Published: 05/18/67

Identification

If Statement

B. Goldberg

I. INTRODUCTION

The if statement may have one of the following forms:

```
if scalar expression then unit-1; else unit-2;
if scalar expression then unit-1;
if scalar expression then; else unit-1;
```

In this discussion, the coding produced for the three formats is illustrated for case 1 through the use of simple statements in which tests are made for equality. (Comparison operations involving other types of expressions are covered in BN.6.03.) The general format of the coding is the same for all cases. Only the coding for the if clause differs. This is illustrated in the examples below.

A. Case 1: Simple Comparison Using No Long String Variables

For simplicity, all variables used in the following statements are single-precision floating-point numbers. (The same coding pattern is produced for fixed-point numbers or short strings.)

Statement 1: if a = b then c = d; else e = f;

Pass 1 Coding

```
        Macro                                                          Comment

ldfl    alias1,bits1,0,xxx,int,auto,0,level,0                    }
ifeqfl  bits1,0,alias2,bits2,0,xxx,int,auto,0,level,0            }  if a = b
golb    alias3,144,0,xxx,con,xxxx,0,level,0                      }

ldfl    alias4,bits4,0,xxx,int,auto,0,level,0                    }
stfl    alias5,bits5,0,xxx,int,auto,0,level,0                    }  then c = d
golb    alias6,144,0,xxx,con,xxxx,0,level,0                      }
dclb    ,alias3,144,0,xxx,con,xxxx,0,level,0

ldfl    alias7,bits7,0,xxx,int,auto,0,level,0                    }
stfl    alias8,bits8,0,xxx,int,auto,0,level,0                    }  else e = f
dclb    ,alias6,144,0,xxx,con,xxxx,0,level,0
```

Here the arrows indicate transfer of control. The first golb macro is produced to provide a false exit; i.e., if a ≠ b, the else coding is executed. If a = b, the then coding is executed and the second golb macro "skips around" the else coding.

Pass 2 Coding

```
        Instruction                     Macro

        fld    sp|alias1                ldfl
        fcmp   sp|alias2                ifeqfl
        tnz    alias3                   golb
"
        fld    sp|alias4                ldfl
        fst    sp|alias5                stfl
        tra    alias6                   golb
alias3: null   "                        dclb
"
        fld    sp|alias7                ldfl
        fst    sp|alias8                stfl
alias6: null   "                        dclb
```

Statement 2: if a = b then c = d;

Pass 1 Coding

```
      Macro                                                          Comment

ldfl    alias1,bits1,0,xxx,int,auto,0,level,0                    } if a = b
ifeqfl  bits1,0,alias2,bits2,0,xxx,int,auto,0,level,0            }
golb    alias3,144,0,xxx,con,xxxx,0,level,0


ldfl    alias4,bits4,0,xxx,int,auto,0,level,0                    }
stfl    alias5,bits5,0,xxx,int,auto,0,level,0                    } then c = d
dclb    ,alias3,144,0,xxx,con,xxxx,0,level,0                     }
```

next macro

Since there is no else code, the golb macro transfers to the next macro. The next macro is also executed following the then code. This eliminates the need for a second golb macro.

Pass 2 Coding

```
          Instruction          Macro

          fld    sp|alias1     ldfl
          fcmp   sp|alias2     ifeqfl
          tnz    alias3        golb
   "
          fld    sp|alias4     ldfl
          fst    sp|alias5     stfl
alias3:   null   "             dclb
```

Statement 3: if a = b then; else c = d;

Pass 1 Coding

```
          Macro                                                    Comment

ldfl      alias1,bits1,0,xxx,int,auto,0,level,0                    } if a = b
ifeqfl    bits1,0,alias2,bits20,xxx,int,auto,0,level,0             }
golb      alias3,144,0,xxx,con,xxxx,0,level,0

golb      alias4,144,0,xxx,con,xxxx,0,level,0                      then
dclb      ,alias3,144,0,xxx,con,xxxx,0,level,0

ldfl      alias5,bits5,0,xxx,int,auto,0,level,0                    }
stfl      alias6,bits6,0,xxx,int,auto,0,level,0                    } else c = d
dclb      ,alias4,144,0,xxx,con,xxxx,0,level,0                     }
```

This coding is the same as that for statement 1, except that the body of the then clause is missing. Note that two golb macros are produced, because of the inclusion of the else statement.

