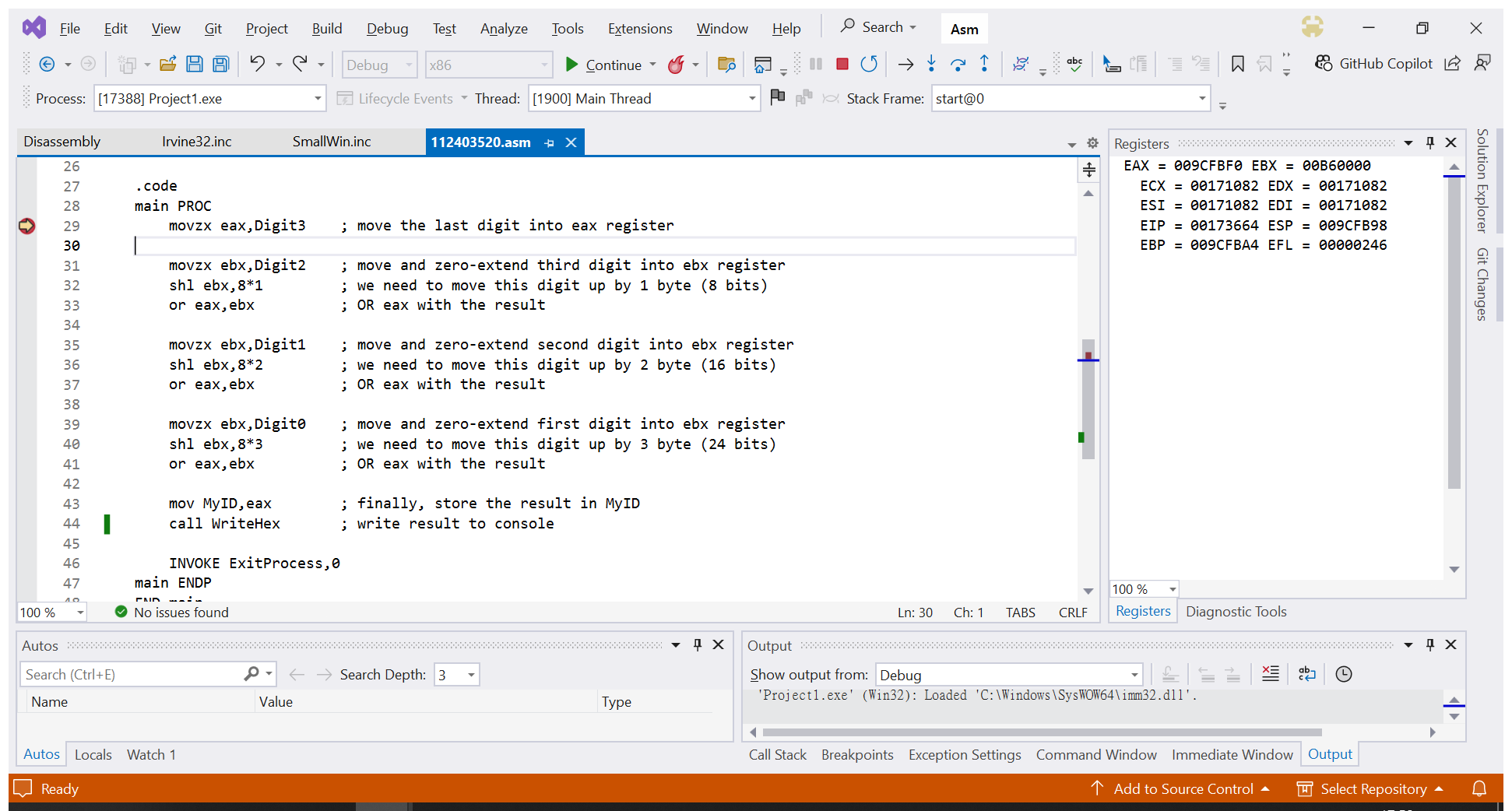
