**Завдання 1:**

Проробити тестові приклади та варіанти індивідуальних завдань. Скласти звіт.

**Завдання 2:**

 Проробити тестові приклади та варіанти індивідуальних завдань. Скласти звіт.

|  |  |  |  |
| --- | --- | --- | --- |
| № | Регістри для пересилки8/16/32 розрядних даних | Дані 8/16/32 | Знакові (-)Беззнакові (+) |
|  4 | DL/DX/AX | 14h /8888h /33338888h | +/-/- |

*Код програми:*

*Випадок А:*

#include "stdafx.h"

#include "iostream.h"

int main(int argc, char\* argv[])

{

unsigned char b=0x14;

unsigned char a=0;

cout << "Before MOVE Assembler: a= "<< a <<" b= "<< b

<<endl << endl;

\_asm {

mov dL,b

mov a,dL

}

cout << "After MOVE Assembler:a= "<< a <<" b= "<< b

<<endl << endl;

printf(" AFTER MOVE a=%i b=%i \n",a,b);

return 0;

}

*Результат програми:*

Before MOVE Assembler: a= b= ¶

After MOVE Assembler:a= ¶ b= ¶

 AFTER MOVE a=20 b=20

Press any key to continue

*Випадок В:*

#include "stdafx.h"

#include "iostream.h"

int main(int argc, char\* argv[])

{

short int b=0x8888;

short int a=0;

cout << "Before MOVE Assembler: a= "<< a <<" b= "<< b

<<endl << endl;

\_asm {

mov ax,b

mov a,ax

}

cout << "After MOVE Assembler:a= "<< a <<" b= "<< b

<<endl << endl;

printf(" AFTER MOVE a=%i b=%i \n",a,b);

return

}

*Результат програми:*

Before MOVE Assembler: a=0 b= -30584

After MOVE Assembler:a= -30584 b= -30584

After MOVE a= -30584 b= -30584

Press any key to continue

*Випадок С:*

#include "stdafx.h"

#include "iostream.h"

int main(int argc, char\* argv[])

{

int b=0x33338888;

int a=0;

cout << " Before MOVE Assembler: a= " << a <<" b= "<< b<< endl << endl;

\_asm {

mov ax, WORD PTR b ; ax<=== молодша частина a

mov bx, WORD PTR b+2 ; bx<=== старша частина a

mov WORD PTR a, ax ; молодша частина x<=== <ax>

mov WORD PTR a+2, bx ; старша частина x<=== <bx>

}

cout << " After MOVE Assembler: a= " << a <<" b= "<< b<< endl << endl;

//printf(" AFTER MOVE a= %i b=%i \n",a,b);

return 0;

}

*Результат програми:*

Before MOVE Assembler: a=0 b= 859015304

After MOVE Assembler:a= 859015304 b= 859015304

Press any key to continue

**Завдання 3:**

Використовуючи асемблерні інструкції додавання та інкременту, обчислити значення виразу. Порівняти отримані дані із даними отриманими в С++.

|  |  |  |  |
| --- | --- | --- | --- |
| № | Вираз для обчислення | Значення a/b/c | Знакові (-)Беззнакові (+) |
| 4 | a+(8b)+c+12 | 14h/0888h /03338888h | +/-/- |

*Код програми:*

#include "stdafx.h"

#include "iostream.h"

int main(int argc, char\* argv[])

{

char a=0x14;

short int b=0x0888;

int c=0x03338888;

int d=0;

printf("a= %i, b= %i, c= %i \n",a,b,c);

\_\_asm {

mov EAX,0

mov EBX,0

mov ECX,0

mov EDX,0

mov AL, a

mov BX,b

mov ECX,c

adc BX,BX

adc BX, b

adc BX, b

adc BX, b

adc BX, b

adc BX, b

adc BX, b

adc EBX, ECX;

adc EBX,12

adc EBX,EAX

mov d,EBX

}

printf("Assembler: d=a+(8b)+c+12= %i \n",d);

d=a+8\*b+c+12;

printf(" C++ : d=a+(8b)+c+12= %i \n",d);

return 0;

