unique and priority case statements

Unique Case

- Unique is an *assertion* which implies:
  - All possible values of `case_expression` are in the `case_items`
  - At simulation run time, a match must be found in `case_items`
  - At run time, only one match will be found in `case_items`
- Unique guides synthesis
  - It indicates that no priority logic is necessary
  - This produces parallel decoding which may be smaller/faster
- Unique usage
  - Use when each `case_item` is unique and only one match should occur.
  - Using a "default" `case_item` removes the testing for non-existent matches, but the uniqueness test remains

`unique` and `priority` case statements

Priority Case

- Priority is an *assertion* which implies:
  - All possible values for `case_expression` are in `case_items`
- Priority guides synthesis
  - It indicates that all other testable conditions are don't cares and may be used to simplify logic
  - This produces logic which is possibly smaller/faster
- Priority usage
  - Use to explicitly say that priority is important even though the Verilog case statement is a priority statement.
  - Using a "default" `case_item` will cause priority requirement to be dropped since all cases are available to be matched.
  - Use of a "default" also indicates that more than one match in `case_item` is OK.
- Priority is a bad name. Case is already a priority structure.