Pass 2 Coding

```
                Instruction                 Macro

          fld     sp|alias1                 ldfl
          fcmp    sp|alias2                 ifeqfl
          tnz     alias3                    golb
"
          tra     alias4                    golb
alias3:   null    "                         dclb
"
          fld     sp|alias5                 ldfl
          fst     sp|alias6                 stfl
alias4:   null    "                         dclb
```

B. Case 2: Simple Comparison Using Long String Expressions in the If Clause

Here an entry to the string operation routine is called to make the comparison. (See BN.6.03). The following macros are then produced to load the result of the comparison (a 1-bit string) into the bit-string register and to provide a false exit.

```
ldbs          alias1,1,0,xxx,int,auto,0,level,0
iflb          alias2,144,0,xxx,con,xxxx,0,level,0
  .
  .
dclb          ,alias2,144,0,xxx,con,xxxx,0,level,0
```

Pass 2 translates this as follows:

```
          eapap     sp|alias1+2,*
          eapbp     sp|alias1,*
          adbbp     ap|0
          lda       bp|0
          anaq      =v1/-1,71/0
          tze       alias2

alias2:   null      "
```

C. Case 3: Use of Short Bit-String Expression in the If Clause

The following source code illustrates this case:

```
declare c bit (5);
if c then go to a;
```

Here the codition is true if any bit of c is a 1; the condition is false only if all bits of c are 0's.

Pass 1 produces the following macros to load the bit string and provide the false exit:

```
ldbs    alias1,5,0,xxx,int,auto,0,level,0
bsbs    5,0,5,0
iflb    alias2,144,0,xxx,con,xxxx,0,level,0
 .
 .
dclb    ,alias2,144,0,xxx,con,xxxx,0,level,0
```

Pass 2 translates this coding as follows:

```
          eapap     sp|alias1+2,*
          eapbp     sp|alias1,*
          adbbp     ap|0
          ldan      bp|0
          anaq      =v5/-1,67/0
          tze       alias2

alias2:   null      "
```

D. Case 4: Use of Long Bit-String Expression in the if Clause

EPL calls the ixbs_ entry to the string operation (stgop_) routine when the scalar expression in the if statement is a long or varying string. (See BN.6.03 for a description of the ixbs_ entry.) The arguments are: the string, "1"b, and a temporary location in which to store the result. This result is a fixed-point number: 0 if there are no 1-bits in the string, or the position of the first 1-bit in the string. Pass 1 then produces the following macros to load this number and to provide a false exit:

```
ldfx    alias1,bits1,0,xxx,int,auto,0,level,0
iflb    alias2,144,0,xxx,con,xxxx,0,level,0
  .
  .
dclb    ,alias2,144,0,xxx,con,xxxx,0,level,0
```

Pass 2 translates this code as follows:

```
        lda     sp|alias1
        tze     alias2
alias2: null    "
```

E. Case 5: Use of a Compound Condition in the If Clause

The coding produced for this case can be understood by examining the following source statement: if i = j & j = k then go tob ;

Pass 1 produces the following macros for this statement:

	Macro	Comment
ldfx	alias1,bits1,0,xxx,int,auto,0,level,0	Loads j
eqfx	bits1,0,alias2,bits2,0,xxx,int,auto,0,level,0	Compares j with k
stbs	alias3,1,0,xxx,int,auto,0,level,0	Stores result in temporary location
ldfx	alias4,bits4,xxx,int,auto,0,level,0	Loads i
eqfx	bits4,0,alias1,bits1,0,xxx,int,auto,0,level,0	Compares i with j
ndbs	1,0,alias3,1,0,xxx,int,auto,0,level,0	Ands the two results
bsbs	1,0,1,0	Truncates answer to a 1-bit string
iflb	alias5,144,0,xxx,con,xxxx,0,level,0	If the answer if a 0, goes to the next statement
golb	alias6,144,0,con,xxxx,0,level,0	If the answer is a 1, goes to b
dclb	,alias5,144,0,xxx,con,xxxx,0,level,0	

Pass 2 translates this code as follows:

	Instruction		Macro/Comment
	lda	sp|alias1	ldfx
	cmpa	sp|alias2	eqfx
	tze	*+3	Result is 10.....0 if j = k
	lda	0,du	00.....0 if j ≠ k
	tra	*+2	
	lda	131072,du	
	eapap	sp|alias3+2,*	stbs
	eapbp	sp|alias3,*	
	adbbp	ap|0	
	sta	bp|0	
	lda	sp|alias4	ldfx
	cmpa	sp|alias1	eqfx
	tze	*+3	Result is 10.....0 if i = j
	lda	0,du	00.....0 if i ≠ j
	tra	*+2	
	lda	131072,du	
	eapap	sp|alias3+2,*	ndbs
	eapbp	sp|alias3,*	
	adbbp	ap|0	
	ana	bp|0	
	anaq	=v1/-1,71/0	bsbs
	tze	alias5	Transfer if false; i.e., either j ≠ k or i ≠ j
	tra	alias6	If true go to b
alias5:	null	"	