```
void __fastcall
Finalize_Only_SET(int i_Set)
{ // выполнение инструкции SET номер i_Set !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
// предполагается, что это мгновенно (время учитывать не надо).....................
// инструкция SET выполняется НЕ ПРОЦЕССОРОМ, а ВХОДНЫМ коммуникатором !!!
 char Set[_SET_LEN],
      adrResult[_ID_LEN],
      adrPredicat[_ID_LEN], // поле предиката
      str[_4096], tmp[_512];
 bool s_isPredicat, // true, если ВЫПОЛНИВШИЙСЯ оператор есть ПРЕДИКАТ
      isPredicat, // true, если ЗАВИСИМЫЙ оператор ПРЕДИКАТ
      flagNot, // true, если в ЗАВИСИМОМ операторе первый символ имени флага предиката "!" или
      flagPredicat, // true, если в имени ЗАВИСИМОГО оператора есть переменная (XXX или !XXX)
      flagPredica_TRUE; // true, если в ЗАВИСИМОМ операторе флаг предиката true (с учётом isNo
//
 if( !Regim ) // не выполнять - закончить счет ------------------------------
  return;
//
 REAL Result = StrToReal( Mem_Instruction[i_Set].adrOp1, i_Set ); // запомнили значение 1-го о
 strcpy( adrResult, Mem_Instruction[i_Set].adrResult ); // строка-адрес результата выполненног
//
 Mem_Instruction[i_Set].flagOp1 = true; // у SET всегда готов!...
//
 s_isPredicat = false; // is_Predicat( Mem_Instruction[i_Set].Set ); // у SET всегда false
//
////////////////////////////////////////////////////////////////////////////
 Add_toData( i_Set, adrResult, Result ); // добавим в Mem_Data[] и для визуализации
////////////////////////////////////////////////////////////////////////////
//
// установим флаг единократного выполнения SET ...............................
//
 Mem_Instruction[i_Set].flagExecOut = true; // установили флаг единичного выполнения
 mS->Cells[6][i_Set+1] = Vizu_Flags(i_Set); // визуализировали это в таблице SG_Sets
//
 t_printf( "-I- %s(){1}: инструкция #%d [%s] выполнена (%s) -I-",
                  __FUNC__, i_Set, Line_Set(i_Set, 1), Get_Time_asLine());
//
////////////////////////////////////////////////////////////////////////////
// установим флаг ГОТОВ у ВСЕХ операндов, совпадающих по имени с адресом adrResult в пуле инст
```

```
//
 strcpy(str, ""); // очистим str
//
////////////////////////////////////////////////////////////////////////////
 for( ULI i=0; i<Really_Set; i++ ) // по всему пулу инструкций в Mem_Instruction[]
  {
   strcpy( Set, Mem_Instruction[i].Set ); // ... так удобнее для дальнейшей работы !
//
   if( is_SET( Set ) ) // это инструкция SET - не обрабатываем ! ---------------
    continue;
//
   strcpy( adrPredicat, Mem_Instruction[i].adrPredicat ); // будем работать с adrPredicat, не
//
   isPredicat = is_Predicat( Set ); // true, если ЗАВИСИМЫЙ оператор - ПРЕДИКАТ
//
//--- проверяем, начинается ли имя флага ПРЕДИКАТА с '!' или '~' ..............
   if ( !isPredicat ) // это инструкция - НЕ ПРЕДИКАТ...
    flagNot = ( adrPredicat[0] == symbolNot_1 || adrPredicat[0] == symbolNot_2 )
              ? true : false; // true, если в поле adrPredicat первый символ '!' или '~'
//
//--- проверяем, совпадает ли имя возвращённой переменной с именем ------------
//--- переменной в поле предиката ЗАВИСИМОЙ инструкции -----------------------
   flagPredicat = false;  // начальная установка
   if( !isPredicat &&  // ЗАВИСИМАЯ инструкция - НЕ ПРЕДИКАТ
       strcmp( adrPredicat, trueLowerCase  ) && // И это НЕ статический true
       strcmp( adrPredicat, falseLowerCase ) )  // И это НЕ статический false
   {
    if( flagNot ) // имя флага предиката НАЧИНАЕТСЯ с '!' или '~'
     if( !strcmp( &adrPredicat[1],adrResult ) ) // начиная со второго символа...
