

Пензенский государственный университет
Кафедра «Вычислительная техника»

ОТЧЕТ

по лабораторной работе №7

По дисциплине: «Арифметические и логические основы
вычислительной техники»

На тему: "Поразрядные логические операции и операции сдвига в
цифровых процессорах"

Выполнили:
студенты группы 20ВВ2
xxxxxxxxxxxxxxxx

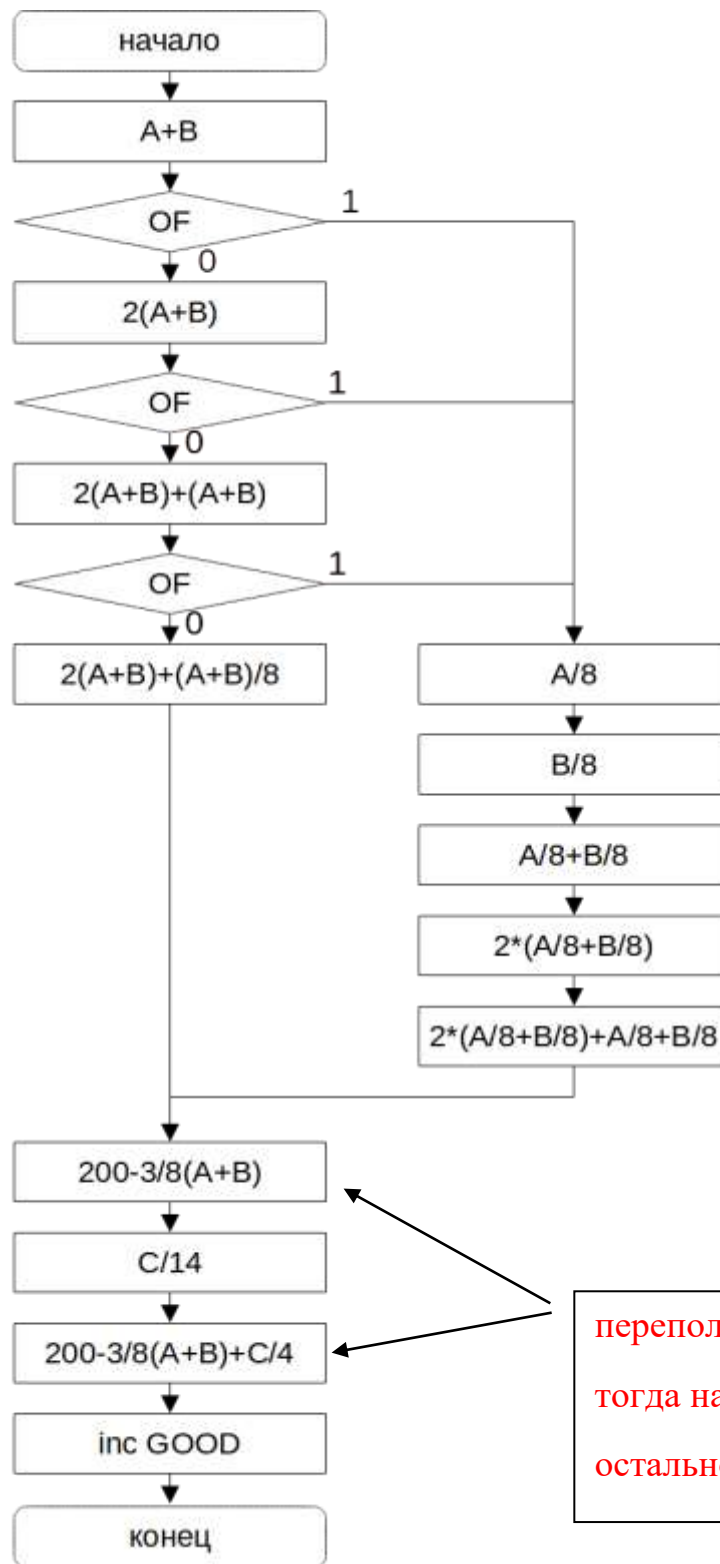
Принял:
старший преподаватель
xxxxxxxxxxxxxxxx

Пенза, 2021

Лабораторное задание:

Для процессора i8086 написать на языке ассемблера программу вычисления выражения $Y = 200 - 3/8 * (A + B) + C/4$. Формат переменных – 16 бит, целые со знаком (занимают одно слово). Умножение и деление на константы делать только с использованием операций сдвига, сложения и вычитания.

