KIM-1 Multiplication and Division

James C Couchman, General Dynamics Corp, Fort Worth Division, POB 748, Fort Worth TX 76101

When I bought a MOS Technology KIM-1 microcomputer to use in a specific control function, it arrived with a set of comprehensive instruction, programming, and hardware books. As soon as I connected a 5 V power supply, I was able to interact with the machine through the hexadecimal keyboard and light-emitting diode (LED) display. It was a bit more difficult to get our Teletype to work with the KIM-1, but with a slight adjustment to the teleprinter timing, the problem was cured.

The KIM-1 is still a real bargain, with features including the 6502 microprocessor, 2 K bytes of read-only memory (containing the Keyboard Input Monitor from which the name is derived), an interval timer, fifteen input and output lines, 1 K bytes of programmable memory (with address logic for 16 K bytes), and probably some features I have not yet discovered.

Since the KIM-1 is programmed in machine language using a set of fifty-six instructions, I believe that the best way to learn to program it is to not just read about it, but do it. One should just start writing code, and, in time, the power of the basic instruction set will really be understood and appreciated.

Once the user is familiar with the capabilities of the KIM-1, he begins to wish that it could do more. One tool that provides more capability is a set of software routines that perform sixteen-bit multiplication and division on the 6502 processor. After I searched for a suitable set of routines, I concluded that I would have to write my own.

To prevent you from having to "reinvent the wheel," I am presenting these routines here. In developing these routines, I enlisted the invaluable assistance of my associates G R Arnett and J R Williamson. These routines should work without much difficulty on other 6502-based computers.

Sixteen-Bit Routines

These routines can multiply and divide two 16-bit signed quantities together and produce a signed 16-bit result. The routines are written as relocatable subroutines.

In multiplication, the high-order byte of the first multiplicand is loaded into hexadecimal location 0000, and the low-order byte into location 0001. The high-order byte of the second multiplicand is put into location 0006, and the low-order byte into location 0007.

In division, the high-order byte of the divisor is loaded into hexadecimal location 0000; the low-order byte into location 0001. The high-order byte of the dividend is placed into location 0006, and the low-order byte is loaded into location 0007. If the value of the divisor is zero, the division routine will return control to the calling program.

For both the multiplication and the division routines, the answer is returned in hexadecimal locations 0002 (high-order) and 0003 (low-order byte). It should not be very hard to change this if need be.

An example of a simple calling routine is shown in listing 1. The calling sequence is essentially the same for both multiplication and for division; only the value contained in the two bytes that follow the jump-to-subroutine (JSR) instruction must be changed.

Listing 1a: *Calling sequence for 16-bit multiply subroutine.*

Address	Code
0007	20 (JSR)
0008	00
0009	01 (multiply)
000A	A9 (LDA)
000B	00
000C	F0
000D	FC

Listing 1b: *Calling sequence for 16-bit divide subroutine.*

Address	Code
0007	20 (JSR)
0008	30
0009	00 (divide)
000A	A9 (LDA)
000B	00
000C	F0
000D	FC

The assembler mnemonics and hexadecimal code for the multiplication subroutine are given in listing 2. The division subroutine is given in similar form in listing 3. The multiplication subroutine is shown in hexadecimal memory-dump form in listing 4; the division code in that form in listing 5.

My collegues and I hope that these programs will help other KIM-1 users. We know that having had them prepared for us would have saved us much time.

Listing 2: *Relocatable subroutine to perform multiplication of 16-bit quantities on the 6502 microprocessor as used in the MOS Technology KIM-1. Both assembler mnemonics and hexadecimal code are given. Entry point is hexadecimal location 0100.*

