

КОНТРОЛЬНАЯ РАБОТА ПО МДК 01.01.

«Системное программирование»

Цель работы:

приобрести практические навыки в разработке, отладке и выполнении программ на ассемблере;

закрепить теоретические знания по использованию различных методов адресации микропроцессорных систем при программировании на языке системного программирования.

Методические указания

Контрольная работа состоит из 2-х задач. В каждой задаче необходимо разработать и отладить программу на ассемблере в системном отладчике DEBUG. В каждом задании приведен пример решения типовой задачи, который рекомендуется изучить и выполнить в пошаговом режиме (это поможет выполнить основное задание).

При отладке индивидуальных задач рекомендуется первоначально присваивать переменным (параметрам циклов) удобные для контрольных расчетов значения.

После решения задачи рекомендуется ответить на контрольные вопросы для закрепления полученных знаний и подготовке к экзамену.

Результаты решения задач представить в печатном или рукописном варианте. По каждой задаче представить схему алгоритма и текст программы с комментариями. Вариант задачи соответствует номеру обучающегося по классному журналу.

1. Задание

1.1. Выполнить программу *линейной* структуры, вычисляющую значение функции Y по формуле $Y = (2 \cdot X + 4 \cdot Z - V)/2 + W + 5 + L$ для следующих исходных данных:

X = 0003; Z = 0002; V = 0004; W = 0005, L = 0005.

```
MOV  AX, 0
MOV  BX, 400
MOV  AX, [BX]
MOV  CL, 2
MUL  CL
PUSH AX
MOV  AX, [BX + 2]
MOV  CL, 4
MUL  CL
POP  DX
```

```
ADD AX, DX
SUB AX, [BX + 4]
MOV CL, 2
DIV CL
MOV AH, 0
ADD AX, [BX + 6]
ADD AX, 5
ADD AX, [BX + 8]
HLT
```

После выполнения программы в регистре AX должно храниться число 14h.

1.2. Выполнить программу *сложной* структуры, вычисляющую значение функции $Y = K!$ для следующих вариантов значений исходных данных:

a) $K = 5$; б) $K = 6$.

```
CS:0300  MOV BX, 600
CS:0303  MOV CX, [BX]
CS:0305  CALL 400
.....
CS:0400  MOV AX, 1
CS:0403  MUL CX
CS:0405  LOOP 403
CS:0407  RET.
```

В зависимости от используемого варианта исходных данных, в регистре AX после выполнения программы должно храниться число:

a) 0078h; б) 02D0h.

2. Индивидуальные задания на контрольную работу

2.1. Разработать, отладить и выполнить программу *линейной* структуры, вычисляющую значение функции Y в соответствии с вариантом.