Общий алгоритм вычисления выражения приведен на блок схеме:



переполнения не будет?
тогда надо доказать!
остальное -- ОК

Листинг программы:

```

; y=200-3/8 (A+B) +C/4
data segment
A dw 698h
B dw 0FAC0h
C dw 7FFFh
Y dw ?

```

```
GOOD db ?  
data ends
```

```
code segment  
assume cs: code, ds:data, ss: nothing  
start:  
mov ax, data; load adress  
mov ds,ax; data segment  
mov GOOD, 0 ;Флаг нормального завершения
```

```
mov ax, A  
add ax, B ;A+B  
jo div8  
mov bx, ax  
sal bx, 1 ;2 (A+B)  
jo div8  
add bx, ax ;3 (A+B)  
jo div8  
mov cl, 3  
sar bx, cl ;3 (A+B) /8  
jmp c4
```

```
div8:  
mov ax, A  
mov cl, 3  
sar ax, cl ;A/8  
mov bx, B  
mov cl, 3  
sar bx, cl ;B/8  
add ax, bx ;A/8+B/8  
mov bx, ax  
sal bx, 1 ;2 (A/8+B/8)  
add bx, ax ;3 (A/8+B/8)  
jmp c4
```

```
c4:  
mov ax, C  
mov cl, 2  
sar ax, cl ;C/4  
jmp result
```

```
result:  
mov cx, 200  
sub cx, bx ;200-3/8 (A+B)  
add cx, ax ;200-3/8 (A+B) +C/4
```

```
mov Y, cx
jmp LABELGOOD
```

```
LABELGOOD:
inc GOOD
```

```
quit:
mov ax, 4c00h;
int 21;
code ends
end start
```

Чтобы выполнить полное тестирование программы, нужны следующие расчёты:

1. Нормальное завершение, 2 случая (для $Y > 0$ и $Y < 0$), установка флага $GOOD = 1$.
2. Переполнение при $A+B$, 1 случай, обработка переполнения (переход к метке $div8$) и установка флага $GOOD = 1$.
3. Переполнение при $3/8*(A+B)$, 2 случая ($2*(A+B)$ и $2*(A+B) + (A+B)$), обработка переполнения (переход к метке $div8$), установка флага $GOOD = 1$.

Переполнение при вычитании $200 - 3/8*(A+B)$ невозможно:

$$c8 - \frac{3}{8} (A+B) = 200 + (-\frac{3}{8} (A+B))$$

Переполнение возникнет при условии:

$$\begin{array}{r} 0000\ 0000\ 1100\ 1000 \\ + 0111\ 1111\ 1000\ 0000 \\ \hline 1000\ 0000\ 0100\ 1000 \end{array}$$

Значит, наименьшее значение $\left[-\frac{3}{8} (A+B)\right]_2 = 0111\ 1111\ 1000\ 0000$

$$\left[\frac{3}{8} (A+B)\right]_2 = 1000\ 0000\ 1000\ 0000_2$$

$$\frac{3}{8} (A+B) = -0111\ 1111\ 1000\ 0000_2 = -7F80_{16} = -32640_{10}$$

$$A+B = -32640 * 8 / 3 = -87040$$

Такой результат при сложении A и B говорит о переполнении, значит, этот случай будет обработан, т. е. будет совершен переход к метке $div8$.

Всего потребуется пять прогонов программы.

