The `LEAQ` Instruction

- `LEAQ` stands for “Load Effective Address Quad”.
  - Even though it is classified as a data movement operation, it can do a limited amount of computation.
  - This instruction is often used to generate code for multiplication by small constants.
- Definition
  - Form: `LEAQ S, D`.
  - Restriction: `S` specifies a memory operand. (So `D` must be a register.)
  - The *effective address* that `S` evaluates to is placed in `D`: $D \leftarrow EA(S)$.

Examples of The `LEAQ` Instruction

- If:
  - R[`%rax`] contains the value x and
  - R[`%rcx`] contains the value y.

- Then:
  - `LEAQ 6(%rax), %rdx` stores the value $6 + x$ in `%rdx`.
  - `LEAQ (%rax, %rcx), %rdx` stores the value $x + y$ in `%rdx`.
  - `LEAQ (%rax, %rcx, 4), %rdx` stores the value $x + 4y$ in `%rdx`.
  - `LEAQ 7(%rax, %rax, 8), %rdx` stores the value $7 + 9x$ in `%rdx`.
  - `LEAQ 0xA(, %rcx, 4), %rdx` stores the value $10 + 4y$ in `%rdx`.
  - `LEAQ 9(%rax, %rcx, 2), %rdx` stores the value $9 + x + 2y$ in `%rdx`.