Вариант	Функция Y	Адреса (16-ричные номера) ячеек памяти и значения хранящихся в них аргументов X, B, C					
		X		B		C	
		Адрес	Значение	Адрес	Значение	Адрес	Значение
1	$(2X + B + 1/C)/3$	0550	5	0552	BA	0554	8
2	$(4X - B + C - 1)/2$	0560	B0	0562	13	0564	17
3	$8X + B - C/3 + 1$	0570	16	0572	9A	0574	B2
4	$(X + B/2 + C - 1) \times 3$	0580	3	0582	8	0584	8B
5	$(2X + B - C)/2$	0590	17	0592	33	0594	27
6	$X/3 - B + 2C + 5$	0600	4	0602	16	0604	10
7	$X/2 - B + 4C + 1$	0610	10	0612	4D	0614	1F
8	$2(X - B + C) - 1/X$	0620	A	0622	CA	0624	FF
9	$(X + B - C)/2$	0630	35	0632	8F	0634	AC
10	$(X + B - C/2) \times 5$	0640	43	0642	2	0644	18
11	$(X + B)/2 - C$	0650	15	0652	FD	0654	5B
12	$(X - 1)/2 + B - C$	0660	1B	0662	3F	0664	32
13	$(X/4 - B + C) \times 2$	0670	14	0672	18	0674	EC
14	$(X + B - C - 1)/3$	0680	11	0682	96	0684	5
15	$2(X + 1) - C/2 + B$	0690	AB	0692	AB	0694	50
16	$(X + B - C/2) \times 6$	0700	B0	0702	CO	0704	A
17	$(X - B/2 + C - 1) \times 7$	0710	14	0712	10	0714	3
18	$3(C/2 - B + X)$	0720	A5	0722	AD	0724	FA
19	$4(X - B/3 - C)$	0730	AF	0732	5E	0734	3D
20	$(X - B)/(C - 1)$	0740	F1	0742	A7	0744	5
21	$(B - 1)/3 + C - B$	0750	2B	0752	4F	0754	12
22	$(C/3 + B - C) \times 3$	0760	12	0762	16	0764	CC
23	$(B + C - X + 8)/6$	0770	8	0772	16	0774	4
24	$3(C - 6) - B/2 + X$	0780	AD	0782	AD	0784	60
25	$C - B - X/2$	0790	B2	0792	C1	0794	A5
26	$B - X/4 - C + 7$	0800	28	0802	16	0804	31
27	$8C + B + X/6$	0810	64	0812	2A	0814	84
28	$(X - C)B/3 + 5$	0820	3F	0822	24	0824	8
29	$3(C + B) + X/4$	0830	42	0832	EE	0834	11
30	$2X + (B - C)/3 + 7$	0840	BD	0842	25	0844	2D
31	$7B - C + X/5$	0850	2	0852	1A	0854	44
32	$(X + C)B/2 - 1$	0860	1F	0862	14	0864	8
33	$2(C + X) - B/3$	0870	21	0872	DE	0874	F1
34	$X/3 - B/2 + C - 1/X$	0890	5F	0892	12	0894	32

Примечание. Значения аргументов и их адреса (задающие смещение ячеек с данными от начала сегмента DS) даны в шестнадцатеричной системе счисления.

2.2. Разработать, отладить и выполнить программу сложной структуры, вычисляющую значение функции Y в соответствии с вариантом.

Вариант	Функция Y	Вариант	Функция Y
1	$Y = \frac{B!}{5^n + B^3}$	2	$Y = \frac{B^4 + A^3}{(B + 1)^m}$
3	$Y = \frac{B^n - 5^3}{(B + 1)!}$	4	$Y = \frac{N! - (2N + K)!}{K!}$
5	$Y = \frac{B^m + 4^5}{K!}$	6	$Y = \frac{(2A + 1)^m}{(K+1)!}$
7	$Y = \frac{N! + (2 - N)!}{B^m}$	8	$Y = \frac{(A+1)!}{2A!}$
9	$Y = \frac{A^3 - 5^5}{A^m}$	10	$Y = \frac{-7! + (K + 2)!}{K!}$
11	$Y = \frac{N! - (2N - 2)!}{(N + 8)!}$	12	$Y = \frac{5^4 - A^6}{(A - 1)^m}$
13	$Y = \frac{A^5 - 5^3}{A^m}$	14	$Y = \frac{N! - (2N - 2)!}{(N + 3)!}$
15	$Y = \frac{7! - (K + 2)!}{K!}$	16	$Y = \frac{B^n}{B^n - 5^3}$
17	$Y = \frac{(2N)! - (n + 2)!}{N! + (3 - N)}$	18	$Y = \frac{A^3 - B^m}{5^m + B^2}$
19	$Y = \frac{A^8 + A^{-8}}{2(A + 1)^m}$	20	$Y = \frac{B^m - A^3}{5^m + B^2}$
21	$Y = \frac{B^m + A^5}{(A + B)^m}$	22	$Y = \frac{B^3 + A^4}{B^m}$
23	$Y = \frac{5^7 - A^5}{(A + 1)^m}$	24	$Y = \frac{B^3}{5^4 - B^n}$
25	$Y = \frac{B^4 - A^m}{5^3 + B^m}$	26	$Y = \frac{A^5 - B^m}{B^m + 5^5}$