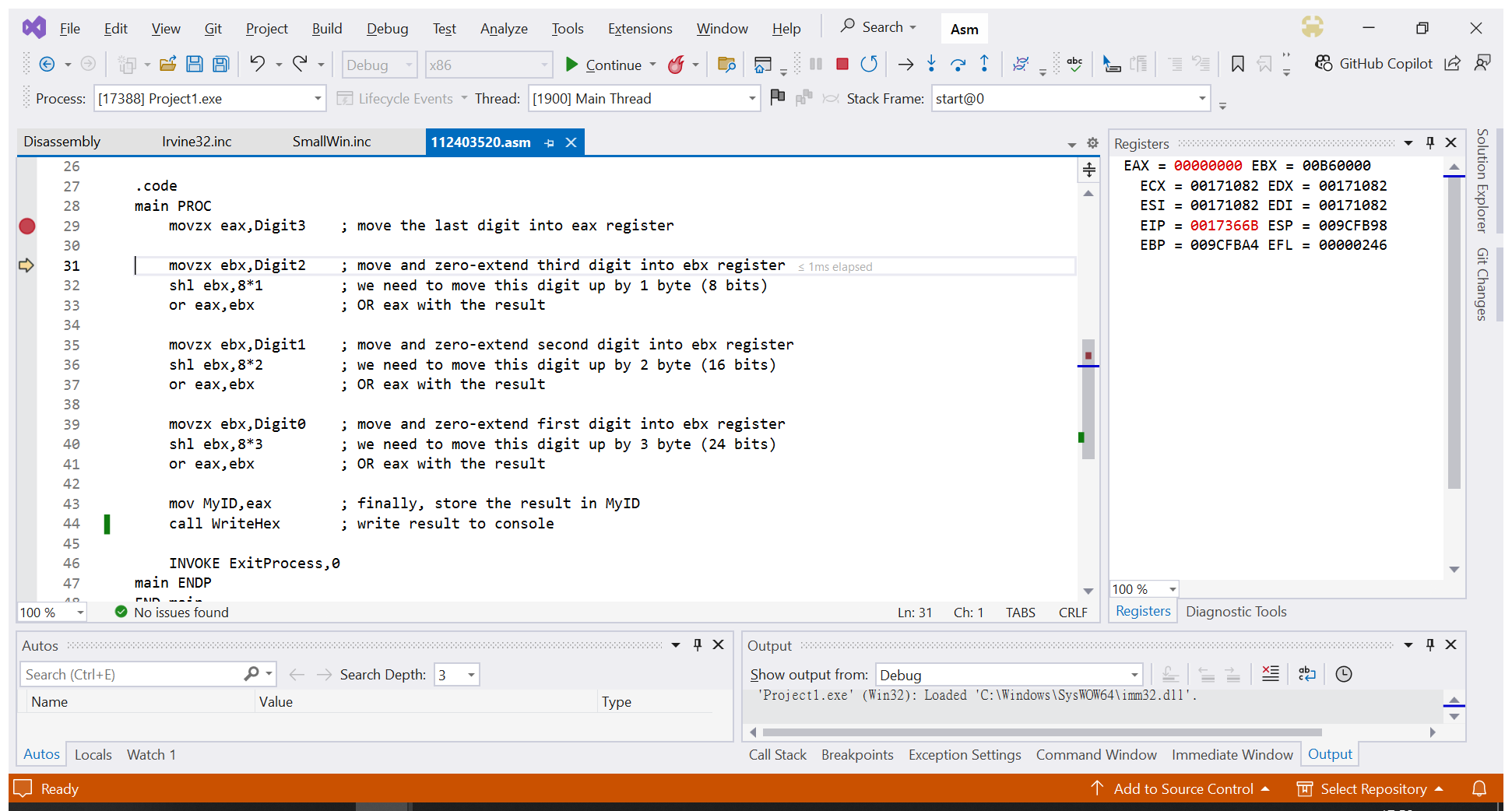
組合語言 作業一

