Awlsim - Funktionsplan (FUP) / Function Block Diagram (FBD)

FUP/FBD implementation (called FUP in the text below) in Awlsim.

Awlsim's FUP tries to resemble the S7 FUP closely. However there are some notable and intentional differences between Awlsim's FUP and S7 FUP. These differences are described below.

Chaining of non-booleans

In Awlsim FUP it is possible to connect integer (or REAL) outputs of elements directly (via wire) to integer (or REAL) inputs. That brings some additional semantics w.r.t. the EN mechanism.

In FUP lots of non-boolean elements carry an EN input and an ENO output. The EN input completely disables the execution of this particular element. That results in implicit semantics for chained elements. In Awlsim FUP the chained elements implicitly inherit the EN signal of their parent elements.

That means an element is only executed, if its EN is TRUE *and* the EN of its parent elements is also TRUE.

Boolean vs. non-boolean outputs in elements with EN input

In addition to that additional semantics are required for boolean outputs in elements with EN input:

- Boolean outputs of FUP boxes are always written to FALSE, if the EN of this FUP box is FALSE. This behavior is an Awlsim extension.
- Integer or REAL (-> non-boolean) outputs of FUP boxes are *not* written at all, if the EN of this FUP box is FALSE. This behavior resembles S7 FUP.

So whether an output is actually written depends on whether it is boolean.

No conversion from AWL to FUP language

There is no way to switch a program source from AWL to FUP language. That is a useless feature, so it won't be implemented in Awlsim.

However, the AWL code that is generated from a FUP diagram can easily be reviewed or even be reused for other purposes.

But an AWL program cannot be converted into a FUP diagram.