

P0087CXZ GMCM MACRO CROSS-ASSEMBLER RELEASE 4.03 DATASET: P 04/25/86 15:22 PAGE 399
 1987 PORT FUEL INJECTION SIMULTANEOUS

SUBROUTIN
 01674500
 01674600
 01674700

*/** SUBROUTINES CALLED : NONE
 **/
 **/
 /***

P0087CXZ GMCM MACRO CROSS-ASSEMBLER RELEASE 4.03 DATASET: P 04/25/86 15:22 PAGE 398
 1987 PORT FUEL INJECTION SIMULTANEOUS

SUBROUTN
 01669300 9510 F36E
 01669400
 01669500
 01669600
 01669700
 01669800
 01669900
 01670000
 01670100
 01670200
 01670300
 01670400
 01670500
 01670600
 01670700
 01670800
 01670900
 01671000
 01671100
 01671200
 01671300
 01671400
 01671500
 01671600
 01671700
 01671800
 01671900
 01672000
 01672100
 01672200
 01672300
 01672400
 01672500
 01672600

P4M1616B SURR
 /***
 **/ 4/18/84 REV A **/
 **/ P4M1616B C2XCG **/
 /***

DESCRIPTION - PERFORMS A 16 X 16 FIXED POINT BINARY MULTIPLY
 FUNCTION:
 THIS ROUTINE PERFORMS THE 16 X 16 MULTIPLY AND
 RETURNS THE MIDDLE TWO BYTES AND THE UPPER
 TWO BYTES.
 MULTIPLICATION IS DONE USING THE MUL INSTRUCTION
 ROUNDING TAKES PLACE FROM THE LOW ORDER BYTE OF
 PRODUCT.
 THE MIDDLE TWO BYTES WILL BE SET TO \$FFFF IF THE
 HIGH ORDER BYTE OF THE PRODUCT IS NON ZERO.

PARAMETERS : NONE
 CALLING ARGUMENTS :
 X REG= MULTPLICAND
 A ACC= MS BYTE OF MULTIPLIER
 R ACC= LS BYTE OF MULTIPLIER
 RETURNING ARGUMENTS:
 A,B = THE MIDDLE TWO BYTES OF THE 32 BIT PRODUCT
 X REG= UPPER TWO BYTES OF THE 32 BIT PRODUCT

SUBROUTN

01674900	9511	F36E 36	P4M1616B	PSH A	
01675000	9512	F36F 3C		PSH X	
01675100	9513	F370 3C		PSH X	
01675200	9514	F371 37		PSH B	
01675300	9515	F372 36		PSH A	
01675400			*		
01675500	9516	F373 30		TSX	
01675600	9517	F374 A6 03		LDA A	3,X
01675700	9518	F376 3D		MUL	
01675800	9519	F377 89 00		ADC A	#0
01675900	9520	F379 A7 05		STA A	5,X
01676000	9521	F378 EC 01		LDA D	1,X
01676100	9522	F37D 3D		MUL	
01676200	9523	F37E EB 05		ADD B	5,X
01676300	9524	F380 89 00		ADC A	#0
01676400	9525	F382 ED 04		STA D	4,X
01676500	9526	F384 A6 00		LDA A	0,X
01676600	9527	F386 E6 03		LDA B	3,X
01676700	9528	F388 6F 06		CLR	6,X
01676800	9529	F38A 3D		MUL	
01676900	9530	F38B E3 04		ADD D	4,X
01677000	9531	F38D 69 06		ROL	6,X
01677100	9532	F38F ED 04		STA D	4,X
01677200	9533	F391 A6 00		LDA A	0,X
01677300	9534	F393 E6 02		LDA B	2,X
01677400	9535	F395 3D		MUL	
01677500	9536	F396 EB 04		ADD B	4,X
01677600	9537	F398 A9 06		ADC A	6,X
01677700	9538	F39A E7 04		STA B	4,X
01677800	9539	F39C ED 02		STA D	2,X
01677900	9540	F39E 4D		TST A	
01678000	9541	F39F 27 05(F3A6)		BEQ	DKM16B
01678100	9542	F3A1 CC FFFF		LDA D	#FFFFFF
01678200	9543	F3A4 ED 04		STA D	4,X
01678300	9544	F3A6 38	DKM16B	PUL X	
01678400	9545	F3A7 38		PUL X	
01678500	9546	F3A8 32		PUL A	
01678600	9547	F3A9 33		PUL B	
01678700	9548	F3AA 31		INS	
01678800	9549	F3AB 39		RTS	
01678900				ENDS	

& P RESERVE SPACE ON STACK FOR MS BYTE OF PRODUCT;
& P RESERVE SPACE ON STACK FOR PARTIAL PRODUCTS
& P MULTIPLICAND TO STACK
& P LS MULTIPLIER TO STACK
& P MS MULTIPLIER TO STACK

& P LS MULTIPLICAND X LS MULTIPLIER

& P LS MULTIPLIER X MS MULTIPLICAND

& P MS MULTIPLIER X LS MULTIPLICAND

& P CLEAR MS BYTE OF PRODUCT;

& P MS MULTIPLIER X MS MULTIPLICAND

& P MS PRODUCT 16 BITS IN A,B;

& P SAVE HIGH 2 BYTES IN X REGISTER

& D HIGH BYTE NONZERO?;

& IF NO GOTO OK: ELSE YES;

& P SET RESULT = \$FFFF;

& P RESTORE STACK;

& P RESTORE HIGHEST BYTES TO X REGISTER;

& P RESULTS RETURNED IN A,B;

& T RETURN;