***Задача 1.*** Домашня робота. Підготовка до МКР 1.

Для 16-розрядного процесора розробити операційну схему (1\*), функціональний микроалгоритм у змістовних термінах (1) для виконання операції множення 16-розрядних додатних двійкових чисел (старший розряд – знаковий). Розробити цифрову діаграму стану регістрів (2) для додатних аргументів. Розробити мікропрограму виконання множення в символічних кодах мікроасемблеру (2). Множення (Z = X х Y) реалізувати заданим способом множення (табл. 1). Обробку знаків аргументів (див. табл. 3) і визначення знаку результату виконати окремо (1). Номери регістрів НОЗП і значення операндів для реалізації мікроалгоритму вибрати з табл. 2 і табл. 3 відповідно. \* - в дужках надано кількість балів за виконання кожного завдання.

|  |  |  |
| --- | --- | --- |
| *Таблиця 1* | | |
| ***h*2** | ***h*1** | **Спосіб множення (*А* х *В*)** |
| 0 | 0 | 1-м способом |
| 0 | 1 | 2-м способом |
| 1 | 0 | 3-м способом |
| 1 | 1 | 4-м способом |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | *Таблиця 2* | | | | | | | ***h*1** | ***h2*** | ***h*4** | Регістри для збереження аргументів. За необхідності подвоєння довжини регістрів використати будь-який вільний регістр (R10 – R13),  або регістр *RQ*. | | Регістри для формування результату \*\* | | ***RGX*** | ***RGY*** | R14.R15 | | 0 | 0 | 0 | *R11* | *R10* | | 0 | 0 | 1 | *R*11 | *R*12 | | 0 | 1 | 0 | R12 | *R13* | | 0 | 1 | 1 | *R*13 | *R10* | | 1 | 0 | 0 | *R*11 | *R*13 | | 1 | 0 | 1 | *R10* | *R*12 | | 1 | 1 | 0 | *R*13 | *R*12 | | 1 | 1 | 1 | *R*12 | *R10* | | |  |  |  |  |  | | --- | --- | --- | --- | --- | | Таблиця 3 | | | | | | h1 | h2 | h3 | X1 | X2 | | 0 | 0 | 0 | –12 | 9 | | 0 | 0 | 0 | 12 | 10 | | 0 | 1 | 1 | 18 | – 6 | | 0 | 1 | 1 | –19 | 5 | | 1 | 0 | 0 | 17 | 13 | | 1 | 0 | 0 | – 6 | 15 | | 1 | 1 | 1 | – 23 | – 11 | | 1 | 1 | 1 | 11 | –21 |   \*\* - в І та ІІІ способі множення для зберігання множника можна задіяти один з регістрів результату (R14.R15), в залежності від напрямку зсуву суми часткових добутків (І спосіб – R14:=Z, R15:=X; ІІ спосіб – навпаки). |