|  |  |  |
| --- | --- | --- |
| *C source code* | *Literal Translation* | *Optimized Translation* |
| uint32\_t k ;  …  for (k = 0; k < 100; k++)  {  foo() ;  }  … | LDR R4,=0  STR R4,k  Top:  LDR R4,k  CMP R4,#99  BHI Done  BL foo  LDR R4,k  ADD R4,R4,#1  STR R4,k  B Top  Done: | LDR R4,=0  Top:  CMP R4,#99  BHI Done  BL foo  ADD R4,R4,#1  B Top  Done:  STR R4,k  ; The STR is needed to update k  ; in its memory location. |
| uint32\_t Fact(uint32\_t n)  {  uint32\_t result ;  result = 1 ;  while (n > 1)  {  result \*= n ;  n-- ;  }  return result ;  } | Fact: ; parameter ‘n’ is already in R0  LDR R1,=1 ; use R1 for temporary ‘result’  Top:  CMP R0,#1 ; (1) There are no references  BLS Done ; to identifiers ‘n’ or ‘result’  MUL R1,R1,R0  SUB R0,R0,#1 ; (2) Since functions use registers  B Top ; for temporaries, there is no  Done: ; need to update them in memory.  MOV R0,R1 ; Copy result into R0 for return  BX LR | |
| uint32\_t Ones(int32\_t x)  {  uint32\_t count ;  count = 0 ;  do  {  if (x < 0) count++ ;  x = x << 1 ;  } while (x != 0) ;  return count ;  } | Ones:  LDR R1,=0 ; use R1 for temporary ‘count’  Top:  CMP R0,#0 ; These three lines are the  IT LT ; code that the loop repeats  ADDLT R1,R1,#1  LSL R0,R0,#1 ; These two lines could be  CMP R0,#0 ; combined using LSLS  BNE Top  MOV R0,R1 ; Copy count into R0 for return  BX LR | |