Address	Mnemonic	Hexadecimal Code
0100	CLC	18
0101	CLD	D8
0102	LDA #0	A9 00
0104	TAX	AA
0105	STA 0002	85 02
0107	STA 0003	85 03
0109	LDA 0000	A5 00
010B	BNE	D0 11
010D	LDA 0001	A5 01
010F	BEQ	F0 0C
0111	CMP #1	C9 01
0113	BNE	D0 1D
0115	LDA 0006	A5 06
0117	STA 0002	85 02
0119	LDA 0007	A5 07
011B	STA 0003	85 03
011D	RTS	60
011E	BPL	10 12
0120	INX	E8
0121	LDA 0001	A5 01
0123	CLC	18
0124	EOR FF	49 FF
0126	ADC #1	69 01
0128	STA 0001	85 01
012A	LDA 0000	A5 00
012C	EOR FF	49 FF
012E	ADC #0	69 00
0130	STA 0000	85 00
0132	LDA 0006	A5 06
0134	BNE	D0 26
0136	LDA 0007	A5 07
0138	BEQ	F0 18
013A	CMP #1	C9 01
013C	BNE	D0 32
013E	DEX	CA
013F	BNE	D0 12
0141	LDA 0001	A5 01
0143	CLC	18
0144	EOR FF	49 FF
0146	ADC #1	69 01
0148	STA 0003	85 03
014A	CDA 0000	A5 00
014C	EOR FF	49 FF
014E	ADC #0	69 00
0150	STA 0002	85 02
0152	RTS	60
0153	LDA 0001	A5 01
0155	STA 0003	85 03
0157	LDA 0000	A5 00
0159	STA 0002	85 02
015B	RTS	60
015C	BPL	10 12
015E	INX	E8
015F	LDA 0007	A5 07

Listing 2 continued on page 215

Listing 2 continued:

```
0161    CLC          18
0162    EOR FF       49 FF
0164    ADC #1       69 01
0166    STA 0007     85 07
0168    LDA 0006     A5 06
016A    EOR FF       49 FF
016C    ADC #0       69 00
016E    STA 0006     85 06
0170    LDA 0000     A5 00
0172    STA 0004     85 04
0174    LDA 0001     A5 01
0176    STA 0005     85 05
0178    LDA 0003     A5 03
017A    CLC          18
017B    ADC 0001     65 01
017D    STA 0003     85 03
017F    LDA 0002     A5 02
0181    ADC 0000     65 00
0183    STA 0002     85 02
0185    SEC          38
0186    LDA 0007     A5 07
0188    SBC #1       E9 01
018A    STA 0007     85 07
018C    LDA 0006     A5 06
018E    SBC #0       E9 00
0190    STA 0006     85 06
0192    CMP #0       C9 00
0194    BNE          D0 E2
0196    LDA 0007     A5 07
0198    CMP #0       C9 00
019A    BNE          D0 DC
019C    DEX          CA
019D    BNE          D0 15
019F    LDA 0002     A5 02
01A1    EOR FF       49 FF
01A3    STA 0002     85 02
01A5    LDA 0003     A5 03
01A7    EOR FF       49 FF
01A9    CLC          18
01AA    ADC #1       69 01
01AC    STA 0003     85 03
01AE    LDA 0002     A5 02
01B0    ADC #0       69 00
01B2    STA 0002     85 02
01B4    RTS          60
```

Listing 3: *Relocatable subroutine to perform division of 16-bit quantities on the 6502 microprocessor of the KIM-1, with assembler mnemonics. Entry point is hexadecimal location 0030.*

Address	Mnemonic	Hexadecimal Code
0030	CLC	18
0031	CLD	D8
0032	LDA #0	A9 00
0034	TAX	AA
0035	STA 02	85 02
0037	STA 03	85 03
0039	LDA 00	A5 00
003B	BNE	D0 05
003D	LDA 01	A5 01
003F	BNE	D0 15
0041	RTS	60
0042	BPL	10 12
0044	INX	E8
0045	LDA 01	A5 01
0047	CLC	18
0048	EOR FF	49 FF
004A	ADC #1	69 01
004C	STA 01	85 01
004E	LDA 00	A5 00
0050	EOR FF	49 FF