Прогоны программы:

- 1) Нормальное завершение, $Y > 0$.

$$A = 1688_{10} = 698_{16} = 110\ 1001\ 1000_2$$

$$B = -1344_{10} = -540_{16} = -101\ 0100\ 0000_2$$

$$[B]_2 = 1111\ 1010\ 1100\ 0000_2$$

$$C = 32767_{10} = 7FFF_{16} = 0111\ 1111\ 1111_2$$

$$Y = 200_{10} - 3_{10}/8_{10} * (1688_{10} - 1344_{10}) + 32767_{10}/4_{10} = 8262_{10} = 2046_{16} = 10\ 0000\ 0100\ 0110_2$$

$$A+B$$

A		0	0	0	0	0	1	1	0	1	0	0	1	1	0	0	0
B		1	1	1	1	1	0	1	0	1	1	0	0	0	0	0	0
A+B	1	0	0	0	0	0	0	0	1	0	1	0	1	1	0	0	0

$$SF = 0; CF = 1; OF = 0; ZF = 0$$

$$2 * (A+B)$$

bx	0	0	0	0	0	0	0	1	0	1	0	1	1	0	0	0
sal bx, 1	0	0	0	0	0	0	1	0	1	0	1	1	0	0	0	0

$$2 * (A+B) + (A+B)$$

$2 * (A+B)$	0	0	0	0	0	0	1	0	1	0	1	1	0	0	0	0
$(A+B)$	0	0	0	0	0	0	0	1	0	1	0	1	1	0	0	0
$2 * (A+B) + (A+B)$	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0

$$3/8 * (A+B)$$

bx	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0
sar bx, 3	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1

$$C/4$$

C	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
sar ax, 2	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1

$$200 + (-3/8 * (A+B))$$

200		0	0	0	0	0	0	0	0	1	1	0	0	1	0	0	0
-bx		1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1
$2-3/8*(A+B)$	1	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1	1

SF = 0; CF = 1; OF = 0; ZF = 0

$$200-3/8*(A+B)+C/4$$

$2-3/8*(A+B)$	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1	1
C/4	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1
Y	0	0	1	0	0	0	0	0	0	1	0	0	0	1	1	0

The screenshot shows the DEBUG interface with the following details:

- Assembly List:**
 - cs:003E A10400 mov ax,[0004]
 - cs:0041 B102 mov cl,02
 - cs:0043 D3F8 sar ax,cl
 - cs:0045 EB01 jmp 0048
 - cs:0047 90 nop
 - cs:0048 B9C800 mov cx,00C8
 - cs:004B 2BCB sub cx,bx
 - cs:004D 03C8 add cx,ax
 - cs:004F 890E0600 mov [0006],cx
 - cs:0053 EB01 jmp 0056
 - cs:0055 90 nop
 - cs:0056 FE060800 inc byte ptr [0008]
 - cs:005A B8004C mov ax,4C00** (Current instruction)
 - cs:005D CD15 int 15
 - cs:005F 0 A B C r Y [bx+si],al
- Registers:**
 - ax: 1FFF
 - bx: 0081
 - cx: 2046
 - dx: 0000
 - si: 0000
 - di: 0000
 - bp: 0000
 - sp: 0000
 - ds: 4ACF
 - es: 4ABF
 - ss: 4ACE
 - cs: 4AD0
 - ip: 005A
- Memory (ds:0000):**
 - 98 06 C0 FA FF 7F 46 20
 - 01 00 00 00 00 00 00 00
 - 00 CF 4A 8E D8 C6 06 08
 - 3 06 02 6
 - 1 E3 70
- Status Bar:** F1-Help F2-Bkpt F3-Mod F4-Here F5-Zoom F6-Next F7-Trace F8-Step F9-Run F10-Menu

$$Y = 0010\ 0000\ 0100\ 0110_2 = 2046_{16} = 8262_{10}$$

2) Нормальное завершение, $Y < 0$.

$$A = 1023_{10} = 03FF_{16} = 11\ 1111\ 1111_2$$

$$B = 1012_{10} = 03F4_{16} = 11\ 1111\ 0100_2$$

$$C = 1008_{10} = 03F0_{16} = 11\ 1111\ 0000_2$$

$$Y = 200_{10} - 3_{10}/8_{10} * (1023_{10} + 1012_{10}) + 1008_{10}/4_{10} = -311_{10} = -137_{16} = -11\ 0001\ 0001_2$$

$$A+B$$

A	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
B	0	0	0	0	0	0	1	1	1	1	1	1	0	1	0	0
A+B	0	0	0	0	0	1	1	1	1	1	1	1	0	0	1	1

