

Отчёт

по лабораторной работе №7

вариант 123

Выполнил: Тимошкин Р. В., группа Р3131
Преподаватель: Абузов Я. А.

Текст задания

Синтезировать цикл исполнения для выданных преподавателем команд. Разработать тестовые программы, которые проверяют каждую из синтезированных команд. Загрузить в микропрограммную память БЭВМ циклы исполнения синтезированных команд, загрузить в основную память БЭВМ тестовые программы. Проверить и отладить разработанные тестовые программы и микропрограммы.

1. ADCSP - Сложить два верхних числа на вершине стека с учетом переноса, результат поместить на стек, установить признаки N/Z/V/C
2. Код операции - 0F10
3. Тестовая программа должна начинаться с адреса 119₁₆

Исходный код синтезируемой команды

Адрес МП	Микрокоманда	Описание	Комментарий
BB	81E0014002	if CR(8) = 1 then GOTO @ E0	Исправленная ячейка интерпретатора
E0	0080009008	SP -> AR	Первое значение стека в DR
E1	0100000000	MEM(AR) -> DR	
E2	0020009001	DR -> BR	Первое значение в BR
E3	0080009408	SP + 1 -> AR	Второе значение стека в DR
E4	0100000000	MEM(AR) -> DR	
E5	80E8011040	if PS(C) = 0 then GOTO ADDSP @ E8	Если нет переноса – пропускаем инструкцию
E6	0001F09421	BR + DR + 1 → DR, N, Z, V, C	Сумма двух верхних чисел + 1, знаки N, Z, V, C
E7	80E9101040	GOTO PUSHRES @ E9	Переход к сохранению результата на стек
E8	0001F09021	BR + DR → DR, N, Z, V, C	Сумма двух верхних чисел, знаки N, Z, V, C
E9	0088009208	~0 + SP -> SP, AR	Результат выражения в стек
EA	0200000000	DR -> MEM(AR)	
EB	80C4101040	GOTO INT @ C4	Переход к циклу прерывания

Описание команды

ADCSP - Сложить два верхних числа на вершине стека с учетом переноса, результат поместить на стек, установить признаки N/Z/V/C. Код операции - 0F10.

Программа

Адрес	Код команды	Мнемоника	Комментарий
119	0x0100	WORD 0x100	ARG1
11A	0xff00	WORD 0xff00	ARG2
11B	0x5b0a	WORD 0x5b0a	ARG3
11C	0x11cd	WORD 0x11cd	ARG4
11D	0x0250	WORD 0x250	ARG5
11E	0x0180	WORD 0x180	ARG6
11F	0x0000	WORD 0x0	CHECK1
120	0x0000	WORD 0x0	CHECK2
121	0x0000	WORD 0x0	CHECK3
122	0x0000	WORD 0x0	FINAL
123	0x0000	WORD 0x0	RES1
124	0x0000	WORD 0x0	RES2
125	0x0000	WORD 0x0	RES3
126	0x0200	CLA	Очистка AC
127	0x0300	CLC	Очистка carry
128	0xde0c	CALL TEST1	Вызов 1 теста
129	0x0200	CLA	Очистка AC
12A	0x0300	CLC	Очистка carry
12B	0xde25	CALL TEST2	Вызов 2 теста
12C	0x0200	CLA	Очистка AC
12D	0x0300	CLC	Очистка carry
12E	0xde3c	CALL TEST3	Вызов 3 теста
12F	0xaf01	LD #1	Загрузка значения 1 (TRUE) в AC

130	0x2eee	AND CHECK1	Проверка результатов тестов
131	0x2eee	AND CHECK2	
132	0x2eee	AND CHECK3	
133	0xeeee	ST FINAL	Сохранения значения проверки
134	0x0100	HLT	Останов
135	0xaee3	TEST1: LD ARG1	Загрузка 1 аргумента теста на стек
136	0x0c00	PUSH	
137	0xeeeb	ST RES1	Сохранения 1 аргумента
138	0xaee1	LD ARG2	Загрузка 2 аргумента теста на стек
139	0x0c00	PUSH	
13A	0x0300	CLC	Установка carry в 1
13B	0x0380	CMC	
13C	0x5ee6	ADC RES1	Сохранение ожидаемого результата
13D	0xeeee5	ST RES1	
13E	0x0300	CLC	Установка carry в 1
13F	0x0380	CMC	
140	0x0f10	WORD 0x0f10	ADCSP
141	0x0800	POP	Проверка результата
142	0x7ee0	CMP RES1	
143	0xf006	BEQ DONE1	Выход из теста с сохранением 0 (FALSE) или 1 (TRUE) в результат.
144	0xeede	ST RES1	
145	0x0800	POP	
146	0x0800	POP	
147	0x0200	CLA	
148	0xeed6	ST CHECK1	
149	0x0a00	RET	
14A	0xeed8	DONE1:ST RES1	
14B	0x0800	POP	
14C	0x0800	POP	

