

İ.T.Ü.
Faculty of Computer and Informatics
Computer Engineering



MICROCOMPUTER LAB REPORT

Lab No : 3
Lab Date : 24.10.2014 Friday
Group : 14
Group Members : 150110040 - Çağatay Erdiz
150110131 - Mert Cingöz
150110801 - Umut Pilge
150110030 - Duran Eren
150110106 - Ahmet Gözüberk

Research Assistant : Hasan Ünlü

A) Introduction

Stack usage of MC6802:

- PC: Program Counter
- SP: Stack Pointer
- XR: Index Register
- JSR: Jump to subroutine (push PC; jump absolute)
- RTS: Return from subroutine (pull PC)
- STS: Store SP
- STX: Store XR
- TSX: $SP + 1 \leftarrow XR$
- PSHA: Push data of A to stack, then $SP - 1 \rightarrow SP$
- PSHB: Push data of B to stack, then $SP - 1 \rightarrow SP$
- PULA: $SP + 1 \rightarrow SP$, then pull data from SP to A
- PULB: $SP + 1 \rightarrow SP$, then pull data from SP to B
- INS: Increment SP
- LDS: Load SP

B) Subroutines and Argument Passing

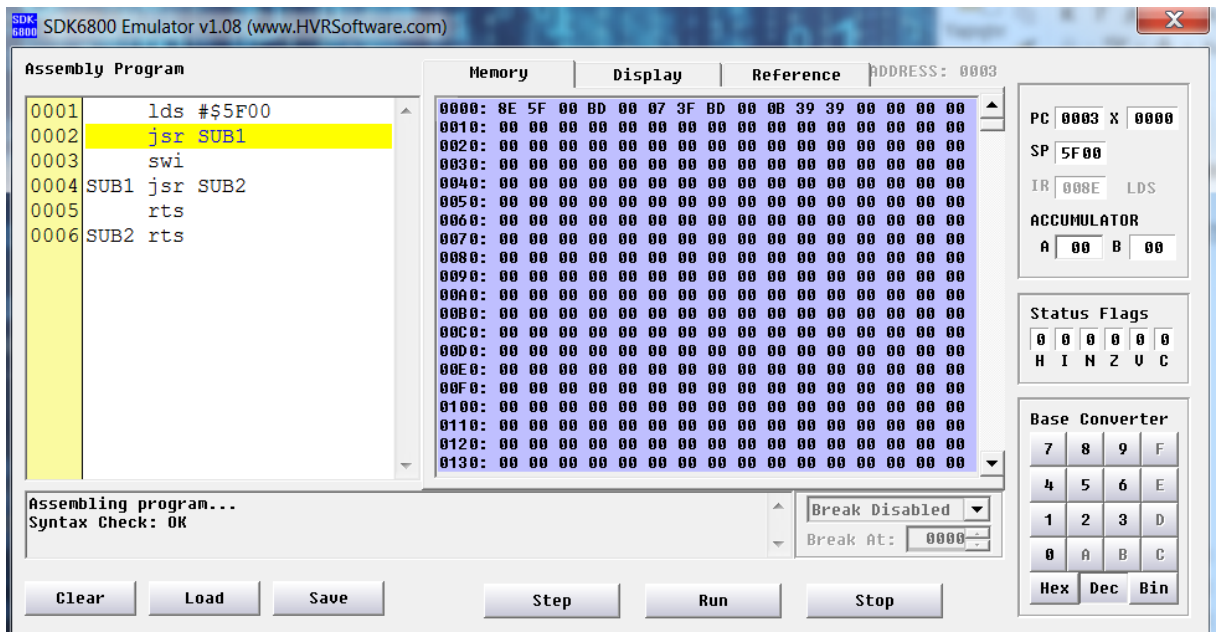
1) Calling Subroutines Lab Code:

```
lds #5F00
jsr SUB1
swi

SUB1 jsr SUB2
    rts

SUB2 rts
```

PC	Stack_Pointer	Memory[Stack_Pointer]
\$0000	\$5EF9	\$00
\$0003	\$5F00	\$00 (after this line: \$06)
\$0007	\$5EFE	\$00 (after this line: \$0A)
\$000B	\$5EFC	\$00
\$000A	\$5EFE	\$0A
\$0006	\$5F00	\$06