}

*Результат програми:*

a=20, b=2184, c=53708936

Assembler: d=a+(8b)+c+12= 53726440

 C++ : d=a+(8b)+c+12= 53726440

Press any key to continue

**Завдання 4:**

Використовуючи асемблерні інструкції додавання/віднімання та

інкременту/декременту, обчислити значення виразу. Порівняти отримані дані із даними

отриманими в С++.

|  |  |  |  |
| --- | --- | --- | --- |
| № | Вираз для обчислення | Значення a/b/c | Знакові (-)Беззнакові (+) |
| 4 | a-(8b-1)+c+12 | 03338888h /0888h /14h | +/-/- |

Код програми:

#include "stdafx.h"

#include "iostream.h"

int main(int argc, char\* argv[])

{

unsigned long int a=0x03338888;

short int b=0x0222;

char c=0x13;

int d=0;

printf("a= %i, b= %i, c= %i \n",a,b,c);

\_\_asm {

mov EAX,0 ;

mov EBX,0 ;

mov ECX,0 ;

mov EDX,0 ;

mov ax,b

adc eax,eax

adc eax,eax

adc eax,eax

dec eax

mov EBX,a

sbb EBX,EAX

mov cl,c

adc EBX,ECX

inc EBX

inc EBX

inc EBX

inc EBX

inc EBX

inc EBX

inc EBX

inc EBX

inc EBX

inc EBX

inc EBX

inc EBX

mov d,EBX

}

printf("Assembler: d=a-(8\*b-1)+c+12= %i \n",d);

d=a-(8\*b-1)+c+12;

printf(" C++ : d=a-(8\*b-1)+c+12= %i \n",d);

return 0;

}

*Результат програми:*

a= 53708936, b= 546, c= 19

Assembler: d=a-(8\*b-1)+c+12= 53704600

 C++ : d=a-(8\*b-1)+c+12= 53704600

Press any key to continue

**Завдання 5:**

Не використовуючи асемблерні інструкції **віднімання**, обчислити значення виразу.

Використати 32 розрядні регістри та команди розширення, або зміни знаку. Множення

замінити додаванням. Усі дані вважати знаковими. Скомпілювати приклад. Запустити

програму у відладчику. Порівняти отримані результати обчислень із аналогічними в С++.

|  |  |  |
| --- | --- | --- |
| № | Вираз для обчислення | Значення a/b/c |
| 4 | 3(a+b-1)-8c+1 | 0x30 /0x0826 /0x0404826 |

*Код програми:*

#include "stdafx.h"

#include "iostream.h"

int main(int argc, char\* argv[])

{

int a=0x30;

int b=0x10826;

int c=0x0404826;

int d=0;

printf("a= %i, b= %i, c= %i \n",a,b,c);

\_\_asm {

//\*\*\*\*\* a+b-1 \*\*\*\*

mov Al,0x30; a

mov EBX,0x10826; b

CBW

CWDE

add EAX,EBX;

dec EAX;

//\*\*\*\*\*3(a+b-1)\*\*\*\*

mov ecx,eax

add eax,eax

add eax,ecx

//\*\*\*\*\* -8c+1 \*\*\*\*\*

mov EDX,c;

add EDX,EDX;

add EDX,EDX;

add EDX,EDX;

neg EDX

add eax,edx

inc EAX

mov d,EAX

}

printf("Assembler: d=3(a+b-1)-8c+1= %i \n",d);

d=3\*(a+b-1)-8\*c+1;

printf(" C++ : d=3(a+b-1)-8c+1= %i \n",d);

return 0;