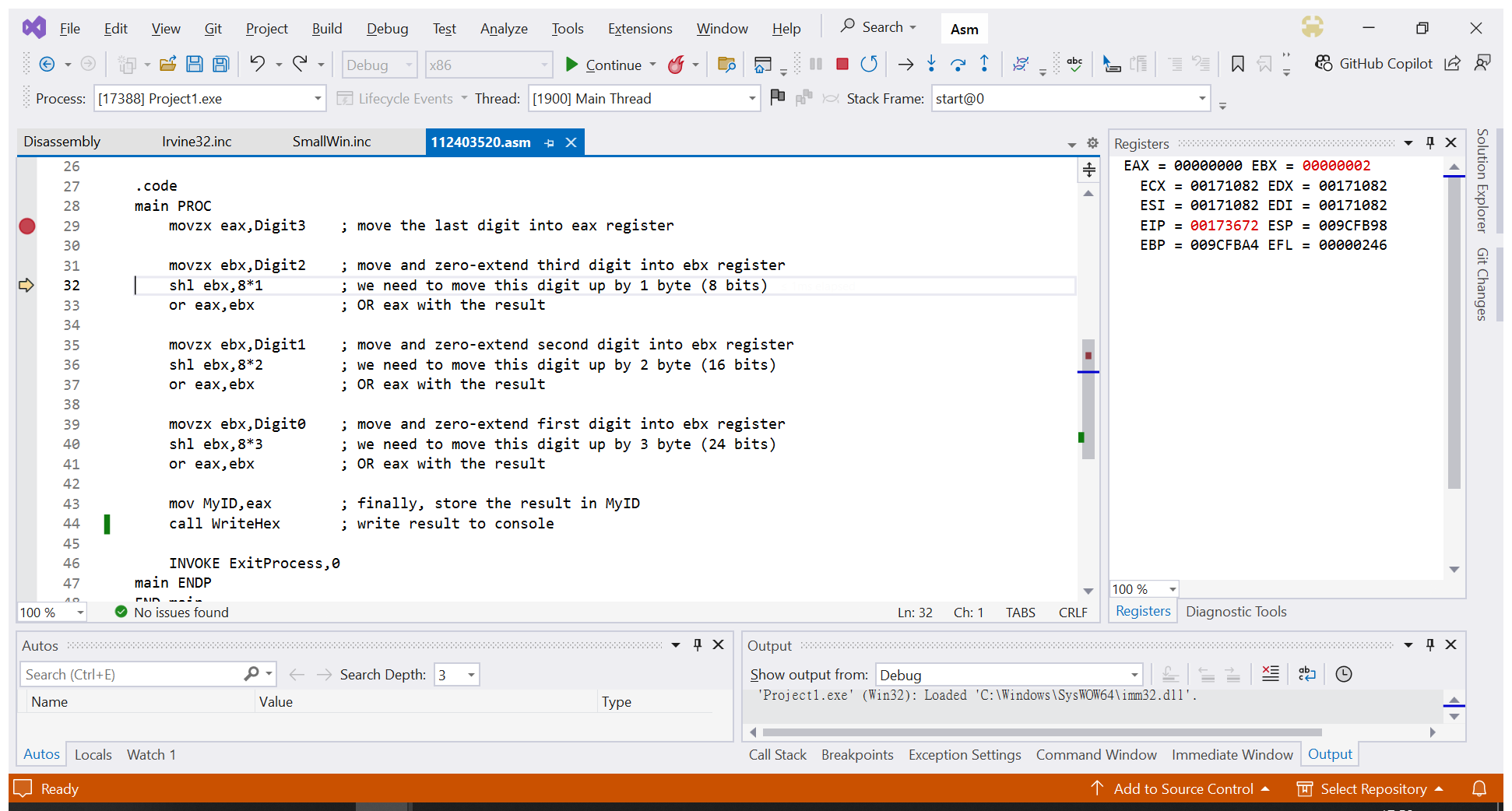
112403520李宥寬

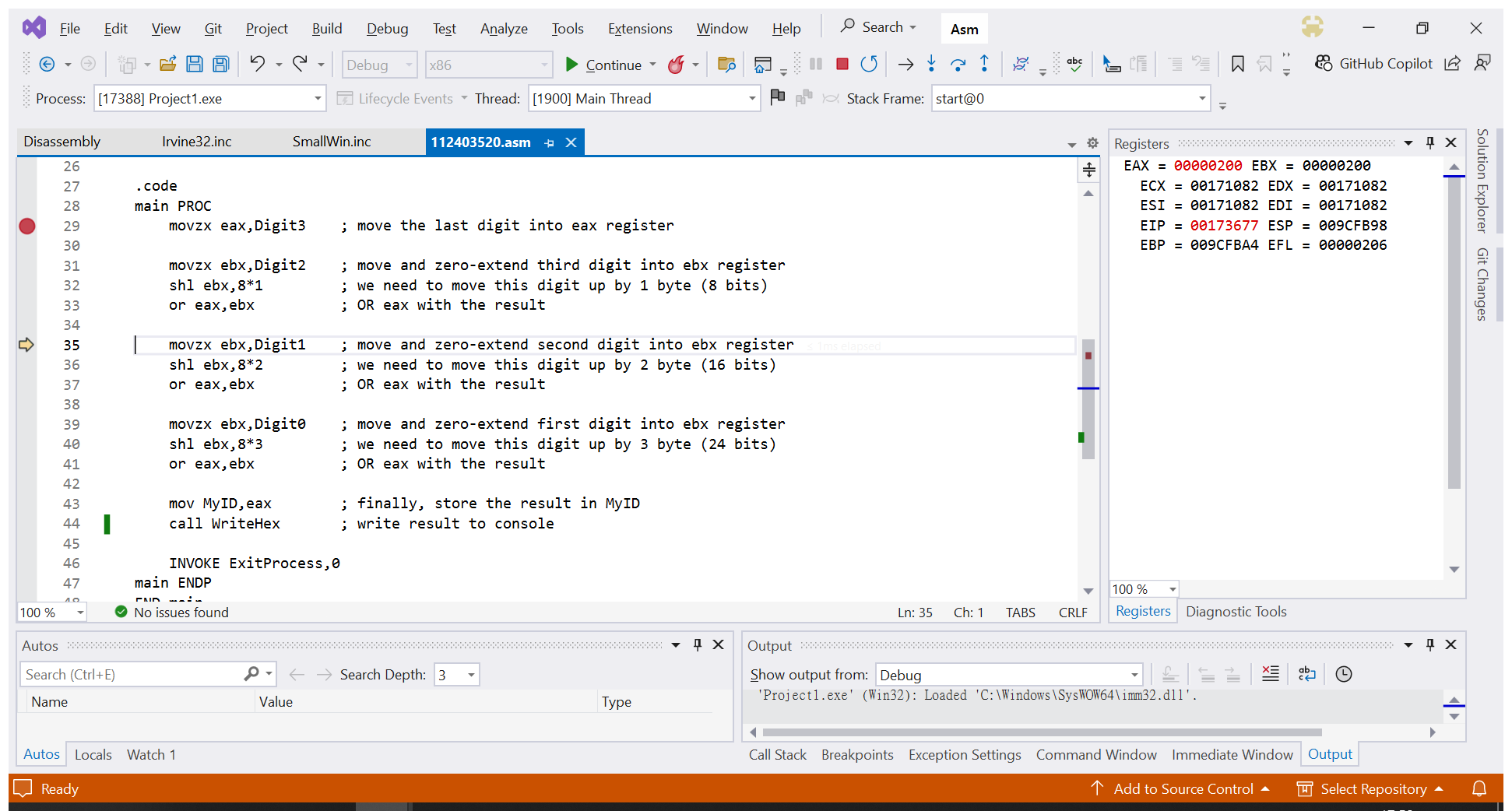
