Concurrent Assertions

Based on clocking !!

- delays based on clock block
- only cycles

assert property ( ) true; else false;

- The property can be defined separetly
- true seldom used

Properties

- boolean expressions
- real, shortreal, realtime, string, event, etc. not allowed

disable iff(expr)
Turns off assertion if true

expr **|->**
trigger. Assertion waits for expr to be true

expr **|=>**
trigger. Assertion waits until next clock after true

expr limits
no += ++ – etc

cycle expressions
##nnn nnn is a number (waits nnn cycles)
##name name is variable, waits amount
##(expr) waits constant expr cycles
##[a:b] waits a to b cycles ($ is forever)

and a **and** b
All sequences matched

or a **or** b
At least one sequence matched

intersect a **intersect** b
both sequences end at the same time

throughout expr **throughout** seq
expr true during seq

within seq1 **within** seq2
seq1 occurs within seq2

first_match **first_match**(seqs)
the first occurance of seqs matches
will take the shortest sequence

expr1 ##1 expr2
match expr1, then expr2 a cycle later

(seq **,** match item**)**
match item --> assignment (local variable)

(seq)[* value **]** consecutive repetition
repeated value times
a:b also

expr [* range**]** true at the end of the range

expr [= range **]** expr true a,b (range) times

sequence name **;**
sequence stuff
endsequence

Handy Functions

- $onehot(expr) one bit high
- $isunknown(expr) any bit x or z
- $rose(expr)
  - lsb changed to 1
- $fell(expr)
  - lsb changed to 0
- $stable(expr)
  - expr didn't change
- $past(expr1,[ticks],[expr2])
  - value ticks in the past
  - clocking enable with expr2