      flagPredicat = true;
//
    if( !flagNot ) // имя флага предиката  НЕ НАЧИНАЕТСЯ с '!' или '~'
     if( !strcmp( adrPredicat,adrResult ) )
      flagPredicat = true;
   }
//
//--- теперь определяем значение Result на true или false и окончательно -------
//--- (с учётом статических true/false) устанавливаем flagPredicatTrue ---------
//
```

```
   flagPredica_TRUE = false; // начальная установка
   if( flagPredicat ) // переменная-предиктор определена, но значение ещё неизвестно
    if( ( flagNot && !Result ) || // имя начинается с '!' или '~' и Result==false
       ( !flagNot &&  Result ) ) // имя Не начинается с '!' или '~' и Result==true
     flagPredica_TRUE = true;
//
//--- отдельно обрабатываем статический true или false -----------------------
   if( !strcmp( adrPredicat, trueLowerCase ) ) // если true...
    flagPredica_TRUE = true;
   if( !strcmp( adrPredicat, falseLowerCase ) ) // если false...
    flagPredica_TRUE = false;
//
////////////////////////////////////////////////////////////////////////////////
//
   switch( Get_CountOperandsByInstruction( Set ) ) // ... число входных операндов инструкции S
   {
    case 1: // ----- один операнд + (возможно) предикат -----------------------
// ----- обрабатываем случай, когда ЗАВИСИМЫЙ оператор - не предикат -----------
           if( !isPredicat ) // ... этот оператор - не предикат ...
           {
            if( MI_adrOp1(i) ) // 1-й операнд ГОТОВ
            {
             MI_flagOp1(i) = true;
             snprintf(tmp,sizeof(tmp), " %d(1|1)", i); strcat(str, tmp); // флаг ГОТОВ у 1-го
             snprintf(tmp,sizeof(tmp), " %d(*|1)", i); strcat(str, tmp); // флаг ГОТОВ у инст
             mS->Cells[6][i+1] = Vizu_Flags(i); // визуализировали ФЛАГИ данной инструкции
            }
//
//----- если это НЕ ПРЕДИКАТНАЯ инструкция, надо ещё проверить истинность поля ПРЕДИКАТА
           if( !flagPredicat &&
               flagPredica_TRUE ) // значение флага ПРЕДИКАТА
           {
//           Mem_Instruction[i].flagPredicat = true; // установили флаг ПРЕДИКТОР_ГОТОВ
            if( !Mem_Instruction[i].flagPredicat_TRUE ) // если не был установлен true...
            {
             Mem_Instruction[i].flagPredicat_TRUE = true; // установили флаг ПРЕДИКТОР_ИСТИНЕ
             snprintf(tmp,sizeof(tmp), " %d(PredTRUE|1)", i); strcat(str, tmp); // флаг true
             mS->Cells[6][i+1] = Vizu_Flags(i); // визуализировали ФЛАГИ данной инструкции
            }
```

```
            }
//
            if( MI_flagOp1(i) && // первый операнд ГОТОВ...
                flagPredica_TRUE ) // ... и флаг предиката есть true
             Add_toBuffer( i ); // добавить ГКВ-команду в буфер команд для исполнения
            } // конец if( !isPredicat )
//
// ----- обрабатываем случай, когда ЗАВИСИМЫЙ оператор суть предикат ----------
/* таких случаев теперь НЕТ.....................................................
            if( isPredicat ) // ... этот оператор - предикат ...
            {
             if( MI_adrOp1(i) ) // 1-й операнд ГОТОВ
             {  // если был установлен true - не надо ЗАНОВО ПЕРЕУСТАНАВЛИВАТЬ ..!
              MI_flagOp1 = true;
              snprintf(tmp,sizeof(tmp), " %d(1|1)", i); strcat(str, tmp); // флаг ГОТОВ у 1-го
              snprintf(tmp,sizeof(tmp), " %d(*|1)", i); strcat(str, tmp); // флаг ГОТОВ у инст
              mS->Cells[6][i+1] = Vizu_Flags(i); // визуализировали ФЛАГИ данной инструкции
             }
//
             if( MI_flagOp1 ) // первый операнд предиката ГОТОВ... у предиката нет предиката
              Add_toBuffer( i ); // добавить ГКВ-команду в буфер команд для исполнения
            } // конец if( isPredicat )
//
// ---- добавляем в текст протокола ------------------------------------------
//
*/
            break; // конец обработки оператора с одним операндом + возможно, предиктор
////////////////////////////////////////////////////////////////////////////////
    case 2: // ----- два операнда + (возможно) предикат ----------------------
// ----- обрабатываем случай ЗАВИСИМОЙ инструкции - НЕ ПРЕДИКАТА --------------
            if( !isPredicat ) // ... не предикат ...