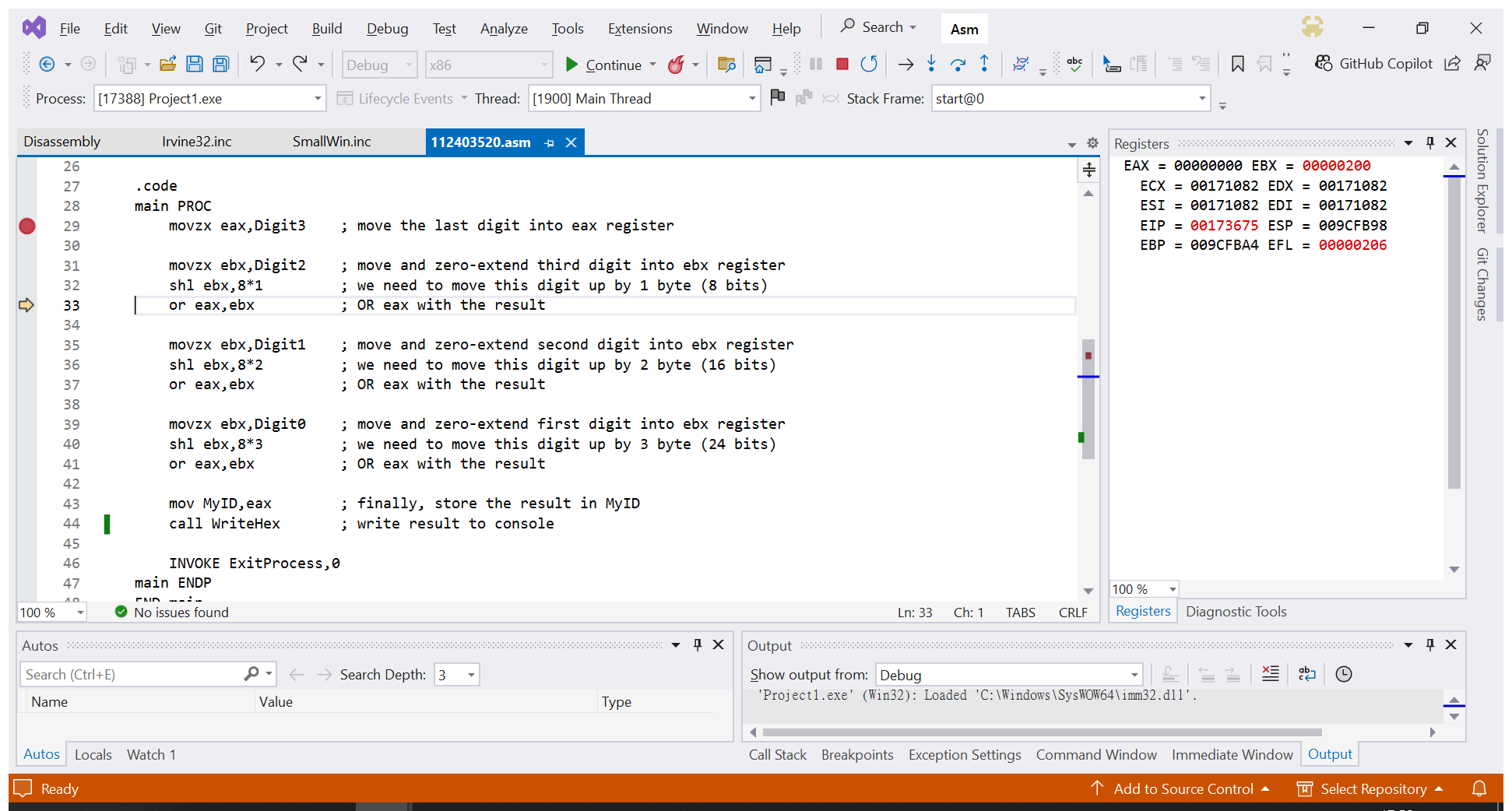
1. **程式流程截圖、程式碼說明**



