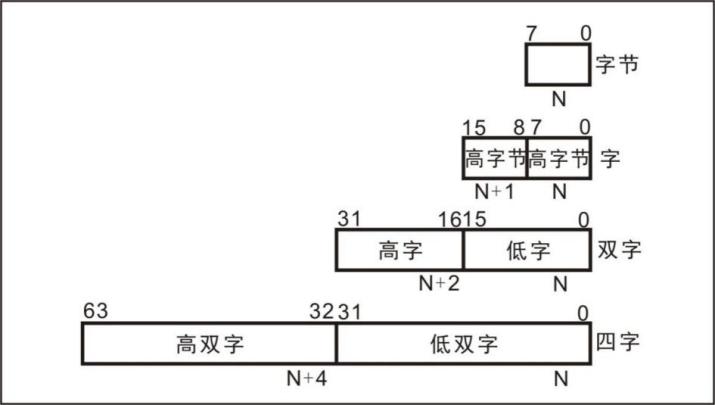
1. 第6页：第一章进制-1.2节进制的运算

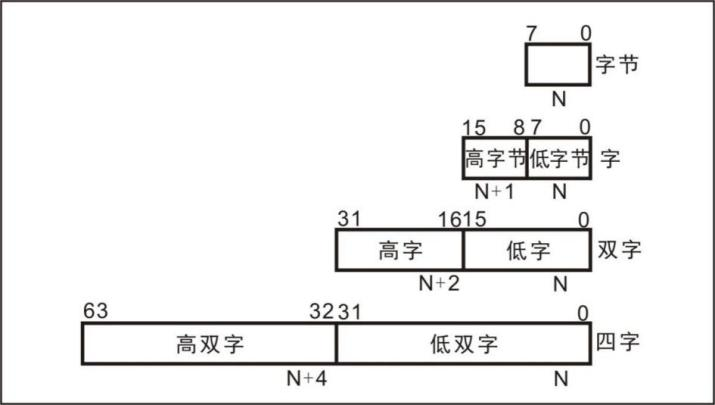
4+5 ：4后面查5个数，所以 4+5 = 10；

更正：4+5 ：4后面查5个数，所以 4+5 = **11**；

1. 第16页第一章1.4节图1-4-5计算机数据宽度



改为：



1. 第33页第一章2.3节

## 2.3.1【32位通用寄存器】

32位处理器中：EAX、ECX、EDX、EBX、ESP、EBP、ESI、EDI这8个寄存器通常存放一般性的数据，被称为通用寄存器。它们都有各自的用途。【本节不做详细介绍，用到时会详细介绍】