Вариант	Функция Y	Вариант	Функция Y
27	$Y = \frac{(K - M)!}{K! + M!}$	28	$Y = \frac{A^7 - A^{-7}}{(2A + 1)^m}$
29	$Y = \frac{N! - (2N - K)!}{K!}$	30	$Y = \frac{A^5 + A^{-5}}{2A^m}$
31	$Y = \frac{A^5 - A^{-5}}{2^m A}$	32	$Y = \frac{A^9 - A^{-9}}{(2A + 1)^m}$
33	$Y = \frac{(K - 2N)! + K!}{(N - 2K)! + 1}$	34	$Y = \frac{3A^4 + (2A)^m}{3A^{m+1}}$

Примечание. Значениями аргументов и их местом размещения (ячейки оперативной памяти или РОНЫ процессора) задаться самостоятельно.

3. Справочные материалы по особенностям программирования и выполнения операций умножения и деления

3.1. При программировании на языке ассемблера операций *умножения* следует помнить, что в микропроцессоре ВМ86 они выполняются в регистре-аккумуляторе АХ и его расширителе DX.

При выполнении данных операций число (множимое, которое может храниться в РОНах или в ячейках памяти), умножается на содержимое (множитель) регистра AL (при операциях над байтами) или на содержимое регистра АХ (при операциях над словами). Результат умножения (произведение) будет возвращаться в регистр АХ, если в операции участвовали однобайтные числа. Если же умножались двухбайтные числа, то произведение будет помещено в два регистра – в АХ и DX, причем младшие разряды (младший байт) этого произведения запишутся в регистр АХ, а старшие разряды (старший байт) – в DX.

Рекомендуемая последовательность действий, которую нужно соблюдать при использовании операции умножения, следующая:

а) *при умножении однобайтных слов:*

обнулить регистр АХ;

заслать множитель в AL;

заслать множимое в младшую или старшую половину любого РОНа, например, в BL (если же множитель находится в ячейке памяти, адресуемой, допустим, через ВХ, то в этой операции нет необходимости);

выполнить операцию умножения `mul BL` или (если множимое находилось в ячейке, адресуемой через ВХ) операцию `mul byte ptr [ВХ]`;

результат умножения (произведение) прочитать в регистре АХ.

Так, например, если множитель (число 3) было записано в AL, а множимое (число 2) – в BL, то после выполнения команды `mul BL` в регистр АХ будет записано произведение 6 ($АХ = 0006$).

б) *при умножении двухбайтных слов:*

обнулить регистры АХ и DX;

заслать множитель в АХ;

заслать множимое в любой РОН, например, в СХ (если же множимое находится в ячейке памяти, адресуемой, допустим, через ВХ, то в этой операции нет необходимости);

выполнить операцию умножения `mul СХ` или (если множимое находилось в ячейке, адресуемой через ВХ) операцию `mul word ptr [ВХ]`;

результат умножения прочитать: в АХ – младшие разряды произведения, а в DX – старшие разряды.

Например, если в АХ было записано число 501h (т. е. $1281_{(10)}$), а в СХ – число 125h (т. е. $293_{(10)}$), то после выполнения команды умножения `mul СХ` будет получено произведение 5BA25h ($375333_{(10)}$). Младшие разряды (с 0 по 15) этого произведения будут помещены в регистр АХ ($АХ = BA25$), а старшие разряды (с 16 по 31) – в регистр DX ($DX = 0005$).

Примечание. Допускается использование команды `mul AX` для возведения в квадрат содержимого регистра `AX`. Однако пользоваться такой командой нужно осторожно, не задавая больших чисел, ибо возможны прерывания по переполнению разрядной сетки.

3.2. При программировании на языке ассемблера операций *деления* следует помнить не только, что они выполняются в регистре-аккумуляторе `AX` и его расширителе `DX`, но и то, что микропроцессор `VM86` оперирует только целыми числами.

Результат деления этим микропроцессором двух целых чисел также не может быть дробным и будет представлен как целое частное и целый остаток. Например, результатом деления $5/2$ будет частное, равное `02`, и остаток, равный `01`, а результатом деления $0Bh/3$ (т. е. $11_{(10)}/3$) будет частное `03` и остаток `02`.

