Приложение B: КОМАНДЫ МИКРОПРОЦЕССОРА СМ 601

Обозначения:
A - accumulator A
B - accumulator B
M - Memory operand (8 bit)
M(16) - 16-bit memory operand
S - Stack pointer
X - Index register
P - processor status register

Флаг состояния процессора:
H - Half Carry
I - Interrupt disable
N - Negative
Z - Zero
V - Overflow
C - Carry
. - unchanged
* - changed
0 - cleared to 0
1 - set to 1

мнемо код / схема действия команды	вид адресации						HINZVC
	Inh	Rel 2	Imm 2	Dir 2	Ind 2	Ext 3	

```
ABA   A+B   -> A  1B  2                                  *.****
— Add B to A

ADCA  A+M+C -> A             89  2 99  3 A9  5 BD  4 *.****
— Add Memory with Carry to A

ADCB  B+M+C -> B             C9  2 D9  3 E9  5 F9  4 *.****
— Add Memory with Carry to B

ADDA  A+M   -> A             8B  2 9B  3 AB  5 BB  4 *.****
— Add Memory to A

ADDB  B+M   -> B             CB  2 DB  3 EB  5 FB  4 *.****
— Add Memory to B

ANDA  A&M   -> A             84  2 94  3 A4  5 B4  4 ..**0.
— Logic AND Memory to A

ANDB  B&M   -> B             C4  2 D4  3 E4  5 F4  4 ..**0.
— Logic AND Memory to B
```

мнемо код	схема действия	вид адресации												HINZVC
команды		Inh		Rel 2		Imm 2		Dir 2		Ind 2		Ext 3		

```
ASL   shift left                          68  7 78  6 ..****
— Arithmetic Shift left M

ASLA  shift left  48  2                                ..****
— Arithmetic Shift left A

ASLB  shift left  58  2                                ..****
— Arithmetic Shift left B

ASR   Aritm shr                           67  7 77  6 ..****
— Arithmetic Shift right M

ASRA  Aritm shr   47  2                                ..****
— Arithmetic Shift right A

ASRB  Aritm shr   57  2                                ..****
— Arithmetic Shift right B

BCC   if C = 0          24  4                          ......
— Branch if no carry

BCS   if C = 1          25  4                          ......
— Branch if carry

BEQ   if Z = 1          27  4                          ......
— Branch if equal

BGE   if N ^ V =0       2C  4                          ......
— Branch if greater or equal

BGT   Z|(N^V) = 0       2E  4                          ......
— Branch if greater

BHI   C | Z = 0         22  4                          ......
— Branch if High

BITA  A & M                 85  2 95  3 A5  5 B5  4 ..**0.
— Bit test M with A

BITB  B & M                 C5  2 D5  3 E5  5 F5  4 ..**0.
— Bit test M with B

BLE   Z|(N^V) = 1       2F  4                          ......
— Branch if Less or Equal
```

мнемо код команды	схема действия	вид адресации Inh	Rel 2	Imm 2	Dir 2	Ind 2	Ext 3	HINZVC

```
BLS   C | Z = 1         23  4                          ......
— Branch if Low or Same

BLT   N ^ V = 1         2D  4                          ......
— Branch if Little

BMI   N = 1             2B  4                          ......
— Branch if Minus

BNE   Z = 0             26  4                          ......
— Branch if Not Equal

BPL   N = 0             2A  4                          ......
— Branch if Plus

BRA   uncond            20  4                          ......
— Branch Always

BSR   Brnch Subrt       8D  8                          ......
— Branch To SubRoutine

BVC   V = 0             28  4                          ......
— Branch if Overflow Clear

BVS   V = 1             29  4                          ......
— Branch if Overflow Set

CBA   A - B       11  2                                ..****
— Compare A with B

CLC   0 -> C      0C  2                                .....0
— Clear Carry

CLI   0 -> I      0E  2                                .0....
— Clear Interrupt mask

CLR   0 -> M                                6F  7 7F  7 ..0100
— Clear Memory

CLRA  0 -> A      4F  2                                ..0100
— Clear A

CLRB  0 -> B      5F  2                                ..0100
— Clear B
```

мнемо код	схема действия	вид адресации						HINZVC
команды		Inh	Rel 2	Imm 2	Dir 2	Ind 2	Ext 3	

```
CLV   0 -> V      0A  2                                  ....0.
— Clear oVerflow

CMPA  A - M                 81  2 91  3 A1  5 B1  4 ..****
— Compare A with M

CMPB  B - M                 C1  2 D1  3 E1  5 F1  4 ..****
— Compare B with M

COM   not M -> M                      63  7 73  7 ..**01
— Compliment (not) memory

COMA  not A -> A  43  2                                ..**01
— Compliment (not) A

COMB  not B -> B  53  2                                ..**01
— Compliment (not) B

CPX   X - M(16)             8C* 3 9C  4 AC  6 BC  5 ..***.
— Compare ind. reg. X with 16-bit Memory

DAA   dec adjust  19  2                                ..****
— Decimal adjust A

DEC   M - 1 -> M                      6A  7 7A  6 ..***.
— Decrement M

DECA  A - 1 -> A  4A  2                                ..***.
— Decrement A

DECB  B - 1 -> B  5A  2                                ..***.
— Decrement B

DES   S - 1 -> S  34  4                                ......
— Decrement Stack Pointer

DEX   X - 1 -> X  09  4                                ...*..
— Decrement Index Reg. X

EORA  A ^ M -> A            88  2 98  3 A8  5 B8  4 ..**0.
— Excluding OR memory to A

EORB  B ^ M -> B            C8  2 D8  3 E8  5 F8  4 ..**0.
— Excluding OR memory to B
```