EAX、EXC、EDX、EBX为数据寄存器；

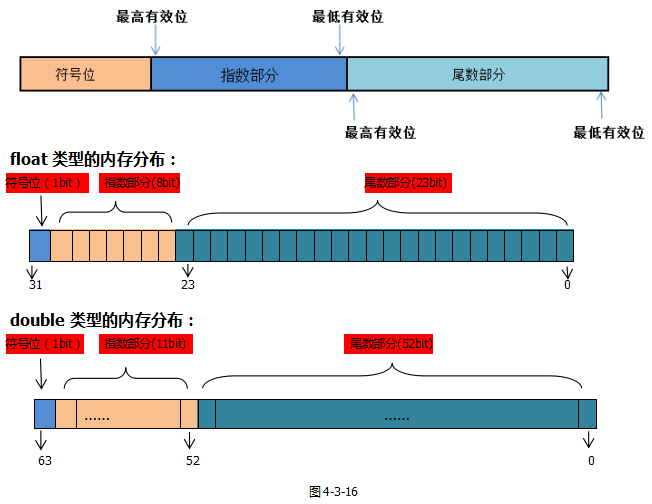
改正为：EAX、ECX、EDX、EBX为数据寄存器；

4、第170页：第二章汇编基础-2.17节JCC指令

最高位相同的两个数作运算，是不会改变OF位；

更正：1、最高位**不**相同的两个数作运算，是不会改变OF位；

5、第249页：第四章变量和数据类型4.3节数据类型



6、第507页：第七章7.3节堆栈图解析生命周期

1. 第162页：第二章2.17 JCC指令

SBB指令：带借位减法,如果CF位为1，将相减的结果再减1。

格式：ADC r/m,r/m/imm 两边宽度一致，且不能同时为内存。ADC改为SBB