$$2^* (A+B)$$

bx	0	0	0	0	0	1	1	1	1	1	1	1	0	0	1	1
sal bx,1	0	0	0	0	1	1	1	1	1	1	1	0	0	1	1	0

$$2^* (A+B)+(A+B)$$

$2^* (A+B)$	0	0	0	0	1	1	1	1	1	1	1	1	0	0	1	1	0
(A+B)	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	1	1
$2^* (A+B)+$ (A+B)	0	0	0	1	0	1	1	1	1	1	1	0	1	1	0	0	1

$$3/8^*(A+B)$$

bx	0	0	0	1	0	1	1	1	1	1	1	0	1	1	0	0	1
sar bx,3	0	0	0	0	0	0	1	0	1	1	1	1	1	1	0	1	1

$$C/4$$

C	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0
sar, 2	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0

$$200+(-3/8^*(A+B))$$

200	0	0	0	0	0	0	0	0	1	1	0	0	1	0	0	0
bx	1	1	1	1	1	1	0	1	0	0	0	0	0	1	0	1
$200+(-3/8^*(A+B))$	1	1	1	1	1	1	0	1	1	1	0	0	1	1	0	1

$$200-3/8*(A+B)+C/4$$

200-3/8*(A+B)	1	1	1	1	1	1	0	1	1	1	0	0	1	1	0	1
C/4	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0
Y	1	1	1	1	1	1	1	0	1	1	0	0	1	0	0	1

The screenshot shows a debugger interface with the following components:

- Menu Bar:** File, View, Run, Breakpoints, Data, Options, Window, Help.
- Assembly Window:**
 - Address: cs:0045 70E2, Instruction: jo 0029, Comment: ax 00FC, c=0
 - Address: cs:0047 03C8, Instruction: add cx, ax, Comment: bx 02FB, z=0
 - Address: cs:0049 890E0600, Instruction: mov [0006], cx, Comment: cx FEC9, s=0
 - Address: cs:004D 70DA, Instruction: jo 0029, Comment: dx 0000, o=0
 - Address: cs:004F EB01, Instruction: jmp 0052, Comment: si 0000, p=0
 - Address: cs:0051 90, Instruction: nop, Comment: di 0000, a=0
 - Address: cs:0052 FE060800, Instruction: inc byte ptr [0008], Comment: bp 0000, i=1
 - Address: cs:0056 B8004C, Instruction: mov ax, 4C00, Comment: sp 0000, d=0
 - Address: cs:0059 CD15, Instruction: int 15, Comment: ds 4ACF
 - Address: cs:005B 0000, Instruction: add [bx+si], al, Comment: es 4ABF
 - Address: cs:005D 0000, Instruction: add [bx+si], al, Comment: ss 4ACE
 - Address: cs:005F 0000, Instruction: add [bx+si], al, Comment: cs 4AD0
 - Address: cs:0061 0000, Instruction: add [bx+si], al, Comment: ip 0056
 - Address: cs:0063 0000, Instruction: add [bx+si], al
 - Address: cs:0065 0, Instruction: id Y [bx+si], al
- Memory Dump:**
 - ds:0000 FF 03 F4 03 F0 03 C9 FE
 - ds:0008 01 00 00 00 00 00 00 00
 - ds:0010 00 CF 4A 8E D8 C6 06 00
- Status Bar:** 1-Help F2-Bkpt F3-Mod F4-Here F5-Zoom F6-Next F7-Trace F8-Step F9-Run F10-Menu

$$[Y]_2 = 1111\ 1110\ 1100\ 1001_2$$

$$Y = -0000\ 0001\ 0011\ 0111_2 = -137_{16} = -311_{10}$$

3) Переполнение при A+B, обработка переполнения (переход к метке div8)

$$A = 31077_{10} = 7965_{16} = 111\ 1001\ 0110\ 0101_2$$

$$B = 15251_{10} = 3B93_{16} = 11\ 1011\ 1001\ 0011_2$$

$$C = 4360_{10} = 17248_{16} = 100\ 0011\ 0110\ 0000_2$$

$$Y = 200_{10} - 3_{10}/8_{10} * (31077_{10} + 15251_{10}) + 17248_{10}/4_{10} = -12858_{10} = -323A_{16} = -0011\ 0010\ 0011\ 1010_2$$

A+B

A	0	1	1	1	1	0	0	1	0	1	1	0	0	1	0	1
B	0	0	1	1	1	0	1	1	1	0	0	1	0	0	1	1
A+B	1	0	1	1	0	1	0	0	1	1	1	1	1	0	0	0

SF = 1; CF = 0; OF = 1; ZF = 0

Произошло переполнение. Переход к метке div8.

