AVG2
NEXT:			
******	* * * * * * * *	* * * * * * * * * * * * *	* * * * * * * * * * * * * * * * * * * *
* Get a	verage o	f two data by	ytes in program memory at
* locat	ion poin	ted to by dat	ta on top of stack
* Outpu	t in A		
* X aff	ected		
* Stack	return	address corre	ected
	ORG	\$B700	
AVG2:	PULX		;Get address of data
	LDAA	0,X	;Get first piece of data
			; (at DATA1) into A
	ADDA	1,X	;A=A + (data at DATA2)
	ASRA		;Divide by 2;Result in A
	INX		;Increment the X index
	INX		; register twice to point
			; to next instruction upon
			; RTS
	PSHX		;Push the corrected address
			; back on stack
	RTS		

LABEL	Address	Value
START	\$B600	\$8E
	\$B601	\$00
	\$B602	\$41
	\$B603	\$BD
	\$B604	\$B7
	\$B605	\$00
DATA1	\$B606	\$37
DATA2	\$B607	\$A3
NEXT	\$B608	

\$003B	
\$003C	
\$003D	
\$003E	
\$003F	
\$0040	\$B6 (Data1 Hi)
\$0041	\$06 (Data1 Lo)

University of Florida	EEL 4744	Drs. E. M. Schwartz &A. Antonio Arroyo
Department of Electrical & Computer Engineering		Professors in ECE
Page 3/5	PARAMETER PASSING	15-Mar-01 11:29 AM

## Solution 3

Pass a pointer to the location of the parameter(s) immediately after the call instruction.

Similar to Solution 2 except now a pointer to the data is passed (instead of the data itself), so that the data does not have to be know at assemble time.

	ORG	\$0001	
LOCAT:	FCB	\$37, \$A3	;Define location of data
			; and some default data
	ORG	\$B600	;Start program at \$B600
START:		#\$0041	;initialize stack pointer
	JSR	AVG3	;Call the subroutine AVG3
DAT_AD:	FDB	LOCAT	;Define the location of
			;data
NEXT:			
			* * * * * * * * * * * * * * * * * * * *
			ytes in program memory at
	-	ted to by po	inter on top of stack
* Outpu			
* X aff			
* Stack	return	address corre	ected
	ORG	\$B700	
AVG3:	PULX		;Get address of pointer to
			; data
	INX		;Increment the X index
	INX		
	TINY		; register twice to point
	TINY		; to next instruction
	INA		; to next instruction ; upon RTS
	PSHX		<pre>; to next instruction ; upon RTS ;Push the corrected address</pre>
			; to next instruction ; upon RTS
			<pre>; to next instruction ; upon RTS ;Push the corrected address</pre>
	PSHX		<pre>; to next instruction ; upon RTS ;Push the corrected address ; back on stack</pre>
	PSHX DECX	0 , X	<pre>; to next instruction ; upon RTS ;Push the corrected address ; back on stack ;Make X point to pointer at</pre>
	PSHX DECX DECX	0,X 0,X	<pre>; to next instruction ; upon RTS ;Push the corrected address ; back on stack ;Make X point to pointer at ; DAT_AD</pre>
	PSHX DECX DECX LDX		<pre>; to next instruction ; upon RTS ;Push the corrected address ; back on stack ;Make X point to pointer at ; DAT_AD ;Put the pointer into X</pre>
	PSHX DECX DECX LDX LDAA	0,X	<pre>; to next instruction ; upon RTS ;Push the corrected address ; back on stack ;Make X point to pointer at ; DAT_AD ;Put the pointer into X ;Get first piece of data</pre>

LABEL	Address	Value
START	\$B600	\$8E
	\$B601	\$00
	\$B602	\$41
	\$B603	\$BD
	\$B604	\$B7
	\$B605	\$00
DATA_AD	\$B606	\$00
DATA2	\$B607	\$01
NEXT	\$B608	

\$003B	
\$003C	
\$003D	
\$003E	
\$003F	
\$0040	\$B6 (DAT_AD Hi)
\$0041	\$06 (DAT_AD Lo)

