

1. Що таке слово стану процесора?
У кожному такті роботи блока обробки даних формуються ознаки результату:
Ознаки Z, N, V, C утворюють **слово стану процесора**.

2. Які ознаки результату формуються в кожному такті роботи БОД?

N – ознака знаку

Z – ознака нульового результату

V – ознака переповнення

C – ознака переносу за межі старшого розряду результату

3. На підставі чого здійснюється розгалуження обчислювального процесу?

На підставі слова стану процесора. Ознаки використовуються для розгалуження обчислювального процесу. Певна ознака результату використовується в мікрокоманді умовного переходу як логічна умова, залежно від значення якої відбувається або не відбувається перехід.

4. Які операції над ознаками результату можна використовувати в регістрах M, N схеми керування станом звусів? Комбінування, заперечення(для формування логічної умови)

5. Які СКЗ3 формує сигнал логічної умови СТ?

Загалом існує 64 варіанти формування СТ основі «старих» та «свіжих» ознак, 14 найпоширеніших з них мають мнемоніки: zo, vo, co, pxorv, corz, m_z, m_v, m_n, m_z, m_z, m_v, m_n, m_z. Інші 50 варіантів задаються комбінаціями розрядів СКЗ3_MI.5-0 макроінструкції для схем керування станом і зсувами до допомогою директиви FIELD.

6. Які сигнали можна під'єднувати до входів L1-L6 мультиплексора умови?

До входів L1 – L6 можна під'єднувати такі сигнали:

Z, C, N, V, RM.Z, RM.C, RM.N, RM.V, RN.Z, RN.C, RN.N, RN.V, CT,

RDM, RDD, INT, IRQ0, IRQ1, IRQ2, IRQ3, IRQ4, IRQ5, IRQ6, IRQ7.

7. Які мнемоніки логічних умов використовують в мікрокомандах ФАМ?

Система команд ФАМ використовує такі мнемоніки логічних умов:

z,
nz,
zo, co, no, vo,
m_z, m_c, m_n, m_v,
m_z, m_c, m_n, m_v,
pxorv,
zorc,
l1, l2, l3, l4, l5, l6,
ct, rdm, rdd, int, irq0, irq1, ..., irq7,

а також те саме з префіксом not.

8. За допомогою якої директиви БМК здійснюється під'єднання сигналів до входів L1-L6 мультиплексора умови?

За допомогою директиви БМК LINK

9. Які мікрокоманди ФАМ забезпечують роботу з мікропрограмами?

Мікрокоманди для організації циклів:

```
{ rfct; } \ повторити цикл за адресою з верхівки стека, якщо (РА/ЛПЦ) ≠ 0
{ rptct addr; } \ повторити цикл за адресою addr, якщо (РА/ЛПЦ) ≠ 0
{ loop cond; } \ перехід до наступної адреси (вихід з циклу), якщо
    \ (cond = 1), інакше – перехід за адресою з верхівки стека
{ twb cond, addr; } \ розгалуження на три напрями
```

Мікрокоманди виклику мікропідпрограм:

```
{ cjs cond, addr; } \ виклик мікропідпрограми за адресою addr, якщо
    \ cond=1
{ crtn cond; } \ повернення з мікропідпрограми, якщо cond=1
{ jsrp cond, addr; } \ виклик мікропідпрограми за адресою addr, якщо
    \ cond=1, інакше – виклик мікропідпрограми за адресою
    \ з РА/ЛПЦ
```

cond → z, nz, zo, co, no, vo, m_z, m_c, m_n, m_v,
m_z, m_c, m_n, m_v, pxorv, zorc,
l1, l2, l3, l4, l5, l6,
ct, rdm, rdd, int, irq0, irq1, ..., irq7,
а також те саме з префіксом not.

Лабораторна робота №3 Відповіді на контрольні запитання

1. Як реалізують складні операції в комп'ютерних системах?
Шляхом їх розкладання на сукупність мікрооперацій, кожній з яких відповідає команда мікроасемблера.
2. Перелічити формати даних, над якими виконують складні операції.
Над цілими розрядними двійковими числами(формат INT) та над числами з плаваючою крапкою (REAL)
3. Як перетворити число з формату INT у формат REAL?

Вихідне число, подане у доповняльному коді, перетворити у прямий код.

Заповнити формат REAL.