A/8

ax	0	1	1	1	1	0	0	1	0	1	1	0	0	1	0	1
sar ax, 3	0	0	0	0	1	1	1	1	0	0	1	0	1	1	0	0

B/8

bx	0	0	1	1	1	0	1	1	1	0	0	1	0	0	1	1
sar bx, 3	0	0	0	0	0	1	1	1	0	1	1	1	0	0	1	0

A/8+B/8

B/8	0	0	0	0	0	1	1	1	0	1	1	1	0	0	1	0
A/8	0	0	0	0	1	1	1	1	0	0	1	0	1	1	0	0
A/8+B/8	0	0	0	1	0	1	1	0	1	0	0	1	1	1	1	0

2*(A/8+B/8)

bx	0	0	0	1	0	1	1	0	1	0	0	1	1	1	1	0
sal bx, 1	0	0	1	0	1	1	0	1	0	0	1	1	1	1	0	0

2*(A/8+B/8)+A/8+B/8

2*(A/8+B/8)	0	0	1	0	1	1	0	1	0	0	1	1	1	1	0	0
A/8+B/8	0	0	0	1	0	1	1	0	1	0	0	1	1	1	1	0
3*(A/8+B/8)	0	1	0	0	0	0	1	1	1	1	0	1	1	0	1	0

C/4

ax	0	1	0	0	0	0	1	1	0	1	1	0	0	0	0	0
sar ax, 2	0	0	0	1	0	0	0	0	1	1	0	1	1	0	0	0

$$200+(-3*(A/8+B/8))$$

200	0	0	0	0	0	0	0	0	1	1	0	0	1	0	0	0
-3*(A/8+B/8)	1	0	1	1	1	1	0	0	0	0	1	0	0	1	1	0
2-3/8*(A+B)	1	0	1	1	1	1	0	0	1	1	1	0	1	1	1	0

$$200+(-3*(A/8+B/8))+C/4$$

200+(-3*(A/8+B/8))	1	0	1	1	1	1	0	0	1	1	1	0	1	1	1	0
C/4	0	0	0	1	0	0	0	0	1	1	0	1	1	0	0	0
Y	1	1	0	0	1	1	0	1	1	1	0	0	0	1	1	0

The screenshot shows the DOS DEBUG program interface. At the top is a menu bar with options: File, View, Run, Breakpoints, Data, Options, Window, Help. Below the menu bar, the CPU type is identified as 80486. The main window displays assembly code with addresses, hex values, and mnemonics. The instruction at address 005A is highlighted. To the right, a register window shows the current state of various registers. At the bottom, a data window shows memory contents. A status bar at the very bottom lists function key shortcuts.

Address	Hex	Mnemonic	Comment
cs:0043	D3F8	sar	ax,cl
cs:0045	EB01	jmp	0048
cs:0047	90	nop	
cs:0048	B9C800	mov	cx,00C8
cs:004B	2BCB	sub	cx,bx
cs:004D	03C8	add	cx,ax
cs:004F	890E0600	mov	[0006],cx
cs:0053	EB01	jmp	0056
cs:0055	90	nop	
cs:0056	FE060800	inc	byte ptr [0008]
cs:005A	B8004C	mov	ax,4C00
cs:005D	CD15	int	15
cs:005F	0000	add	[bx+sil],al
cs:0061	0000	add	[bx+sil],al
cs:0063	0	add	[bx+sil],al

Register	Value
ax	10D8
bx	43DA
cx	CDC6
dx	0000
si	0000
di	0000
bp	0000
sp	0000
ds	4ACF
es	4ABF
ss	4ACE
cs	4AD0
ip	005A

Address	Hex	Comment
ds:0000	65 79 93 3B 60 43 C6 C0	egY; 'C'
ds:0008	01 00 00 00 00 00 00 00	
ds:0010	BB CF 4A 8E D8 C6 06 08	1 1/2 JO + +
	3 06 02 6	+ +
	1 E3 70	p 0 + typ