2) Argument Passing Lab Code:

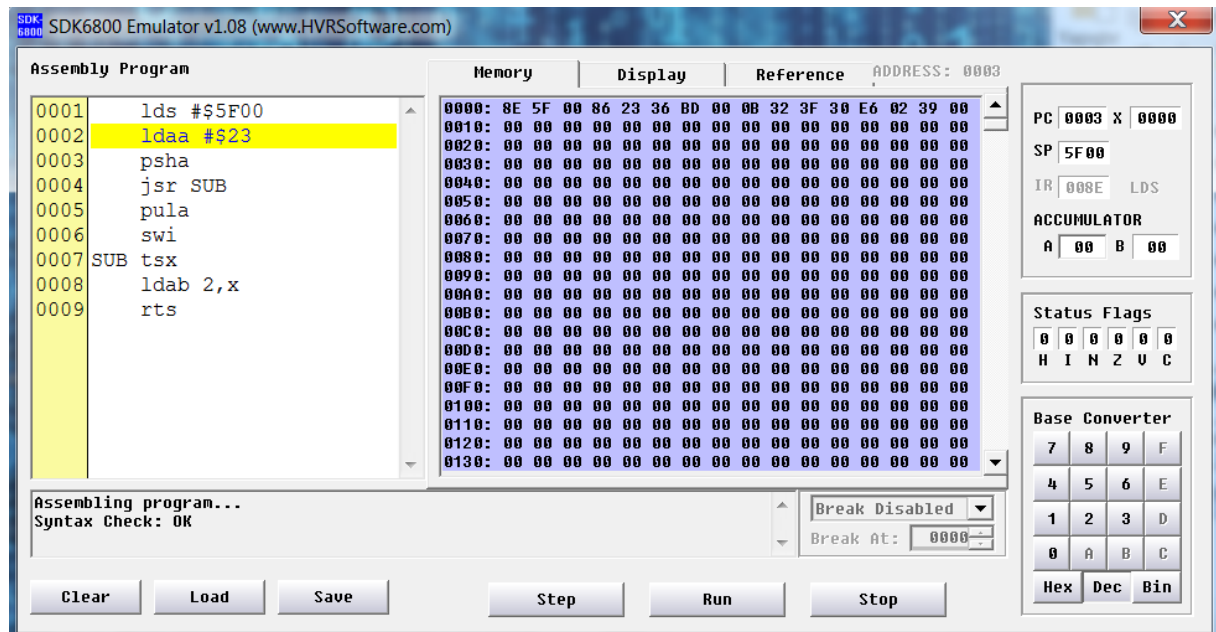
```

lds #$5F00
ldaa #$23
psha
jsr SUB
pula
swi

SUB  tsx
     ldab 2,x
     rts

```

PC	Stack_Pointer	Memory[Stack_Pointer]
\$0000	\$5EF9	\$00
\$0003	\$5F00	\$00
\$0005	\$5F00	\$00 (after this line: \$23)
\$0006	\$5EFF	\$00 (after this line: \$09)
\$000B	\$5EFD	\$00
\$000C	\$5EFD	\$00
\$000E	\$5EFD	\$00
\$0009	\$5EFF	\$09
\$000A	\$5F00	\$23



C) Experiment

1) Recursive Fibonacci Assembly Code:

```

.org $0
lds #$5f00    ;set stack pointer
ldaa #$07     ;calculate fifth element
psha
jsr  fbnc
pula
staa $0100    ;store result at $0100
bra  finish

fbnc tsx
ldaa 2,x      ;argument is taken
cmpa #$02     ;if n is less than 2, return
blt  term
ldaa 2,x      ;take argument
deca         ;n-1
staa 2,x      ;store (n-1)
psha         ;send (n-1) to fib
jsr  fbnc     ;run fib(n-1)
pulb        ;pull return value
tsx
ldaa 2,x      ;now Acca has n-1
deca         ;n-2
stab 2,x      ;store (fib(n-1))
psha         ;send n-2 to fib
jsr  fbnc     ;run fib(n-2)
pulb        ;take return value
tsx

```

```
        ldaa 2,x    ;load acca with fib(n-1) was stored before
        aba        ;result = fib(n-1)+fib(n-2)
        staa 2,x    ;store this stack location to return value
term rts        ;return

finish .end
```