IF-THEN functions in Inventor using SIGN function

In Inventor, how can you assign values to Y according to the following conditions without using user-defined VBA functions?

```
IF (X > 0 in) AND (X < 46 in) THEN
    Y = 15 in
ELSE IF (X >= 46 in) AND (X < 55 in) THEN
    Y = 17 in
ELSE IF (X >= 55 in) AND (X <= 60 in) THEN
    Y = 19 in
ELSE
    Y = 10 in
```

Inventor has a function called SIGN(..) which returns 1 (ul) if the argument is greater than zero, and 0 (ul) if the argument is less than or equal to zero. The argument may have units like in, mm or radian, or it may be unitless.

This SIGN function can be used to set up complex IF-THEN formulas. We may need to define intermediate Boolean variables, for which we give names like if_1, if_2, if_n etc. These variables (also called predicate variables) have only one of two possible values (0 or 1), representing if a particular condition is true (corresponds to value 1) or false (corresponds to value 0).

The basic building blocks of this system[§] consist of four formulae given in 1 to 4 below. In the following, X and a are variables with same units, e.g. inch.

```
1.    IF   X < a  THEN                          if_n = sign( a – X )
           if_n = 1
      ELSE
           if_n = 0

2.    IF   X <= a  THEN                       if_n = 1 - sign( X - a )
           if_n = 1
      ELSE
           if_n = 0

3.    IF   X > a  THEN                          if_n = sign( X - a )
           if_n = 1
      ELSE
           if_n = 0

4.    IF   X >= a  THEN                       if_n = 1 - sign( a – X )
           if_n = 1
      ELSE
           if_n = 0
```

We can use these building blocks to form compound logical statements.

§ Although some of these functions can be obtained from the others, it is helpful to treat them as “basic” building blocks.

5. Logical AND (if more than two conditions are "connected" by AND's you can extend the same idea)

```
IF if_1 is true AND if_2 is true THEN          if_n = if_1 * if_2
      if_n = 1
ELSE
      If_n = 0
```

6. Logical OR (if more than two conditions are "connected" by OR's you can extend the same idea)

```
IF if_1 is true OR if_2 is true THEN   if_n = sign( if_1 + if_2 )
      if_n = 1
ELSE
      If_n = 0
```

7. Logical NOT (NOT true ≡ false)

```
IF if_1 is false THEN                            if_n = 1 - if_1
      if_n = 1
ELSE
      If_n = 0
```

8. Exact Equality

```
IF   X = a  THEN                     if_n = (1 - sign( X - a )) *
      if_n = 1                                (1 - sign( a – X ))
ELSE
      if_n = 0
```

9. Equality with tolerance

```
IF   X = a ± tol THEN          if_n = (1 - sign( X - a – tol )) *
      if_n = 1                          (1 - sign( a – tol – X ))
ELSE
      if_n = 0
```

10. Equality with tolerance (excluding the two extremes)

```
IF   X > a - tol AND X < a + tol THEN
      if_n = 1                         if_n = sign( X - a – tol)*
ELSE                                            sign( a + tol -X)
      if_n = 0
```

Now we can give the answer for the example shown at the beginning. For this, it is helpful to define three "if variables."

```
if_1 = sign(X) * sign(46 in - X)
if_2 = (1 ul - sign(46 in - X)) * sign(X - 55 in)
if_3 = (1 ul - sign(55 in - X)) * (1 ul - sign(X - 60 in))
Y    = 10 in + if_1 * (15 in – 10 in) + if_2 * (17 in – 10 in)
             + if_3 * (19 in – 10 in)
```

Parameter Name	Unit	Equation	Nominal Va...
Model Parame...			
User Parameters			
X	in	56 in	56.000000
if_1	ul	sign(X) * sign(46 in - X)	0.000000
if_2	ul	(1 ul - sign(46 in - X)) * sign(55 in - X)	0.000000
if_3	ul	(1 ul - sign(55 in - X)) * (1 ul - sign(X - 60 in))	1.000000
Y	in	10 in + if_1 * (15 in - 10 in) + if_2 * (17 in - 10 in) + if_3 * (19 in - 10 in)	19.000000

Simulating a Visual Basic CASE Statement

Consider the following VB code snippet (all parameters in mm units):

```
Select case Size
      Case 8
            GuideThk = 0.6
            GuideDia = 11.5
      Case 10
            GuideThk = 0.6
            GuideDia = 14.5
      Case 16
            GuideThk = 0.8
            GuideDia = 22.5
End Select
```

This is equivalent to the following IF-THEN statement:

```
If Size = 8 then
      GuideThk = 0.6
      GuideDia = 11.5
Else if Size = 10 then
      GuideThk = 0.6
      GuideDia = 14.5
Else if Size = 16 then
      GuideThk = 0.8
      GuideDia = 22.5
End if
```

This can be expressed in terms of “if” variables, using item #7 (exact equality) listed earlier:

```
if_1 = (1 ul - sign(Size – 8 mm)) * (1 ul - sign(8 mm - Size))
if_2 = (1 ul - sign(Size – 10 mm)) * (1 ul - sign(10 mm - Size))
if_3 = (1 ul - sign(Size – 16 mm)) * (1 ul - sign(16 mm - Size))
GuideThk = if_1 * 0.6 mm + if_2 * 0.6 mm + if_3 * 0.8 mm
GuideDia = if_1 * 11.5 mm + if_2 * 14.5 mm + if_3 * 22.5 mm
```

Parameter Name	Unit	Equation	Nominal Value
Model Parameters			
User Parameters			
Size	mm	10 mm	10.000000
if_1	ul	(1 ul - sign(Size - 8 mm)) * (1 ul - sign(8 mm - Size))	0.000000
if_2	ul	(1 ul - sign(Size - 10 mm)) * (1 ul - sign(10 mm - Size))	1.000000
if_3	ul	(1 ul - sign(Size - 16 mm)) * (1 ul - sign(16 mm - Size))	0.000000
GuideThk	mm	if_1 * 0.6 mm + if_2 * 0.6 mm + if_3 * 0.8 mm	0.600000
GuideDia	mm	if_1 * 11.5 mm + if_2 * 14.5 mm + if_3 * 22.5 mm	14.500000

Alternatively, we can use item #10 (equality with tolerance, excluding extremes) to make the equations for if variables, a little bit simpler:

```
t = 1e-5 mm
if_1 = sign(Size – 8 mm + t) * sign(8 mm – Size + t)
if_2 = sign(Size – 10 mm + t) * sign(10 mm – Size + t)
if_3 = sign(Size – 16 mm + t) * sign(16 mm – Size + t)
GuideThk = if_1 * 0.6 mm + if_2 * 0.6 mm + if_3 * 0.8 mm
GuideDia = if_1 * 11.5 mm + if_2 * 14.5 mm + if_3 * 22.5 mm
```

Parameter Name	Unit	Equation	Nominal Value
Model Parameters			
User Parameters			
Size	mm	10 mm	10.000000
t	mm	0.00001 mm	0.000010
if_1	ul	sign(Size - 8 mm + t) * sign(8 mm - Size + t)	0.000000
if_2	ul	sign(Size - 10 mm + t) * sign(10 mm - Size + t)	1.000000
if_3	ul	sign(Size - 16 mm + t) * sign(16 mm - Size + t)	0.000000
GuideThk	mm	if_1 * 0.6 mm + if_2 * 0.6 mm + if_3 * 0.8 mm	0.600000
GuideDia	mm	if_1 * 11.5 mm + if_2 * 14.5 mm + if_3 * 22.5 mm	14.500000