Успешное завершение

F1-Help F2-Bkpt F3-Mod F4-Here F5-Zoom F6-Next F7-Trace F8-Step F9-Run F10-Menu

$$[Y]_2 = 1100\ 1101\ 1100\ 0110_2$$

$$Y = -0011\ 0010\ 0011\ 1010_2 = -323A_{16} = -12858_{10}$$

4) Переполнение при $2*(A+B)$, обработка переполнения (переход к метке div8)

$$A = 13912_{10} = 3658_{16} = 11\ 0110\ 0101\ 1000_2$$

$$B = 11315_{10} = 2C30_{16} = 10\ 1100\ 0011\ 0000_2$$

$$C = 28984_{10} = 7138_{16} = 111\ 0001\ 0011\ 1000_2$$

$$Y = 200_{10} - 3_{10}/8_{10} * (13912_{10} + 11315_{10}) + 28984_{10}/4_{10} = -2013_{10} = -7DD_{16} = -111\ 1101\ 1101_2$$

$A+B$

A	0	0	1	1	0	1	1	0	0	1	0	1	1	0	0	0
B	0	0	1	0	1	1	0	0	0	0	1	1	0	0	0	0
A+B	0	1	1	0	1	0	0	0	1	0	0	0	1	0	0	0

$2*(A+B)$

bx	0	1	1	0	1	0	0	0	1	0	0	0	1	0	0	0
sal bx, 1	1	1	0	1	0	0	0	1	0	0	0	1	0	0	0	0

Произошло переполнение. Переход к метке div8.

$A/8$

ax	0	0	1	1	0	1	1	0	0	1	0	1	1	0	0	0
sar ax, 3	0	0	0	0	0	1	1	0	1	1	0	0	1	0	1	1

$B/8$

bx	0	0	1	0	1	1	0	0	0	0	1	1	0	0	0	0
sar bx, 3	0	0	0	0	0	1	0	1	1	0	0	0	0	1	1	0

$A/8+B/8$

B/8	0	0	0	0	0	1	1	0	1	1	0	0	1	0	1	1
-----	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

A/8	0	0	0	0	0	1	0	1	1	0	0	0	0	1	1	0
A/8+B/8	0	0	0	0	1	1	0	0	0	1	0	1	0	0	0	1

$$2*(A/8+B/8)$$

bx	0	0	0	0	1	1	0	0	0	1	0	1	0	0	0	1
sal bx, 1	0	0	0	1	1	0	0	0	1	0	1	0	0	0	1	0

$$2*(A/8+B/8)+A/8+B/8$$

2*(A/8+B/8)	0	0	0	1	1	0	0	0	1	0	1	0	0	0	1	0
A/8+B/8	0	0	0	0	1	1	0	0	0	1	0	1	0	0	0	1
3*(A/8+B/8)	0	0	1	0	0	1	0	0	1	1	1	1	0	0	1	1

$$C/4$$

ax	0	1	1	1	0	0	0	1	0	0	1	1	1	0	0	0
sar ax, 2	0	0	0	1	1	1	0	0	0	1	0	0	1	1	1	0

$$200+(-3*(A/8+B/8))$$

200	0	0	0	0	0	0	0	0	0	1	1	0	0	1	0	0	0
-3*(A/8+B/8)	1	1	0	1	1	0	1	1	0	0	0	0	0	1	1	0	1
200+(-3/8*(A+B))	1	1	0	1	1	0	1	1	1	1	0	1	0	1	0	1	1

$$2+(-3*(A/8+B/8))+C/4$$

200+(-3*(A/8+B/8))	1	1	0	1	1	0	1	1	1	1	0	1	0	1	0	1	1
C/4	0	0	0	1	1	1	0	0	0	1	0	0	1	1	1	1	0
Y	1	1	1	1	1	0	0	0	0	0	1	0	0	0	1	1	1

$$2^* (A+B)+(A+B)$$

$2^* (A+B)$	0	1	1	1	1	1	1	0	1	0	0	0	0	0	0	0
$(A+B)$	0	0	1	1	1	1	1	1	0	1	0	0	0	0	0	0
$2^* (A+B)+(A+B)$	1	0	1	1	1	1	0	1	1	1	0	0	0	0	0	0

