SBC, TSX and TXS
Instructions of the 6800 and 6502

by B.T.G. Tan

B. T. G. Tan, Department of Physics, National University of Singapore.

The 6800 and 6502 microprocessors have very similar architectures and instruction sets. The common instructions of these two microprocessors have been listed elsewhere,[1,2] and the purpose of this note is to draw attention to three common instructions which have subtle but important differences in their operation on the two MPUs. These are:

SBC	Subtract with carry (on the 6800 the mnemonic is followed by A or B indicating the accumulator concerned)
TSX	Transfer stack pointer to index register
TXS	Transfer index register to stack pointer

SBC

The differences in the carry-bit operation for the subtract instruction of the 1802 and Z80 microprocessors have been pointed out by Merrin.[3] Likewise, the differences in SBC between the 6800 and 6502 are due to the different ways in which the two MPUs utilize the carry bit of their status registers. Though SBC is nominally described as subtract with carry, a better description would be subtract with borrow.

The add with carry instruction, ADC, is also common to the 6800 and 6502, and functions identically on them. The operation can be represented as

$$A + M + C$$

where A is the contents of the accumulator, M the operand, and C the carry bit of the status register. The result is put into the accumulator and the resulting carry bit is stored in C. The resulting carry bit will be 1 only if the result exceeds 255. In the case of the 6502, ADC is the only addition instruction, and if an add without carry is required, the carry bit must be cleared first by preceding ADC with CLC (clear carry bit).

To understand the SBC instruction better, let us assume that the status registers of both MPUs have another status bit called the borrow bit. The operation of the SBC instruction can then be represented as

$$A - M - B$$

where B is the borrow bit. B, as well as M, is subtracted from A, and the result is put into the accumulator. The borrow bit resulting from the operation will be stored in B. The resulting borrow bit will be 1 only if the absolute value of (M + B) is greater than that of A. The result of the operation goes into the accumulator. The function of B is to facilitate multi-byte subtraction when borrow bits from one 8-bit subtraction have to be utilized in the subtraction for the next higher 8-bits.

If the subtraction without borrow is required, the borrow bit must be cleared before SBC. We can invent two pseudo-instructions

SEB	Set borrow bit
CLB	Clear borrow bit

to enable us to manipulate B, analogously to SEC and CLC for C. Hence we must precede SBC with CLB for a subtract without borrow.

In the 6800 and 6502 MPUs, there is no separate borrow bit in their status registers. The carry bit C does double duty as a borrow bit, so that it is better described as a carry/borrow bit. In the 6800, the borrow bit is identical with the carry bit, so that the SBC operation can be represented as

$$A - M - C$$

and the borrow bit generated is directly put into C. The instructions SEB and CLB correspond to SEC and CLC respectively.

In the 6502, the borrow bit is the inverse of the carry bit, so that SBC is

$$A - M - \overline{C}$$

and the borrow bit generated is inverted before being put into C. Hence the SEB instruction corresponds to CLC and CLB to SEC. For a subtraction without borrow, SBC would have to be preceded by SEC.

The difference is thus best understood by considering the borrow bit B as entirely separate from C and then taking $B = C$ for the 6800 and $B = \overline{C}$ for the 6502. The implementation of the borrow bit for the 6502 is different from the 6800 in order that arithmetic operations on the 6502 would give more consistent results for C.

For example, the arithmetic operations 5 - 3 and 5 + (-3), using the two's complement in the second operation, would lead to different values for C in the case of the 6800. These operations would give the same values for C in the case of the 6502, since its borrow bit is the inverse of C. In Table 1 (below), the two operations are listed for each MPU. showing the corresponding values of the accumulator and the carry bit after each instruction. These operations were verified on SWTPC 6800 and Apple II microcomputers. The 6809, which is the upgraded version of the 6800, appears to implement the borrow bit in a similar manner to the 6800, that is with the borrow bit the same as the carry bit.

TSX and TXS

The transfer instructions between the stack pointer SP and index register X, TSX and TXS are common to both 6800 and 6502 but have important differences. In both MPUs, the current value of SP points to the location one address below the bottom of the stack. These registers are 16-bits long in the 6800, and 8-bits long in the 6502.

