smv to core

module combination

1. The expression shown is written in module combination and module instance forms.

2. The expression shown is written in module combination and module instance forms.

3. The expression shown is written in module combination and module instance forms.

4. The expression shown is written in module combination and module instance forms.

---


core to smv

In Hebrew:

-ba'amos batatevos

by translating all the words in Hebrew to English.

-module combination

In Hebrew:

-ba'amos batatevos

by translating all the words in Hebrew to English.

-module combination

In Hebrew:

-ba'amos batatevos

by translating all the words in Hebrew to English.

-module combination

In Hebrew:

-ba'amos batatevos

by translating all the words in Hebrew to English.
5. מעריכים במדיות

We define the following operations on the vector a: $a[2][4] := \ldots$. The operations performed on the elements of the vector a are:

- $\text{HOLD_PREVIOUS}$: Hold the previous element of the vector.
- $\text{core}$: Core of the vector.
- $\text{processes}$: Processes of the vector.
- smv: smv processes.
- $\text{HOLD_PREVIOUS}$: Hold previous element of the vector.
- $\text{smv}$: smv processes.
- $\text{process}$: Process of the vector.
- x := \ldots: Initial state of the vector.
- $\text{next}(x) := \ldots$: Next state of the vector.
- $\text{A}$, $\text{B}$: Constants of the vector.

The operations performed on the vector are:

- $\text{HOLD_PREVIOUS}$: Hold the previous element of the vector.
- $\text{smv}$: smv processes.
- $\text{core}$: Core of the vector.
- $\text{processes}$: Processes of the vector.
- x := \ldots: Initial state of the vector.
- $\text{next}(x) := \ldots$: Next state of the vector.
- $\text{A}$, $\text{B}$: Constants of the vector.

The operations performed on the vector are:

- $\text{HOLD_PREVIOUS}$: Hold the previous element of the vector.
- $\text{smv}$: smv processes.
- $\text{core}$: Core of the vector.
- $\text{processes}$: Processes of the vector.
- x := \ldots: Initial state of the vector.
- $\text{next}(x) := \ldots$: Next state of the vector.
- $\text{A}$, $\text{B}$: Constants of the vector.
An example for the “DEFINE” problem
There are some cases dealing with DEFINE in smv that Eyal doesn’t take care. The main reason for that is that in core it's impossible to send a Parameter that is originated in DEFINE as a parameter to a module, as the following core program demonstrates:

```
MODULE SYSTEM()
{
    -- if we use the first line of the following two, parse error will
    -- occur:The actual parameter a is beyond the scope of the module!
    -- The second line works properly.

    DEFINE a := 6;
    -- VAR a : integer INITVAL 6;

    A(a)
}

MODULE A( b : integer)
{
    VAR
        c : integer;
    TRANS t:
        enable: true;
        assign: c':=b;

}
```