程式初始化

將第四個數字移到eax

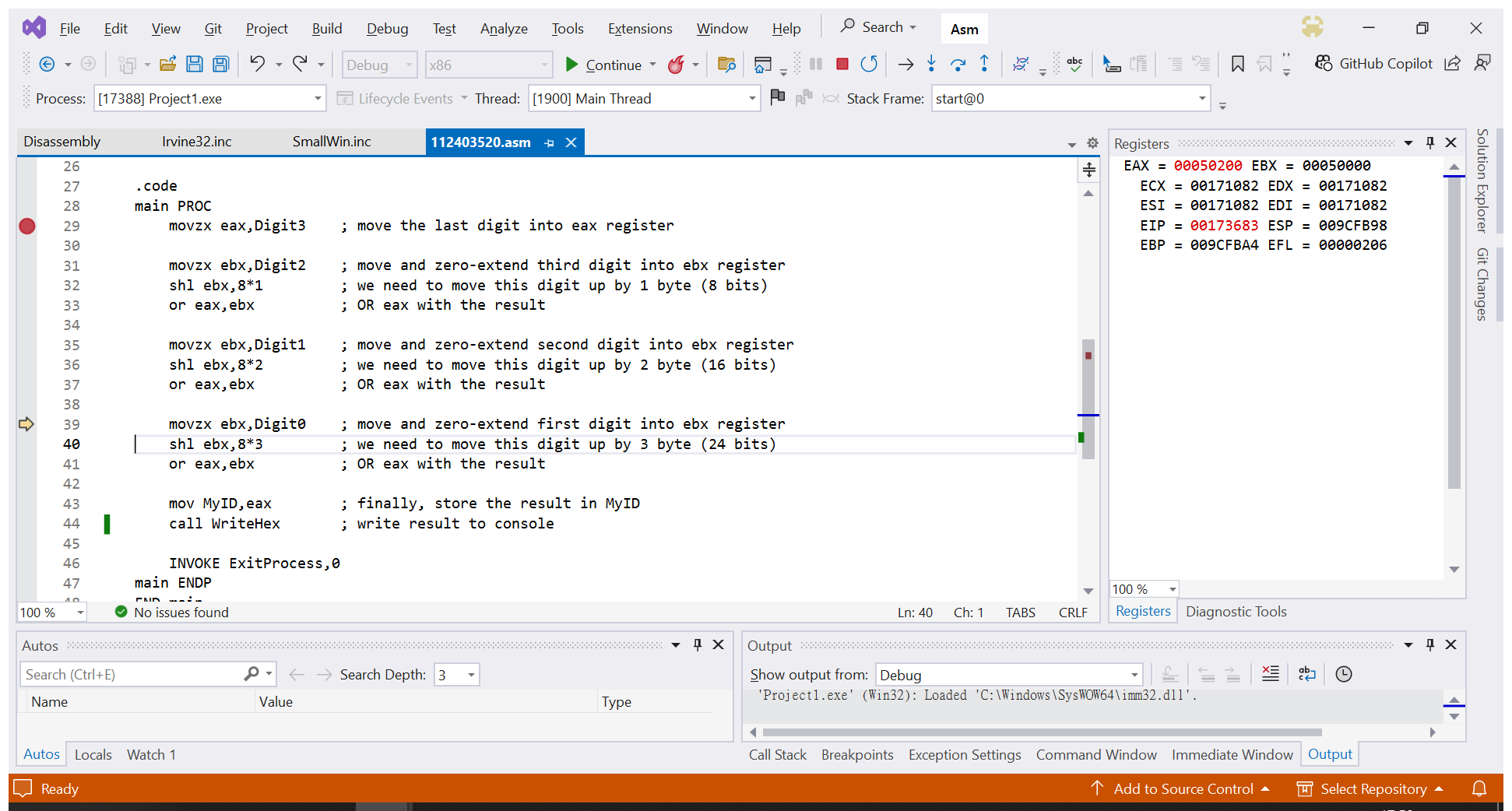