            {
             if( MI_adrOp1(i) ) // 1-й операнд ГОТОВ
             {
              MI_flagOp1(i) = true;
              snprintf(tmp,sizeof(tmp), " %d(1|2)", i); strcat(str, tmp); // флаг ГОТОВ у 1-го
              mS->Cells[6][i+1] = Vizu_Flags(i); // визуализировали ФЛАГИ данной инструкции
             }
//
```

```
            if( MI_adrOp2(i) ) // 2-й операнд ГОТОВ
            {
             MI_flagOp2(i) = true;
             snprintf(tmp,sizeof(tmp), " %d(2|2)", i); strcat(str, tmp); // флаг ГОТОВ у 2-го
             if( MI_flagOp1(i) &&
                 MI_flagOp2(i) )
              { snprintf(tmp,sizeof(tmp), " %d(*|2)", i); strcat(str, tmp); } // флаг ГОТОВ у
             mS->Cells[6][i+1] = Vizu_Flags(i); // визуализировали ФЛАГИ данной инструкции
            }
//
//----- если эта инструкция-НЕ ПРЕДИКАТ, надо проверить флаг (и его значение) поля предиктора
          if( !flagPredicat && // это инструкция НЕ ПРЕДИКАТ
              flagPredica_TRUE ) // значение флага ПРЕДИКАТА
          {
//           Mem_Instruction[i].flagPredicat = true; // установили флаг ПРЕДИКТОР_ГОТОВ
            if( !Mem_Instruction[i].flagPredicat_TRUE ) // если не был установлен true...
            { // если был установлен true - не надо ЗАНОВО ПЕРЕУСТАНАВЛИВАТЬ ..!
             Mem_Instruction[i].flagPredicat_TRUE = true; // установили флаг ПРЕДИКТОР_ИСТИНЕ
             snprintf(tmp,sizeof(tmp), " %d(PredTRUE|1)", i); strcat(str, tmp); // флаг true
             mS->Cells[6][i+1] = Vizu_Flags(i); // визуализировали ФЛАГИ данной инструкции
            }
          }
//
            if( MI_flagOp1(i) && // первый операнд ГОТОВ...
                MI_flagOp2(i) && // второй операнд ГОТОВ...
                flagPredica_TRUE ) // ... и флаг предиката есть true
             Add_toBuffer(i); // добавить ГКВ-команду в буфер команд для исполнения
          } // конец if( !isPredictor )
//
// ----- обрабатываем случай ЗАВИСИМОЙ инструкции - ПРЕДИКАТА -----------------
//           else // так "КРАСИВШЕ" с точки зрения синтаксиса языка, но ЛОГИЧНЕЕ - см. след.
          if( isPredicat ) // ... предикат ...
          {
            if( MI_adrOp1(i) ) // 1-й операнд ГОТОВ
            {
             MI_flagOp1(i) = true;
             snprintf(tmp,sizeof(tmp), " %d(1|2)", i); strcat(str, tmp); // флаг ГОТОВ у 1-го
             mS->Cells[6][i+1] = Vizu_Flags(i); // визуализировали ФЛАГИ данной инструкции
            }
```

```
//
            if( MI_adrOp1(i) ) // 2-й операнд ГОТОВ
            {
             MI_flagOp2(i) = true;
             snprintf(tmp,sizeof(tmp), " %d(2|2)", i); strcat(str, tmp); // флаг ГОТОВ у 2-го
//
             if( MI_flagOp1(i)  &&
                 MI_flagOp2(i) )
              { snprintf(tmp,sizeof(tmp), " %d(*|2)", i); strcat(str, tmp); } // флаг ГОТОВ у
             mS->Cells[6][i+1] = Vizu_Flags(i); // визуализировали ФЛАГИ данной инструкции
            }
//
            if( MI_flagOp1(i) && // первый операнд ГОТОВ...
                MI_flagOp2(i) && // второй операнд ГОТОВ...
                flagPredica_TRUE ) // ... и флаг предиката есть true
             Add_toBuffer(i); // добавить ГКВ-команду в буфер команд для исполнения
           } // конец if( isPredictor )
//
           break; // конец обработки оператора с двумя операндами + возможно, предикат
////////////////////////////////////////////////////////////////////////////////
   default: break;  // кроме 1 или 2 операнда у оператора...
//
   } // конец switch по числу операндов у i-того оператора --------------------
```