14D	0xaf01	LD #1	
14E	0xeed0	ST CHECK1	
14F	0x0200	CLA	
150	0x0a00	RET	
151	0xaec9	TEST2: LD ARG3	Загрузка 1 аргумента теста на стек
152	0x0c00	PUSH	
153	0xeed0	ST RES2	Сохранения 1 аргумента
154	0xaec7	LD ARG4	Загрузка 2 аргумента теста на стек
155	0x0c00	PUSH	
156	0x0300	CLC	Установка carry в 0
157	0x5ecc	ADC RES2	Сохранение ожидаемого результата
158	0xeecb	ST RES2	
159	0x0300	CLC	Установка carry в 1
15A	0x0f10	WORD 0x0f10	ADCSP
15B	0x0800	POP	Проверка результата
15C	0x7ec7	CMP RES2	
15D	0xf006	BEQ DONE2	Выход из теста с сохранением 0 (FALSE) или 1 (TRUE) в результат.
15E	0xeec5	ST RES2	
15F	0x0800	POP	
160	0x0800	POP	
161	0x0200	CLA	
162	0xeebd	ST CHECK2	
163	0x0a00	RET	
164	0xeebf	DONE2:ST RES2	
165	0x0800	POP	
166	0x0800	POP	
167	0xaf01	LD #1	
168	0xeeb7	ST CHECK2	
169	0x0200	CLA	

16A	0x0a00	RET	
16B	0xaebl	TEST3: LD ARG5	Загрузка 1 аргумента теста на стек
16C	0x0c00	PUSH	
16D	0xeeb7	ST RES3	Сохранения 1 аргумента
16E	0xaeaf	LD ARG6	Загрузка 2 аргумента теста на стек
16F	0x0c00	PUSH	
170	0x0300	CLC	Установка carry в 1
171	0x0380	CMC	
172	0x5eb2	ADC RES3	Сохранение ожидаемого результата
173	0xeeb1	ST RES3	
174	0x0300	CLC	Установка carry в 1
175	0x0380	CMC	
176	0x0f10	WORD 0x0f10	ADCSP
177	0x0800	POP	Проверка результата
178	0x7eac	CMP RES3	
179	0xf006	BEQ DONE3	Выход из теста с сохранением 0 (FALSE) или 1 (TRUE) в результат.
17A	0xeeaa	ST RES3	
17B	0x0800	POP	
17C	0x0800	POP	
17D	0x0200	CLA	
17E	0xeea2	ST CHECK3	
17F	0x0a00	RET	
180	0xeea4	DONE3:ST RES3	
181	0x0800	POP	
182	0x0800	POP	
183	0xaf01	LD #1	
184	0xee9c	ST CHECK3	
185	0x0200	CLA	
186	0x0a00	RET	

Описание программы

Программа тестируем синтезированную команду на трех разных тестах и записывает 1, если результат сходится и 0, если нет.

Область допустимых значений

Значение ARG должны находиться в промежутке $(-2^{14}; 2^{14} - 1)$

Однако для специфичных случаев (Тестирование NZVC флагов) допускается $(-2^{15}; 2^{15} - 1)$

Методика проверки

1. Открыть БЭВМ в формате dual ``java -Dmode=dual -jar bcomp-ng.jar``
2. На основе help и таблицы микрокоманд перенести нужные микрокоманды в БЭВМ
 - MW 0080009008
 - MW 0100000000
 - MW 0020009001
 - MW 0080009408
 - MW 0100000000
 - MW 80E8011040
 - MW 0001F09421
 - MW 80E9101040
 - MW 0001F09021
 - MW 0088009208
 - MW 0200000000
 - MW 80C4101040
3. Открыть режим ввода ассемблера
4. Загрузить команды в БЭВМ
5. Написать после кода END и нажать Enter
6. Нажать на Run (F9)
7. Запустить основную программу с адреса 0x119 (F7)
8. Удостовериться, что значение FINAL $(122_{16}) = 0001_{16}$

Вывод

В ходе выполнения лабораторной работы я изучил алгоритм синтеза собственной команды БЭВМ с помощью микропрограмм и методику проверки сделанной программы.