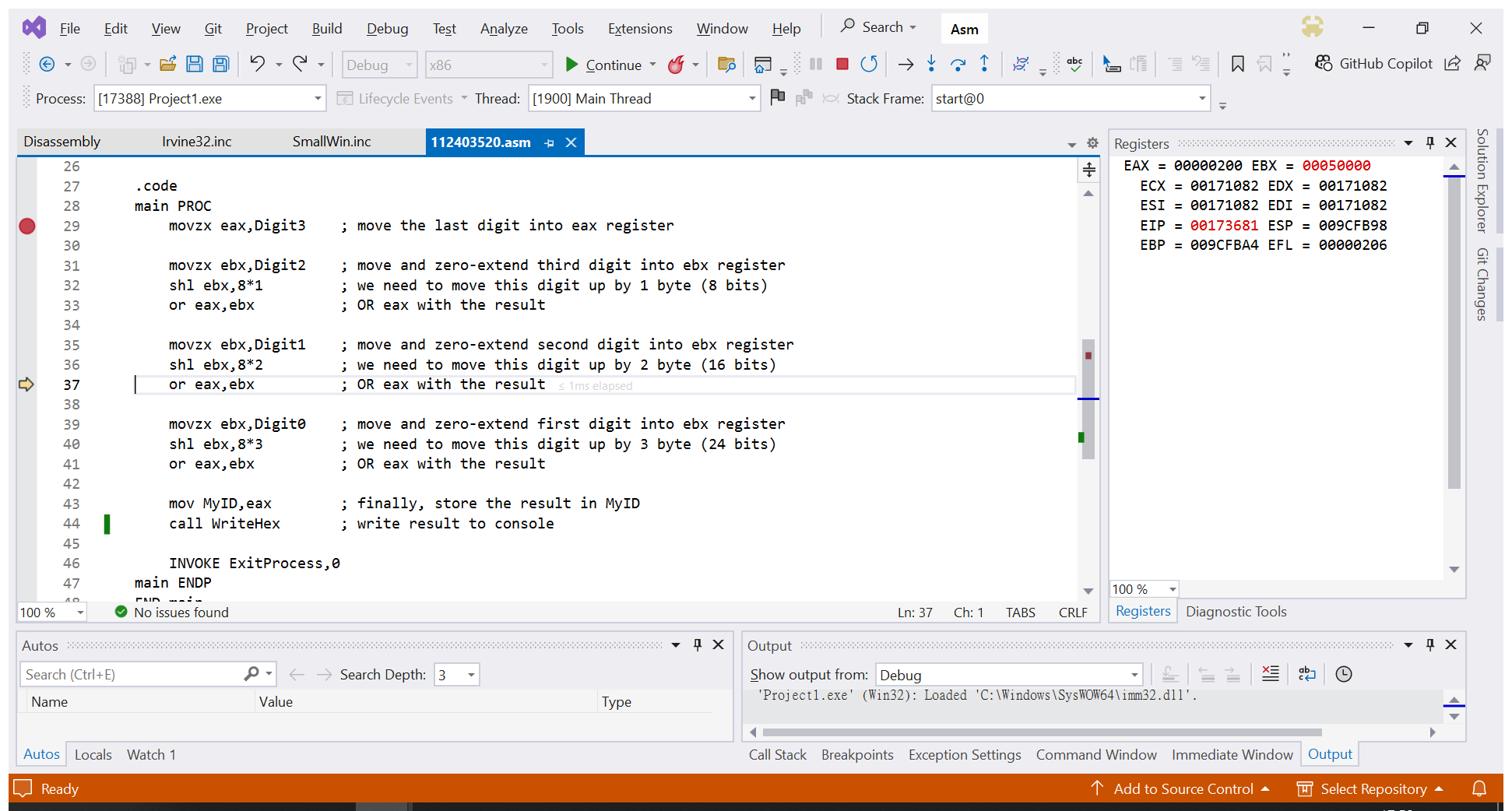
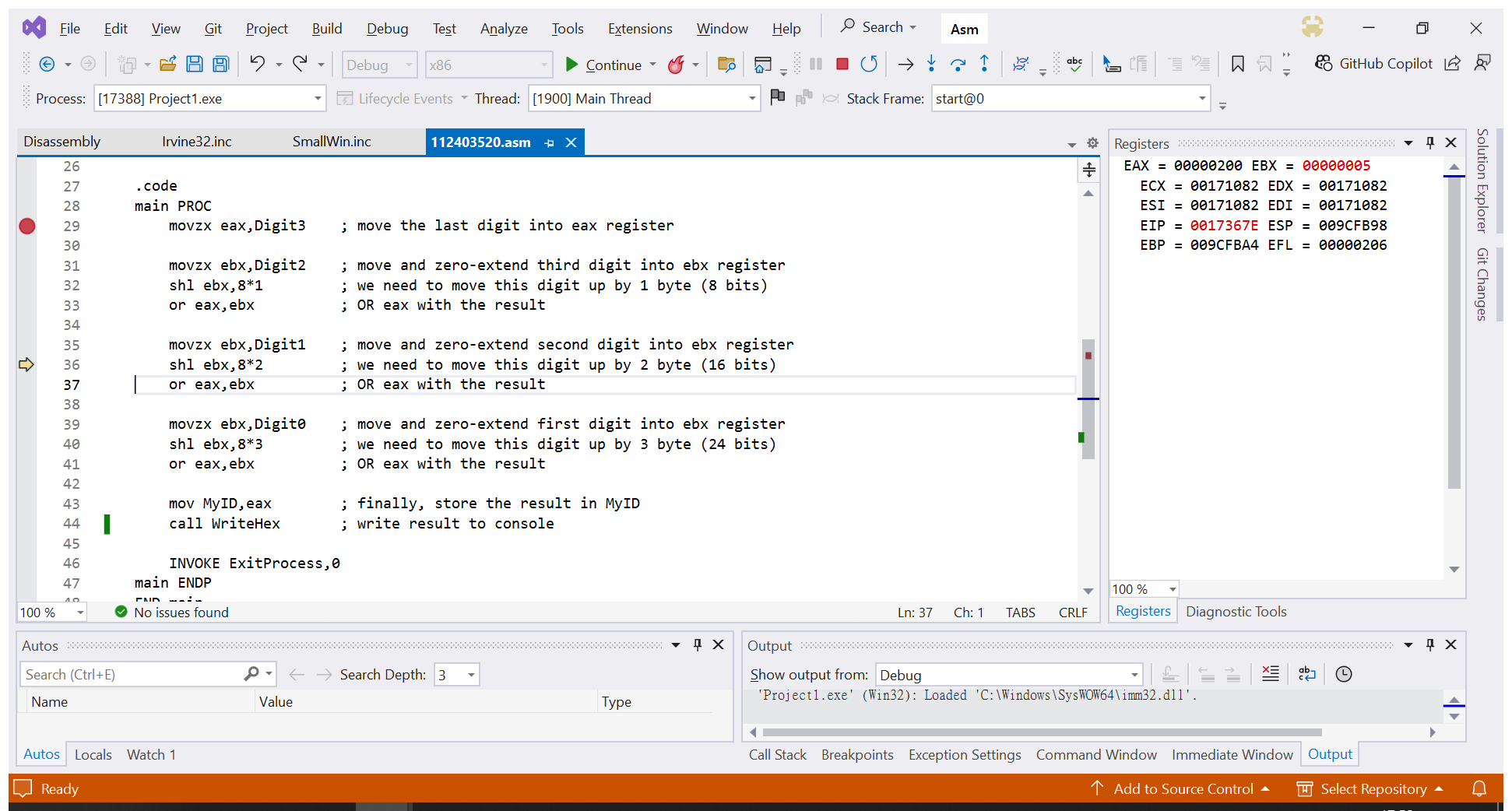