Listing 3 continued on page 216

Listing 3 continued:

```
0052    ADC #0      69 00
0054    STA 00      85 00
0056    LDA 06      A5 06
0058    BNE         D0 26
005A    LDA 07      A5 07
005C    BEQ         F0 18
005E    CMP #1      C9 01
0060    BNE         D0 32
0062    DEX         CA
0063    BNE         D0 12
0065    LDA 01      A5 01
0067    CLC         18
0068    EOR FF      49 FF
006A    ADC #1      69 01
006C    STA 03      85 03
006E    LDA 00      A5 00
0070    EOR FF      49 FF
0072    ADC #0      69 00
0074    STA 02      85 02
0076    RTS         60
0077    LDA 01      A5 01
0079    STA 03      85 03
007B    LDA 00      A5 00
007D    STA 02      85 02
007F    RTS         60
0080    BPL         10 12
0082    INX         E8
0083    LDA 07      A5 07
0085    CLC         18
0086    EOR FF      49 FF
0088    ADC #1      69 01
008A    STA 07      85 07
008C    LDA 06      A5 06
008E    EOR FF      49 FF
0090    ADC #0      69 00
0092    STA 06      85 06
0094    LDA 03      A5 03
0096    CLC         18
0097    ADC #1      69 01
0099    STA 03      85 03
009B    LDA 02      A5 02
009D    ADC #0      69 00
009F    STA 02      85 02
00A1    SEC         38
00A2    LDA 01      A5 01
00A4    SBC 07      E5 07
00A6    STA 01      85 01
00A8    LDA 00      A5 00
00AA    SBC 06      E5 06
00AC    STA 00      85 00
00AE    LDA 00      A5 00
00B0    BMI         30 08
00B2    BNE         D0 E0
00B4    LDA 01      A5 01
00B6    BNE         D0 DC
00B8    BEQ         F0 0D
00BA    SEC         38
00BB    LDA 03      A5 03
00BD    SBC #1      E9 01
00BF    STA 03      85 03
00C1    LDA 02      A5 02
00C3    SBC #0      E9 00
00C5    STA 02      85 02
00C7    DEX         CA
00C8    BNE         D0 15
00CA    LDA 02      A5 02
00CC    EOR FF      49 FF
00CE    STA 02      85 02
00D0    LDA 03      A5 03
00D2    EOR FF      49 FF
00D4    CLC         18
00D5    ADC #1      69 01
00D7    STA 03      85 03
00D9    LDA 02      A5 02
00DB    ADC #0      69 00
00DD    STA 02      85 02
00DF    RTS         60
```

Listing 4: *Multiplication subroutine in hexadecimal memory-dump form.*

```
: 18010018D8A900AA85028503A500D011A501F00CC901D01DA506850974
: 18011802A5078503601012E8A5011849FF69018501A50049FF69000B1D
: 1801308500A506D026A507F018C901D032CAD012A5011849FF69010A0B
: 1801488503A50049FF690085026 0A5018503A50085026 01012E8A5086F
: 1801600718 49FF69018507A50649FF69008506A500850AA501850501B
: 18017 8A503186501 8503A5026500850236A507E9018507A506E90007C0
: 1801908506C900D0E2A507C900D0DCCAD015A50249FF8502A503490BE6
: 1801A8FF18690185 03A5026900850260E9008D0600C900D0D8AD070962
: 0000080008    0
```

Listing 5: *Division subroutine in hexadecimal memory-dump form.*

```
: 18003018D8A900AA85028503A500D005A501D015601012E8A5011808C7
: 18004849FF69018501A50049FF69008500A506D026A507F018C9010992
: 180060D032CAD012A5011849FF69018503A50049FF69008502 60A50A00
: 18007801 8503A50085026 01012E8A5071849FF69018507A5049FF08AA
: 1800904900 8506A5011869018503A5026900850238A501E5078501073 5
: 1800A8A500E5068500A5003008D0E0A501D0DCF00D38A503E901850B00
: 1800C003A502E9008502CAD015A50249FF8502A50349FF18690185 0A0E
: 1800D803A5026900850260002A3F3C3B3F1E3E3E3EFE1E3E7E7A7F07B2
: 0000080008 ■
```