В операциях над однобайтными числами деление осуществляется в регистре-аккумуляторе `AX`, т. е. содержимое этого регистра делится на любое другое 8-битовое число, которое может храниться в других `РОНах` или в ячейках памяти.

Результат деления будет возвращаться в этот же регистр `AX`, причем частное будет помещено в младшую его половину `AL`, а остаток – в старшую половину `AH`.

В операциях над двухбайтными числами деление осуществляется в регистрах `AX` и `DX`, т. е. содержимое этих регистров делится на любое другое 16-битовое число, которое может храниться в других регистрах общего назначения или в ячейках памяти.

Результат деления будет возвращаться в эти же регистры `AX` и `DX`, причем частное будет помещено в регистр `AX`, а остаток – в регистр `DX`.

Для рассмотренных ранее примеров:

а) $5/2$ – в `AX` запишется результат `0102`, где содержимое `AL` = `02` является частным, а содержимое `AH` = `01` является остатком;

б) $Bh/3$ – в `AX` запишется результат `0203`, т. е. частное `03` запишется в `AL`, а остаток `02` будет помещен в `AH`.

При делении 16-битовых чисел, например, числа `0501h` на число `0100h` (т. е. $1281_{(10)} / 256_{(10)}$) результат деления будет помещен: частное `05` – в регистр `AX` (`AX` = `0005`), а остаток `01` – в регистр `DX` (`DX` = `0001`).

Во избежание возникновения ошибок деления и срабатывания при этом, как правило, системы прерывания процессора, нужно заботиться о том, чтобы регистры `AX` и `DX` перед началом выполнения операции деления были очищены от других, хранившихся там ранее данных (т. е. обнулены). Особенно это касается регистра `DX`.

Сформулируем теперь *рекомендуемую последовательность действий*, которую нужно соблюдать при использовании операции деления:

а) *при делении однобайтных чисел:*

обнулить регистр `AX`;

делимое записать в AX или в AL;

делитель записать в младшую или старшую половину любого РОНа, например, в CL (если же делитель находится в ячейке памяти, адресуемой, например, через BX, то в этой операции нет необходимости);

выполнить операцию деления `div CL` (или операцию `div byte ptr [BX]`, если делитель находился в ячейке, адресуемой через BX);

результат деления прочитать в AX, причем частное в – AL, а остаток в – AH.

б) при делении двухбайтных чисел:

обнулить регистры AX и DX;

записать делимое в AX;

записать делитель в любой (кроме DX) РОН, например в CX (если же, как и в предыдущем примере, делитель находится в ячейке памяти, адресуемой, например, через BX, то в этой операции также нет необходимости);

выполнить операцию деления `div CX` (или операцию `div word ptr [BX]`, если делитель находился в ячейке, адресуемой через BX);

результат деления прочитать в AX и в DX, причем частное – в регистре AX, а остаток – в регистре DX.

Примечания:

1. В случаях, если делимое меньше делителя (например 2/3), получаемое дробное частное округляется до целого путем отбрасывания дробной части результата. Например, результатом деления числа 2 на число 3 будет частное 0 и остаток 2, т. е. в регистр AX будет записано 0200. (В случаях, если требуется получить (возвратить) первоначальное значение делимого 2, следует обратную делению операцию умножения выполнять по алгоритму: $0 \times 3 + 2 = 2$).

2. Если частное превышает разрядность аккумулятора (больше FF или FFFF) или делитель является нулем, генерируется прерывание типа 0, а частное и остаток не определены.

3. При возникновении прерывания выполняются следующие действия:

содержимое регистра флагов помещается в стек;

флаги IF и TF сбрасываются в 0;

содержимое сегментного регистра CS помещается в стек;

в регистр CS загружается слово из памяти по адресу 00002;

в регистр IP загружается слово из памяти по адресу 00000.

В результате этих действий МП переходит к подпрограмме обработки прерывания типа 0, полный адрес CS:IP которой берется из ячеек 00002 и 00000.