In the 6800, TSX increments the contents of SP by 1 and then transfers them to X. The reason for this appears to be to make X point to the actual bottom

6800				6502			
	ACCA	C	B=C		ACC	C	$B=\overline{C}$
LDAA #$05	05	D	D	LDA #$05	05	D	D
CLC	05	0	0	SEC	05	1	0
SBCA #$03	02	0	0	SBC #$03	02	1	0
LDAA #$05	05	D	D	LDA #$05	05	D	D
CLC	05	0	0	CLC	05	0	1
ADCA #$FD	02	1	1	ADC #$FD	02	1	0

Table 1.
5 minus 3 subtraction operations for 6800 and 6502. D indicates don't care state.

of the stack. This is presumably for cases when indexing operations on the stack are to be carried out. Conversely, TXS decrements the contents of X by 1 before transferring them to SP, so that if X pointed to the bottom of the stack, SP would then point to the next location below the bottom.

The 6502's TSX and TXS instructions do not perform the increment and decrement operations; they are just straightforward data transfer instructions. Why is this so? SP is an 8-bit register, and its contents ZZ are always interpreted as pointing to the page 1 of memory, $01ZZ. The effective address in indexed addressing is found by adding the 8-bit contents of X to the operand. In the zero-page indexed mode, the effective address is in page zero; in the absolute indexed mode, the page of the effective address depends on the 16-bit operand.

Because of these complications, the TSX and TXS instructions of the 6502 do not appear to be intended to facilitate indexed operations on the stack, as the transfers are effected without incrementing and decrementing. The primary function of TSX and TXS appears to be to effect data transfer to and from SP via X. This is in fact the only way to do so, since the 6502 lacks LDS (load to stack pointer) and STS (store from stack pointer) instructions. These instructions are part of the 6800's instruction set.

The 6809 MPU, which is the upgraded version of the 6800, has the instructions TFR S,X and TFR X,S which are equivalent to TSX and TXS, respectively. In the 6809, the stack pointer points to the bottom of the stack, and not to the location below the bottom as in the case of the 6800. Thus, the two transfer instructions do not perform an increment or decrement before the transfer, and are more like their 6502 equivalents, but for a quite different reason.

Conclusions

The 6800 and 6502 MPUs are so similar that it is worth looking at both together when studying one of them. I have taken this approach in my microprocessor teaching, using a MPU model consisting of the common features of the two MPUs.[2] This MPU model has one 8-bit accumulator, one 8-bit index register, one 16-bit program counter, one 8-bit stack pointer, and one 8-bit status register. It has the same six addressing modes of the 6800, except that indexed addressing is restricted to the zero page only. The instruction set is made up of the 43 instructions common to both 6800 and 6502. These instructions include SBC, TSX, and TXS, and it is important that those using both the 6800 and 6502 understand the small but important differences in these instructions.

Program FLIP

```
comment $

        PROGRAM FLIP

        This program flips from the color-adapter to the monochrome-adapter
        and back again.  The program has no stack segment so that it can be
        converted into a COM file.  This generates an error upon linking which
        should be ignored.

        Simson L. Garfinkel, 1983

        $

data    segment at 40h
        org 16
equip   db ?
data    ends

cseg    segment para 'code'
        org  100h                  ;I want to use this as a COM file
start   proc far
        assume cs:cseg,ss:cseg,ds:data

        mov  ax,cs
        mov  ss,ax
        mov  sp,stacke             ;get end of stack

        push ds
        xor  ax,ax
        push ax                    ;save how to get home
        mov  ax,data               ;use ds to reference segement at 0
        mov  ds,ax
        mov  ah,equip
        and  ah,30h
        cmp  ah,30h                ;check to see if we are running mono or color
        jne  mono
        jmp  color                 ;we are in mono - switch to color
mono:   or   equip,30h             ;switch to monochrome interface
        mov  ax,2
        int  10h                   ;generate interrupt to reset board
        mov  ah,1
        mov  cx,12*256+13          ;set cursor for lines 12 + 13
        int  10h                   ;
        ret                        ;go back to ms-dos
color:  and  equip,0cfh
        or   equip,010h            ;switch to color interface
        mov  ax,4                  ;clear graphics memory
        int  10h
        mov  al,0
        int  10h                   ;select 40x25 BW text, no color burst
        mov  ah,1
        mov  cx,6*256+7            ;set cursor for lines 6 & 7
        int  10h
        ret
start   endp
        db 16 dup (?)              ;stack area
stacke  label near                 ;End of stack
cseg    ends
        end start
```

References

[1] Levanthal, Lance R. *6502 Assembly Language Programming*, Osborne/McGraw-Hill, 1979, p. 3-107.

[2] Tan, B.T.G. "Common instructions of the 6800 and 6502," *Dr. Dobb's Journal of Computer Calisthenics & Orthodontia*, Volume 4, Number 9, October 1979, pp. 38-39.

[3] Merrin, Stephen. "Addition and Subtraction: The 1802 Versus the Z80," *Byte*, Volume 6, Number 3, March 1981, pp. 224-228.

DDJ