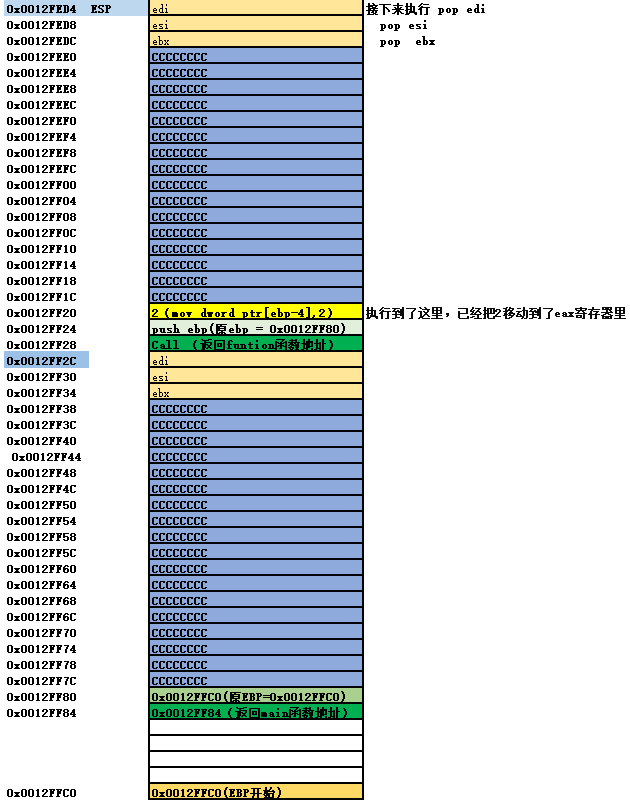


图7-3-18（演示示例代码CH07\_3\_3(-)）

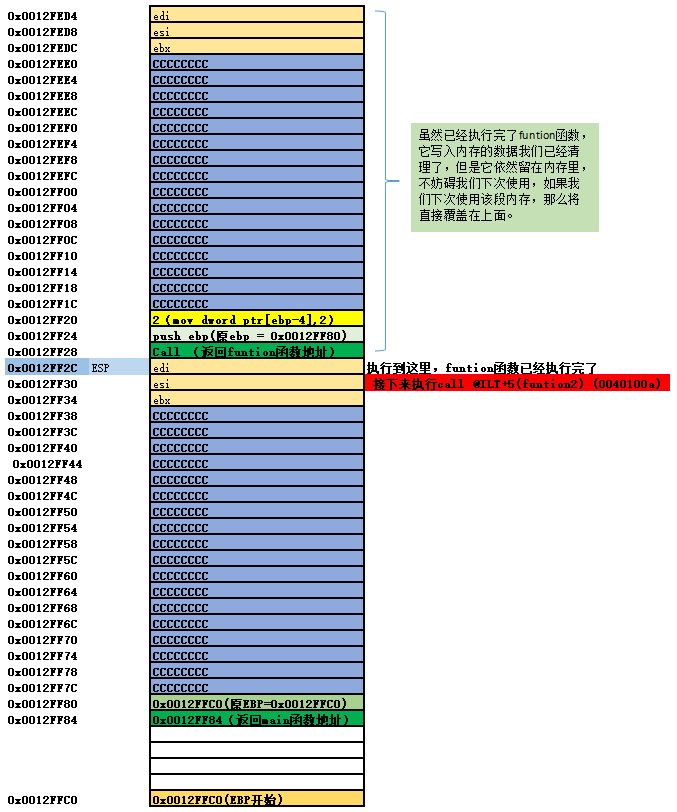


图7-3-19（演示示例代码CH07\_3\_3(二)）

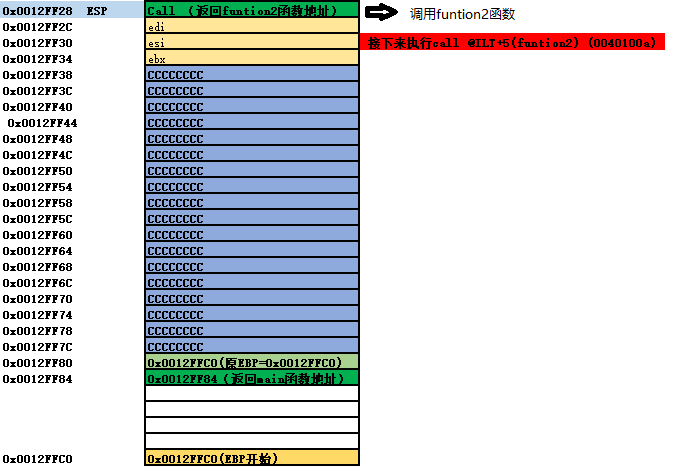


图7-3-20（演示示例代码CH07\_3\_3(三)）

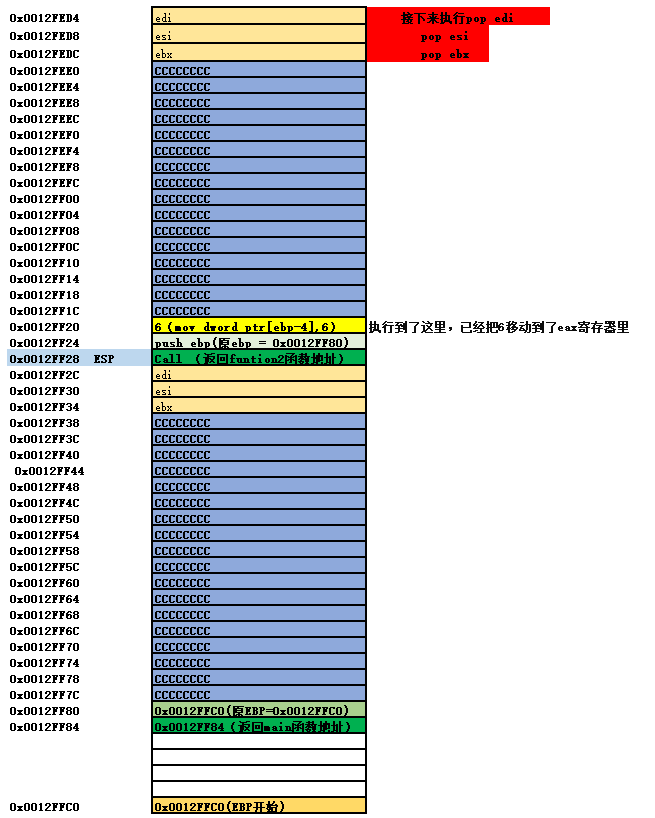
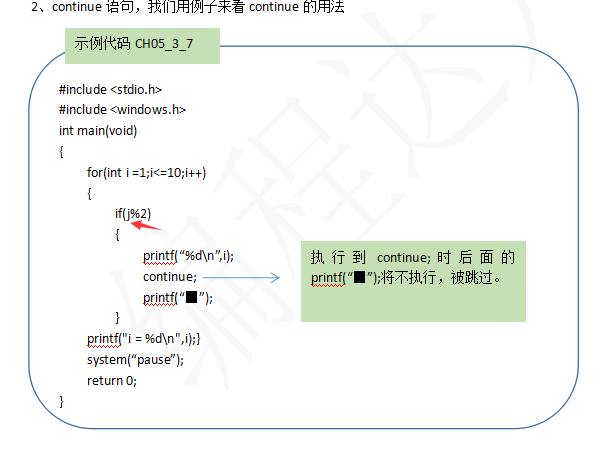