знак мантиси залишити без змін

знайти порядок числа шляхом пошуку крайньої лівої одиниці серед значущих розрядів числа у ПК. Позиція цієї одиниці дорівнює порядку числа. Порядок числа у двійковому коді записати в молодші розряди старшого байта формату REAL;

знайти значущі розряди мантиси. Для цього знайдену старшу одиницю (у ПК) разом з усіма цифрами, розміщеними праворуч від неї, перенести у молодший байт формату REAL;

4. Подати задане число у форматі REAL.
5. Охарактеризувати основні способи множення чисел у прямому коді.
 - з молодших розрядів множника і зсувом суми часткових добутків
 - з молодших розрядів множника і зсувом вліво множеного
 - зі старших розрядів і зсувом вліво суми часткових добутків
 - множення зі старших розрядів множника і зсувом аправо множеного
6. Як виконують множення чисел з плаваючою крапкою?

Співмножники подають в форматі REAL.

Множення включає такі етапи:

- формування знаку результату(додавання за модулем два знакових розрядів мантис)
- додавання порядків(характеристик) числа
- множення мантис(множення дробових чисел без знаку у ПК)
- нормалізація результату(можливе порушення нормалізації мантиси результату вправо)

7. Охарактеризувати етапи ділення чисел з плаваючою крапкою.

Ділене та дільник подають в форматі REAL.

Ділення включає такі етапи:

- формування знаку результату(додавання за модулем два знакових розрядів мантис)
- додавання порядків(характеристик) числа шляхом віднімання від порядку діленого порядку дільника
- ділення мантис(множення дробових чисел без знаку у ПК)
- нормалізація результату

8. Навести алгоритм ділення дробових чисел, поданих у прямому коді, без відновлення остачі.

- 1) Від діленого X відняти дільник Y . Якщо результат $R_0 \geq 0$, то старша цифра z_0 частки, яка належить до розряду цілих, дорівнює 1, а якщо $R_0 < 0$, то нулю.
- 2) Подвоїти остачу.
- 3) Якщо подвоєна остача $R_1 < 0$, то додати, а якщо $R_1 \geq 0$, то відняти дільник. Якщо новоотримана остача $R_{i+1} \geq 0$, то чергова цифра частки $z_{i+1} = 1$, а якщо $R_{i+1} < 0$, то $z_{i+1} = 0$.
- 4) Повторити п.2 та п.3 $n-1$ раз.

9. Як усунути порушення нормалізації вліво(вправо) під час виконання операції над числами в форматі REAL?

При реалізації цього алгоритму може виявитися, що $z_0 = 1$. Це свідчить про порушення нормалізації вліво. Для отримання правильного результату необхідно отриману частку $z_0 z_1 \dots z_{n-1}$ зсунути вправо на один розряд, а порядок результату збільшити на одиницю.

Лабораторна робота №4. Відповіді на контрольні запитання

1. Дати визначення емуляції.
Емуляція – це спосіб виконання системи команд однієї ЕОМ(одного процесора) за допомогою іншої.
2. На чому ґрунтується емуляція системи команд?
Емуляція ґрунтується на принципі мікропрограмного керування. Вона дозволяє реалізувати будь-яку складну систему команд(систему команд високого рівня) за допомогою простої(примітивної) системи команд. При цьому кожній команді системи команд високого рівня відповідає мікропрограма, складена з команд системи команд низького рівня, яка зберігається в мікропрограмній пам'яті.
3. Назвати компоненти програмістської моделі віртуального процесора, призначеного для емуляції системи команд.
Десять 16-розрядних регістрів(REGa-REGj)
Слово стану PSW(Program Status Word)
Показчик стека SP(Stack Pointer)
Програмний лічильник PC (Program Counter)
4. Перелічити етапи емуляції команд.
 - 1) Читання програмного лічильника PC з ОП
 - 2) Читання команди Асемблера з ОП
 - 3) Розпаковування команд
 - 4) Читання операндів з ОП
 - 5) Виконання мікропрограми, яка знаходиться в ПМК й відповідає зчитаній з ОП команді Асемблера
 - 6) Читання PSW з ОП
 - 7) Формування нового PSW (модифікація) і запис його в ОП
 - 8) Модифікація програмного лічильника PC
5. Охарактеризувати структуру мікропрограми емуляції команди.
 - 1) Розміщення в ОП команди Асемблера, PC, PSW, операндів
 - 2) Читання PC з ОП, читання команди, розпаковування команди, читання операндів
 - 3) Перехід на мікропрограму виконання команди
 - 4) Читання PSW з ОП, модифікація та запис PSW в ОП, запис результату в ОП, модифікація PC
 - 5) Задання початкової адреси мікропрограми
 - 6) Мікропрограма реалізації операції
6. Як формується слово стану віртуального процесора під час емуляції команди?
7. Як перевірити результат роботи мікропрограми емуляції?
Після виконання мікропрограми емуляції в ОП будуть змінені значення в PC ($PC:=PC+1$) та (PSW (нові ознаки результату), а також регістр результату.