Cliff votes yes - but I would like to see the following friendly amendments incorporated to correct typos in text and in examples.

Page 2: clause 15.12
WAS: (thereby executing the Observe region
CORRECTED: (thereby executing the Observe**d** region

Page 4 - comment in example code just before clause 15.14.1
WAS: equivalent to ##1 cb.v <= expr
CORRECTED: equivalent to ##1 cb.v <= expr**1**

//----------
This amendment may not be friendly enough but there are serious example typos that would confuse readers if we release this in June. Are these corrections sufficiently friendly???
Page 5 - initial block just before clause 15.14.2 - This describes synchronous drives, not nonblocking assignments. Also, to execute in the Re-NBA region (per the comment) this all has to be in a program, not a module. Also, an inout port cannot be a logic type (multiple drivers).

The one proposed correction that I am not 100% sure of is the statement cb.b <= **cb.**a; I believe we need the clocking block sampled version of a for this assignment (the description and comments still appear to be correct).

WAS:
```
logic a, b, c;
clocking cb @(posedge clk);
  inout a;
  output b;
endclocking

initial begin
  a <= c; // The value of a will change in the Re-NBA region
  b <= a; // b is assigned the value of a before the change
end
```

CORRECTED:
```
program (
  inout        a,
  output logic b,
  input  logic c, clk);

  clocking cb @(posedge clk);
    inout a;
    output b;
  endclocking

  initial begin
    cb.a <= c;    // The value of a will change in the Re-NBA region
    cb.b <= cb.a; // b is assigned the value of a before the change
  end
endprogram
```
//----------

Page 5 - 2nd example in 15.14.2 - The paragraph describes updates in the Re-NBA region, so I believe this should be a program and not a module.

WAS:
```
module m;
  bit a = 1'b1;
```

```
  default clocking cb @(posedge clk);
    output a;
  endclocking

  initial begin
    ## 1;
    cb.a <= 1'b0;
    @(x); // x is triggered by reactive stimulus running in the same time step
    cb.a <= 1'b1;
  end
endmodule
```

CORRECTED:

```
program m;
  bit a = 1'b1;

  default clocking cb @(posedge clk);
    output a;
  endclocking

  initial begin
    ## 1;
    cb.a <= 1'b0;
    @(x); // x is triggered by reactive stimulus running in the same time step
    cb.a <= 1'b1;
  end
endprogram
```

Page 6 - 2nd to last paragraph - proposed clarification (nonblocking assignments and synchronous drives schedule into the NBA and Re-NBA regions - blocking assignments schedule into Active and Re-Active regions).
WAS:
If a synchronous drive and a procedural assignment write to the same variable in the same time step, the writes shall take place in an arbitrary order.

CLARIFIED:
If a synchronous drive and a procedural **nonblocking** assignment write to the same variable in the same time step, the writes shall take place in an arbitrary order.

Page 7 - Last sentence of 16.2.1 - Typo
WAS:
Code in **module**, **interface**, or **package** scope may execute as part ~~either of~~ the Active region set or the Reactive region set processing.

CORRECTED:
Code in **module**, **interface**, or **package** scope may execute as part **of either** the Active region set or the Reactive region set processing.