}

*Результат програми:*

a= 48, b= 67622, c= 4212774

Assembler: d=3(a+b-1)-8c+1= -33499184

 C++ : d=3(a+b-1)-8c+1= -33499184

Press any key to continue

**Завдання 6:**

Використовуючи асемблерні інструкції MOV, ADD, ADC, INC, SUB, SBB, DEC, NEG,MUL, IMUL обчислити значення виразу (операцію множення додаванням не замінювати). Виконати завдання спочатку для випадку беззнакового (MUL), а потім для знакового (IMUL) множення. При виконанні завдання, максимально оптимізувати асемблерний код (зробити мінімальну кількість інструкцій). Пояснити розбіжності рахунків Асемблера та С++, якщо такі були отримані. Порівняти отримані дані із даними отриманими в С++.

|  |  |  |
| --- | --- | --- |
| № | Вираз для обчислення | Значення a/b/c |
| 4 | a\*c-(8\*b-1)+c+12 | А) 0x1A2F /0x107A /0x0BCAБ)0x0B0E0101 /0x0B785C78 /0x0F34F67F |

*Коди програм:*

*Випадок А1:*

#include "stdafx.h"

#include "iostream.h"

int main(int argc, char\* argv[])

{

short a=0x1A2F;

short b=0x107A;

short c=0x0BCA;

int r=0;

printf("a= %X, b=%X, c= %X \n",a,b,c);

\_\_asm {

mov EAX,0 ;

mov EBX,0 ;

mov ECX,0 ;

mov EDX,0 ;

mov AX,a ;

mov BX,b ;

mov CX,c ;

imul EAX,ECX;

mov r,EAX

imul EBX,8;

dec EBX

sub r, EBX

add r, ECX;

add r, 12;

}

printf("Assembler: r=a\*c-(8\*b-1)+c+12 =>> r = %8X \n",r);

r=a\*c-(8\*b-1)+c+12;

printf(" C++ : r=a\*c-(8\*b-1)+c+12=>> r = %8X \n",r);

return 0;

}

*Результат програми:*

a= 1A2F, b=107A, c= BCA

Assembler: r=a\*c-(8\*b-1)+c+12 =>> r = 134361D

 C++ : r=a\*c-(8\*b-1)+c+12=>> r = 134361D

Press any key to continue

*Випадок А2:*

#include "stdafx.h"

#include "iostream.h"

int main(int argc, char\* argv[])

{

 short a=0x1A2F;

 short b=0x107A;

 short c=0x0BCA;

int rh=0;

int rl=0;

int r=0;

printf("a= %X, b= %X, c= %X \n",a,b,c);

\_\_asm {

 mov edx,0

 mov ax,a

 mov bx,c

 mul bx

 mov rl,eax

 mov rh,edx

 mov cx,b

 mov ax,0

 mov dx,0

 mov ax,8

 mul cx

 dec eax

 mov ebx,0

 mov ecx,0

 mov ebx,rl

 mov ecx,rh

 sub ebx,eax

 sub ecx,edx

 mov eax,0

 mov ax,c

 add eax,ebx

 add eax,12

 mov rl,eax

 mov rh,ecx

 }

printf("Assembler: r=a\*c-(8\*b-1)+c+12=>>=%8X%8X\n",rh,rl);

r=a\*c-(8\*b-1)+c+12;

printf(" C++ : r=a\*c-(8\*b-1)+c+12=>> %8X \n",r);

return 0;

}

*Результат програми:*

a= 1A2F, b= 107A, c= BCA

Assembler: r=a\*c-(8\*b-1)+c+12=>>= 134 361D

 C++ : r=a\*c-(8\*b-1)+c+12=>> 134361D

Press any key to continue

*Випадок В1:*

#include "stdafx.h"

#include "iostream.h"

int main(int argc, char\* argv[])

{

int a=0x0B0E0101;

int b=0x0B785C78;

int c=0x0F34F67F;

int rH=0;

int rL=0;

int r=0;

printf("a= %X, b= %X, c= %X \n",a,b,c);

