

Программирование

Вопрос: Нисходящее проектирование программы. Технология модульного программирования, ее отличительные особенности, достоинства и недостатки. Область применения технологии модульного программирования. Правила декомпозиции решаемой задачи (правило семи). Актуальна ли технология модульного программирования для технологии объектно-ориентированного программирования?

Ответ: Технология модульного программирования основывается на нисходящем проектировании программы, когда решаемая задача разбивается на несколько более простых, функционально законченных подзадач. Решение каждой из таких подзадач реализуется с помощью соответствующей функции. Для разумной декомпозиции решаемой задачи можно использовать "правило семи", предложенное психологами. Применительно к программированию это правило можно сформулировать следующим образом.

Каждый оператор в теле функции должен содержать не более 7 строк, а число таких операторов должен быть не более 7. Таким образом, тело функции не должно содержать более 50 строк. Число параметров функции также не должно превышать семи.

Технология модульного программирования является дальнейшим развитием более ранней технологии структурного программирования. Вместе с тем, эта технология актуальна и для более поздней технологии объектно-ориентированного программирования. Она позволяет обоснованно определить состав методов в класса.

ЭВМ и периферийные устройства

Вопрос: Поясните назначение и функции основных элементов операционного блока МП: АЛУ, РОН, регистра признаков.

Ответ: Операционный блок МП предназначен для выполнения некоторого функционально полного набора логических и арифметических операций. Как правило, в его состав входят АЛУ, буферные регистры операндов, регистр результата (аккумулятор), регистр признаков и блок регистров общего назначения (РОН). Комбинационная схема, являющаяся основой АЛУ, содержит двоичный сумматор и набор логических схем. В АЛУ выполняются несколько простейших арифметических (сложение, вычитание) и поразрядных логических (И, ИЛИ, НЕ и др.) операций. Операции по обработке данных, для которых в ОБ отсутствуют аппаратные средства, выполняются программно с помощью процедур. Такие процедуры реализуются в виде последовательности простых операций ОБ, т.е. выполняются на более высоком уровне организации вычислительного процесса, чем уровень регистровых передач.

Кроме универсального АЛУ, МП может содержать одно или несколько специализированных АЛУ. В качестве последних обычно используют блоки аппаратного умножения и деления, а также блоки для выполнения операций с плавающей точкой. Наличие специализированных АЛУ естественно увеличивает сложность СБИС МП, но за счет выполнения дополнительных операций на уровне регистровых передач производительность МП повышается.

Важной составляющей ОБ современных МП является блок внутренней памяти, реализованный в виде набора программно доступных регистров, называемых регистрами общего назначения (РОН). Время обращения к РОН меньше, чем к любым другим устройствам памяти, поэтому память на РОН называется *сверхоперативной*, а устройство, в виде которого она реализована, - сверхоперативным запоминающим устройством (СОЗУ). Число РОН в МП невелико (6 - 16), тем не менее, их наличие в среднем существенно ускоряет выполнение операций. При наличии блока РОН операнды команд могут размещаться в одной из двух запоминающих сред - ОП или СОЗУ. Использование СОЗУ позволяет исключить значительную часть обращений МП к ОП через общую СШ. С одной стороны, это повышает производительность за счет более быстрого обращения к СОЗУ, с другой стороны, появляется возможность параллельно с работой МП использовать СШ для обмена информацией между другими устройствами ВМ. Используя специальные команды, пользователь может либо записывать информацию в РОН, либо считывать ее из РОН при выполнении различных арифметических и логических операций.

В большинстве ранних моделей МП один из общих регистров выделялся в качестве главного регистра. Наделение главного регистра, называемого *аккумулятором* или регистром результата, особыми функциями позволяло реализовать ОБ в виде одноадресного устройства. В таком ОБ один из исходных операндов арифметических и логических операций обязательно должен размещаться в аккумуляторе и в него же помещается результат. Другой операнд названных операций может находиться в памяти или РОН. Входные данные поступают в аккумулятор с внутренней шины МП, а аккумулятор, в свою очередь, может посылать данные на эту шину.

Признаки операций АЛУ (флаги), характеризующие результаты вычислений, запоминаются в одноименных флагах регистра признаков. Типичными признаками являются: нулевой результат, наличие переноса, переполнение, четность, знак и некоторые другие.

Флаг нулевого результата ZF (Zero Flag) устанавливается в 1 при нулевом значении результата. При ненулевом результате ZF=0.

Флаг переноса CF (Carry Flag) запоминает значение переноса (заёма) при сложении (вычитании) операндов. В некоторых МП флаг CF используется для запоминания выдвигаемого бита при сдвиге операнда.

Флаг переполнения OF (Overflow Flag) фиксирует переполнение разрядной сетки результата при выполнении операций со знаковыми числами.

Флаг четности или паритета PF (Parity Flag) фиксирует наличие четного числа единичных разрядов в младшем байте результата. Этот флаг часто используют для контроля при передаче данных.

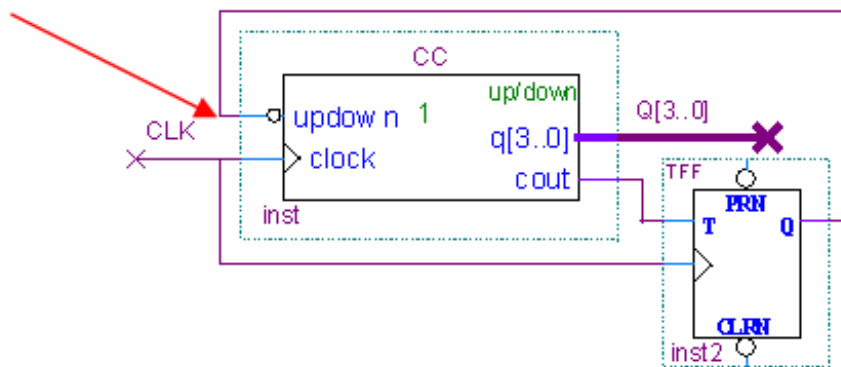
Флаг знака SF (Sign Flag) дублирует значение старшего бита результата. При выполнении операций в дополнительных кодах флаг SF соответствует знаку числа.

В большинстве случаев признаки результата используются для программного управления последовательностями выполняемых команд при разветвлениях и циклах.

Среди регистров общего назначения часто выделяют регистры, используемые в качестве базовых и индексных регистров. В ряде способов адресации содержимое указанных регистров участвует в формировании исполнительных адресов операндов в памяти.

Схемотехника

Вопрос: В каком состоянии будет находиться приведенный на рисунке счетчик, если исходно Q[3..0] и триггер в схеме находились в состоянии 0 и прошло 130 тактов CLK? (Вход updown счетчика – инверсный)



Ответ: Исходно счетчик находится в состоянии H'00'. Когда счетчик сосчитает пятнадцать импульсов, на его выходе cout установится сигнал высокого уровня и следующий, шестнадцатый такт, установит триггер в «1», а счетчик в состояние H'00' и режим работы на вычитание. На выходе счетчика cout установится «1». Семнадцатый тактовый сигнал установит счетчик в состояние H'FF' и триггер в состояние «0». Далее ситуация будет повторяться. Четные такты будут устанавливать счетчик в состояние H'00', а нечетные в состояние H'FF'. Таким образом, 130 такт установит счетчик в состояние H'00'.