МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В. Г. ШУХОВА» (БГТУ им. В.Г. Шухова)

Кафедра программного обеспечения вычислительной техники и автоматизированных систем

Лабораторная работа №2

по дисциплине: «Архитектура вычислительных систем»

тема: «Структура команд процессора»

Выполнил студент группы ПВ-222

Короткунов Александр Александрович

Проверили

Осипов Олег Васильевич

Белгород 2024 г.

Цель работы: изучить структуру команд процессора, научиться составлять машинный код простейших команд.

Вариант 8

```
OR AX, DX
MOV SI, 14789h
ADD AL, [ESI + 8]
CMP BYTE PTR [EBP + 4], 'j'
MOV AX, [EBX + EDI + 17h]

BB 6400
B8 7800
```

Команда 1: `OR AX, DX`
Команда выполняет операцию логического `или` над соответствующими парами битов операндов `AX` и `DX`. Оба операнда имеют регистровую адресацию. Код операции данной команды - `10`. Размер пересылаемых данных равен 16 битам, поэтому `w = 1` и к коду операции добавляется префикс `0x66`. Первый операнд указывает на регистр, поэтому `d = 1`. Операндов в памяти нет, поэтому `mod = 11`. Поле `reg` отвечает за первый операнд `AX`, значит `reg = 000`, а поле `r/m = 010`. Поля данной команды кодируются следующим образом:

Префикс	КОП	d	w	mod	reg	r/m
	10	1	1	11	000	010
0x66	0x0B			0xC2		

Сначала идёт префикс `0x66`. После префикса идут поля `КОП`, `d` и `w`, которые образуют первый байт `0x0B = 1011`. Далее идут поля `mod`, `reg` и `r/m`, которые кодируют операнды `AX` и `DX`. Таким образом, код операции выглядит как `66:0BC2`. Размер команды - 3 байта.

Команда 2: `MOV SI, 14789h`
Команда выполняет пересылку шеснадцатеричного числа `14798` в регистр `SI`. Первый операнд имеет регистровую адресацию, а второй является непосредственным операндом. Код операции данной команды - `1011`, размер пересылаемых данных равен 16 байтам, поэтому `w = 1`. Так как размер равен 16 байтам, значение `14798` обрезается и трансформируется в `4789` - старший байт отрезается. Регистру `SI` соответствует поле `reg = 110`. Число `4789` кодируется двумя байтами `0x47 = 01000111` и `0x89 = 10001001`. Байты числа в коде команды представлены в обратном порядке, поэтому сначала идёт младший байт `0x89`, а следом за ним старший - `0x47`.

Префикс	КОП	w	reg	14789h	
	1011	1	110	10001001	01000111
0x66	0xBE			0x89	0x47

Сначала идёт префикс `0x66`. После префикса идут поля `КОП`, `w` и `reg`, которые образуют первый байт `0xBE = 10111110`. Непосредственный операнд кодируется двумя байтами. Таким образом, код операции выглядит как `66:BE 8947`. Размер команды - 4 байта.

Команда 3: `ADD AL, [ESI + 8]`
Команда выполняет сложение байтов из регистра `AL` и из памяти по адресу `DS:[ESI+8]` и запись результата в `AL`. Первый операнд имеет регистровую адресацию, а второй - базовую адресацию со смещением. Код данной команды - `00`. Мы передаём значение из памяти в регист, поэтому `d = 1`. Размер пересылаемых данных - 8 бит, поэтому `w = 0`. Смещение является однобайтовым, поэтому `mod = 01`. Поле `reg` соответствует первому операнду `AL`, поэтому `reg = 000`, а `r/m = 110`, так как соответствует регистру `ESI`. Десячтичное число `8` отвечает за смещение, переведём его в шеснадцатеричную форму `8 = 0x08`.

КОП	d	w	mod	reg	r/m	8
00	1	0	01	000	110	
0x02			0x46			0x08

Таким образом, код операции выглядит как `0246 08`. Размер команды - 3 байта.

Команда 4: `CMP BYTE PTR [EBP + 4], 'j'`
Команда выполняет сравнение двух операндов. Первый операнд указывает на ячейку памяти, адрес которой находится в регистре `EBP` со смещением `4`. Второй операнд представляет из себя ASCII символ, который является непосредственным операндом. Команде `CMP` соответствует код операции `10000000/111`. `mod = 01`, так как имеется однобайтовое поле смещения. `r/m = 101` - эффективный адрес равен значению в регистре `EBP`. Смещение равно значению `4 = 0x04`. ASCII символ `j = 0x6A`.

КОП	mod	КОП	r/m	4	'j'
10000000	01	111	101		
0x80	0x7D			0x04	0x6A

Команда 5: `MOV AX, [EBX + EDI + 17h]`
Команда выполняет пересылку слова из памяти по адресу `DS:[EBX+EDI+17h]` в регистр `AX`. Первый операнд имеет регистровую адресацию, второй - базово-индексную адресацию со смещением. Для данной команды код операции - `100010`. `d = 1`, так как данные пересылаются из памяти в регистр. `w = 1`, так как происходит пересылка 16-битных значений. `mod = 01`, так как смещение - однобайтовое. `reg = 000`, так как первый операнд соответствует регистру `AX`. `r/m = 100`, так как эффективный адрес задаётся в байте `SIB`, который добавляется к коду команды. Поля `SIB` имеют значения `scale = 00`, `index = 011` (`EBX`), `base = 111` (`EDI`). Смещение кодируется одним байтом - `0x17`. Также к команде добавляется префикс `0x66`, так как происходит пересылка 16-битных значений.
Поля данной команды кодируются следующим образом:

КОП	d	w	mod	reg	r/m	scale	index	base	17h
100010	1	1	01	000	100	00	011	111	
0x8B			0x44			0x1F			0x17

Таким образом, код операции выглядит как `66:8B441F 17`. Размер команды - 5 байт.

Машинный код 1: `BB 6400`
Первый байт - `0xBB = 0b10111011`. Код операции `1011` соответствует команде `MOV`, один из операндов которой имеет регистровую адресацию, второй - непосредственную.
Разложим команду на части:

КОП	w	reg	6400	
1011	1	011		
0xBB			0x64	0x00

Машинный код 2: `B8 7800`
Первый байт - `0xB8 = 0b10111000`. Код операции `1011` соответствует команде `MOV`, один из операндов которой имеет регистровую адресацию, второй - непосредственную.
Разложим команду на части:

КОП	w	reg	7800	
1011	1	000		
0xB8			0x78	0x00

Вывод: изучили структуру команд процессора, научились составлять машинный код простейших команд.