

CS 351

Design of Large Programs

Executor Framework

October 21, 2021

Assembly Language

MONITOR FOR 6802 1.4 9-14-80 TSC ASSEMBLER PAGE 2

```
C000                    ORG     ROM+$0000 BEGIN MONITOR
C000 8E 00 70    START    LDS     #STACK

*****
* FUNCTION: INITA - Initialize ACIA
* INPUT: none
* OUTPUT: none
* CALLS: none
* DESTROYS: acc A

0013            RESETA   EQU   %00010011
0011            CTLREG   EQU   %00010001

C003 86 13    INITA    LDA A   #RESETA   RESET ACIA
C005 B7 80 04            STA A   ACIA
C008 86 11            LDA A   #CTLREG   SET 8 BITS AND 2 STOP
C00A B7 80 04            STA A   ACIA

C00D 7E C0 F1            JMP     SIGNON    GO TO START OF MONITOR

*****
* FUNCTION: INCH - Input character
* INPUT: none
* OUTPUT: char in acc A
* DESTROYS: acc A
* CALLS: none
* DESCRIPTION: Gets 1 character from terminal

C010 B6 80 04    INCH    LDA A   ACIA        GET STATUS
C013 47            ASR A            SHIFT RDRF FLAG INTO CARRY
C014 24 FA            BCC    INCH        RECIEVE NOT READY
C016 B6 80 05            LDA A   ACIA+1    GET CHAR
C019 84 7F            AND A   #$7F        MASK PARITY
C01B 7E C0 79            JMP    OUTCH     ECHO & RTS

*****
* FUNCTION: INHEX - INPUT HEX DIGIT
* INPUT: none
* OUTPUT: Digit in acc A
* CALLS: INCH
* DESTROYS: acc A
* Returns to monitor if not HEX input

C01E 8D F0    INHEX    BSR    INCH        GET A CHAR
C020 81 30            CMP A   #'0        ZERO
C022 2B 11            BMI    HEXERR    NOT HEX
C024 81 39            CMP A   #'9        NINE
C026 2F 0A            BLE    HEXRTS    GOOD HEX
C028 81 41            CMP A   #'A       
C02A 2B 09            BMI    HEXERR    NOT HEX
C02C 81 50            CMP A   #'a       
C02E 2B 08            BMI    HEXERR    NOT HEX
```

Assembly Language Pitfalls

- Hard to write
- Harder to read
- Even harder to understand
- Error prone

A Better Way

High level languages such as Java, C#, Haskell, Rust, etc.

- Offer automatic memory management
- No direct pointer manipulation
- Easier to write
- Easier to read
- Easier to understand
- Relieves many mental burdens from the programmer

Concurrency Assembly Language

- Thread
- wait
- notify
- notifyAll

Concurrency Assembly Language Pitfalls

- Hard to write
- Harder to read
- Even harder to understand
- Error prone

A Better Way

Executors, tasks, streams and `java.util.concurrent`

- Offer automatic thread management
- No direct thread manipulation
- Easier to write
- Easier to read
- Easier to understand
- Relieves many mental burdens from the programmer