\_\_asm {

mov EAX,a ;

mov EBX,b ;

mov ECX,c ;

mov EDX,0 ;

imul EAX,ECX;

mov rL, EAX

imul EBX,8;

dec EBX

sub rL, EBX

add rL, ECX;

add rL, 12;

}

printf("Assembler: r=a\*c-(8\*b-1)+c+12=>> = %8X \n",rL);

r=a\*c-(8\*b-1)+c+12;

printf(" C++ : r=a\*c-(8\*b-1)+c+12=>> \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ %8X \n",r);

return 0;

}

*Результат програми:*

a= B0E0101, b= B785C78, c= F34F67F

Assembler: r=a\*c-(8\*b-1)+c+12=>> = E78F884B

 C++ : r=a\*c-(8\*b-1)+c+12=>> \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ E78F884B

Press any key to continue

*Випадок В2:*

#include "stdafx.h"

#include "iostream.h"

int main(int argc, char\* argv[])

{

 int a=0x0B0E0101;

 int b=0x0B785C78;

 int c=0x0F34F67F;

int rh=0;

int rl=0;

int r=0;

//r=a\*c-(8\*b-1)+c+12;

printf("a= %X, b= %X, c= %X \n",a,b,c);

\_\_asm {

 mov edx,0

 mov eax,0

 mov eax,a

 mov ebx,c

 mul ebx

 mov rl,eax

 mov ecx,b

 mov eax,0

 mov edx,0

 mov eax,8

 mul ecx

 dec eax

 mov ebx,0

 mov ecx,0

 mov ebx,rl

 sub ebx,eax

 mov eax,0

 mov eax,c

 add eax,ebx

 add eax,12

 mov rl,eax

 }

printf("Assembler: r=a\*c-(8\*b-1)+c+12=>>=%8X\n",rl);

r=a\*c-(8\*b-1)+c+12;

printf(" C++ : r=a\*c-(8\*b-1)+c+12=>> \_ \_%8X \n",r);

return 0;

}

*Результат програми:*

a= B0E0101, b= B785C78, c= F34F67F

Assembler: r=a\*c-(8\*b-1)+c+12=>>=E78F884B

 C++ : r=a\*c-(8\*b-1)+c+12=>> \_ \_E78F884B

Press any key to continue

**Завдання 7:**

Обчислити заданий цілочисельний вираз. Змінні вважати цілочисельними знаковими. Розрядність регістрів – 32 біти, даних 8 або 16 біт. Розглянути окремо кожен випадок. Використати вивчені арифметичні операціїі. Обчислення виразу реалізувати у вигляді функції на С++ та Асемблері (аналогічно прикладу 3) для кожного із розглядуванихвипадків вхідних даних. Формат результату залежить від специфіки виразу.

(2\*с + d - 52)/(а/4 + 1)

*Код програми:*

#include "iostream.h"

#include "conio.h"

#include "stdio.h"

short int ci16(short int a, short int c, short int d)

{return (2\*c+d-52)/(a/4+1);}

short int ass16(short int a, short int c, short int d)

{

char r1=0;

 \_\_asm

 {

mov al,c

mov bl,2

mul bl

mov bl,0

mov bl,d

sub al,bl

mov cl,23

add al,cl

mov cl,0

mov cl,al

mov al,0

mov al,a

mov bl,0

mov bl,4

div bl

dec al

mov dl,0

mov dl,al

mov al,0

mov al,cl

div dl

mov r1,al

 }

 return r1;

}

void main()

{

 short int a, c, d;

 cout<<"(2\*c+d-52)/(a/4+1)\n";

 cout<<"\n-->>\n";

 cout<<"\nVvedit a(16bit)=";

 cin>>a;

 cout<<"\nVvedit c(16bit)=";

 cin>>c;

 cout<<"\nVvedit d(16bit)=";

 cin>>d; cout<<"\n-->>";

 cout<<"\n(2\*c-d+52)/(a/4-1)-16bit="<<ci16(a,c,d)<<"\n";

 getch();

}