**Практическая работа № 6**

|  |  |  |  |
| --- | --- | --- | --- |
| Выполнил | Студент гр. МФ-11 | Лазаренко А. В. | 24.03.2021 |
| Проверил | Ст. преподаватель | Меняйлов Е. С. |  |

**Тема работы:**

«Операции сдвига в языке Ассемблер. Совместное использование языка Ассемблер и языков высокого уровня».

**Цель работы:**

1) Ознакомиться и освоить работу с командами операций сдвига на языке

Ассемблер

2. Выполнить задание согласно варианту. Результаты вычислений вывести

на экран в двоичной системе счисления.

**Задание:**

1) Проанализировать задание, составить алгоритм выполнения необходимых расчетов.

2) Написать на языке ассемблера программу, выполняющую поставленную задачу.

3) Протестировать программу на различных входных данных.

**Вариант 11**

**Номер 1:**

/\* Algorithm:

\* 1) declaring necessary variables "number", "mask" (initialised with 1 (00000001)), "result" (for storing the value obtained after the operations);

\* 2) declaring the necessary char arrays "Number" (for storing the binary code of the \* entered number) and "Result" (for storing the binary code of the changed number);

\* 3) asking a user to assign the value to a variable "number";

\* 4) while a user keeps entering value, which is equal to zero or negative, he is asked \* to do it again with corresponding message;

\* 5) putting the value of a variable "number" in a CX register, copy it to the BX \* register with "mov";

\* 6) searching for the eleventh bit of the number; if it equals to one, initialising BX \* register with the number, which bits were shifted right for 11 positions;

\* 7) if value stored in the BX register is >= to 1, it becomes 1, else it is equal to zero;

\* 8) putting the value stored in a BX register in a variable "result";

\* 9) outputting the result of the program: initial number, its binary code, its value obtained after operations, its binary code (either 00000001 or 00000000).

\*/

#include <iostream>

using namespace std;

int main(void)

{

short number;

short mask = 1;

short result = 0;

char Number[255] = "";

char Result[255] = "";

cout << "Enter the number: ";

cin >> number;

while (number <= 0)

{

cout << "You entered the number wrong (it should be a positive one). Consider reentering." << endl;

cout << "Enter the number: ";

cin >> number;

}

\_\_asm

{

mov CX, number;

mov BX, CX;

sar BX, 11;

and BX, mask;

mov result, BX;

}

\_itoa\_s(number, Number, 2);

printf("Initial number: %d In binary: \t%016s;\n", number, Number);

\_itoa\_s(result, Result, 2);

printf("Operation result:%4d In binary: \t%016s.\n", result, Result);

return 0;

}

**Результаты выполнения:**

1) 

2) 

3) 

**Номер 2:**

/\* Algorithm:

\* 1) declaring necessary variables "number", "result" (for storing the value obtained after the operations);

\* 2) declaring the necessary char arrays "Number" (for storing the binary code of the entered number) and "Result" (for storing the binary code of the changed number);

\* 3) asking a user to assign the value to a variable "number";

\* 4) while a user keeps entering value, which is equal to zero or negative, he is asked to do it again with corresponding message;

\* 5) putting the value of a variable "number" in a ECX register, copy it to the EAX register with "mov";

\* 6) shifting left bits of the number stored in EAX register for (max register bit number (31) - the far right of the given field (23));

\* 7) shifting right bits of the number stored in EAX register for (the far right of the given field (13) - (max register bit number (31) - the far right of the given field (23)));

\* 8) putting the value stored in a EAX register in a variable "result";

\* 9) outputting the result of the program: initial number, its binary code, its value obtained after operations, its binary code.

\*/

#include <iostream>

using namespace std;

int main(void)

{

int number;

int result = 0;

char Number[255] = "";

char Result[255] = "";

cout << "Enter the number: ";

cin >> number;

while (number <= 0)

{

cout << "You entered the number wrong (it should be a positive one). Consider reentering." << endl;

cout << "Enter the number: ";

cin >> number;

}

\_\_asm

{

mov ECX, number;

mov EAX, ECX;

shl EAX, 31 - 23;

shr EAX, 13 + (31 - 23);

mov result, EAX;

}

\_itoa\_s(number, Number, 2);

printf("Initial number: %d In binary: \t%032s;\n", number, Number);

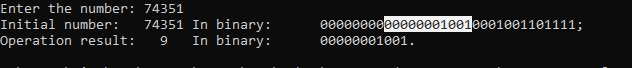
\_itoa\_s(result, Result, 2);

printf("Operation result:%4d In binary: \t%011s.\n", result, Result);

return 0;

}

**Результаты выполнения:**

1) 

2)



3)



**Задание\*:**

**Номер 2:**

#include <iostream>

using namespace std;

int main(void)

{

unsigned int number;

unsigned int result;

char Number[255] = "";

char Result[255] = "";

scanf\_s("%u", &number);

result = number | 0;

result <<= 8;

result >>= 21;

\_itoa\_s(number, Number, 2);

printf("Initial number: %u In binary: \t%032s;\n", number, Number);

\_itoa\_s(result, Result, 2);

printf("Operation result: %4u In binary: \t%011s.\n", result, Result);

return 0;

}

**Результат выполнения:**



**Номер 1:**

#include <iostream>

using namespace std;

int main(void)

{

int mask = 2048; // 2\*\*11 = 2048

unsigned int number;

unsigned int result;

char Number[255] = "";

char Result[255] = "";

scanf\_s("%u", &number);

result = number & mask;

result >>= 11;

\_itoa\_s(number, Number, 2);

printf("Initial number: %u In binary: \t%032s;\n", number, Number);

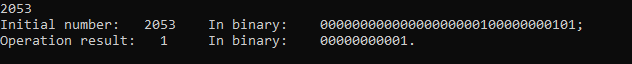
\_itoa\_s(result, Result, 2);

printf("Operation result: %3u In binary: \t%011s.\n", result, Result);

return 0;

}

**Результат выполнения:**



**Вывод:**

Я ознакомился и освоил работу с командами операций сдвига на языке Ассемблер.