TYPE 579: NESTED FORCING FUNCTION

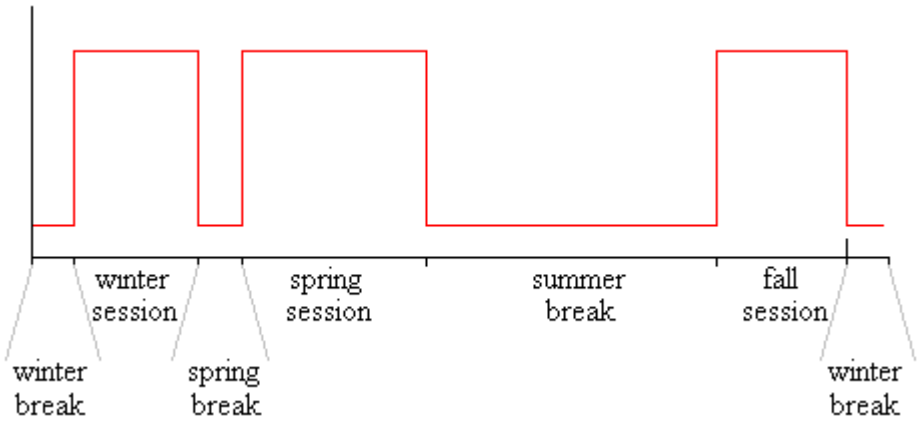
General Description

The nested forcing function eliminates the multiple TYPE 14 forcing functions and numerous equations needed to set the schedule for an input that changes throughout the year. This forcing function allows the user to "tier" the forcing functions such that the schedule of the tertiary function depends on the value of the secondary and primary forcing functions.

Detailed Description

The standard method for dealing with complex forcing functions in TRNSYS is to define multiple forcing functions for each period in question and then write one or more equations that pick which one of the forcing functions should be active at the current time. For example, to set the occupancy of a school building, three forcing functions might be defined. The first would repeat on an annual basis and would determine whether the school was in session or on vacation. The second might repeat on a weekly basis and determine whether the current time was a weekend or a weekday. The third might repeat on a 24 hour basis and determine what hours the school was occupied. An equation would then be written, multiplying the forcing function outputs together to form an annual occupancy profile.

To alleviate this complication, Type579 allows the user to define multiple tiers (up to 5) of forcing functions where lower tiers are only active if their upper tiers allow them to be. The repeat times (or lengths) of each forcing function cycle are then defined in decreasing order with the longest cycle occurring first. Once all the cycle lengths have been defined, the user specifies the number of ON periods that occur in each cycle. Taking the example of a school year, the first (longest) cycle might be the one that sets an output of 1 if the school is in session and a 0 if the school is out of session. If the school has a summer break, a winter break and a spring break, then there would be three ON periods as shown in Figure 1.



The next set of parameters that the user must enter are the start and stop times of each ON period in the cycle. The cycle's repeat time has already been set by specifying the length of each tier or cycle.

Once the primary cycle is defined, the user then proceeds on to the next longest cycle, specifying it in much the same manner. The major difference between the primary cycle and all subsequent cycles is that with non primary cycles, the user specify the number of ON cycles for when the superceding cycles are ON and when they are OFF. Taking the secondary cycle as an example, the user would have to

specify the number of ON periods when the primary cycle is ON, then the number of ON periods when the primary cycle is OFF.

While not complex, Type579 can be confusing at first; it is perhaps best explained through the use of the school example. A school's spring break begins April 1st (hour 2160) and lasts one week, ending hour 2328. Summer break begins on June 1st (hour XX) and ends September 1st (hour XX) Winter break begins December 1st (hour 8016) and ends December 31st (hour 8760). During times when the school is on break. The building is unoccupied. During non break times, the building is occupied Monday through Friday from 8 AM to 4 PM on Saturdays from 8 AM till noon.

Our forcing function will output a 1 if the building is occupied. In this example, there are three tiers (cycles) in the forcing function, a yearly cycle, a weekly cycle and a daily cycle. Since there are three breaks and three sessions in the school year, the first cycle has 3 ON periods. The number of ON periods in the second (weekly) cycle will depend on the day of the week that the simulation year begins. For example, if January 1 falls on a weekend day then there will be one ON period. If January 1 falls on a weekday there will be two ON periods because the occupied time will be split by a weekend. For the purposes of this example, it is assumed that January 1 falls on a Monday and that consequently, there are two ON periods in the secondary cycle. The daily cycle has only one ON period.

```
PARAMETERS 25
3     !number of tiers (one yearly, one weekly, one daily)
8760  !length of tier 1 (yearly)
168   !length of tier 2 (weekly)
24    !length of tier 3 (daily)
3     !number of ON periods in tier 1 (yearly)
0     !start of first ON period for yearly cycle
2160  !end of first ON period for yearly cycle
2328  !start of second ON period for yearly cycle
3624  !end of second ON period for yearly cycle
5088  !start of third ON period for yearly cycle
8016  !end of third ON period for yearly cycle
0     !number of ON periods in tier 2 (weekly) when tier 1 is OFF
2     !number of ON periods in tier 2 (weekly) when tier 1 is ON
0     !start of first ON period in weekly cycle when tier 1 is ON
24    !end of first ON period in weekly cycle when tier 1 is ON
72    !start of second ON period in weekly cycle when tier 1 is ON
168   !end of second ON period in weekly cycle when tier 1 is ON
0     !number of ON periods in tier 3 (daily) when tiers 1 and 2 are OFF
0     !number of ON periods in tier 3 (daily) when tier 1 is OFF and tier 2 is ON
1     !number of ON periods in tier 3 (daily) when tier 1 is ON and tier 2 is OFF
8     !start of first ON period in daily cycle when tier 1 is ON and tier 2 is OFF
12    !end of first ON period in daily cycle when tier 1 is ON and tier 2 is OFF
1     !number of ON periods in tier 3 (daily) when tiers 1 and 2 are ON
8     !start of first ON period in daily cycle when tier 1 and 2 are ON
16    !end of first ON period in daily cycle when tier 1 and 2 are ON
```

Note that when a cycle contains no ON periods, the user does not need to specify any start and stop times for that cycle.

The INPUTS to the model are the number of occupants present in the building during each of the cycle combinations. The INPUTS proceed from all cycles OFF to all cycles ON, following a Boolean progression. In the case of the school example, 8 inputs will be needed. The first indicates the number of people present in the building when all cycles are OFF (break, weekend, night). For the second input, the shortest cycle is set to ON; in other words the second INPUT would be the number of occupants in

the building during break weekend days. The progression continues, the last INPUT being the number of occupants during school sessions, on a weekday and during occupied hours. If 100 people are in the school during classes and 20 people are in the school on Saturdays, the INPUT list would be as follows:

```
INPUTS 8
0     !occupancy during school break, weekend nights
0     !occupancy during school break, weekend days
0     !occupancy during school break, weekday nights
0     !occupancy during school break, weekday days
0     !occupancy during school session, weekend nights
20    !occupancy during school session, weekend days
0     !occupancy during school session, weekday nights
100   !occupancy during school session, weekday days
```

There is a default limit of 25 ON cycles for each instance of Type579. The limits of 5 tiers and 25 ON cycles can be increased by modifying the Type579 source code if you have a Fortran compiler or by contacting your distributor if you do not.

Example

Type579 is used in the following Simulation Studio example:
\trnsys16\TESS Models\Studio Projects\Utility Library\Tiered Forcing Function.tpf