мнемо код команды	схема действия	вид адресации Inh	Rel 2	Imm 2	Dir 2	Ind 2	Ext 3	HINZVC
INC	M + 1 -> M					6C 7	7C 6	..***.
— Increment Memory								
INCA	A + 1 -> A 4C 2							..***.
— Increment A								
INCB	B + 1 -> B 5C 2							..***.
— Increment B								
INS	S + 1 -> S 31 4							.....
— Increment Stack Pointer								
INX	X + 1 -> X 08 4							...*..
— Increment Index reg. X								
JMP	jump					6E 4	7E 3	.....
JSR	jump Subrtn					AD 8	BD 9	.....
— Jump to SubRoutine								
LDAA	M -> A			86 2	96 3	A6 5	B6 4	..**0.
— Load A								
LDAB	M -> B			C6 2	D6 3	E6 5	F6 4	..**0.
— Load B								
LDS	M(16) -> S			8E* 3	9E 4	AE 6	BE 5	..**0.
— Load Stack Pointer								
LDX	M(16) -> X			CE* 3	DE 4	EE 6	FE 5	..**0.
— Load Index reg. X								
LSR	Logic shr					64 7	74 6	..0***
— Logical Shift Right								
LSRA	Logic shr 44 2							..0***
— Logical Shift Right A								
LSRB	Logic shr 54 2							..0***
— Logical Shift Right B								
NEG	0 - M -> M					60 7	70 6	..****
— Negate M								

мнемо код	схема действия	вид адресации						HINZVC
команды		Inh	Rel 2	Imm 2	Dir 2	Ind 2	Ext 3	

```
NEGA  0 - A -> A  40  2                                    ..****
— Negate A

NEGB  0 - B -> B  50  2                                    ..****
— Negate B

NOP   NoOPeration 01  2                                    ......

ORAA  A | M -> A            8A  2 9A  3 AA  5 BA  4       ..**0.
— OR A

ORAB  B | M -> B            CA  2 DA  3 EA  5 FA  4       ..**0.
— OR B

PSHA  Push A      36  4                                    ......
— Push A

PSHB  Push B      37  4                                    ......
— Push B

PULA  Pull A      32  4                                    ......
— Pull A

PULB  Pull B      33  4                                    ......
— Pull B

ROL   rol C<-M<-C                       69  7 79  6       ..****
— Rotate Left

ROLA  rol C<-B<-C 49  2                                    ..****
— Rotate Left A

ROLB  rol C<-B<-C 59  2                                    ..****
— Rotate Left B

ROR   ror C->M->C                       66  7 76  6       ..****
— Rotate Right

RORA  ror C->A->C 46  2                                    ..****
— Rotate Right A

RORB  ror C->B->C 56  2                                    ..****
— Rotate Right B
```

мнемо код	схема действия	вид адресации						HINZVC
команды		Inh	Rel 2	Imm 2	Dir 2	Ind 2	Ext 3	

```
RTI   Rtrn Intrpt 3B 10                                        ******
— Return from Interrupt

RTS   Rtrn Subrtn 39  5                                        ......
— Return from SubRoutine

SBA   A - B -> A  10  2                                        ..****
— Subtract B from A

SBCA  A-M-C -> A             82  2 92  3 A2  5 B2  4           ..****
— Subtract with Carry from A

SBCB  B-M-C -> B             C2  2 D2  3 E2  5 F2  4           ..****
— Subtract with Carry from B

SEC   1 -> C                                                   .....1
— Set Carry flag

SEI   1 -> I                                                   .1....
— Set Interrupt flag

SEV   1 -> V                                                   ....1.
— Set Overflow flag

STAA  A -> M                   97  4 A7  6 B7  5               ..**0.
— Store A

STAB  B -> M                   D7  4 E7  6 F7  5               ..**0.
— Store B

STS   S -> M(16)               9F  5 AF  7 BF  6               ..**0.
— Store Stack Pointer

STX   X -> M(16)               DF  5 EF  7 FF  6               ..**0.
— Store index register X

SUBA  A - M -> A             80  2 90  3 A0  5 B0  4           ..****
— Subtract from A

SUBB  B - M -> B             C0  2 D0  3 E0  5 F0  4           ..****
— Subtract from B

SWI   Soft Intrpt 3F 12                                        .1....
— Software Interrupt
```

мнемо код	схема действия	вид адресации						HINZVC
команды		Inh	Rel 2	Imm 2	Dir 2	Ind 2	Ext 3	

```
TAB   A -> B      16  2                                  ..**0.
— Transfer A to B

TAP   A -> P      06  2                                  ******
— Transfer A to Processor status

TBA   B -> A      17  2                                  ..**0.
— Transfer B to A

TPA   P -> A      07  2                                  ......
— Transfer Processor status to A

TST   M - 0                                 6D  7 7D  6 ..**00
— Test Memory

TSTA  A - 0       4D  2                                  ..**00
— Test A

TSTB  B - 0       5D  2                                  ..**00
— Test B

TSX   S + 1 -> X  30  4                                  ......
— Transfer Stack pointer to X

TXS   X - 1 -> S  35  4                                  ......
— Transfer X to Stack pointer

WAI   Wait Intrpt 3E  9                                  .1....
— Wait for Interrupt
```