•	of Florida	& Computer Enginee	EEL 4744	Drs. E. M. Schwartz &A. Antonio Arroyo Professors in ECE
Page 4/5	of Electrical	& Computer Enginee	PARAMETER PASSING	15-Mar-01 11:29 AM
Solution 4		on the stack prior to	o the call. (PSH)	
r uss the p			,, · (1)	
	ORG	\$B600	;Start program at \$B600	
START:	LDS	#\$0041	; initialize stack pointer	r
	LDAA	#\$37	;Load data onto stack	
	PSHA			
	LDAA PSHA	#\$A3		
	JSR	AVG4A	;Call the subroutine AVG4	4A/AVG4B
	•••			
			* * * * * * * * * * * * * * * * * * * *	* * *
		-	puts on the stack	
_	affecte	umulator A		
	modifie			
beach	ORG	\$B700		
AVG4A:	PULX		;Save return address	
	PULA		;Get second piece of data	a
	PULB			
	ABA		;A=A+B	
	ASRA		;Divide by 2	
	PSHX		;Fix stack for return	
	חשפ		; address	
	RTS			
******	* * * * * * * *	****	* * * * * * * * * * * * * * * * * * * *	* * *
* Get t	he avera	ge of the in	puts on the stack	
* Outpu	t in acc	umulator A		
* A,X a	ffected			
* Stack	unmodif			
	ORG	\$B700		
AVG4B:	TSX	0 v	; IX = SP+1 & IX points to	o the stack
	LDAA ADDA	2,X 3,X	;A = $1^{st}$ parameter ;A = A + $2^{nd}$ parameter	
	ADDA ASRA	<b>υ</b> , Δ	;A = A + 2 parameter ;Divide by 2	
	RTS		, Divide by 2	
_				

University				Drs. E. M. Schwartz &A. Antonio Arroyo
Department Page 5/5	of Electrical	& Computer Engineer	PARAMETER PASSING	Professors in ECE 15-Mar-01 11:29 AM
Solutio	n 5			
Pass th	e addres	s of the para	ameter(s) on the stack pric	or to the call. (PSH)
	ORG	\$0001		
LOCAT1:		\$37	;Define location of data	
	ORG	0006н		
LOCAT2:		\$A3	; and some default data	
	ORG	\$B600	;Start program at \$B600	
START:	LDS	#\$0041	;initialize stack pointer	
	LDX	#LOCAT2	;Put location of 2nd data	
	PSHX		; onto stack	
	LDX	#LOCAT1	;Put location of 1st data	
	PSHX		; onto stack	
	JSR	AVG5A	;Call the subroutine AVG54	A/5B
* * * * * * *	••• ******	* * * * * * * * * * * * *	* * * * * * * * * * * * * * * * * * * *	* *
* Get a	verage o	f numbers po:	inted to by addresses	
	stack			
* Outpu	t in acc	umulator A		
		ted; stack mo	odified	
	ORG	\$B700		
AVG5A	PULX		;Save return address	
	PULY		;Get address of 1st data	
	LDAA	О,Ү	;Put first data in A	
	PULY		;Get address of 2nd data	
	ADDA	О,Ү	;A=A + 2nd data	
	ASRA		;Divide by 2	
	PSHX		;Fix stack for return add.	
	RTS			
*****	******	* * * * * * * * * * * * *	* * * * * * * * * * * * * * * * * * * *	*
* Get a	verage o	f numbers po	inted to by addresses	
	stack	r numberb po.		
		umulator A		
		ted; stack u	nmodified	
	ORG	\$B700		
AVG5B:	TSX	72700	;IX = SP+1 & IX points to	the stack
	LDY	2,X	;Get address of 1st data	
	LDAA	0,Y	;Put first data in A	
	LDY	4,X	Get address of 1st data	
	ADDA	0,Y	;A=A + 2nd data	
	ASRA		;Divide by 2	
	RTS			

٠

What if you wanted the subroutine to have no effect on any registers? Can you think of any other ways to pass data? How about using RAM for variables! •