說明:

將第三個數字移到EBX，左移8 bit，再OR EAX 上原本的第四位數字。

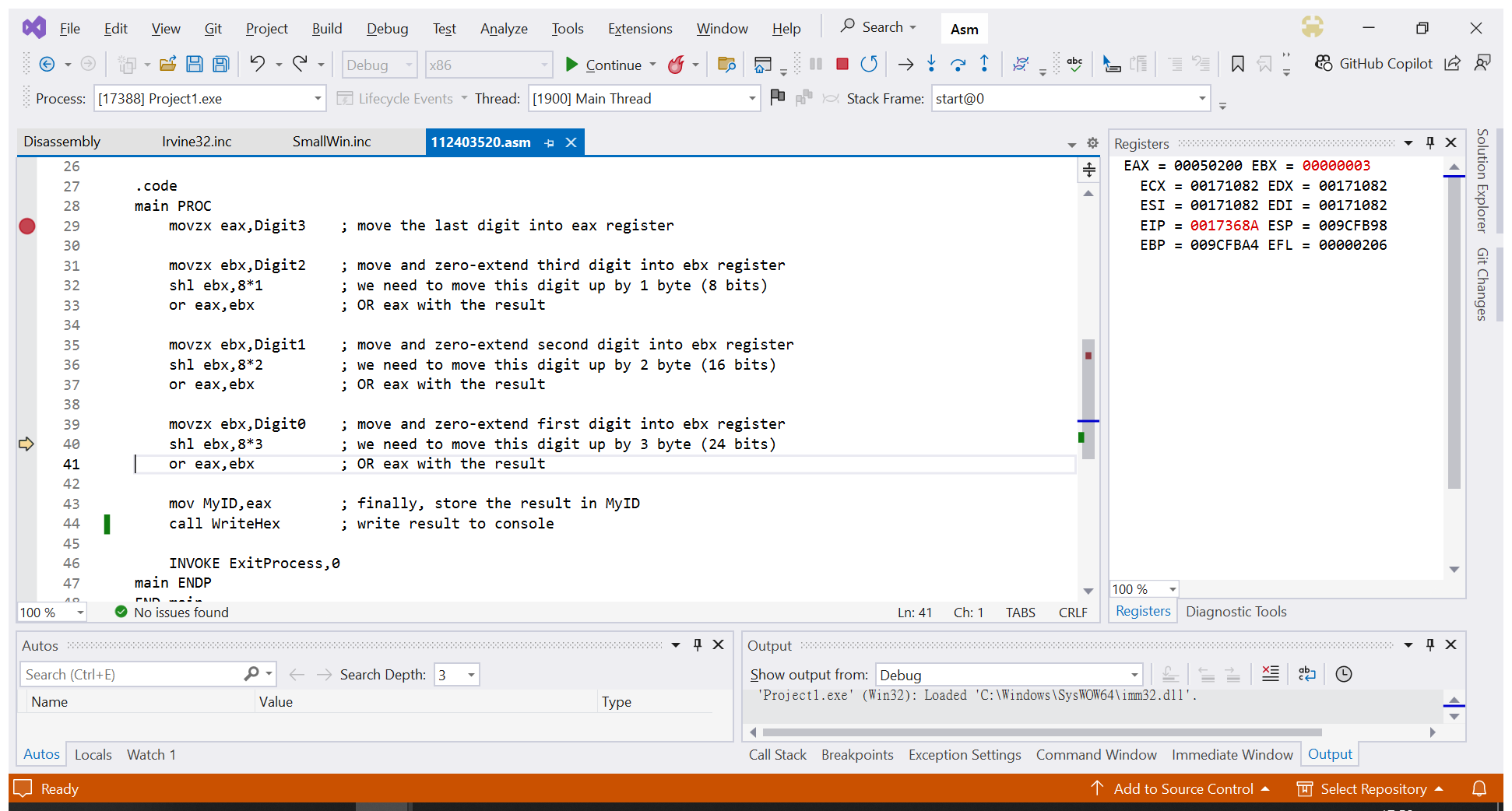
所以EAX由低到高的3、4位數就會是第三個數字，而1、2位還是第四個數字



說明:

將第二個數字移到EBX，左移16 bit，再OR EAX。

所以EAX由低到高的5、6位數就會是第二個數字，而1~4位保留原本的值

A screenshot of a computer

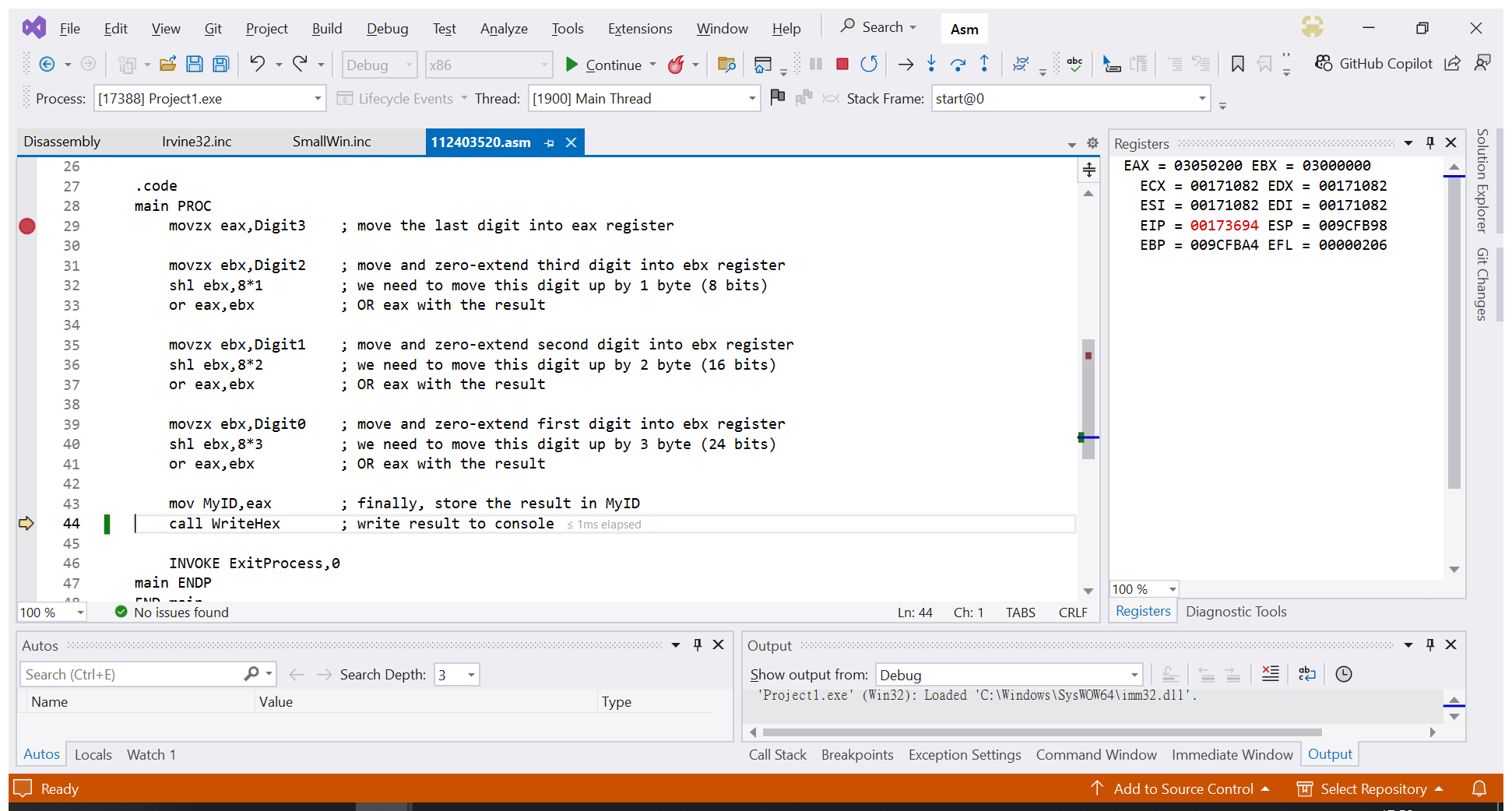
Description automatically generated A computer screen shot of a computer

Description automatically generated

說明:

將第一個數字移到EBX，左移24 bit，再OR EAX。

所以EAX由低到高的7、8位數就會是第一個數字，而1~6位保留原本的值



A screenshot of a computer

Description automatically generated

將儲存在EAX的結果複製到MyID，並輸出到Console

1. **完成的程式畫面截圖**

可以看到左下角Watch的MyID和輸出到Console的值皆為正確的(3520->03050200)

**A screenshot of a computer

Description automatically generated**

完整程式碼

A screenshot of a computer program

Description automatically generated

1. **作業心得**

這次的組合語言作業讓我深入學習了 bitwise OR 的運用。透過搭配 AND、SHL 和 SHR 指令，我能夠有效實現各種 bit masking 操作，精確提取所需的位元值，從而提高資料儲存和運用的效率。

此外，這次作業中我還使用了 MOVZX 指令，這個指令讓我能夠將較小資料型別的數據轉移到較大的暫存器中，同時自動在高位元補零，這樣的特性不僅提高了程式的可讀性，也大大簡化了數據處理的過程。這些學習經驗不僅增強了我對組合語言的理解，也提升了我在資料處理方面的能力。