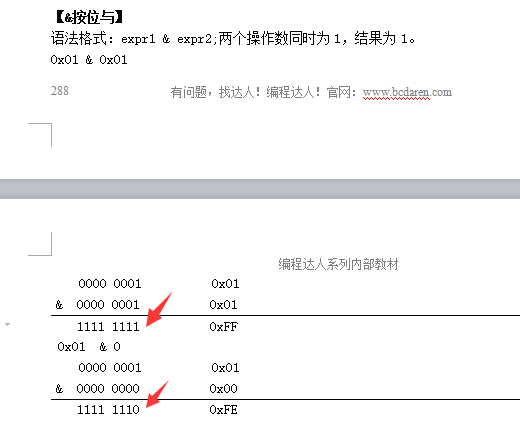
图7-3-21（演示示例代码CH07\_3\_3(四)）

1. 第354页：第五章5.3 for语句 5.3.3【循环语句的几种退出方式】

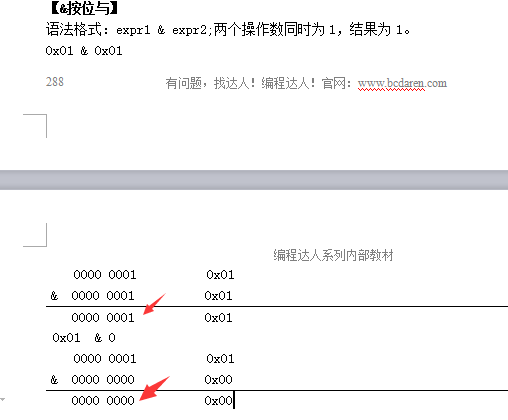


if(j%2)应改为ifi%2)

1. 第289页：第四章 变量、数据类型 4.6.5【位运算符】



改为：

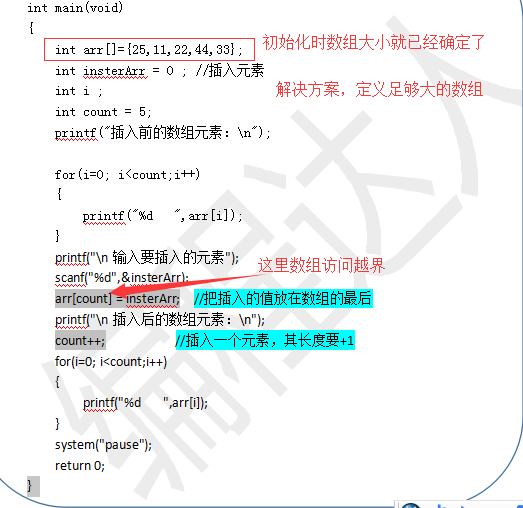


1. 第404页：第六章 数组6.2.4【数组元素的删除和插入】
2. 第88页:第二章汇编基础2.9内存复制

当DF位为0时，STOS指令会把AL/AX/EAX存器存储的数据，存储到[EDI]指定的内存地址相对应的数据中去。相应的EDI存储的数据会增加相应的宽度1、2、4，直至AL/AX/EAX存器存储的数据变为0；应为：AL/AX/EAX存器存储的数据不变

当DF位为1时，STOS指令会把AL/AX/EAX寄存器存储的数据，存储到[EDI]指定的内存地址相对应的数据中去。相应的EDI存储的数据会减少相应的宽度1、2、4，直至AL/AX/EAX存器存储的数据变为0。应为：AL/AX/EAX存器存储的数据不变

我们通常用STOS来进行初始化，我们将需要初始化的值放在AI/AX/EAX寄存器中，将需要初始化的内存地址储存到EDI中。



更正为：

#include <stdio.h>  
#include <windows.h>  
int main(void)  
{  
int arr[MAX\_PATH] = {25,11,22,44,33};  
int instetArr = 0 ;  
int count = 5;  
int i = 0;  
printf("插入前的数组元素:\n");  
for (i ;i<count;i++)  
{  
printf("%d   ",arr[i]);  
  
}  
printf("\n输入要插入的元素:\n");  
scanf("%d",&instetArr);  
arr[count++] = instetArr;  
for (i=0;i<count;i++)  
{  
printf("%d   ",arr[i]);  
}  
  
system("pause");  
return 0;  
}

结果：

