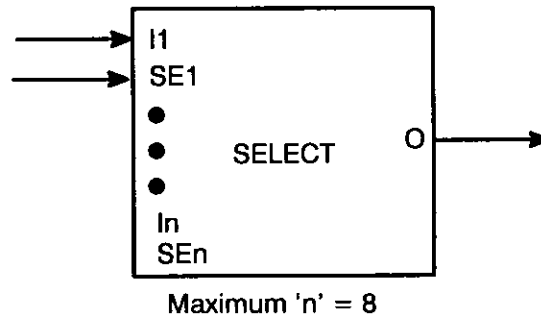


32.0 SELECT

This function can be used in AutoMax Control Block tasks and UDC Control Block tasks.



Function

OUTPUT will equal the sum of all selected inputs. If no SELECT is TRUE, OUTPUT = 0.

Program Statement

```
CALL SELECT(INPUT1 = input1%,           &  
            SELECT1 = select1@....,     &  
            INPUTn = inputn%,           &  
            SELECTn = selectn@,         &  
            OUTPUT = output%)           &
```

Inputs

I1 (INPUT1) =

INTEGER signal input 1. This parameter must be specified.

SE1 (SELECT1) =

BOOLEAN signal 1 select. This parameter must be specified. When TRUE, INPUT1 is algebraically summed with other selected inputs, producing OUTPUT.

In (INPUTn) =

INTEGER signal input n. This is an optional parameter. A maximum of 8 inputs can be specified.

SEn (SELECTn) =

BOOLEAN signal select. This is an optional parameter. However, a SELECT is required for every INPUT. When TRUE, (INPUTn) is algebraically summed with other selected inputs, producing OUTPUT.

Outputs

O (OUTPUT) =

INTEGER signal output. This parameter must be specified.

Notes

1. The order in which (INPUTn) and (SELECTn) pairs are entered is not important. However, it is required that, for "m" channels used, channels 1 through "m" be programmed. In other words, the channels used must be contiguous beginning with channel 1. A compilation error will occur if this requirement is not met.

The following is a correct statement:

```
CALL SELECT (INPUT1 = INA%,SELECT1 = SELECTA@,    &
             INPUT2 = INB%,SELECT2 = SELECTB@,    &
             INPUT3 = INC%,SELECT3 = SELECTC@,    &
             INPUT4 = IND%,SELECT4 = SELECTD@,    &
             OUTPUT = output%)
```

The following is also correct, illustrating that the order of entry is not important:

```
CALL SELECT (INPUT2 = INB%,SELECT2 = SELECTB@,    &
             INPUT4 = IND%,SELECT4 = SELECTD@,    &
             INPUT3 = INC%,SELECT3 = SELECTC@,    &
             INPUT1 = INA%,SELECT1 = SELECTA@,    &
             OUTPUT= output%)
```

The following is an incorrect statement because the channels are not contiguous:

```
CALL SELECT (INPUT1 = INA%,SELECT1 = SELECTA@,    &
             INPUT3 = INC%,SELECT3=SELECTC@,    &
             INPUT4 = IND%,SELECT4=SELECTD@,    &
             OUTPUT = output%)
```

2. **Overflow Handling:** During block execution, (INPUTn) and (SELECTn) pairs are read and processed in sequential order beginning with INPUT1 and SELECT1. Since the sum of the selected inputs is calculated as a 32-bit value and inputs are integer quantities (16-bit values), an overflow cannot occur during these intermediate additions (given an 8 input channel restriction).

When all consecutive sequential input channels are processed, the final sum (OUTPUT) must be within the range -32768 to +32767 or an error will be logged and the result will be clamped to -32768 or +32767, producing OUTPUT.