Произошло переполнение. Переход к метке div8.

$$A/8$$

ax	0	0	1	0	0	1	1	0	0	0	0	0	0	0	0	0
sar ax, 3	0	0	0	0	0	1	0	0	1	1	0	0	0	0	0	0

$$B/8$$

bx	0	0	0	1	1	0	0	1	0	1	0	0	0	0	0	0
sar bx, 3	0	0	0	0	0	0	1	1	0	0	1	0	1	0	0	0

$$A/8+B/8$$

B/8	0	0	0	0	0	0	1	1	0	0	1	0	1	0	0	0
A/8	0	0	0	0	0	1	0	0	1	1	0	0	0	0	0	0
A/8+B/8	0	0	0	0	0	1	1	1	1	1	1	0	1	0	0	0

$$2^*(A/8+B/8)$$

bx	0	0	0	0	0	1	1	1	1	1	1	0	1	0	0	0
sal bx, 1	0	0	0	0	1	1	1	1	1	1	0	1	0	0	0	0

$$2^*(A/8+B/8)+A/8+B/8$$

$2^*(A/8+B/8)$	0	0	0	0	1	1	1	1	1	1	0	1	0	0	0	0
A/8+B/8	0	0	0	0	0	1	1	1	1	1	1	0	1	0	0	0
$3^*(A/8+B/8)$	0	0	0	1	0	1	1	1	1	0	1	1	1	0	0	0

C/4

ax	0	0	1	1	0	0	0	0	0	0	1	0	0	0	0	0
sar ax, 2	0	0	0	0	1	1	0	0	0	0	0	0	1	0	0	0

$2+(-3*(A/8+B/8))$

200	0	0	0	0	0	0	0	0	0	1	1	0	0	1	0	0	0
$-3*(A/8+B/8)$	1	1	1	0	1	0	0	0	0	0	1	0	0	1	0	0	0
$2+(-3/8*(A+B))$	1	1	1	0	1	0	0	1	0	0	0	1	0	0	0	0	0

$2+(-3*(A/8+B/8))+C/4$

$2+(-3*(A/8+B/8))$	1	1	1	0	1	0	0	1	0	0	0	1	0	0	0	0	0
C/4	0	0	0	0	1	1	0	0	0	0	0	0	1	0	0	0	0
Y	1	1	1	1	0	1	0	1	0	0	0	1	1	0	0	0	0

File View Run Breakpoints Data Options Window Help

[.] CPU 80486

cs:0043 D3F8 sar ax,cl

cs:0045 EB01 jmp 0048

cs:0047 90 nop

cs:0048 B9C800 mov cx,00C8

cs:004B 2BCB sub cx,bx

cs:004D 03C8 add cx,ax

cs:004F 890E0600 mov [0006],cx

cs:0053 EB01 jmp 0056

cs:0055 90 nop

cs:0056 FE060800 inc byte ptr [0008]

cs:005A B004C mov ax,4C00

cs:005D CD15 int 15

cs:005F 0000 add [bx+si],al

cs:0061 0000 add [bx+si],al

cs:0063 0 A B C id Y [bx+si],al

ax 0C08 c=0

bx 17B8 z=0

cx F518 s=0

dx 0000 o=0

si 0000 p=0

di 0000 a=0

bp 0000 i=1

sp 0000 d=0

ds 4ACF

es 4ABF

ss 4ACE

cs 4AD0

ip 005A

ss:0008 6864

ss:0006 742E

ss:0004 706C

ss:0002 6568

ss:0000 6474

Успешное завершение

F1-Help F2-Bkpt F3-Mod F4-Here F5-Zoom F6-Next F7-Trace F8-Step F9-Run F10-Menu

$$[Y]_2 = 1111\ 0101\ 0001\ 1000_2$$

$$Y = -0000\ 1010\ 1110\ 1000_2 = -AE_{16} = -2792_{10}$$

Вывод: создали программу на языке ассемблера для вычисления выражения $Y=200-3/8*(A+B)+C/4$. Умножение и деление на константы реализовано операциями сдвига, сложения и вычитания. Исключение – переполнение - обработано для пересчета и получения верного результата.