

proposal for 1420

AC-SV.

August 21, 2006

SECTION 17.11.4

Replace

RESTRICTION 4: For every recursive instance of property q in the declaration of property p , each actual argument expression e of the instance satisfies at least one of the following conditions:

- e is itself a formal argument of p .
- No formal argument of p appears in e .
- e is passed to a formal argument of q that is typed and the set of values for the type is bounded.

With:

RESTRICTION 4: For every recursive instance of property q in the declaration of property p , each actual argument expression e of the instance satisfies ~~at least~~ one of the following conditions:

- e is itself a formal argument of p .
- No formal argument of p appears in e .
- ~~e is passed to a formal argument of q that is typed and the set of values for the type is bounded.~~

Replace:

```
property check_write_data_beat
(
  expected_data, // [0:127]
  tag, // [3:0]
  i // [3:0]
);
first_match
(
  ##[0:$]
  (
    (data_valid && (data_valid_tag == tag))
    ||
    (retry && (retry_tag == tag))
  )
)
```

```

    )
  )
  |->
  (
    (
      (data_valid && (data_valid_tag == tag))
      |->
      (data == expected_data[i*8+:8])
    )
    and
    (
      if (retry && (retry_tag == tag))
      (
        1'b1 |> check_write_data_beat(tag, expected_data, 4'h0)
      )
      else if (last_data_valid)
      (
        1'b1 |> check_write_data_beat(tag, expected_data, i+4'h1)
      )
      else
      (
        ##1 (retry && (retry_tag == tag))
        |>
        check_write_data_beat(tag, expected_data, 4'h0)
      )
    )
  )
);
endproperty

with

property check_write_data_beat
(
  logic [0:127] expected_data,
  logic [3:0] tag,
  logic [3:0] i
);
logic [3:0] l_i; // pass argument i by value
(1'b1, l_i = i) ##0
first_match
(
  ##[0:$]
  (
    (data_valid && (data_valid_tag == tag))
    ||
    (retry && (retry_tag == tag))
  )
)
)

```

```

|->
(
  (
    (data_valid && (data_valid_tag == tag))
    |->
    (data == expected_data[i*8+:8])
  )
  and
  (
    if (retry && (retry_tag == tag))
    (
      1'b1 | => check_write_data_beat(tag, expected_data, 4'h0)
    )
    else if (last_data_valid)
    (
      1'b1 | =>
      check_write_data_beat(tag, expected_data, 1_i+4'h1i+4'h1)
    )
    else
    (
      ##1 (retry && (retry_tag == tag))
      | =>
      check_write_data_beat(tag, expected_data, 4'h0)
    )
  )
);
endproperty

```

Annex E.5 :

Replace

RESTRICTION 4: For every recursive instance of property q in the declaration of property p , each actual argument expression e of the instance satisfies at least one of the following conditions:

- e is itself a formal argument of p .
- No formal argument of p appears in e .
- e is passed to a formal argument of q that is typed and the set of values for the type is bounded.

With:

RESTRICTION 4: For every recursive instance of property q in the declara-

tion of property p , each actual argument expression e of the instance satisfies ~~at least~~ one of the following conditions:

- e is itself a formal argument of p .
- No formal argument of p appears in e .
- ~~e is passed to a formal argument of q that is typed and the set of values for the type is bounded.~~