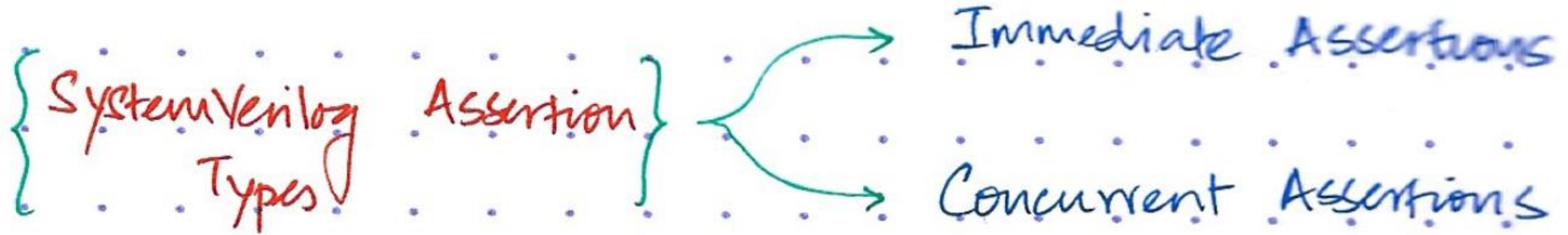


SystemVerilog Assertions (SVA)

```
fifo_will_never_goes_full: assert property(  
    @(posedge clk) !fifo_is_full  
);
```

```
memory_write: assert property(  
    @(posedge clk) disable iff(reset)  
    en ==> mem[$past(addr)] == $past(data)  
);
```

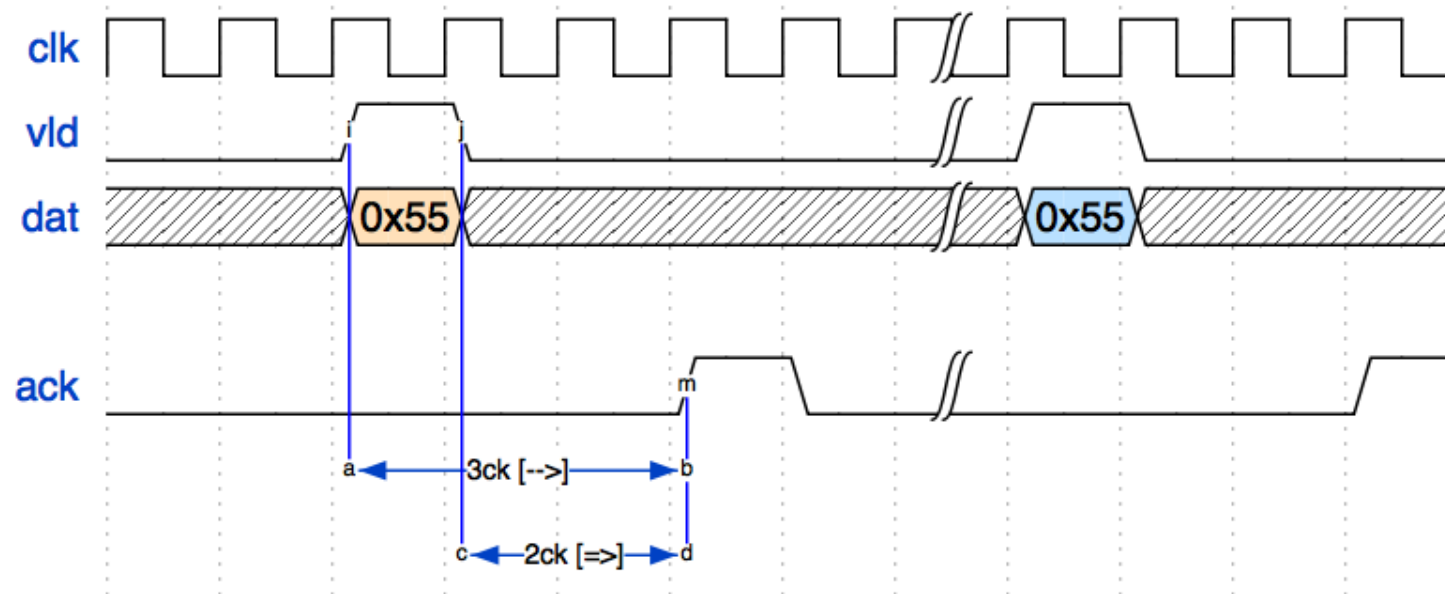
Типы ассершенов



Структура concurrent assertion

```
assertion_label: assert property(  
    @( <event_trigger> ) disable iff( <disable_condition> )  
    <property_body>  
)  
else begin ...; end // <- эта строка опциональна
```

Операторы импликации в concurrent assertion



// If inputs vld=1 and dat=8'h55, then ack is high 3 cycles later.

// The RHS is evaluated from the same cycle LHS is true

```
valid_gnt_chk: assert property(@posedge (clk) disable iff (rst)
    (vld && dat == 8'h55) |-> ##3 ack);
```

// The RHS is evaluated one clock cycle after LHS is true

```
valid_gnt_chk: assert property(@posedge (clk) disable iff (rst)
    (vld && dat == 8'h55) |=> ##2 ack);
```

Операторы

Оператор	Применение
##n ##[m:n]	операторы задержки на фиксированное число тактов и диапазон тактов соответственно
!, , &&	булевы операторы
->	совмещённая импликация
=>	несовмещённая импликация lication

Системные функции

Функция	Применение
\$rose	возвращает true, если младший бит выражения изменился в 1. В противном случае, возвращает false.
\$fell	возвращает true, если младший бит выражения изменился в 0. В противном случае, возвращает false.
\$stable	возвращает true, если значение выражения не изменилось. В противном случае, возвращает false.
\$past(expression, num_cycles)	возвращает значение выражения expression num_cycles циклов назад.
\$countones	возвращает число единиц в выражении
\$onehot	возвращает true если ровно один бит равен 1. Если ни один из бит не равен 1 или таких бит больше одного, возвращает false
\$onehot0	возвращает true, если ни одного бита, или только один равен 1.
\$isunknown	возвращает true, если хотя бы один бит в выражении равен X или Z.

Подключение ассершенов

```
bind binding_module_name: binding_instance_name
    bound_module_name bound_instance_name(portlist);
    ^- optional | can be .* if ports are the same
```

```
bind binding_instance_name bound_module_name bound_instance_name(portlist);
```